



云效2020 云效流水线 Flow

文档版本: 20220713



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。	⚠ 危险 重置操作将丢失用户配置数据。
⚠ 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。	會告 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 <i>,</i> 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 {alb}	表示必选项,至多选择一个。	switch {act ive st and}

目录

1.云效流水线 Flow 是什么	11
2.初识云效流水线「Flow」	13
2.1. 「企业」是什么	13
2.2. 「流水线」是什么	13
2.3. 「Flow」快速上手教程	13
3.流水线	14
3.1. 创建流水线	14
3.2. 流水线源	15
3.2.1. 流水线源	15
3.2.2. 配置代码源	16
3.2.2.1. 配置代码源	16
3.2.2.2. 企业 SSH 公钥配置最佳实践	16
3.2.3. 代码提交触发	17
3.2.3.1. 代码提交触发	17
3.2.3.2. 触发事件与过滤条件	18
3.2.3.3. 云效Codeup 中配置 WebHook	19
3.2.3.4. 阿里云 Code 中配置 WebHook	20
3.2.3.5. 码云 Gitee 中配置 WebHook	20
3.2.3.6. Github 中配置 WebHook	21
3.2.3.7. Gitlab 中配置 WebHook	21
3.2.3.8. Coding 中配置 WebHook	21
3.2.3.9. 通用 Git 中配置 WebHook	22
3.2.4. 配置 Jenkins 源	22
3.2.5. 配置 Flow 流水线源	24
3.3. 编排流水线	28
3.4. 触发流水线运行	30

3.4.1. 触发流水线运行	30
3.4.2. 定时触发流水线	31
3.4.3. Webhook 触发流水线	31
3.5. 查看最近运行	34
3.6. 批量操作	35
3.7. 流水线分组	35
4.代码管理	38
4.1. 分支模式	38
4.2. 代码合并	44
5.构建	46
5.1. 构建集群	46
5.1.1. 构建集群	46
5.1.2. 北京和中国(香港)构建集群	46
5.1.3. 自有构建集群	46
5.1.3.1. 自有构建集群	47
5.1.3.2. 阿里云 ECS 构建集群	49
5.1.3.3. 自有主机构建集群	51
5.2. 构建语言支持	54
5.3. 设置依赖下载访问凭证	55
5.4. 构建容器镜像	58
5.4.1. 构建容器镜像	58
5.4.2. 构建阿里云容器镜像	58
5.4.3. 海外集群构建容器镜像	63
5.5. 构建制品	67
5.5.1. 构建制品	69
5.5.2. Maven 仓库	70
5.5.3. Go 镜像代理	70
5.5.4. NPM 仓库	70

5.6. 制品管理	71
5.6.1. 制品管理	71
5.6.2. 归档至云效公共存储空间	71
5.6.3. 上传至 Packages 通用制品仓库	73
5.6.4. 上传至阿里云 OSS	75
5.6.5. 上传至 Maven 私服	79
5.7. 自定义镜像构建	81
5.8. 移动端构建	82
5.8.1. 移动端构建	82
5.8.2. Android 构建	83
5.8.3. 微信小程序构建发布	84
6.部署	88
6.1. 新建部署任务	88
6.2. 主机部署	88
6.2.1. 主机部署	89
6.2.2. 主机组	89
6.2.2.1. 主机组	90
6.2.2.2. 部署到阿里云 ECS	92
6.2.2.3. 部署到公网主机	95
6.2.2.4. 部署到免费体验主机	97
6.2.3. 部署配置	101
6.2.4. 部署策略	103
6.2.5. 查看部署详情	103
6.2.6. Docker 部署	105
6.2.7. 部署历史 和 回滚	106
6.3. Kubernetes 部署	107
6.3.1. Kubernetes 部署	107
6.3.2. Kubernetes 服务连接	107

6.3.2.1. Kubernetes 服务连接	109
6.3.2.2. 获取 Kubernetes 集群证书文件	111
6.3.2.3. 如何手动生成 Kubernetes Config 文件	112
6.3.3. Kubernetes Kubectl 发布	113
6.3.3.1. Kubernetes Kubectl 发布	113
6.3.3.2. 如何在 YAML 中使用自定义变量	113
6.3.3.3. Kubernetes常用资源 YAML 规范参考	114
6.3.3.4. 查看 Kubernetes 部署日志	115
6.3.4. Kubernetes 镜像升级	116
6.3.5. Kubernetes 分批发布	117
6.3.6. Kubernetes 蓝绿发布	119
6.3.7. ACK Nginx Ingress 灰度发布	122
6.3.8. Helm Release 部署	128
6.3.9. ASM 蓝绿发布	130
6.3.10. Kubernetes发布常见问题	139
6.3.11. Kubernetes发布和运维职责分离	140
6.3.12. 部署历史 和 回滚	140
6.4. EDAS 应用发布	140
6.5. EDAS Kubernetes 应用发布	144
6.6. SAE 应用发布	147
6.7. 阿里云函数计算 FC 发布	150
6.8. 资源编排服务 ROS 发布	152
6.9. 无构建部署	156
7.质量检测	158
7.1. 代码扫描	158
7.1.1. 代码扫描能力	158
7.1.2. 使用代码扫描	158
7.1.3. 全量扫描和增量扫描	159

7.1.4. P3C代码扫描	159
7.1.4.1. P3C代码扫描	160
7.1.4.2. 指定扫描规则集	160
7.1.4.3. 指定或排除目录扫描	160
7.1.5. 使用自定义代码扫描规则	161
7.1.5.1. 使用自定义代码扫描规则	161
7.1.5.2. ESLint自定义规则	161
7.1.5.3. TypeScript自定义规则	162
7.1.5.4. Java P3C自定义规则	163
7.2. 单元测试	164
7.3. 质量红线	165
8.变量和缓存	168
8.1. 环境变量	168
8.2. 运行选择变量	174
8.3. 通用变量组	175
8.4. 缓存设置	177
9.通知	181
9.1. 使用 Webhook 插件发送通知	181
9.2. 钉钉机器人发送群消息	181
9.3. 企业微信机器人发送群消息	183
9.4. 飞书机器人发送群消息	185
10.企业管理	188
10.1. 权限管理	188
10.1.1. 企业权限与角色管理	188
10.1.2. 流水线成员权限	189
10.1.3. 主机组成员权限	190
10.1.4. 流水线分组权限	190
10.2. 流水线模板管理	191

10.3. 任务组管理	194
10.4. 任务组批量升级	197
10.5. 步骤管理	200
10.6. 授权管理	201
10.7. 服务连接管理	203
11.自定义步骤 CLI 工具	206
11.1. 自定义步骤 Flow CLI 子命令: step	206
11.2. 步骤 yaml 描述语言	210
11.3. 步骤 Yaml 描述语言场景补充	211
11.4. Flow cli使用常见问题	212
11.5. 创建一个带红线功能的测试步骤	214
12.常见问题	220
12.1. 构建问题	220
12.1.1. Java 构建失败—构建依赖缺失	220
12.1.2. Java 构建缓慢—未设置有效缓存目录	221
12.1.3. Java 构建缓慢—删除缓存文件	223
12.1.4. Java 构建缓慢—自定义 Maven 配置	224
12.1.5. Java 构建缓慢—下载海外依赖	224
12.1.6. 镜像构建缓慢—从海外下载基础镜像	225
12.1.7. 构建物上传失败	226
12.1.8. 本地调试优化	228
12.1.9. 自有构建集群缓存清理	229
12.1.10. Composer 版本升级	230
12.1.11. 流水线运行过程中定义环境变量	230
12.2. 部署问题	231
12.2.1. 主机部署—环境变量包含特殊字符	231
12.3. 管理功能问题	233
12.3.1. 授权管理—添加多个阿里云账号的授权服务	233

12.3.2. 授权管理—RAM 账号授权失败	235
12.3.3. 通过钉钉进行人工卡点审批	235
12.4. Flow IP 地址	סכר
12.5. 部署常见问题	238
12.6. 错误:"超时取消"问题排查	238
12.7. 构建常见错误排查	239
13.迁移	241
13.1.从CodePipeline迁移	241
14.计费规则	246
14.1. 并发数和构建时长	246
15.最佳实践	249
15.1. 构建加速	249

1.云效流水线 Flow 是什么



「流水线」,又名「Flow」,是一款企业级、自动化的研发交付流水线, 提供灵活易用的持续集成、持续验证、 持续发布功能,帮助企业高质量、高效率的交付业务。 流水线是持续交付的载体,通过构建自动化、集成自动化、验证自动化、部署自动化,完成从开发到上线过程的持续交付。通过持续向团队提供及时反馈,让交付过程高效顺 畅。

更多「云效」产品,查看:云效快速入门

为什么选择「Flow」

快速上手配置

相比较自建的持续交付工具,使用 Flow 仅仅需要几步,就可以快速搭建 CICD 流程。 Flow 为你提供了开箱即用的构建环境,提供了数十种通用的流水线模版帮助你快速创建流水线,从此,你的企业无需花费精力维护你的 CICD 工具。 同时,我们提供了支持各种场景的最佳实践文章供你参考,帮助你更快上手使用。Flow语言专项场景

各种覆盖各种工具和技术栈

Flow 作为一款企业级的自动化交付流水线,全面覆盖研发场景中涉及的技术栈和工具链。

构建		代码扫描		部署	
Java 构建	Maven	Java 安全扫描	PHP Metrics 静	主机部署	ECS应用部署
Java 构建, 压缩包	Maven 构建,并上…	Java 安全扫描	PHP Metrics 静态	主机部署	ECS应用部署
たつ 获取GIT版本 获取GIT版本	Python 构建 Python 构建 上传	nede JavaScript 代 JavaScript 代码扫描	Python 代码扫描 Python 代码扫描	主机部署(元构 使用脚本发布到主机	Docker部署 主机Docker部署
C-) Serverless Devs	GO 构建	源码漏洞检测	Python依赖安 扫描Python依赖文	ACK Nginx Ingr	Kubernetes 发布
使用Serverless De	Go 构建	Sourcebrella Pinp		基于ACK Nginx In	用于发布 Kuberne
NET Core 构建	Node.js 构建	Python 代码安	GO 代码扫描	Kubernetes 镜	Kubernetes 蓝
.Net Core 构建,	Node.js 构建	使用Bandit扫描代	Go 代码扫描	升级 Kubernetes	基于Istio进行应用
GCC 构建 GCC 构建, 压缩包	php PHP 构建 PHP 构建上传到仓库	TSLint 代码扫描 TypeScript 代码扫	Android 代码扫描 Android 代码扫描	Kubernetes 分… Kubernetes 分批…	Helm Release 使用代码库中的 H

1. 流水线源

Flow 支持将业界通用的代码仓库作为流水线的触发源,并且支持通过代码提交、Tag 创建、合并请求等代码仓库事件触发流水线运行。

同时, Flow 的流水线也支持通过其他工具直接触发运行,如:自建 Jenkins 服务/Flow 流水线

2. 构建工具

Flow 作为一款云原生时代的流水线工具,通过容器技术彻底摆脱了对于虚拟机构建环境的依赖,您甚至可以根据您的使用需求,在同一条流水线上使用不同的构建环境。此 外,提供了各种语言的容器环境,满足不同的构建使用场景。更多内容可查看<mark>构建语言支持</mark>。

3. **自动化测试**

为保障企业整体研发流程的质量,Flow 提供代码扫描、 安全扫描和各种自动化测试能力,支持人工测试卡点、自动化验证卡点等多种质量红线,确保业务质量。更多内容可 查看代码扫描能力。

4.部署能力

Flow 支持不同国家,不同云厂商以及专有云环境发布。无论你的企业使用的虚拟主机 or Kubernetes,都可以使用 Flow 实现轻松发布。另外,通过灰度发布、分批发布的策 略,最大限度的避免了不稳定发布对用户的影响, 保障业务交付的稳定。更多内容可查看主机部署和Kubernetes 部署

5. 阿里云深度集成

Flow 全面对接阿里云 ECS/ACK/ACR/OSS/EDAS/SAE/FC 等多种阿里云云服务,可以通过流水线串联阿里云产品整体使用路径。

强大的研发分支管理能力

阿里巴巴在 DevOps 的最佳实践也体现在分支管理能力上,Flow 支持将常用的研发模式融入流水线,将企业 DevOps 的整个体系流程化。

丰富而灵活的模版

Flow 提供了数十种通用的流水线模版帮助你快速创建流水线,与此同时,支持企业定制自有模版来管理企业持续集成和持续交付流程。

服务于企业

Flow 致力于为企业提供优质的服务:

- 通过企业管理组织架构和企业数据。
- 提供安全加密机制保障企业数据安全。
- 复杂的流程通过可视化编排和结果展现,保障交付可见可控可度量。
- 企业级权限体系,对交付资源和成员进行精细控制,保障交付的安全性。
- 和钉钉,邮件,站内消息打通,提供即时反馈。
- 专业的客户服务支持。

如何收费

Flow 致力于为企业用户提供优惠且稳定的 CICD 服务,因此为企业提供了一定额度的免费资源使用量。更多计费规则可查看并发数和构建时长。

2.初识云效流水线「Flow」 2.1.「企业」是什么

什么是企业

企业空间是为企业或者组织设定的一个专属区域,每个企业空间相互独立。管理员可以在企业空间中对企业人员、企业数据进行管理。 使用「Flow」前,你需要先创建一个企业。

如何创建企业

如果你是云效的企业用户,你不需要重新创建企业空间,只需要前往当前企业空间的应用中心内安装「Flow」,即可以立即使用了。 对于非云效的企业用户,创建企业空间也很简单。

首先前往Flow 官网,点击登录按钮,按照提示进行操作。

2.2.「流水线」是什么

什么是流水线

流水线是「Flow」提供的自定义流程编排工具,通过构建,部署,测试,管控等组件化能力,把从开发到交付的各项工作串联起来,从而让企业轻松的实现持续交付 。

如何使用流水线

如果,你还没有完成了云效企业创建,请参考 "「企业」是什么"一节注册企业。 如果,你已经完成了云效企业创建,请直接登录「Flow」官网。

流水线的基本概念

- 输入源: 持续交付的原始物料, 如 Git 仓库。
- 阶段: 在流水线中需要按顺序执行的一组任务的集合,一个阶段可以是手动运行也可以是自动运行的。阶段之间串行执行;
- 任务: 在阶段中具体需要完成的动作,任务之间可以串行执行也可以并行执行,目前任务包含:代码扫描,单元测试,构建,部署,合并代码,人工审核等功能。这些任 务分为两类,一类是与工作区相关的(AgentJob)以及与工作区无关的(Agentless Job),工作区相关任务会在固定的工作目录中执行用户定义的任务。
- 步骤:步骤作为Flow最底层的核心能力,所有的流水线都是通过步骤的编排组合而成的。

具体使用请参考"流水线"一章。

2.3. 「Flow」快速上手教程

提前准备

- 1. 直接使用 Flow 示例代码库
- 阿里云 Code 上 Flow 基于 SpringBoot 的示例代码库: https://code.aliyun.com/code-template/spring-boot。
- 2. 在代码库里面 Flow 已经准备来应用部署启动的脚本,可以直接使用
- 可以参考示例的部署脚本 deploy.sh https://code.aliyun.com/code-template/spring-boot/blob/master/deploy.sh。
- 3. 注册并登陆 「Flow」

步骤一: 创建流水线

创建流水线

- 步骤二: 在部署任务中添加部署主机
- 步骤三:运行流水线

3.流水线 3.1. 创建流水线

点击流水线列表右上角的"新建流水线"按钮,开始创建流水线

☆ 我的星标	我的星标			(1) =↓ ► ◎ 新建	流水线
⑧ 我参与的	流水线名称	最近运行状态	最近运行阶段	运行人	最近运行时间	
♀ 全部流水线	飞流持续集成	#1 - 🥑	测试 构建	🥏 杨超越	2019-09-24	*
设置信息	行云日常环境持续交付	#1 - 😒	測试 构建 部署 ●○	🌝 杨超越	2019-09-24	*
C LINE	飞流生产环境持续交付	#15 - 🥑	測试 部署 合并主干 新増阶 ◎○○	刘 刘昊然	2019-09-24	*
	行云持续集成	#1 - 😒	測试 构建 ● ○	● 吴彦祖	2019-09-23	*
	飞流日常环境持续交付	#14 - 🥑	测试 构建 日常部署	🥏 杨超越	2019-09-24	*
	所思日常环境持续交付	#2 - 🥑	构建 部署	🌸 刘德华	2019-09-23	*
	飞流预发环境持续交付	#12 - 🤌	测试 构建 部署 。●	🐽 周杰伦	2019-09-23	*
	行云生产环境持续交付	#87 - 🤌	測试 部署 合井主干 新増阶… ●○	刻 刘吴然	2019-09-24	*

选择流水线模版

新建流水线时,选择对应的开发语言,可以查看当前语言下的默认流水线模版。

☆ 我的星标	我的星标			FL 🛛 🕄	新建流水
 ② 我参与的 □ 全部流水线 	1	选择流水线模板	×	最近运行时间	
	5 空流水线	PHP	杨超越	2019-09-24	*
设置信息	🐇 Java	PHP 测试、构建、部署到主机	杨超越	2019-09-24	*
⑥ 企业设置	JS NodeJS	代码扫描 4028 0738	刘吴然	2019-09-24	*
	GO GO	T ^{ay} xe naver	吴彦祖	2019-09-23	*
	Python	10) m	杨超越	2019-09-24	*
	55 共元	PHP 测试、构建 代码扫描	刘德华	2019-09-23	*
		1,5,4535,88	周杰伦	2019-09-23	*
		测试	刘昊然	2019-09-24	*

选择代码源

选择完模版后,可以选择你使用的代码源,作为流水线的开始的源头输入。

源	● 新建源	测试			新建代码源			× 常部署	新的阶段
5.2. (Jan 19) and a state			选择代码源						
C−3 flow/flow-engine		+	〇	(2) 阿里云	G 码云	Git	Github	3 日常部4	日 ●新約任务
		Ŷ	Bitbucket	Gitlab	Coding				
			正在使用	可里云账号为:	усу				
			代码仓库				新建代码	库	
			git@code.aliyu	in.com:1024/de	ocker-08191.git		~		
			分支						
			请选择分支				~		
			代码提交触发				a		
							取消 确认		

配置代码源的介绍和使用,可查阅"流水线源"篇。 需要启用代码提交触发,可查阅"代码提交触发"篇。

完成新建

完成创建后,默认进入到流水线的编排界面,可以继续流水线的输入源和流程编排。

源	新建源	测试	构建	日常部署	新的阶段
C-J flow/flow-engine			3描 🗈 🤣 Java构建, 1	7压缩包 🖪 🔗 E	常部者 🚦 💿 新的任务
		尽 Maven单元	测试		
1				2	
源				编排区	

流水线编排的介绍,可查阅"<mark>编排流水线</mark>"篇。

3.2. 流水线源

3.2.1. 流水线源

流水线源,作为 Flow 流水线的输入源头,可以从 文件托管服务 或者 其他第三方服务 中获取原料,并将这些原料放入到工作区。 流水线会对输入获取并放入工作区的原料进行处理,处理的目的可能是执行的内容可能是编译打包、执行自动化任务、版本制作。 流水线处理输出的结果可能包括软件的测试报告、压缩文件、目录、文档等等。 目前 Flow 支持的流水线源类型:

- 代码源
 - 配置代码源
- Jenkins 源
 配置 Jenkins 源
- Flow 流水线
- 配置 Flow 流水线源

后续 Flow 还将继续推出其他类型的流水线源,敬请期待:

- OSS
- 制品仓库

3.2.2. 配置代码源

3.2.2.1. 配置代码源

在流水线的编辑页面, 点击「添加流水线源」按钮,可以为流水线添加输入代码源,如下所示:



Flow支持将业界通用的代码仓库作为流水线的触发源,包括:

- 云效Codeup
- 阿里云Code
- 自建 Git
- 码云Gitee
- Git lab 公有云
- Git hub 公有云
- Git hub 企业版
- Bit bucket

Coding

如需配置代码提交触发,请参考文档"代码提交触发"

3.2.2.2. 企业 SSH 公钥配置最佳实践

对于使用了 自建 Gitlab、Bitbucket、Gitlab、Coding、通用 Git 的用户除了直接配置用户名和密码来进行代码源授权以外,还可以使用企业 SSH 公钥,如下图所示:

〈 返回 流水线 2020-07-27		基本信息 流程配置 制	触发设置 变量和缓存				仅保存	保存并运行
源	执行命令		新阶段	添加流水线源				×
				代码源	选择代码源			
riow-example/spring-boot	Kubernetes &/b	34(1)%) &		Jenkins Flow流水线	O 自建Gitlab	示例代码源	阿里云Code	G 码云
十添加代码源					<mark>③</mark> 云效Codeup	③ 通用Git	Github	\$
					代码仓库 ⑦			
					请输入代码行	这库		
					默认分支 ⑦			
					请输入分支			
					选择凭证类型	○ 企业公组		
					0 17661	0 TTT		
					Ssh-rs	a AAAAB3NzaC1	yc2EAAAADAQA	复制 重置
				源 加				

拷贝企业 SSH 公钥,并添加到 Git 仓库的中。以 自建 Gitlab 为例,为了能够使用企业 SSH 公钥同步代码,需要将企业 SSH 公钥绑定到 Git 仓库个人账号上。由于每个公钥在 同一个代码平台上仅可配置一次,因此最佳实践是将企业管理员将 SSH 公钥配置到企业公共账号,Flow 通过公共账号同步代码。

3.2.3. 代码提交触发

配置 WebHook 后,在相应的代码地址和分支上提交代码后就可以触发流水线的运行了。

如何配置

1. 编辑代码源,开启代码提交触发监听,并将对应的 WebHook 复制出来。

				新建代码源	l.	\times		
新建源	阶段1	选择代码源						
	B - Ø	(つ) 阿里云	⑤ 码云	Git	Github	Bitbucket		
		W Gitlab	Coding					
			当前账号绑	定的阿里云账号	为【果薇sl】			
未配置源		代码仓库				新建代码库		
		请选择代码仓	库			~		
		分支						
		请选择分支				~		
		开启监听			1			
		A https://	flow-openapi.te	ambition.com/s	cm/webhook/e	2AinGZD6tkiOT	•	
		将该地址复制到代	码仓库WebHook监	听代码事件			-	

访问对应的代码库地址,在代码库 --> Settings (设置) ---> Webhooks,添加上一步复制的 WebHook 地址。
 不同的代码源添加 WebHook 的方式可以参看子文档。

通过上述方式配置 WebHook 后,在相应的代码地址和分支上提交代码后就可以触发流水线的运行了。

3.2.3.1. 代码提交触发

配置 WebHook 后,在相应的代码地址和分支上提交代码后就可以触发流水线的运行了。

如何配置

1. 编辑代码源,开启代码提交触发监听,并将对应的 WebHook 复制出来。

MUTER 2021-00-22	基本信息 流程配置	触发设置 变量和缓存	了。
认水线源	构建	添加流水线源	:
p flow-example/spring-boot 冷 master ☐ flow-example_spring-b +添加流水线源	存 Java 构建上传	代码源 Jenkins Flow流水线	 合庫 spring-boot ■ ■

2. 访问对应的代码库地址,在代码库--> Settings (设置)---> Webhooks,添加上一步复制的 WebHook地址。不同的代码源添加 WebHook的方式可以参看子文档。 通过上述方式配置 WebHook后,在相应的代码地址和分支上提交代码后就可以触发流水线的运行了。

3.2.3.2. 触发事件与过滤条件

你可以使用触发事件和过滤条件的配置来对代码源Webhook事件进行过滤。

不同代码源支持的触发事件

	阿里云Code	云效Codeup	通用Git	Github	Gitlab	Coding	Bitbucket	自建Gitlab
代码提交	D	٥	D	٥	D	D	٥	D
Tag创建						D		
合并请求完成后	D	0	D	0	D	0	0	D
合并请求 新建/更 新	٥		٥		٥	٥		٥

分支或标签过滤

通过配置 分支或标签过滤,可以对代码源不同触发事件的分支进行过滤,下面针对每一种事件进行解释:

代码提交

从本地 push 分支到远程分支(或在代码管理平台的Web端进行文件修改),比如从本地的master push到 origin/master。Flow 会将 push 的目标分支与过滤条件进行匹配, 如果匹配成功,则会触发流水线,并使用 push 的目标分支作为流水线运行时使用的分支。

Tag创建

从本地 push tag 到服务端 tag(或在代码管理平台的 Web 端进行 Tag 创建),比如将本地的tag release/0.0.1 push 服务端。Flow 会将 push 的目标 Tag 名称与过滤条件进 行匹配,如果匹配成功,则会触发流水线,并使用目标 Tag 作为流水线运行时使用的代码版本。

合并请求 新建/更新

在代码管理平台的 Web 端创建合并请求,Git Lab 或者 Codeup支持此类事件。比如从远程的 dev 分支(源分支)发起一个合并请求到 master 分支(目标分支)。Flow 会将 合并请求的目标分支与过滤条件进行匹配,如果匹配成功,则会触发流水线,并使用源分支作为流水线运行时使用的分支。

对于 Codeup 来说,也可以使用 git-repo来触发合并请求的创建。与普通的合并请求相比,使用 git-repo 创建合并请求的源分支不是一个固定的分支,而是一个特殊的 ref, 比如refs/merge-requests/123/head。对于这种情况,Flow 在流水线运行时,会将该 ref 检出到一个临时的分支,进行后续的构建。

合并请求完成后

与合并"合并请求新建/更新"相比,唯一的差别是: Flow 使用目标分支作为流水线运行时使用的分支。

过滤条件的匹配规则

代码提交触发的过滤条件支持正则表达式。

你可以输入一个普通分支,比如master或者dev。也可以使用Java语言支持的正则表达式语法,以下是几个示例。 如果想匹配多个分支,比如 master 和 dev,可以输入 master[dev 如果想匹配以 feature 开头的分支,可以输入 ^feature.* 完整的正则表达式支持请参见:https://docs.oracle.com/javase/8/docs/api/java/util/regex/Pattern.html。

正则表达式在线验证工具: https://www.freeformatter.com/java-regex-tester.html。

代码路径过滤

通过配置 代码路径过滤,可以对代码源不同触发事件的代码路径进行过滤,目前仅支持 CodeUp 代码源 Push 事件。当代码源触发事件发生时,只有代码路径过滤规则下的代 码发生更新才可以触发流水线运行,填写过滤条件支持正则表达式。

3.2.3.3. 云效Codeup 中配置 WebHook

前序操作

1. 在 Flow 流水线,点击编辑流水线,点击编辑代码源。

在云效 Codeup 配置 WebHook

1. 在流水线的代码源中开启代码提交触发监听,Flow 会自动将 Webhook 插入对应代码库中。



11 🕃 首页 >	haha 👻 🔺			邀请	🍊 体验透镜	Q、搜索当前	库或跳转	
设置								
基本	设置	N	lebhooks					● 新建 Webho
分支	设置	允	许开发人员通过订阅代码库事件(分支推送、 	标签推送等)来触发构建部署	更新镜像等操作			10.0-
推送	规则设置	(1)		描述	肥友事件 推送事件	行效批送車の	△ ++	操作
合并	请求设置		nup://now-openapi.aliyun.com/sc		1田达+野1牛	机並推达事件	口汁用水争件	
Web	hooks							
部署	密钥							
集成	与服务							
大文	件存储							
仓库	备份 Beta							

3.2.3.4. 阿里云 Code 中配置 WebHook

前序操作

- 1. 在 Flow 流水线,点击编辑流水线,点击编辑代码源。
- 2. 开启代码提交触发监听,并将对应的 WebHook 复制出来。

在阿里云 Code 配置 WebHook

- 1. 访问阿里云 Code 的代码库地址,在代码库 --> Settings (设置)---> WebHooks 中
- 2. 将流水线复制出来的 WebHooks 添加 URL

		为了更好的为您提供服务, 云效 邀请您使用持续交付相关功能。云效结合ECS、EDAS等服务为您提供完备的发布、部署、测试全研发流程,大大提升您的研发效率. 了解更多>
ñ	首页	
	设置	
e,	WebHooks	WebHooks
		WebHooks用于在项目发生相关事件时通知外部服务器。
¢;	Services	
•	分支	链接 http://example.com/trigger-ci.json
	«	
		約次 ● 推送事件
		1位区に注意の作用では1000年間の1月1日の1日の1日の1日の1日の1日の1日の1日の1日の1日の1日の1日の1日の1
		□ 你就是这事件 新品质性法会使时接触发出链接
		增加新评论时将触发此链接
		□ 缺陷事件
		缺陷创建时将触发此链接
		□ 合并请求事件
		合并请求创建时将触发此链接
		□ 构建事件
		构建状态改变时将触发此链接
		- シート
		18 M WEBLOOKS

3.2.3.5. 码云 Gitee 中配置 WebHook

前序操作

- 1. 在 Flow 流水线,点击编辑流水线,点击编辑代码源。
- 2. 开启代码提交触发监听,并将对应的 WebHook 复制出来。

在码云 Gitee 配置 WebHook

1. 访问 码云 Gitee的代码库地址,在代码库 --> Settings (设置)---> WebHooks 中

2. 将流水线复制出来的 WebHooks 添加 URL

3.2.3.6. Github 中配置 WebHook

前序操作

- 1. 在 Flow 流水线,点击编辑流水线,点击编辑代码源。
- 2. 开启代码提交触发监听,并将对应的 WebHook 复制出来。

在 Github 配置 WebHook

- 1. Github 的代码库地址,在代码库 --> Settings (设置) ---> Webhooks中,点击 Add Webhook
- 2. 将 Flow 流水线上复制的 WebHook 地址,填入 Payload URL
- 3. 将 Content Type 选择为 "application/json"



3.2.3.7. Gitlab 中配置 WebHook

前序操作

- 1. 在 Flow 流水线,点击编辑流水线,点击编辑代码源。
- 2. 开启代码提交触发监听,并将对应的 WebHook 复制出来。

在 Gitlab 配置 WebHook

- 1. 访问Gitlab的代码库地址,在代码库 --> Settings (设置) ---> Integrations 中
- 2. 将流水线复制出来的 WebHook 添加 **URL**



3.2.3.8. Coding 中配置 WebHook

前序操作

- 1. 在 Flow 流水线,点击编辑流水线,点击编辑代码源。
- 2. 开启代码提交触发监听,并将对应的 WebHook复制出来。

在 Coding 配置 WebHook

1. 访问Coding的项目地址,4 2. 将流水线复制出来的 Webl	E项目> 项目设置> 开发者选项> WebHook中 Hook 添加 URL							
☆ > 示例项目 → > 设置	/ 开发者选项 / WebHook / 新建 WebHook			搜索		[]	ġ ¢	~
← 项目设置	新建 WebHook							
A 项目与成员	URL	令)	牌					
ı 项目协同	请输入 URL		非必填					
<♪ 开发者选项	内容类型	内	容格式					
	application/json ~	c	coding			~		
	 広听事件 ② 项目协同 项目协同通知迭代、軍項 (需 求、任务和缺陷)相关事件 任务 (旧版) 任务 (旧版) 创建、开关、编 辑、指派、标签等操作 学 Push Git push 到仓库 事件开关 开启声事件触发时将发送请求. 	章、删	 成员 ·项目内 设置 ✓ MR 合并请 	成员的新增、删除、权限 求进行创建、关闭	 制品/ 制品发 持续集 止、穷 	车 布、更新版 <u> </u> 	* 触发、停	
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	新建 WebHook 取消							

3.2.3.9. 通用 Git 中配置 WebHook

#### 前序操作

- 1. 在 Flow 流水线,点击编辑流水线,点击编辑代码源。
- 2. 开启代码提交触发监听,并将对应的 WebHook 复制出来。

#### 在通用 Git 配置 WebHook

- 1. 访问Git的代码库地址,在代码库 --> Settings (设置) ---> Integrations 中
- 2. 将流水线复制出来的 WebHook 添加 URL

← → C 🔒 gitlab.com/richrenf/hello-world	d/-/settings/integrations			Θ
🦊 GitLab Projects 🗸 Groups 🗸 Activity	Milestones Snippets Analytics 😰 🗸	C ~ Search or jump to Q D 🏌 C	≌ <b>0</b> ~	
H hello-world	richrent > hello-world > Integrations Settings			
û Project	Integrations	URL		
C Repository	Webhooks can be used for binding events when something is happening within the project	http://example.com/trigger-ci.json		
Issues	poloce.	Secret loken		
13 Merge Requests		Use this token to validate received payloads. It will be sent with the request in the X-Gitlab-Tok	en	
√g CI/CD		Trigger		
Operations		Push events This URL will be triggered by a push to the repository		
🖰 Packages		Branch name or wildcard pattern to trigger on (leave blank for all)		
🔲 Wiki		Tag push events This URL will be triggered when a new tag is pushed to the repository.		
🔏 Snippets		Comments		
Settings		This URL will be triggered when someone adds a comment		
General		Confidential Comments This URL will be triggered when someone adds a comment on a confidential issue		
Members				
Integrations		This URL will be triggered when an issue is created/updated/merged		
Repository		Confidential Issues events This URL will be triggered when a confidential issue is created/updated/merged		

# 3.2.4. 配置 Jenkins 源

Poll SCM		Ø	
勾建			
Execute shell		×	
Command e	cho hello world > 'a.txt'		
Se	e the list of available environment variables	高級	
增加构建步骤 👻			
构建后操作			
Archive the ar	tifacts	×	
用于存档的文件	‡ a.txt	0	
		高级	
增加构建后操作步骤			

Flow 支持将您的 Jenkins 服务配置为流水线源,当流水线触发运行时,Flow 会根据您的配置下载 Jenkins 任务的构建产物并执行后续任务。你的构建产物需要在 Jenkins 的任务 中先进行归档操作。

在流水线编排页,点击"添加源",在新建流水线弹框中选择"Jenkins",输入您的 Jenkins 公网地址、用户、密码,完成账号绑定。

▼ 源(0)	十 阶段1		新建流水线源	×	
暂未设置,点击 添加	<b>D</b> -\$	代码源 Jenkins	请先绑定Jenkins账号		
▼ 变量 (0)	+				
暂未设置,点击 添加		1			
			why ①	•	
			<b>堺</b> 定		
				取消 保存	
₽ 触发设置					G

选择 Jenkins 任务 Job 名称和对应默认版本,完成配置。

圖 流水线编排 流水线 2020-0	04-07				保存保存并运行
▼ 源(0)	+ 801	91	新建流水线源	×	
暂未设置,点击 <b>添加</b>	0-	代码源 	久 账号: p://5a * > =	绑定新账号 解除规定	
▼ 变量 (0)	g +	1		~]	
暂未设置,点击 添加			InstBuild	~	
				_	
				_	
				取消保存	
					G

请确保您的 Jenkins 服务配置了公网出口。

Jenkins 🕨	ENABLE AUT	O REFRESH
e New Item		
Reople	Manage Jenkins	
Build History		
🏕 Manage Jenkins	New version of Jenkins (2.249.2) is available for download (changelog). Or Upgrade Autom	atically
🍓 My Views		
New View	Warnings have been published for the following currently installed components. Configure which of these warnings are	shown
	Jenkins 2.204.2 core and libraries:	
Build Queue 📼	Multiple security vulnerabilities in Jenkins 2.251 and earlier, LTS 2.235.3 and earlier	
No builds in the queue.	Multiple security vulnerabilities in Jenkins 2.227 and earlier, LTS 2.204.5 and earlier Multiple security vulnerabilities in Jenkins 2.244 and earlier, LTS 2.235.1 and earlier	
Build Executor Status		
1 Idle	Configure System	
2 Idle	Comigue giocal octango and parto.	
	Configure Global Security	
	Secure Jenkins; define who is allowed to access/use the system.	
	Slobal Tool Configuration	
	Configure tools, their locations and automatic installers.	
	-	
	Reload Configuration from Disk	
	Discard all the loaded data in memory and reload everything from tile system. Useful when you modified config files directly on disk.	
	🚄 Manage Plugins	
	Add. remove disable or enable pluging that can extend the functionality of lenking	

# 3.2.5. 配置 Flow 流水线源

#### 原理说明

在实际应用过程中,存在构建环节和部署环节分离的场景。

比如,需要对开发环境、测试环境、生产环境分别进行部署,但是每个环境的部署频率是不同的。如果将多个环境的部署环境配置在同一条流水线,则会存在多次重复、取消 的情况。如果将多个环境配置成多条流水线,则每个环境在部署过程中会出现重复构建的现象。

因此,Flow 提供了流水线之间触发的能力,您可以将流水线 A 设置为流水线 B 的触发源,当流水线 B 被触发运行时,Flow 会根据您的配置下载源流水线 A 中的构建产物并执行后续任务。

在流水线编排页,点击"添加源",在新建流水线弹框中选择"Flow 流水线",选择源流水线 和默认版本,当流水线被执行时就会下载源流水线默认版本的构建产物。

#### ♫ 注意

流水线 A 若想下载流水线 B 的构建产物,B 需要将构建产物进行归档,归档完成后流水线 B 的构建产物会和流水线运行记录绑定。可查看归档至云效公共存储空间。 如果使用 Packages 通用制品仓库管理制品文件,上传至 Packages 通用制品仓库并不会将制品归档,因此如果想实现多流水线之间的制品传递,需要在上传完 Packages 后 对制品文件进行归档操作。

同时, 流水线之间也可以自动触发, 打开触发事件开关, 可选择自动触发的事件。

< 返回 触发流水线		基本信息 流程配置 触发设置 变量和缓存	仅保存	将并运行
源	命令	新阶段	编辑流水线源 ①	>
⑦ 源流水线 2020-07-16	● Ģ 执行命令(预发)	◎ 新的任务	源流水线 ⑦	
⊩ lastSuccessBuild			源流水线 2020-07-16(1017319)	~
十添加代码源			默认版本 ⑦	
			最后一次成功执行	~
			触发事件	
			道听源流水线事件触发	
			鮑发事件 🗌 成功 🗌 失敗 🗹 取消	

配置完成后,点击保存运行,会下载源流水线的构建产物至工作区;若要将构建产物进行部署,需要将工作区的构建产物上传之后,才可选择部署。



#### 示例

场景说明: 流水线 A 生成产物, 流水线 B 获取构建产物后, 发布至 ECS。 流水线 A 中进行 Java 构建, 并对构建产物进行归档操作。

く 返回     流水线A	基本信息	<b>流程配置</b> 触发设置 变量和缓存	仅保存保存并运行
源	构建	新阶段	编辑 谊 🛛 🕹 🕹
p flow-example/spring-boot ▶ master +添加代码源	ら Java 构建上传	● 新的任务	<pre>1 # moven build default command 2 mvn -B clean package -Dmaven.test.skip=true -Dautoconfig.skip 3 # gradle build default command 5 # ./gradlew build 6 7 # ant build default command</pre>
			<ul> <li>- 构建物上传</li> <li>步骤名称</li> </ul>
		1	<ul> <li>内建物上传</li> <li>上传方式</li> <li></li></ul>
			□ 制品中包含打包路径的目录 ⑦

流水线 B 将 A 配置为流水线源,将下载到的构建产物重新归档,并部署至 ECS。

く 返回 - 流水线B	基本信息	<b>流程配置</b> 触发设置 变量和缓存		仅保存 保存并运行
源	执行命令	新阶段	编辑流水线源 ⑪	×
중 流水线A ト lastSuccessBuild	执行命令	<ul> <li>新的任务</li> </ul>	源流水线 ③ 流水线A(1474357) 默认版本 ③ 最后一次成功执行 触发事件	
〈 返回   流水线B	基本信。	思 <b>流程配置</b> 触发设置 变量和缓存		仅保存保存并运行
源	执行命令 🖉	新阶段	编辑 ⑰	×
济水线A	ゆりまた	● 新的任务	任务步骤 ③ 添加步骤	
	● 并行任务		- 构建物上传	٥

十添加代码源		步骤名称	
		构建物上传	
	(1)	上传方式 <ul> <li>」均档至云效公共存储空间(供后续任务拉取并部署</li> <li>企业私有通用制品仓库(制品将上传Packages统一制品名称 ⑦</li> <li>Artifacts_\$(PIPELINE_JD)</li> </ul>	)⑦ -管理) ⑦
		打包路径 ⑦ default ] 制品中包含打包路径的目录 ⑦	]⊕⊝
		任务插件	
		₩ 208.00198811+	

〈 返回    流水线B	基本信息 流程配置	触发设置 变量和缓存	仅保存	保存并运行
执行命令 🖉	部署	新阶段	编辑 ⑪	×
			任务名称	
<ul> <li>●</li> <li>♀</li> <li>执行命令</li> <li>●</li> </ul>	<i>♀</i> 主机部署	● 新的任务	主机部署	
● 并行任务			✔ 部署时下载制品	
		$\frown$	制品	
		(1)	制品名称.Artifacts_\${PIPELINE_ID}	
			主机组	新建主机组
			新的主机组	~
			①当前主机组已有关联流水线,请谨慎操作	
			部署配置	
			下载路径 ②	
			/home/admin/app/package.tgz	
			执行用户 ⑦	
			root	
			您可以在部署脚本中直接使用环境变量,无需 Base64解码处理。但是如果你的环境变量中包符,您将无法正常使用您的环境变量,查看帮	再进行 2含特殊字 ³ 助
			部署脚本(?)	27

# 3.3. 编排流水线

通过流水线编排,你可以定义持续交付的自动化流程,将构建,部署,测试,管控等组件化能力进行编排和串通,完成从开发到上线过程的自动化流程。

Flow 流水线提供了以下编排能力:

- 阶段: 在流水线中需要按顺序执行的一组任务的集合, 一个阶段可以是手动运行也可以是自动运行的。阶段之间串行执行;
- 任务:在阶段中具体需要完成的动作,任务之间可以串行执行也可以并行执行,目前任务包含:代码扫描,单元测试,构建,部署,合并代码,人工审核等功能。
- 步骤:步骤作为 Flow 最底层的核心能力,所有的流水线都是通过步骤的编排组合而成的。

#### 增加阶段

在流水线编排页面,点击"+"可以添加阶段,点击后,会生成一个新的阶段,并默认生成当前阶段中的一个流水线任务,通过选择任务模版,可以快速创建当前阶段中的流 水线任务。



任务新建后,点击任务卡片,可以通过任务抽屉进行任务配置。

in more search juvu	四音 加以大致 2018-08-0	2		VR 17 VR1271451	
源	新建源	阶段1	阶段2	Java代码扫描	
O richrenf/hello-wo	rld	+ B laveff Et		1 任务名称	
}∞ master				Java代码扫描	
		- ひ Maven単元	测试	任务步骤	
		及 Java代码	目描	● 添加步骤	
				▶ Java代码规约扫描	
				任务插件	
				<ul> <li>添加描件</li> </ul>	
				删除任务	
				该任务将被彻底删除,注意这是不可逆操作,任务下所有数据将会删除	
				副 除	

#### 在阶段中添加任务

#### 增加并行任务

在流水线编排页面,hover到一个阶段,会出现"+并行任务",hover并点击后,可以选择任务模版,完成创建。



#### 添加串行任务

在流水线编排页面, hover 到一个任务, 会出现 "+", 既可以在该任务的前后添加串行任务, 可以选择任务模版, 完成创建。

☐ 流水线编排 流水线 2019-0	08-27				保存保存并运行
源 6	新建源 阶段1		阶段2	阶段3	新的阶段
<ul> <li>aonecloud-pre-rdcuitest/n</li> <li>master</li> </ul>	ew	<ul> <li>● Java代码扫描</li> <li>● #行任务</li> </ul>	■	12	部署 ● ● 新的任务
	Ģ	Maven单元测试 ③ 并行任务			
*					

# 3.4. 触发流水线运行

飞流支持不同类型的触发策略,你可以根据自己的使用场景,选择适合的方式来触发流水线运行

当前支持的触发策略包以下几种:

- 手工触发,用户可在流水线上点击"运行"来进行手动执行。
- 代码提交触发
- 定时触发
- Webhook 触发

## 3.4.1. 触发流水线运行

Flow 支持不同类型的触发策略,你可以根据自己的使用场景,选择适合的方式来触发流水线运行 当前支持的触发策略包以下几种:

- 手工触发,用户可在流水线上点击"运行"来进行手动执行。
- 代码提交触发
- 定时触发流水线
- Webhook 触发流水线

可以通过流水线触发信息,查看流水线是如何触发的

最近运行 ì	运行历史						∅ 编辑	🗩 运行 🛛 🗶 💷
运行记录	状态	代码信息		运行详情	触发信息		持续时间	\$ V
#13	0	master	3 >	命令 ●	流水线定	的自动触发	11秒	查看
#12	•	master	3 >	命令 ●	流水线定	的自动触发	9秒	查看
#11	0	master	3 >	命令 ●	流水线定	的自动触发	12秒	查看
#10	0	master	3 >	命令 ●	在 spring	g-boot-example 提交代码	. 10秒	查看
#9	0	master	3 >	命令 ●	1 在 spring	g-boot 提交代码至 maste	10秒	查看
#8	0	master	2 >	命令 ●	流水线定	的自动触发	9秒	查看
#7	0	master	2 >	命令 ●	sm) sm)	<b>ytest</b> 在 smy-test 完成合	10秒	查看
#6	0	master	2 >	命令 ●	s smj	<b>ytest</b> 在 smy-test 新建/更	8秒	查看
#6				命令			4.0.El	*=

## 3.4.2. 定时触发流水线

### 定时触发

通过定时配置,可以周期性的自动触发流水线的执行。在编辑流水线时,可以点击定时运行,然后配置定时配置。

	● 新建源	测试		构建		新的阶段	
		<b>9</b>	Java代码扫描			● 新的任务	
			定时开关		已开启 🌑		
未調	EWS		日期选择 2 星期日 2 星期五 触发方式	星期一 🗌 星期二 🗌 星期三 星期六	星期四		
			<ul> <li>周期触发</li> <li>00:00 ~ 0</li> </ul>	单次触发 11:00 ① 间隔 30分钟	~		
			_	B	消 确认		
	1)						

# 3.4.3. Webhook 触发流水线

### ↓ 注意

若你希望通过代码源触发流水线运行,请通过 流水线代码源->开启代码提交触发 完成配置。Webhook 触发用于代码源以外的第三方工具或自有系统触发流水线运行。

#### Webhook 触发设置

通过 Webhook 触发,用户通过任意第三方系统触发流水线运行。在编辑流水线时,可以点击触发设置,开启 Webhook 触发。

〈 返回   流水线 2021-10-25	基本信息 流程配置 <b>触发设置</b> 变量和缓存	保存并运行
Webhook <b>触发</b> 定时触发	Webhook <b>触发</b> 外部系统通过Webhook将环境参数传给流水线并触发运行, 查看文档	
	将Webhook复制到你的触发器(代码仓库触发请通过 流水线代码源->开启代码提交触发 完成配置) http://flow-openapi.aliyun.com/pipeline/webhook/4mVS Xs03TXOu	

#### Webhook 触发流水线运行

在启用流水线 Webhook 后可以使用以下命令直接触发流水线运行

curl --header "Content-Type: application/json" --request POST --data '{}' https://flow-openapi.aliyun.com/pipeline/webhook/your-webhook

#### 触发流水线运行的同时支持将外部环境变量。可以按以下操作配置: • 在流水线变量中,配置你希望传递的变量值,如

变量		<b>变量</b>		/= >+ 10 66 /1	何外印体田注	此亦是 大美子松	
通用变量组		进过正义坏现变重头	现流水线过程正制化,可以任执	们过程的性	的阶段使用这	些变重,宣有又怕	
缓存		字符变量 🔍 🔍 🕤 新建变量					
		参数	默认值	私智	密模式⑦	运行时设置⑦	操作
	$(\mathbf{I})$	abc	123	0			Ŵ
		运行选择变量					🔍 🔮 新建变量
		参数	默认值		选项		操作
				没有数据			

● 开启 Webhook 触发,调用并使用 dat a 信息传递触发参数,修改运行时变量值,如

curl --header "Content-Type: application/json" --request POST --data '{"abc": "456"}' https://flow-openapi.aliyun.com/pipeline/webhook/your-webhook

• 可以看到, 示例中的环境变量 abc 初始化为 123, 触发后的流水线变量值变化为了 456

最近运:执行命令			×	; <b>请</b> 三菜单
1 ⊘ 运行	执行命令(1s)		न	载
<ul> <li>♥ 申请运行</li> <li>■ 请理工作</li> <li>■ 請理工作</li> </ul>	际境(2s) [executionStep begins at 2021-11- 【(Os) [INF0] PATH=/usr/local/sbin:/usr/ [11:54:32] [SUCCESS] 获取 Credenti	30 11:54:32] local/bin:/usr/sbin:/usr/bin:/sbin:/bin .al 成功		
<ul> <li></li></ul>	F(1s)         [11:54:32] [WARNING]未指定工作路径,           1s)         [11:54:32] [INF0] 执行步骤	使用默认路径/root/workspace/code		
● 缓存上传(	15) [11:54:32] [TMF0] PIPELINE_TD=152 [11:54:32] [TMF0] PIPELINE_TMHE=満 [11:54:32] [IMF0] BUILD_NUMBER=1 [11:54:32] [IMF0] BUILD_NUMBER=1 [11:54:32] [IMF0] WORK_SPACE=/roo [11:54:32] [IMF0] PR0JECT_DIR=/roo [11:54:32] [IMF0] PLUGIN_DIR=/roo [11:54:32] [IMF0] PLUGIN_DIR=/roo [11:54:32] [IMF0] 出合可規的大計文 [11:54:32] [IMF0] 出合可規的大計文 [11:54:32] [IMF0] 出行用户命令 [11:54:32] [IMF0] 出行用户命令 [11:54:32] [User Command] + echo [11:54:32] [SUCCESS] 步骤运行成功	9451 1d6f5032838333284f91c2 t/workspace od/workspace/code t/plugins 239833 准备		

对于任意第三方系统可以通过以上方式触发流水线运行

#### ACR 镜像 Push 触发流水线

#### 使用触发器触发流水线

对于使用了阿里云容器镜像服务(ACR)的用户可以通过 ACR 的触发器在镜像更新后自动触发流水线运行。

如下所示,在 ACR 镜像仓库触发器设置页面,使用飞流的 Pipeline Webhook 作为触发器 URL,并根据实际情况选择触发策略:

三(つ阿里云)	☆ 工作台		Q 授索 費用 工单 ICP 备案 企业 支持 App 🖸 🎝 🕁 🕜 简体 📀
容器镜像服务 / 实例列表 /	/ <b>镜像仓库</b> / 触发器		
← cpj-repos	sitory		华东1(杭州)   公开   自动构建仓库   🗸 正常 🌲 部署
基本信息	创建 ① 触发器使	创建触发器	×
构建仓库授权	名称	类型	镜像推送触发器
触发器	flow_trigger	• 名称	flow-trigger 12/32 可记录 修政 制除
镜像版本			长度为 1-32 个字符,只支持字母、数字、下划线
镜像同步		* 触发器 URL	http://flow- openapi.aliyun.com/pipeline/webhook/4m /m0DK // 68/1024
		* 触发方式	<ul> <li>● 全部触发 ○ 表达式触发 ○ Tag触发</li> <li>● 合部触发 ○ 表达式触发 ○ Tag触发</li> </ul>

配置完成后,可以通过 Push 镜像到该仓库触发流水线运行。

#### 在流水线中获取触发镜像

为了获取当前触发流水线的镜像实例,用户需要在流水线变量中添加名为 ACR_PUSH_IMAGE 并设置默认值,当 ACR 触发器触发该流水线构建时,该变量会自动映射为 Push 镜 像的完整地址,从而在后续的步骤中通过该变量完成镜像相关的其它操作:

<b>变量</b> 通用变	量组	<b>变量</b> 通过定义环境变量实现流水线过程定制化,可以在执行过	拉程的任何阶段使用	]这些变量, <b>查看</b> 文	【档	
缓存		字符变量			Q 😁 新	「建变量
	新建变量				×	
	参数 ACR_PUSH_IMAGE ● 新建变量	默认值 registry.cn-hangzhou.aliyuncs.com/coj-	私密模式⑦	运行时设置⑦	操作 ①	建变量
		没有	订数据	取消	添加	

# 3.5. 查看最近运行

访问流水线列表,会在列表中展示当前流水线的【最近一次运行结果】的状态和运行缩略图。

☆ 我的星标		我的星标				Fl 🛛 🚯 🕷	<b>ī建流水线</b>
<ol> <li>     我参与的   </li> </ol>		流水线名称	最近运行状态	最近运行阶段	运行人	最近运行时间	
♀ 全部流水线	(1)-	▶ 飞流持续集成	#1 - 🥥	测试 构建	🥏 杨超越	2019-09-24	*
设置信息		行云日常环境持续交付	#1 - 😒	測试 构建 部署 ●○○	🥏 杨超越	2019-09-24	*
O LINE		飞流生产环境持续交付	#15 - 🤨	測试 部署 合并主干 新増阶… ● ○ ○ ○	刘 刘昊然	2019-09-24	*
		行云持续集成	#1 - 😢	週试 构建 • •	● 吴彦祖	2019-09-23	*
		飞流日常环境持续交付	#14 - 🥑	测试 构建 日常部署	🌝 杨超越	2019-09-24	*
		所思日常环境持续交付	#2 - 🥑	构建 部署	🌸 刘德华	2019-09-23	*
		飞流预发环境持续交付	#12 - 🤌	测试 构建 部署 。●	💿 周杰伦	2019-09-23	*
		行云生产环境持续交付	#87 - 🥝	測试 部署 合井主干 新増阶 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	🔊 刘吴然	2019-09-24	*

点击某条流水线,会展示当前流水线最近一次的运行结果,包括以下内容:

- 当前流水线的触发信息上下文,包括:当前流水线运行状态,运行人,开始时间,运行时间,流水线执行锁对应的代码信息等
- 当前流水线的具体执行阶段,任务和对应的操作

最近运行 运行历史			● 运行	三菜
#10 🥑 等待中 🛞 I	观消 測试	构建	日常部署	
<ul> <li>C-&gt; flow/flow-engine</li> <li>§∞ master</li> <li>-∞ 7027078</li> </ul>	<ul> <li>● Java代码规约扫描</li> <li>4 0 0 4</li> <li>总数 回答 严重 一般</li> </ul>	◎ 构建 application.tgz	日常部署 部署暂停,请进入部署详情确认后,点击 继续部署	
	同扫描报告 []日志 44秒	[] 日志 43秒	✓ 部署详情 14分9秒	
	· Maven単元激试               ①             · 100%             · 通过             · 通过			
运行人 💿 杨月	日起			
开始时间 2019-09-24 21:00	06			

# 3.6. 批量操作

- 流水线列表,支持批量操作
- 运行
- 删除
- 添加成员
- 删除成员

☆ 我的星标	我参与的				₽Ų	新建流2
. 我参与的	■ 已选择3项	行 ① 删除	风 添加成员 风 删除成员			¢
♀ 全部流水线	流水线 2019-11-08	-	-	-	-	☆
	☑ 流水线 2019-11-07	#1 - 🥑	代码扫描	🕐 石磊	2019-11-08	
	流水线 2019-10-22	#2 - 😣	測试 构建 部署	🛞 石磊	2019-11-08	
	☑ 流水线 2019-10-22	#1 - 🙁	测试 构建	🛞 石磊	2019-11-08	
	☑ 流水线 2019-10-22	#1 - 😣	測试 构建 部署 ● ○ ○	🛞 石磊	2019-11-08	
	预发环境测试流水线	#11 - 🥝	测试 构建 部署	(1) 石磊	2019-11-08	

# 3.7. 流水线分组

Flow 支持通过分组分类管理流水线,同时支持在分组上设置角色权限,帮助您更方便的管理企业流水线。

### 流水线分组

您可以创建多个流水线分组,并将企业的流水线移动至对应分组下进行管理。

							邀请 🌔 体验透镜	+ 99 <del>1</del> ?
⇒ 我的流水线	۲	分组1				🗶 💿 🛛 新建	流水线 Q ☴ 文	
》 全部流水线			流水线名称	最近运行状态	最近运行阶段	运行人	最近运行时间	۵
<b>流水线分组</b>	+		流水线 2021-01-29					☆ (≥)
分组1			Demo库流水线样例	#3 - 🥑	命令 ●	🗙 光晕	2021-01-27 15:40	☆ ⊙ …
分组2			流水线1					☆ (≥)
		1						
ର <b>ା ଜେଲ</b>		4	1 8条/面					

分组支持操作 创建/删除/编辑,需要具有流水线分组管理权限。默认企业拥有者和管理员拥有该权限。

〈 返回	1	全业设置	>
工具箱	角色设置		新建角色
& 通用变量组	<b>一 拥有者</b> 我的角色 ~	着管理员 个 成员	默认角色 ~
□ 标签管理			
🔁 授权管理	▶ 外部成员 ✓	<ul> <li>企业Maven配置管理</li> <li>所有管理权限</li> </ul>	
🗐 主机组管理		<ul> <li>○ 流水线组管理</li> </ul>	
※ Kubernetes集群管理	(1)	☑ 所有管理权限	
应 流水线模板管理		✓ Kubernetes集群管理	
🖂 任务组管理		<ul> <li>✓ 创建kubertenes集群</li> <li>✓ 所有管理权限</li> </ul>	
123 步骤管理		保存	
2 构建集群管理			
管理员设置			
<u> 风</u> 角色设置			

### 分组权限

流水线分组支持设置 成员/部门 权限。

- 在分组上设置成员角色,分组下的流水线自动继承分组权限。
- 在分组上设置为所有权限,还可以对分组进行管理操作(分组内添加成员 or 添加流水线)。
|          |   |         |                         |                                               | 邀请    | 🧯 体验透镜 | + 99+ ? |
|----------|---|---------|-------------------------|-----------------------------------------------|-------|--------|---------|
| ⋚ 我的流水线  | ۲ | 分组1     |                         | X 😒                                           | 新建流水线 | o° ≞↑  |         |
| 会 全部流水线  |   | □ 流水线名称 | 成员列表                    | Q。 <table-cell-rows> 新増成员 ─</table-cell-rows> | 运行人   | 创建人    | 最近运行时间3 |
| 流水线分组    | + | _ 流水线1  | 测试部门                    | 默认为 查看 ~                                      |       | 🗶 光晕   |         |
| 分组1      |   |         | <b>光晕</b><br>成员隶属于:测试部门 | 所有权限 ~                                        |       |        |         |
| 分组2      |   |         | smytest<br>成员隶属于: 测试部门  | 查看 ~                                          |       |        |         |
| ① 未分组流水线 |   |         |                         |                                               |       |        |         |
|          |   |         |                         |                                               |       |        |         |
|          |   |         |                         |                                               |       |        |         |
|          |   |         |                         |                                               |       |        |         |
|          |   |         |                         |                                               |       |        |         |
|          |   |         |                         |                                               |       |        |         |

具体权限规则可查看:<mark>流水线分组权限</mark>

# 4.代码管理 4.1. 分支模式

#### 什么是分支模式

Flow对分支模式提供了强有力的支持:用户可以只需要关心集成和发布哪些 feature 分支,而对 release 分支创建和管理、分支合并等一系列工作,可以托付给 Flow 完成。 本节内容详细介绍分支模式下,各(类)分支的使用方式。

#### master 代表最新发布版本

一般情况下, master 分支代表最新发布版本。当需要最新发布版本的内容时, 直接取分支末端即可。 不论其他哪类分支,都建议一般从 master 分支创建,并且经常从 master 分支合并,以便跟上"潮流",减少将来集成时的各种问题,比如代码合并冲突。 每当软件正式发布前,系统会确保它基于 master 最新。 每当软件正式发布后,系统会把相应内容合并回 master,以便让 master 分支始终代表最新发布版本。 一般来说,使用者不要直接"写"东西到master分支。把"写"的工作交给系统适时自动完成。

#### 在各 feature 分支上开发

一条 feature 分支(又称变更分支、开发分支),通常用来承载一个缺陷的修复,或者一个需求(如果不是很大的话)的开发,或者任务分解后一个任务的开发。 一般来讲,基于 master 分支最新版本创建 feature 分支。然后在 feature 分支上开发、测试,直到这个 feature 功能完成,质量 OK,准备好去集成和发布。

#### release 分支上的集成

release 分支用于集成和发布。基于 master 分支最新版本创建一条 release 分支,然后把想要集成的各条feature分支合并到这条release分支,进行部署和测试工作。 如果有新的 feature 分支要加入本次集成,那就把它也合并进这条 release 分支,然后再次部署并测试。

如果测试发现问题,就到 feature 分支上修复,然后把它再次合并到 release 分支,把修复带到 release 分支。

当然如果一个 feature 的问题太多太大,那干脆就放弃它。也就是说,新建一条 release 分支,把其他 feature 分支都合并过去,唯独不再合并这条 feature 分支。 就像 master 分支一样,release 分支也是由系统自动管理的。 使用者不要直接在上面改代码,代码修改请总是在 feature 分支完成。

#### release 分支上的发布上线

当 release 分支上的质量足够好,本次想上线的功能也都具备之后,就要考虑发布上线的问题啦。如前面讲的,发布上线前,会确保它基于基础分支(常见的如 master)最 新。而发布后会把 release 分支合并回 master,让 master 代表最新发布版本。

以上几节介绍的内容,见下图:



### 多个环境/流程时

假定要想集成发布上线,要经过日常测试环境上的测试这个流程,还要经过预发环境上的测试这个流程,那么两个流程用一条 release 分支就有些不合适。因为两个流程可能 同时在测不同的 feature 分支集合。

分支模式用这个办法避免这个问题:每一个测试环境,也就是每个流程,关联它自己的 release 分支。日常测试、预发测试这两个环境(也就是两个流程),分别关联两条 release 分支。这样就不会相互影响。推而广之,为正式运行环境,也对应一条release分支。也就是说,每个环境都有对应的 release 分支。

当把集成成果从一个环境传递到下一个环境时,就是把一个环境下已合并到一起的 feature 分支,再往另一个环境对应的 release 分支上合并一遍……这么做有点儿笨。系统 实际的做法是,基于 master 分支创建另一个环境对应的 release 分支,然后把前一个环境对应的 release 分支合并到新的 release 分支上。

本节介绍的内容,对应下图:



以上就是关于分支模式这种研发模式的原理性介绍,以下我们看一下如何在流水线中使用分支模式。

## 如何在流水线中使用分支模式

## 编排流水线

流水线的新建方式其他流水线相同,当新建流水线时选择了「开启分支模式」,就会自动创建包含【分支管理器】的分支模式流水线。

- 1. 新建流水线
- 2. 添加代码源,以使用「云效Codeup」为例,选择代码库,选择「开启分支模式」,然后点击「添加」



3. 添加完成后,在「流程配置」页面可以看到第一个阶段「分支管理器」。在分支管理器中设置基础分支,基础分支默认是 master。基础分支是发布分支的创建来源。发布 分支从基础分支创建,然后合并运行分支。「分支管理器」只能是在第一个阶段配置,且在这个阶段不能配置并行任务。

<ul> <li>公 返回 進井枝 2020-08-11</li> <li>3</li></ul>	基本値 東程 単次设置 文単印紙存 株行命令 単理語 ● 多 単行命令 22
---------------------------------------------------	-------------------------------------------------

## 运行流水线

流水线配置完成后,就可以开始运行了。

1. 在运行配置中,添加运行分支

く 返回 瀧水銭 2020-06-11	基本信息	流程配置 触发设置 变量和缓存 ————————————————————————————————————	×	仅保存 保存并
0011d8903/testflow0609	代码源 分支 https://codeup.aliyun.com/5	状态 #95bf7fe17c0e0001fd8903/testflow0609.git ● 添加运行分支	操作 【 <b>务</b> 【暂无运行分支	
8,637104.0k +			取消 运行	

2. 进入添加运行分支对话框,选择运行分支。若在代码源选择的其他代码库,这里输入运行分支。

< 返回 流水线 2020-06-11		本信息 流程配置 触发设置 变量	和缓存			仅保存	保存并运行
2	2	添加运行分支		×			
001fd8903/testflow0609 b	28				18		
⊳ master	https://codeup.aliyun.com/5e95						
	运行分支 br001			~			
十添加代码源			返	回 満加			
< 返回 進水域 2020-06-11	<u>*</u>	本信息 <b>流程配置</b> 触发设置 安康	和總存			汉保存	保存并运行
<ul> <li>( 返回) 進水线 2020-06-11</li> </ul>	¥ 2	本信息 <b>美程配置</b> 触发设置 变量 运行配置	40歳79	×		汉保存	<u>廉</u> 存并运行
< 返回 通水线 2020-06-11 差	<del>لا</del> ۳	本信息 <b>流程配置</b> 触发设置 受量 <b>运行配置</b> 分支	10879 14.5	×		17.68.75	像得并运行
< 返回 道水地 2020-06-11 源 •	基 1 【代初週 0e0001fd8903/testflow0609	本信息 <b>流程配置</b> 始末设置 安물 <b>运行配置</b> 과文 br001	8067 18.5 19.5 19.5 19.5 10.5	×	£\$	<b>288</b>	<b>保存并</b> 违行
<ul> <li>第二章 面状线 2020-06-11</li> <li>通 0.011/dB903/testflow0600 ト ト master</li> </ul>	基 パ 「代現源 」	本信息 <mark>発育配置</mark> 松水设置 交量 道行配置 かえ broot ・ 添加回行分支	10歳存 14.0 16集成	× Hitti Ū		仪保存	保存并進行
<ul> <li>( 返回) 進水域 2020-06-11</li> <li>第</li> <li>001/d8903/Restflow0609 ト ト master</li> <li>+ 流館代码源</li> </ul>	至 1 (代現源 ee0001fdB903/testflow0609	本仮思 <u>教育配置</u> 松光以至 安重 通行配置 分支 br001 ・ 添加油行分支	R067 10.5 行集成 10	× Heft T	E.#	17.94 <b>1</b> 7	象有并运行
<ul> <li>         ( 近回 進水线 2020-06-11         </li> <li> </li></ul>	基 代码源 000001/d8903/testflow0609	本信息 <mark>決容配置</mark> 秋米以直 交量 运行配置 分支 br001 ● 添加运行分支	1087 142 193.4	× Hr T M	£8)	22課程	象存并运行
<ul> <li>《 返回 進水线 2020-06-11</li> <li>■</li></ul>	来 【行玩派 	本信息 <b>美丽配度</b> 触激设度 安星 运行配置 动变 boot 企 添加油行分变	0007 455 193.4 10	X Hrt Û	£\$	0.84 0	(\$P\$7.#12.0)
<ul> <li>《 返回 激光线 2020-06-11</li> <li>◎0011dB903/testflow0600 ト ト master</li> <li>+ 添加代码源</li> </ul>	2 【代現源 …0e0001fd8903/testflow0609	本瓜島 <u>東和配置</u> 秋米役画 安曇 运行配置 かえ boo1 ・ 添加面行分支	0067 X.S. 199.4 R	X Brit Tr M Brit Tr M Brit Tr M	5.5	议保养	<b>除守井运</b> 行
<ul> <li>《 返回 》 迎大线 2020-06-11</li> <li>◎0011d8903/testflow0609 ▶</li> <li>▶ master</li> <li>+ 淡和代码源</li> </ul>	著 「代話派 	本信息 実現記室 於太心点 交易 运行記 度 2013 1001 ① 添加送行分支	0007 第25 行集成 取	× 18/7 10 19 19	5.8	<b>2.84</b>	操行并运行
< 返回 週末线 2020-06-11 源 ● ► master + 淡加代現源	2 代研惑 0e0001dB903/testflow0609	本信息 現相配置 触来设置 定量 运行配置 かの01 ・ 添加油行分支	10.67 30.65 特集成 取	× Haft Tr M Haft Tr	:5	0.##	保守并运行
< 返回 近水地 2020-06-11 源 ● …001/d8903/testfow0609 ト ト master +※20ft6現意	2 代研惑 …0e0001/dB903/testflow0609	本信息 読程記名 触来设定 史書 送行記品 かつ の 添加法行分支	10.67 35.6 19.9.4 10	× Heft T 78 Heft T	:5	0.847	<b>张</b> 得并运行

可以添加多个分支

	73			×	
	Correct				
🕒001fd8903/testflow0609 ≽	代码源	分支 br001	状态	操作	E <b>%</b>
► master	0e0001fd8903/testflow0609	br002	行来成	w th	
+添加代码源		● 添加运行分支			
				取消运行	

运行分支添加完成后,就可以开始运行。在「分支管理器」卡片中可以查看执行结果及日志。若合并冲突,需要根据提示解决冲突后继续运行。
 通过「源」的「查看分支」或「分支管理器」卡片的「分支详情」可以查看创建的 release 分支及运行分支信息。

	Ⅲ 🔽 首页 > 流水线 2020-06-11 🕶	*			透清 🕂	∆ 0°
	最近运行 运行历史	()+###	10.000		<ul> <li>&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;&gt;</li></ul>	A,成员 ∃
	通行人 予絶时间 2020-06-11 09 44:19 持续时间 36秒 ▼ 須(1) ●e0001/d8903/testflow0609 ト release/20200611-0944392_ e982111 ト 査者分支	<ul> <li>◆ 分支管理器 進行成功</li> <li>ト 9文7年 □ 日本 335</li> </ul>	<ul> <li>● 执行命令 进行中</li> <li>● 取用 ① 日志 28</li> </ul>			
観光前子 溢行法         分生信息         ○ 副子 ② 細田 保護者           #2 ● 近行活动         素方分支: release/2020061-094439293615472_release_1007935_2         近日         2 ● 近行活动           第50日         ● alyuzhi         大         シス         0 回行 ② 細田 保護者         ● 「           第50日         ● alyuzhi         大         ● 「         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○         ● ○ <t< td=""><td></td><td></td><td></td><td></td><td></td><td></td></t<>						
** ● minikov 数年分変: release/20200611-094439239515472_release_1007935_2 通行入 ● alpxz#i 新設期 2020-06-11 094439 中の21 3ecs9-4b769393577ecb0cff1516443.2 已集成 中の21 3ecs9-4b769393577ecb0cff1516443.2 已集成 中の21 4ec589467286aba0acc7df0ab98e6001 東街 下信は38/20200611-0944392 ト 電音の変	Ⅲ 💽 前页 > 現未結 2020-06-11 -	*			• •	<u>م</u> ا
开版时间 2020-06-11 09-44-19 片版时间 6499 ・ 通(1) ・ celosof(25020657-ceD/cH15/68463.4) と日東点 - celosof(25020657-0944592 - celosof(25020657-0944592 - celosof(25020657-0944592 - celosof(25020657-0944592 - celosof(25020657-0944592 - celosof(25020657-0944592) と ていいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいいい	※ 資页 > 遺水线 2020-06-11 • 最近進行 進行历史	*	分支信息	×		<u>۵</u> گ به بقع =
持续封词 ・ 遺(1) ・	<ul> <li>註意 ) 微水线 2020-06-11 •</li> <li>磁坯运行 运行历史</li> <li>#2 ③ 运行成功</li> <li>运行人 ④ allyuzh</li> </ul>	* 業 業 系 分支: release/20200011-09443921 直行分支	分支信息 33815472_release_1007935_2	×	<b>6</b> 0 80 7 88	<u>ح</u> ک پ هم ع
★ 須(1)     ★ 16001 @cc68462266eba30ace7df0ab38a6001.    已集成     ★ refeaseg/20200671-0944392     ★ refeaseg/20200671-0944392     ★ 原告分支	<ul> <li>         ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・</li></ul>	2 2 2 2 2 2 2 2 2 2 2 2 2 2	分支信息 33815472_release_1007935_2 committd	×		۵ . ه هم =
executids903,Atestflow0609             x.itl                 x.itl                 releaser/20200611-0944392               executids903,Atestflow0609                 releaser/20200611-0944392               x.itl                 releaser/20200611-0944392               x.itl	王王 (三) 百元 > 武水城 2020-06-11 ・ 単近道行 道行历史           君2 (2) 道行成功           道行人           夏300-06-11 0-44-110           分数時間         2020-06-11 0-44-110           持数時間         2020-06-11 0-54-110	* 変布分支: release/2020001-09443921 語行分支 分支 b002	yżdał           committid           committid           żacałock/269938377ecboch115/684b3	х. Еща	<b>0</b> 200 0 200 7 202	۵ (۲) هم هم =
	#近畿行 遠行历史         編近畿行 遠行历史         #2<2<通行成功	* 変布分支: releasen/20200611-09443921 逆行分支 分変 bro02 br001	>żyżdła           committid           scało-cło/26936377ecb0cff196864b3           ccb69462e8eab30ece7xf0ab9886001	КС 28.6 28.6	(1) (1) (1) (1) (1) (1) (1) (1) (1) (1)	а Ф Ала =

4. 再次运行时,可以选择继续添加分支或删除已集成分支。



删除已集成分支,执行流水线时将会进行以下操作:

- 1.基于分支管理器中设置的基础分支(如 master),创建新的 release 分支
- 2. 除了该特性分支外的其他在云效配置中的其他分支合并到 release 分支
- 3. 基于 release 分支的最新内容运行流水线

# 4.2. 代码合并

步骤或者任务中可以添加 代码合并 步骤。

				保存保存并运行 ×
源(1)     • 🖝 👻	代码扫描	新阶段	编辑	×
● …/new-java-on-ecs 诊触发	・ G Java 代码扫描	● 新的任务	任务名称 Java 代码扫描	
全局变量(0) 🔹 🔻			任务步骤 赤加步骤	
暂未设置,点击 添加			代码     >       覆盖率     >       測试     >       构建     >       发布     >	检测是否落后于描定分支 代码合并 Git标签
			静态扫描 >> 下载 >> 上传 >>	:意这是不可逆操作,任务下所有数据将会删除
② 定时运行				

## 云效2020

			保存保存并运行 ×
源(1) 🕒 👻	代码扫描	新阶段	编辑 ×
● …/new-java-on-ecs ☆	日	● 新的任务	<ul> <li>添加步骤</li> </ul>
<b>全局变量(0)</b> ● ▼ 暫未设置,点击 添加			<ul> <li>てた9百升列及</li> <li>少環名称</li> <li>代码合并预发</li> <li>目标分支</li> <li>master</li> <li>図 期除源分支</li> <li>マ 只删除匹配的分支</li> <li>、</li> </ul>
〇 定时运行			任务播件 ③ 添加插件  删除任务 该任务将被彻底删除,注意这是不可逆操作,任务下所有数据将会  则  融

代码合并会将当前触发流水线的分支合并到目标分支。用户还可以选择合并后是否删除源分支。用户也可以输入一个字符串或者正则表达式,只有匹配的分支才会被删除。 完整的正则表达式支持请参见:https://docs.oracle.com/javase/8/docs/api/java/util/regex/Pattern.html。 正则表达式在线验证工具:https://www.freeformatter.com/java-regex-tester.html。

# 5.构建

# 5.1. 构建集群

Flow 为了支持用户个性化的构建场景,提供了三种类型的构建集群的能力支持。用户在流水线编排时,可以为任务设置不同的构建集群。

## 5.1.1. 构建集群

Flow 为了支持用户个性化的构建场景,提供了三种类型的构建集群的能力支持。用户在流水线编排时,可以为任务设置不同的构建集群。

## 云效构建集群

北京和中国(香港)构建集群

#### 自有构建集群

自有构建集群

## 5.1.2. 北京和中国(香港)构建集群

为解决用户下载海外构建依赖或者连接海外服务的场景需求,Flow分别提供了北京构建集群和中国(香港)构建集群。

## 使用构建集群

在任务节点选择构建集群,默认使用北京构建集群。



## 5.1.3. 自有构建集群

如果有特殊的构建环境需求,用户也可在 Flow 中创建并使用私有的构建集群。 云效流水线 Flow 支持使用用户阿里云 ECS 或者自有主机作为构建集群。 对于上述的主机类型,需要确认以下配置:

- 1. 支持的 OS 类型: Linux
- 2. 推荐配置: 4 核 8G 或以上
- 3. 安装并启动 Docker Daemon: https://docs.docker.com/engine/install/
- 4. 验证 Docker 运行状态:在主机上执行命令 sudo docker ps; echo \$?,输出应该为 0
- 5. 对于使用阿里云ECS主机作为构建集群的情况,需要确保云助手处于运行状态。

## 创建自有构建集群

通过 构建集群管理 -> 新建构建集群,开始创建自有构建集群。

III 流水线 Flow					🛨   🤐 🧷 😻
☆ 我的星标	构建集制	样管理			♡ 三↓ ● 新建构建集群 1
风 我参与的		构建集群名称名称	主机数量	包藏町间	\$
⇒ 全部流水线		新的构建集群	1台	2020-05-22	C T
企业管理		普通成员2建的构建集群	1台	2020-05-22	C T
為 通用变量组 □ 标签管理		预发环境测试私有构建集群0514	2 台	2020-05-14	C t
□ 主机组管理					
🗇 流水线模板管理					
☑ 任务组管理					
四 步骤管理					
构建集群管理					
③ 企业设置					

## 图1.新建构建集群

你可以创建两种类型的主机组:

#### 使用自有构建集群

在任务节点选择构建集群,切换为自有构建集群。

	291 P71 F.A.		编辑 📅
Java 代码扫描 ● 《 Java 构建上传 Maven 单元测试	• • <u>Eth</u>	<ul> <li>新約任券</li> </ul>	日月日日         日日日         日日日         日日日         「二〇日日         「二〇日         「二〇日

图2.使用构建集群

## 5.1.3.1. 自有构建集群

如果有特殊的构建环境需求,用户也可在 Flow 中创建并使用私有的构建集群。 云效流水线 Flow 支持使用用户阿里云 ECS 或者自有主机作为构建集群。 对于上述的主机类型,需要确认以下配置:

- 1. 支持的 OS 类型:Linux
- 2. 推荐配置: 4 核 8G 或以上

3. 安装社区版docker, Docker Daemon: https://docs.docker.com/engine/install/

#### 4. 验证 Docker 运行状态,在主机上执行命令

docker run --rm --entrypoint=/bin/sh -v /var/run/docker.sock:/var/run/docker.sock registry.cn-beijing.aliyuncs.com/build-steps/runner:latest && echo 0

, 输出应该为 0

## 5. 对于使用阿里云ECS主机作为构建集群的情况,需要确保云助手处于运行状态。

## 创建自有构建集群

#### 通过 构建集群管理 -> 新建构建集群,开始创建自有构建集群。

☆ 我的星标	构建集群管理			∇ 三↓ ⑤ 新建构建集群 1
A. 我参与的	内建集群名称名称	主机数量	包藏时间	8
♀ 全部流水线	新的构建集群	1台	2020-05-22	C 1
企业管理	普通成员2建的构建集群	1台	2020-05-22	C t
<ul> <li>3. 週用支重磁</li> <li>○ 标签管理</li> </ul>	预发环境测试私有构建集群0514	2台	2020-05-14	C 1
☐ 主机组管理				
☑ 流水线模板管理				
☑ 任务组管理				
11 步骤管理				
内建集群管理				
⑧ 企业设置				
				Ø

## 你可以创建两种类型的主机组:

- 阿里云 ECS 构建集群
- 自有主机构建集群

## 使用自有构建集群

在任务节点选择构建集群,切换为自有构建集群。

《 返回 預发演示构建集群+编排页 基本信息 <b>建程配置</b> 触发设置 变量 仅保存 保存并指行
HE         HE<



## 5.1.3.2. 阿里云 ECS 构建集群

云效流水线 Flow 支持将阿里云 ECS 添加为私有构建集群。你可以按照以下方式,将你的阿里云 ECS 加入构建集群,从而让流水线任务使用其进行构建。

## 创建阿里云 ECS 类型构建集群

1. 新建构建集群,选择【主机类型】为"阿里云ECS"。

			-	
	选择3	E机类型	×	
☆ 我的星标				⑦ 三↓ ⑤ 新建构建集群
风 我参与的			a)	
♀ 全部流水线		-	05 22	ार्थ भीग
			00-22	
			05-22	2 1
	1)		05-14	6 1
	阿里云ECS	目有主机		
	阿里云提供的性能早越、稳定可靠、弹 性扩展的 laaS 级别云计算服务	非同里云的公网王机,包括其他公有云 主机或者可以联通公网的自有主机		
☑ 流水线模板管理				
□ 仕旁翅管埋		Э		
回步骤管理				
■ 构建集群管理				
(2) 企业设置				

2. 选择【服务授权】和【地区】,展示【可添加的主机列表】,可【新建服务授权】。

III 流水线 Flow	
	<
☆ 我的星标	◎ 新建物理集群 (
. ²⁶ . 我参与的	rdc-rdc的ecs服务授权
♀ 全部流水线	地区 05-22 匠 位
企业管理	华东1 (杭州) ~
。通用变量组	□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □
□ 标签管理	実所の         P         主机名         占用他风         主机状态         05-14         ビ 音
🗊 主机組管理	- Lingtoffeld 10216211218 worker.kge 本主上用 工学
☑ 流水线模板管理	PuppiceUse 192:100:112:10 Workerでas 単水口方 正帝     Life 14:21:00:112:10 Workerでas 単水口方 正帝
☑ 任务组管理	□ PupPedutyA 192.100.112.1 Worker*A05 ▼A口方 工作
□ 步骤管理	□ PUDHAUUUX 192.100.112.4 WUIKEPKoo ● 水口/9 正帝
〓 构建集群管理	i-bpHdct0yk i92.i06.i12.a worker+kos ●末白用 正常
② 企业设置	□ i-bp1lih4svu 192.168.0.68 worker-k8s ● 未占用 正常
	Lipp1lh4svu 192.168.0.67 worker-k8s ● 未占用 正常
	□ I-bp1c26e4j 172.16.96.219 EDAS-scale ●未占用 正常
1 10 10 10 10 10 10 10 10 10 10 10 10 10	● 新不支持関始主机 上一步 下一步

3. 选择你想要添加至构建集群的 ECS 主机。

※ 流水线 Flow		(1) (1) (1) (1) (1) (1) (1) (1) (1) (1)
	<	
☆ 我的星标	服务授权 🕥 新建服务授权	▽ 三↓   ● 新建构建集群
.鸟、我参与的	rdc-rdc的ecs服务授权 ~	a 😒
♀ 全部流水线	地区	05-22
企业管理	华东1 (杭州) ~	
。通用变量组	可添加的主机列表 📀 💦 🔍 🖓	05-22 / U W
□ 标签管理	□ 実例ID IP 主机名 占用情况 主机状态	05-14 🗹 🗇
🗊 主机组管理		
🗇 流水线模板管理		
☑ 任务组管理	L-hn1dn10kk 192168112.4 worker-kRs ● 表古田 正常	
回 步骤管理	LholdetDuk 102162112.8 worker.k2e e.e.e.m TC	
■ 构建集群管理	i hpthdotfs 1021601120 i7bn1bdo15 a未上用 工業	
③ 企业设置	□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	
	□-bp1r25edi 1721696219 EDAS-scale ● 未占用 正常	
		e e

4. 编辑构建集群信息,包括构建集群名称、标签;点击保存,即可完成构建集群的创建。

//////		
	< 阿里云ECS·构建集群信息 ×	
合 我的星标	名称 🛛 🖓 🗄	↓ ● 新建构建集群
风 我参与的	新的构建集群	
⇒ 全部流水线	标签 (1)	
	请选择	C T
企业管理	± /n 05-22	re tu
A。通用变量组	199 1191 12 148名 人田慎淳 主点任务 操作	
□ 标签管理		C T
🗇 主机组管理	Риричоку 192.100.112.4 Wolker-Kos Ф АСЛЭ ША Ш	
☑ 流水线模板管理	为了提高成功,会在ECS上安装StarAgent 详情宣看文档     上一步     保存	
☑ 任务组管理		
回 步骤管理		
内建集群管理		
⑥ 企业设置		
^ 新手仕券 0/2		

5. 至此,你可在流水线中任务节点中选择使用该主机组了。

< 返回 预发演示构建集群+编排页	基本信息 流程配置 触发设置 变量	仅保存 保存并运行
単行 単行行) なん 単行行う なん 単行行う なん し	8000 • • • • • • • • • • • • • • • • • • •	編載 ① >> 日本1000000000000000000000000000000000000

## 5.1.3.3. 自有主机构建集群

云效流水线 Flow 支持将自有主机添加为私有构建集群。你可以按照以下方式,将您的自有主机加入构建集群,从而让流水线任务使用其进行构建。

## 创建自由主机类型构建集群

1. 新建构建集群,选择 【主机类型】为 "自有主机"。

	选择主机	机类型	×	
☆ 我的星标				▽ 三↓ ● 新建构建築群
²⁰ 我参与的			2	
异 全部流水线				
			05-22	C V
企业管理			05-22	1
為 通用变量组	(1)		05-14	ार्थः भीग
□ 标签管理	阿里云ECS	自有主机		
主机组管理	阿里云提供的性能卓越、稳定可靠、弹 性扩展的 laaS 级别云计算服务	非阿里云的公网主机,包括其他公有云 主机或者可以联通公网的自有主机		
☑ 流水线模板管理				
☑ 任务组管理	$\odot$	$_{\odot}$		
回 步骤管理				
= 构建集群管理				
② 企业设置				
V LINE				
ヘ 新手任务 0/2				

2. 展示【可添加的主机列表】,如果想要添加新的自有主机至【可添加的主机列表】,请复制主机添加指令。

III 流水线 Flow						<ul> <li>930</li> <li>930</li> <li>930</li> <li>931</li> <li>932</li> <li>933</li> <li>934</li> <li>935</li> <li>935</li></ul>
	<	自有主机·添加主机		×		
☆ 我的星标	可添加的主机列表			9, 7	2	7 ☴↓ 🕒 新建构建集群
风 我参与的	二 実例ID IF	p 主机名	占用情况	主机状态		
♀ 全部流水线	271d622c-3 (	内网)192.16 iZbp14ot0yk	● 未占用	正常	MMErce	~~ ~/ ++
6.5.M18	61361269-3 (	内网)192.16 iZbp14ot0yk	● 未占用	正常	<b>阿圭云ECS</b>	le w
2 次田市市4日	68261ac2-9 (	内网)192.16 iZbp14ot0yk	● 未占用	正常	阿里云ECS	C T
	6c7eea4d-0 (I	内网)192.16 iZbp14ot0yk	● 未占用	正常	阿里云ECS	C T
同 主机组管理	17caa635-fa (	内网)192.16 iZbp14ot0yk	● 未占用	主机无法连接		
② 流水线模板管理	3b1c02e1-8 (	内网)192.16 iZbp14ot0yk	● 未占用	主机无法连接		
☑ 任务组管理	453c1cd6-1 (	内网)192.16 iZbp14ot0yk	● 未占用	主机无法连接		
回 步骤管理	a765e439-d (	内网)192.16 iZbp14ot0yk	● 未占用	主机无法连接		
= 构建集群管理	如何导入新机器到「可添加」	列表」?				
⑧ 企业设置	1.请复制下列命令到需要添加的主机」 2.执行1-2分钟后,新机器会自动导入	上,该命令运行 <b>14分54秒</b> 后过期 、「可添加列表」里				
	bash «(curl http://agent-install 'http://pre-devops.aliyuncs.com 8d35- d78361/11995&sign=x2348oo3			日复制		
			Ŀ	- <b>步</b>		0

3. 请打开你需要添加进构建集群的主机的命令行,并将前序操作中复制的命令,在你的自有主机上执行(请注意,主机需要能访问公网)。



4. 主机添加指令执行后,等待1-2分钟后,自有主机会导入【可添加的主机列表】,选择你想要添加至主机组的自有主机。

III 流水线 Flow							🕑 🤐 🕐 🚈
	<	自行	与主机 · 添加主机		×		
☆ 我的星标	可添加的主机列表				Q, 7		▽ 三↓ ⑤ 新建构建集群
风 我参与的	二 实例ID	IP	主机名	占用情况	主机状态		
⇒ 全部流水线	271d622c-3	(内网)192.16	iZbp14ot0yk	● 未占用	正常	EIE = 500	~
	61361269-3	(内网)192.16	iZbp14ot0yk	● 未占用	正常	阿里云ECS	Le w
企业管理	68261ac2-9	(内网)192.16	iZbp14ot0yk	● 未占用	正常	阿里云ECS	C 1
	6c7eea4d-0	(内网)192.16	iZbp14ot0yk	● 未占用	正常	阿里云ECS	C t
	17caa635-fa	(内网)192.16	iZbp14ot0yk	● 未占用	主机无法连接		
	3b1c02e1-8	(内网)192.16	iZbp14ot0yk	● 未占用	主机无法连接		
	453c1cd6-1	(内网)192.16	iZbp14ot0yk	● 未占用	主机无法连接		
四 为限管理	a765e439-d	(内网)192.16	iZbp14ot0yk	● 未占用	主机无法连接		
二 和建集群管理	如何导入新机器到「可述	Shn列来」?					
(2) 企业设置	1.请复制下列命令到需要添加的; 2.执行1-2分钟后,新机器会自动	主机上,该命令运行 14 1号入「可源加列表」重	分54秒 后过明				
· · · · ·	bash <[curl http://agent-ir 'http://pre-devops.aliyunc 8d35- d78361f1995&sign=x234				同期		6
					上一步下一步		<b>e</b>

5. 编辑构建集群信息,包括构建集群名称、标签;点击保存,即可完成构建集群的创建。

			0.5.4.0 Ab 10	after 2010 After alls				
	<		目有土机·构建	栗群1音思		×		
☆ 我的星标	名称						8	● 新建构建集群
风 我参与的	新的构建集群						主机类型	
♀ 全部流水线	(1) 标签						RMEroc	CA - 44
	请选择					~	<b>阿圭</b> 五EC3	le w
企业管理	主机						阿里云ECS	C 1
為 通用变量组	实例ID	IP	主机名	占用情况	主机状态	操作	WWIEros	C4 197
□ 标签管理	68261ac2	(内网)192.16	iZbp14ot0y	● 未占用	正常	Û	<u>阿圭云EC3</u>	Ľ ₩
◎ 主机组管理								
☑ 流水线模板管理					上一老	步 保存		
☑ 任务组管理								
四 步骤管理								
III 构建集群管理								
③ 企业设置								
								P
15 E / T # 0/2								

6. 至此,你可在流水线中任务节点中选择使用该主机组了。

〈 返回   預发演示构建集群+编排页	基本信息 <b>流程配置</b> 触发设置 变量	<b>仅保存</b> 保存并运行
時日 Java ft 単日の Maven 単元間式 maven	BRIG 9 <u><u><u><u>x</u></u>(f.6)</u> 1 1</u>	編纂 音 ×      任务名称     安任务     「交任务     「「「「「「」」」」     「「」」     「」」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「     「」     「」     「」     「」     「」     「     「」     「     「」     「     「」     「」     「」     「     「     「     「」     「     「」     「」     「     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「」     「     「     「     「     「     「     「     「     「     「     「     「     「     「     「     「     「     「     「     「     「     「       「         「

## 5.2. 构建语言支持

云效流水线 Flow 通过多种构建步骤,支持不同语言的构建能力。下面描述构建环境中安装的软件清单,及不同语言场景下的使用方法。

## 公共软件

所有构建环境都包含的软件:

- 1. gcc/g++ 4.8.4
- 2. make 3.8.1
- 3. curl 7.35.0
- 4. wget 1.15
- 5. unzip 6.00
- 6. git 1.9.1

7. python 3.5 (不在PATH中, 需要使用 /usr/alibaba/install/python-3.5.0/bin/python3 来引用)

## 各语言构建环境中的软件及使用方法

语言	支持版本
<ul> <li>Java</li> <li>Maven 2.2.1, 3.2.5, 3.5.2, 3.6.1, 3.6.3</li> <li>Gradle Flow 未提供内置 Gradle, 建议使用 GradleWrapper。</li> <li>修改gradle/wrapper/gradle-wrapper.properties文件, 将其中的distributionUrl修改 为 Flow 提供的镜像地址, 如: "distributionUrl=https\://rdc-public-software.oss- cn-hangzhou.aliyuncs.com/gradle/gradle-6.5-bin.zip",可将其中'6.5'可替换为以下 版本。</li> <li>云效提供的gradle镜像版本: 4.4.1, 4.5.1, 4.6, 4.7, 4.8.1, 4.9, 4.10.3, 5.3.1, 5,4.1, 5.5.1, 5.6.4, 6.5, 6.8.4</li> </ul>	<ul> <li>JDK 1.6</li> <li>JDK 1.7</li> <li>JDK 1.8</li> <li>JDK 1.9</li> <li>JDK 10</li> <li>JDK 11</li> </ul>
NodeJS • 安装了python 2.7, 默认置于PATH中,支持node-gyp编译。 • cnpm 6.1.1	<ul> <li>node6.11.3 npm3.10.10 yarn1.15.2</li> <li>node7.10.0 npm4.2.0 yarn1.15.2</li> <li>node8.13.0 npm6.4.1 yarn1.15.2</li> <li>node9.11.2 npm5.6.0 yarn1.15.2</li> <li>node10.15.3 npm6.4.1 yarn1.15.2</li> <li>node11.15.0 npm6.7.0 yarn1.15.2</li> <li>node12.2.0 npm6.9.0 yarn1.15.2</li> <li>node12.18.4 npm6.14.6 yarn1.15.2</li> <li>node13.14.0 npm6.14.4 yarn1.15.2</li> <li>node14.8.0 npm6.14.7 yarn1.15.2</li> <li>node14.8.0 npm6.14.7 yarn1.15.2</li> </ul>

<b>漫言</b>	支持版本
Python	<ul> <li>2.7</li> <li>3.5</li> <li>3.6</li> <li>3.7</li> </ul>
PHP • componser 1.8.6	<ul> <li>5.6</li> <li>7.0</li> <li>7.1</li> <li>7.2</li> <li>7.3</li> <li>7.4</li> </ul>
Golang • go-wrapper	<ul> <li>1.8</li> <li>1.9</li> <li>1.11</li> <li>1.12</li> <li>1.13</li> </ul>
C/C++	• gcc/g++ 10.2 • gcc/g++ 9.3 • gcc/g++ 8.4
.net Core	• SDK 3.1

对应的构建能力都是通过基础构建机支持,基础环境已经支持了上述构建能力,如上述能力无法支持你的构建需求,请你联系我们的客服以获取更多的支持。

# 5.3. 设置依赖下载访问凭证

## 环境变量

在构建过程中,需要设置仓库的访问凭证并下载构建依赖,为保护访问凭证的安全性,Flow 支持通过环境变量设置访问凭证,关于环境变量请查看:<del>环境变量</del> 当然,作为企业统一管控的仓库访问凭证,我们更建议您使用通用变量组进行管理,关于通用变量组请查看:通用变量组

```
以下,提供了如何使用环境变量配置 Java、NPM 和 GoLang 仓库访问凭证的指南。
```

#### Java

1. 在 settings.xml 中添加私库的地址和账号/密码(环境变量)。

<servers></servers>
<server></server>
<id>rdc-releases</id>
<username>\${MVN_USERNAME}</username>
<pre><password>\${MVN_PASSWORD}</password></pre>
<server></server>
<id>rdc-snapshots</id>
<username>\${MVN_USERNAME}</username>
<pre><pre>cpassword&gt;\${MVN_PASSWORD}</pre></pre>
<profiles></profiles>
<profile></profile>
<activation></activation>
<activebydefault>true</activebydefault>
<id>rdc-private-repo</id>
<repositories></repositories>
<repository></repository>
<id>rdc-releases</id>
<url>https://packages.aliyun.com/maven/repository/2000093-release/</url>
<repository></repository>
<id>rdc-snapshots</id>
<url>https://packages.aliyun.com/maven/repository/2000093-snapshot/</url>

将其中相关的 <url> 内容替换成您自己的私库地址。

-	
2.	在沇水线甲配置账号密码的外境受量。

返回 通过环境变量构建时下载私有包		基本信息 流程配置 触发设置	变量		仅保存
<b>交量</b> 通用 <b></b> 中最相	<b>变量</b> 通过定义环境变量实现流水线过程定制化,可	可以在执行过程的任何阶段使用这些变。 <b>查看文档</b>			
AUT 13 49, 000-103	配置变量组				Q。 💿 新建变量
	參数	默认值	私密模式②	运行时设置②	操作
	MVN_USERNAME				ŵ
	MVN_PASSWORD				1

3. 在流水线运行命令中,使用 settings.xml 。



NPM 和 GoLang

1. 在流水线中配置账号密码的环境变量。

< 返回 go 测试		基本信息 流	程配置 触发设置 变量和缓存			仅保存				
<b>变量</b> 通用变量组	<b>变量</b> 通过定义环境变量实现流水线过	<b>安重</b> 通过定义环境变量实现流水线过程定制化,可以在执行过程的任何阶段使用这些变量, <b>查看文档</b>								
緩存	字符变量	字符变量								
	参数	服大议	认值	私密模式②	运行时设置⑦	操作				
	GIT_USERNAME	***	*****			Ŵ				
	GIT_PASSWORD	••••				Ŵ				
	运行选择变量					Q。 🕒 新建变量				
	参数		默认值	选项		操作				
			没有	数据						

## 2. 在 npm install 或 go get 之前,添加如下命令。

git config --global url."https://\${GIT_USERNAME}:\${GIT_PASSWORD}@codeup.aliyun.com".insteadOf "https://codeup.aliyun.com"

< 返回	go 测试		基本信息 流程配置	<b>置</b> 触发设置 变量和缓存		仅保存	保存并运	行
源		构建	新阶段		编辑 ①			×
	+ 添加代码演	ゆ Go 构建 ゆ Node.js 构建	新的任约	*	<ul> <li>云效北京伪建集群</li> <li>任务步骤</li> <li>⑦ 添加步骤</li> <li>⑦ 添加步骤</li> <li>⑦ ⑤加步骤</li> <li>⑦ ⑤加季</li> <li>① ○</li> <li>② ◎ ◎</li> <li>② ◎ ◎</li> <li>② ◎ ○</li> <li>② ● ○</li> <li>② ● ○</li> <li>○ ○</li> <li>○</li> <li>○</li></ul>	/.cn 1.com ://3{GIT_USI .dlyun.com .dlyun.com 228 go gonic	()	

〈 返回	go 测试		基本信息 流程配置 触发设置 3	变量和缓存		仅保存保存并运行
	go 潮试 + 添加代码源	构建	基本信息 燕尾配置 触发设置 3 新阶段 ● 新的任务	交量和維存	<ul> <li>編目①</li> <li>TSLEMENT ()</li> <li>云次北京构建集群</li> <li>G効ルテル</li> <li>不加ルテル</li> <li>小の信点: 本利建</li> <li>少磁名称</li> <li>Node.js: 林建</li> <li>功磁名称</li> <li>14.8</li> <li>Nd&amp;@@ ①</li> <li>2 strptk.vour compand here:</li> <li>3 comp install git-https://codeu. /node: expressjs.git</li> <li>公 优先使用级仔 ①</li> </ul>	(文保存)         保存并运行           ····································
					任务插件	
其中:						

GIT_USERNAME : 是您在第1步中配置的私有 git 仓库的用户名环境变量的名称。

GIT_PASSWORD : 是您在第1步中配置的私有 git 仓库的密码环境变量的名称。

将其中的 codeup.aliyun.com 部分替换成您实际使用的 git 仓库的域名,比如 github.com

## 5.4. 构建容器镜像

Flow 通过 Docker 构建组件,提供对各语言的通用镜像构建能力,让用户可以通过使用代码库中的 Dockerfile 构建容器镜像,并推送到阿里云或者自定义的容器镜像仓库中。

## 构建阿里云容器镜像

## 海外集群构建容器镜像

## 5.4.1. 构建容器镜像

Flow 通过 Docker 构建组件,提供对各语言的通用镜像构建能力,让用户可以通过使用代码库中的 Dockerfile 构建容器镜像,并推送到阿里云或者自定义的容器镜像仓库中。

## 构建阿里云容器镜像

构建阿里云容器镜像

## 5.4.2. 构建阿里云容器镜像

Flow 通过 Docker 构建组件,提供对各语言的通用镜像构建能力,让用户可以通过使用代码库中的 Dockerfile 构建容器镜像,并推送到阿里云容器镜像仓库中。

## 前置条件

- 1. 开通阿里云容器镜像服务或者其它可使用的容器镜像仓库服务
- 2. 代码源中包含需要构建的 Dockerfile 文件

#### 构建 Docker 镜像并推送到阿里云容器镜像服务

下面以 Java 项目镜像构建为例进行说明。

在Flow首页新建流水线,并选择" Java测试,构建镜像 "的流水线模板。

★ 我的星标	我的星标			三↓ ● 新建流水线
④ 我参与的			-	
♀ 全部流水线		选择流水线模板	×	
NO. INTERNATION	C 空流水线 Java 测试、	构建镜像		
设置信息	👙 Java			
[9] 企业设置	Node.js	# +5 2# 5% /0		
	php PHP	193 XE 192 198		
	60 Go		_	
	Python Node.js			
	88 其他 React 构建上	_传到 OSS		
	伊田扫	#		
	1/1/1/1	构建上传		
	测试			

在流水线创建向导中选择好代码源之后,进入流水线编辑页面。点击"阿里云镜像构建"展开任务详情。

R	⊕ 新建源		构建镜像	1	新的阶段	编辑	
G moo/docker-b	build-sample 代	码扫描	- P (	阿里云镜像构建 ①	日 🕒 新的任务	任务名称	
8∼ moater						阿里云镜像构建	
	n ŝ	自元测试				任务步骤	
						- 镜像构建并推送至阿里云镜像仓库	0
						步骤名称	
						镜像构建并推送至阿里云镜像仓库	
						选择授权 (2)	新建服务授
							~
						选择授权不能为空	
							~
						区域不能为空 合库	
						会處不能为空	~
						42 Apr	0

为了能够将构建的镜像正常 Push 到阿里云容器镜像服务的仓库中,需要 Flow 与你的阿里云账号建立连接,点击"新建服务连接",然后点击" RAM 授权"。

源	● 新建源		构建镜像	新的阶段	编辑		
G moo/docker	r-build-sample				阿里云	镜像构建	
}⊸ master		代码扫描	♀ 阿里云镜像构建 ①	□ ● 新的任务	任务步骤		
		n 单元测试	新	建服务授权	$\times$	骤	
			服务授权类型			建并推送至阿里云镜像仓库	
			阿里云容器镜像服务			Я	
			服务授权名称			建并推送至阿里云镜像仓库	
			砧木的服务连接	1	7/20	R	新建服务
			阿里云授权	Ļ		不能为空	
			当刖账号绑定的阿里云账号为 zheng	gy,点击便用该账号进行 RAM授权			
				取消	保存		
					仓库不	能为空	
					标签		
					标签不	能为空	
<b>10</b>					Dock	orfilo 98/3	

## 这里会要求登录阿里云账号,并确认授权,同意授权后会自动跳转回 Flow 页面。

溫馨是示:如唐條改角色权限,请前往RAM控制台角色管理中设置,需要注意的是,错误的配置可能导致Teambition无法积取到必要的权限。 X
Teambition请求获取访问您云资源的权限 下方是系统创建的可以Teambition使用的角色, 授权后, Teambition得有对您云资源相应的访问权限。
▲ AliyunTeambitionDefaultRole 服語: Teambioの放送使用皮肉合意来问题在其他因产品中的资源 即問題者: 「最近了apparted」合意的原因意思

## 点击"同意授权",并填写服务授权的名称,然后点击"保存"。完成并建立服务连接。

End William Contraction	Booker 194Ecompte					PK 13	PRIJVIANI
源	🕕 新建源		构建镜像	新的阶段	编辑		
G moo/docke	er-build-sample				阿里云镜像构建		
i‰ master		代码扫描	♀ 阿里云镜像构建 ①	● 新的任务	任务步骤		
		n 单元测试	新建	服务授权	×		
			服务将权类型		建并推送	至阿里云镜像仓库	0
			阿里云容器镜像服务		Ť.		
			服务授权名称		建并推送	至阿里云镜像仓库	
			砧木的服务连接		7/20 g		新建服务指
			四用二抵收				
			阿圭乙按快 当前账号绑定的阿里云账号为 zhengy	/,且已完成RAM授权	1 不能为空		
					+		9
				取消	保存为空		
							,
					仓库不能为空		
					标签		
					标签不能为空		
10	中时进行				Dockerfile 36/2		

继续选择 "区域"和"仓库",并设置镜像的 TAG 标签。在 Dockerfile 路径中需要填写构建镜像所用的 Dockerfile 相对路径。在 ContextPath 则填写 Docker 构建时的执行上下文。

原	+ 新建源		构建镜像	新的阶段	编辑	
☐ moo/docker-build-s № master	sample	代码扫描	<b>み</b> 阿里云镜像构建	0 日 9 新的任务	> 镜像构建并推送至阿里云镜像仓库 也要文和	
		n 单元测试			镜像构建并推送至阿里云镜像仓库	
		F&			选择授权	新建服务授
					砧木的服务连接	~
					区域	
				1—	> cn-hangzhou	~
					仓库	
				2—	registry.cn-hangzhou.aliyuncs.com/k8s-mirrors/dock	er-build-s v
					标签	(
				(3)—	→ \${DATETIME}	
					Dockerfile路径	(
				(4)—	Dockerfile	
				$\frown$	ContextPath	(

点击"保存并运行流水线",开始运行流水线任务,在流水线运行成功后,可以看到当前流水线的运行结果。

A CONTRACT OF A CONTRACT. CONTRACT OF A CONTRACT. CONTRACT OF A CONTRACT OF A CONTRACT OF A CONTRACT OF A CONTRACT. CONTRACT OF A CONTRACT OF A CONTRACT OF A CONTRACT. CONTRACT OF A CONTRACT OF A CONTRACT OF A CONTRACT. CONTRACT OF A CONTRACT OF A CONTRACT. CONTRACT OF A CONTRACT OF A CONTRACT. CONTRACT OF A CONTRACT OF A CONTRACT OF A CONTRACT. CONTRACT OF A CONTRACT OF A CONTRACT. CONTRACT OF A CONTRACT. CONTRACT OF A CONTRACT OF A CONTRACT. CONTRACT OF A CONTRACT. CONTRACTACT OF A CONTRACT. CONTRAC			
最近运行 运行历史			⊙ 运行
#1 🥑 运行成功	测试	构建镜像	
G moo/docker-build-sample	⊘ Java 代码扫描	阿里云镜像构建	
‰ master -o- 2807d6f	2 0 0 2 ^{总数} 阳塞 严重 一般	运行成功	
	□ 扫描报告 □ 日志 178	● 目志 5分3秒	
	Ø Maven 单元测试		
	0/0 0 0% 测试用例 失敗 跳过 通过率		
	⑦ 测试报告 []日志 22分27秒		
运行人 🚥 砧木			
开始时间 2019-09-26 20:27:42			
运行时上 27 <b>公</b> 21种			

点击任务卡片上的日志按钮,可以查看任务的运行详情以及日志输出。

最近运行 运行历题		阿里云镜像构建 ※	编辑	<b>凤</b> 成员	三菜单
#1 A 1747 st) Th	构建	镜像构建并推送至阿里云镜像仓库(287s)			
	⊘ 环境分派(1s)	 Jatast: Pulling from library/nginy			
G moo/docker-build	⊘ 环境初始化(1s)				
‰ master	⊘ 清理工作区(2s)				
-o- 2807d6f	⊘ 克隆代码(5s)	 Digest: sha256:aeded0f2a861747f43a01cf1018cf9efe2bdd02afd57d2b11fcc7fcadc16ccd1			
	❷ 镜像构建并推送至阿里云镜像	Status: Image is up to date for nginx:latest			
		Successfully built f949e7d76d63			
		docker push registry.cn-hangzhou.aliyuncs.com/k8s-mirrors/docker-build-sample:2019-09-26-20-50-25			
		The push refers to a repository [registry.cn-hangzhou.aliyuncs.com/k8s-mirrors/docker-build-sample] 509a5aadaeeh: Prenaring			
	$\bigcirc$	3bb51901dfa3: Preparing			
	(1)	2db44bce66cd: Preparing			
		509a5ea4aeeb: Mounted from kBs-mirrors/kube-app			
		3bb51901dfa3: Mounted from k8s-mirrors/kube-app 2db44bca66cd: Mounted from k8s-mirrors/kube-app			
		2019-09-26-20-50-25: digest: sha256:7524258ec0d96eca13d701837ae34c05cd741c100c790bb46e53d322721fbb91 size:			
		1545 上传镜像配置文件			
		Total num: 1, size: 97. Dealed num: 0, OK size: 0, Progress: 0%			
		Total num: 1, size: 97. Dealed num: 0, OK size: 97, Progress: 100%			
运行人		Succeed: Total num: 1, size: 97. OK num: 1(upload Times). 0.053674(s) elapsed			
工始日本的 2019-		[INFO] 执行用户命令成功			
7TXERY[8] 2019-		[executionStep ends at 2019-09-26 20:55:09.349]			

在阿里云容器服务控制台中可以查看最新构建的容器镜像。

<	<b>docker-build-samp</b> 李东1(杭州)   私有   本培会库	le   ● 正常	~				10 10 10 10 10 10 10 10 10 10 10 10 10 1
88	镜像版本		(1)				818
受权	版本	現像ID @	状态	Digest	领象大小 🛛	最后更新时间	1
8 版本	2019-09-26-20-50-25	f949e7d76d63	0 IR	7524258ec0d96eca13d701837ae34c 05cd741c100c790bb46e53d322721f bb91	48.320 MB	2019-09-26 20:55:08	安全扫描   服信息   同步
<b>1</b> 5							

## 图9.查看构建镜像

对于使用启动镜像仓库服务的用户,可以在任务步骤中选择"镜像构建并推送至自定义镜像仓库",并通过用户名和密码的方式将镜像推送到自定义仓库中。

## 5.4.3. 海外集群构建容器镜像

Flow 目前提供了海外集群(默认为中国(香港)区域)用于加速海外镜像的拉取和构建。

#### 前置条件

- 1. 确认代码仓库可以被海外集群访问。
- 2. 代码仓库中包含 Dockerfile,且 Dockerfile 中依赖的镜像源可被海外集群访问。

3. 开通海外<mark>阿里云容器镜像服务</mark>或者其它可使用的容器镜像仓库海外服务,推荐优先使用中国(香港) region 的阿里云容器镜像服务。

## 构建 Docker 镜像并推送到阿里云容器镜像服务

下面以 Java 项目镜像构建为例进行说明。

在 Flow 首页新建流水线,并选择" Java测试,构建镜像"的流水线模板。

★ 我的星标	我的星标		曰↓ ● 新建流水线
② 我参与的	VF 10 - VF 10	***	
♀ 全部流水线	选择流水线相	吴攸 二	
とり開始自	➡ 空流水线 Java 测试、构建镜像		
収重活志	🛓 Java		
101 企业设置	Node.js	北京市会会	
	Php PHP	行为建筑国家	
	GO GO		
	Python Node.js		
	88 其他 React 构建上传到 OSS		
	代码扫描		
		构建上传	
	测试		

在流水线创建向导中选择好代码源之后,进入流水线编辑页面。点击"阿里云镜像构建"展开任务详情。

			21 XU
▼ 灝(1) +	构建镜像 1	新阶段	编辑
/flow-example/spring-boot			任务名称
≫ master aven 単元測试	日 4 阿里云视像构建	€ 新的任务	阿里云镜像构建
▼ 変量 (0)		2	☑ 使用海外集群构建
			任务步骤
暂未设置,点击 添加			☆ 添加步骤
			✓ Docker构建-service-connection
			步骤名称
			Docker构建-service-connection
			选择授权 (3) 6
			rdc-rdc 的服务连接
			区域
			新加坡
			仓库
			registry.ap-southeast-1.aliyuncs.com/rdctest/jintang-test

为了能使用中国(香港)集群进行构建,需要勾选"使用海外构建集群"。然后点击"新建服务连接",然后点击"RAM 授权"。

8	● 新建源		构建镜像	新的阶段	编辑		
G moo/docker-k	build-sample				阿里云	镜像构建	
‰ master	ť	代码扫描	<b>ゆ</b> 阿里云镜像构建 <b>①</b>	日 ③ 新的任务	任务步骤		
	n	单元测试	新建服务授	权	$\times$	骤	
			阳冬语如米刑			建并推送至阿里云镜像仓库	0
			阿里云容器镜像服务			я	
			服务调切之称			建并推送至阿里云镜像仓库	
			私太 <b>的服务</b> 连接		7/20	a	新建服务授机
			אנצא פיאמניארויקע	Ψ	1/20	~	
			阿里云授权	Ļ			`
			当前账号绑定的阿里云账号为 zhengy,点击	更用该账号进行 RAM授权			
					_		~
				取消	保存	为空	
							~
					仓库不	能为空	
					标签		(
					1- 00-7	- Administration	
(i) <del>(</del>	0+2=4=				标查个 Dock	·能力全 erfile路径	0

这里会要求登录阿里云账号,并确认授权,同意授权后会自动跳转回 Flow 页面。

温馨提示:如需修改角色权限。请前往RAM控制台角色管理中设置,需要注意的是,错误的配置可能导致Teambilition	n无法获取到必要的	的权限。			>	<	
Teambition请求获取访问您云资源的权限 下力是系统创建的可供Teambilion使用的角色,提取员、Teambilion拥有对您云贯源粗质的访问权限。							
AliyunTeambitionDefaultRole 题话:Teambition就认使用成角色果切词简在其他云产品中的资源 权限题述:用于Teambition服务就认角色的授权调略			 	 	~		
1 同意投权 取消							

点击"统一授权",并填写服务授权的名称,然后点击"保存"。完成并建立服务连接。

Į.	新建源		构建镜像	新的阶段	编辑		
G moo/docker-	-build-sample				阿里云	镜像构建	
}∞ master		代码扫描		日 ① 新的任务	任务步骤	R	
		n 单元测试	新建服务授	权	×	骤	
			服务将权举型			建并推送至阿里云镜像仓库	0
			阿里云容器镜像服务			3	
			服务授权名称			建并推送至阿里云镜像仓库	
			砧木的服务连接		7/20	x	新建服务授权
			阿里云拇权		$\bigcirc$		~
			当前账号绑定的阿里云账号为 zhengy,且已	完成RAM授权	( <b>1</b> )	不能为空	
					Ļ		~
				取消	保存		
							~
					仓库不	能为空	
					标签		0
					标签不	<b>能</b> 为容	
(X) =	P0+2=4=				Docke	erfile路径	(?

继续选择 "区域"和"仓库",并设置镜像的 TAG 标签。在 Dockerfile 路径中需要填写构建镜像所用的 Dockerfile 相对路径。在 ContextPath 则填写 Docker 构建时的执行 上下文。

■ mUA 线编用 DOCKe	Sample			M IF MIFFIEIJ
源	● 新建源	构建镜像	新的阶段 编辑	
G moo/docker-build-s	ample		→ 镜像构建并推送至阿里云镜像仓库	
ĵ⊷ master	代码扫描	日 · · · · · · · · · · · · · · · · · · ·	● 新的任务 步骤名称	
	n 单元测试		镜像构建并推送至阿里云镜像仓库	
	王务		选择授权	新建服务
			砧木的服务连接	
			区域	
			1 cn-hangzhou	
			仓库	
			2 registry.cn-hangzhou.aliyuncs.com/	k8s-mirrors/docker-build-s
			标签	
			3 \${DATETIME}	
			Dockerfile路径	
			(4) Dockerfile	
			ContextPath	
			ContextPath	

点击"保存并运行流水线",开始运行流水线任务,在流水线运行成功后,可以看到当前流水线的运行结果。

<b>最近运行</b> 运行历史			● 运行
¥1 🥑 运行成功	测试	构建镜像	
🕞 moo/docker-build-samp	le	✓ 阿里云镜像构建	
‰ master -œ 2807d6f	2 0 0 总数 阻塞 严重	2 运行成功	
	▶ 扫描报告 □ 日志	178 日志 5分3秒	
	✓ Maven 单元测试		
	0/0 0 0	<b>0%</b> 通过第	
	同 测试报告 [] 日志	2527秒	
运行人	663 砧木		
千始时间 2019-09-26	20:27:42		
医结束	07/\0454		

点击任务卡片上的日志按钮,可以查看任务的运行详情以及日志输出。



#### 在阿里云容器服务控制台中可以查看最新构建的容器镜像。

			Q 25%			费用 工单 备案 企业	支持与服务 🖸 🗘 🗑	🕐 🏫 🕅 🎯
<	docker-build-sampl 李东1 (杭州)  私有   本地合库	le  ● 正常	$\sim$					● 部菁应用
基本信息	镜像版本		(1)					8185
仓库授权	版本	現像10 @	状态	Digest	银梁大小 \varTheta	最后更新时间		操作
触双器 镜像版本	2019-09-26-20-50-25	f949e7d76d63	◎ 正常	7524258ec0d96eca13d701837ae34c 05cd741c100c790bb46e63d322721f bb91	48.320 MB	2019-09-26 20:55:08		安全扫描   瓜信息   同步   删除
镜像同步								< 1 >
								Bâ

对于使用启动镜像仓库服务的用户,可以在任务步骤中选择"镜像构建并推送至自定义镜像仓库",并通过用户名和密码的方式将镜像推送到自定义仓库中。

# 5.5. 构建制品

Flow 通过各种构建组件,对各种语言提供了制品打包能力,让用户可以快速的使用流水线构建制品,并通过后续的部署任务进行部署。

## 创建构建任务

在流水线中通过添加构建任务,可以构建出制品,用于部署。飞流提供了多种方式快捷的使用构建能力:

1)新建流水线时,选择构建相关模版;

2)编辑流水线时,添加构建任务;

## 流水线模版创建构建任务

新建流水线时,选择对应的开发语言,可以查看当前语言下的默认流水线模版,选择带有"构建"节点的流水线模版,即可快速使用构建能力。

☆ 我的星标	我的星标			EL 🚺 🚺	新建流力
<ul> <li>② 我参与的</li> </ul>		选择流水线模板	×	最近运行时间	
⇒ 全部流水线	5 空流水线	PHP	杨超越	2019-09-24	*
设置信息 ② 企业设置	🐇 Java	PHP 测试、构建、部署到主机	杨超越	2019-09-24	*
	JS NodeJS	代码扫描	刘吴然	2019-09-24	*
	GO GO	构建 部署 测试	吴彦祖	2019-09-23	*
	Python		杨超越	2019-09-24	*
	品其他	PHP 测试、构建	刘德华	2019-09-23	*
		代码扫描 构建	周杰伦	2019-09-23	*
		测试	刘昊然	2019-09-24	*

## 图1.流水线模版创建

#### 添加构建任务创建构建任务

在流水线编排中,可以通过添加任务,在模版中选择带有"代码扫描"的模版,可以快速的添加代码扫描任务

源	C 200-200 3	942 <del>4</del>	hana 洪垠仁冬米刑	口谷前蜀	新的阶段		
← flow/flow-engine	9、请输入任务名和	● 新的任务					
	空模板	测试	测试				
	代码扫描 测试	Maven 单元測试 Maven 单元測试	<b>Gradle 单元测试</b> Gradle 单元源试	nede NodeJS 单元测试 NodeJS 单元测试			
	测试构建	Python 单元测试 Python 单元测试	PHP Codeception 单元 PHP Codeception 单元测试	PHP Unit 单元测试 PHP Unit 单元测试			
	镜像构建部署	- <b>GO 单元测试</b> Go 单元测试					
	代码	测试构建					
	执行命令	<b>Java 単元測试、构建</b> Java单元測试、构建、构建	nede NodeJS 构建测试上传 NodeJS 单元测试、构建、	nede React 构建测试上传 React 构建测试并上传至 Oss			
		Python 镜像构建测试 Python 镜像构建及单元测试	Python 打包测试 Python 打包及单元测试	<b>Go 构建测试上传</b> Go 构建测试并上传			
		107 E					

## 图2.添加构建任务

## 配置构建任务

用户可以通过在自定义构建脚本的方式配置构建任务。

## 构建中使用环境变量

可在构建脚本中使用的环境分为系统变量和用户自定义的全局变量,系统变量包含:

\$PIPELINE_ID	#	流水线ID
\$PIPELINE_NAME	#	流水线名称
\$BUILD_NUMBER	#	流水线运行实例编号
\$EMPLOYEE_ID	#	<b>触发流水线用户</b> ID
\$PROJECT_DIR	#	代码库根路径,默认为/root/workspace/code

## 5.5.1. 构建制品

Flow 通过各种构建组件,对各种语言提供了制品打包能力,让用户可以快速的使用流水线构建制品,并通过后续的部署任务进行部署。

## 创建构建任务

云效2020

在流水线中通过添加构建任务,可以构建出制品,用于部署。Flow 提供了多种方式快捷的使用构建能力:

- 1. 新建流水线时,选择构建相关模版;
- 2. 编辑流水线时,添加构建任务;

## 流水线模版创建构建任务

新建流水线时,选择对应的开发语言,可以查看当前语言下的默认流水线模版,选择带有"构建"节点的流水线模版,即可快速使用构建能力。

☆ 我的星标	我的星标			et 🛛 🕤 🕯	<b>斤建流水线</b>
<ul> <li></li></ul>		选择流水线模板	×	最近运行时间	
⇒ 主即加小线	♀ 空流水线	РНР	杨超越	2019-09-24	*
设置信息 1 ⁶ 3 企业设置	🐇 Java	PHP 测试、构建、部署到主机	杨超越	2019-09-24	*
	JS NodeJS	代码扫描	刘吴然	2019-09-24	*
	GO GO	雷··语 题《叶 近版	吴彦祖	2019-09-23	*
	Python		杨超越	2019-09-24	*
	20 具他	PHP测试、构建 代码扫描	刘德华	2019-09-23	*
		构建	周杰伦	2019-09-23	*
		测试	刘昊然	2019-09-24	*
		< 1 2 >			

## 添加构建任务创建构建任务

在流水线编排中,可以通过添加任务,在模版中选择带有"代码扫描"的模版,可以快速的添加代码扫描任务

源	C actors a	20 2 <del>2</del> 9	************************************		新的阶段		
(-) flow/flow onging			远挥仕穷失望	~			
l⊱ master	9、请输入任务名称	■ 新約任务					
	空模板	测试	测试				
	代码扫描 测试	Maven 単元測试 Maven 单元測试	Gradle 单元测试 Gradle 单元测试	nede NodeJS 单元测试 NodeJS 单元测试			
	测试构建 1	Python 单元测试 Python 单元测试	PHP Codeception 单元 PHP Codeception 单元测试	PHP Unit 单元测试 PHP Unit 单元测试			
	镜像构建	<b>○CO</b> Go 単元測试 Go 単元測试					
	代码	测试构建	测试构建				
	执行命令	<b>Java 单元測试、构建</b> Java单元測试、构建、构建	nede NodeJS 构建测试上传 NodeJS 单元源试、构建、	nede React 构建测试上传 React 构建测试并上传至 Oss			
		Python 镜像构建测试 Python 镜像构建及单元测试	Python 打包测试 Python 打包及单元测试	<b>Go 构建测试上传</b> Go 构建测试并上传			
		\$f170					

## 配置构建任务

用户可以通过在自定义构建脚本的方式配置构建任务。

#### 构建中使用环境变量

可在构建脚本中使用的环境分为系统变量和用户自定义的全局变量,系统变量包含:

SPIPELINE ID # 流水线	D
\$PIPELINE NAME # 流水线	名称
	运行实例编号
\$EMPLOYEE_ID # 触发流れ	<b>K线用户</b> ID
\$PROJECT_DIR # 代码库相	<b>艮路径,默认为</b> /root/workspace/cod

## 5.5.2. Maven 仓库

#### Maven 配置

Flow 会自动帮你代理阿里云 Maven 公库 和 云效企业私库 Packages,若您需要使用企业自建的 Maven 私库,或者需要使用自定义的 Maven 配置,可以通过本文完成设置。 Flow 支持通过三种方式设置 settings.xml 完成 Maven 配置:

- 代码库根目录 settings.xml 文件
- Flow 企业设置 -> Maven 设置中上传 settings.xml 文件
- 云效默认的 Maven 配置 settings.xml 文件
- 目前, Maven 配置仅影响 Java 构建, 优先级如下:
- 1. Flow 会优先使用构建代码库的根目录下 Maven 的 settings.xml 文件中配置的依赖库,该配置仅影响与此代码库相关的流水线。同时需要在构建指令中,指定setting.xml 文件。

mvn -s settings.xml clean package

2. 如果用户在企业设置 -> Maven 设置中选择自定义 Maven 配置,且上传了 settings.xml 文件,则该企业内所有的 Java 构建均使用此配置文件。



3. 如果用户项目代码库的根目录没有 Maven settings.xml 且没有选择企业自定义 Maven 配置, Flow 构建时会为用户自动生成一个 settings.xml 文件,并且配置连接阿里云 企业私有仓库和阿里云公共代理仓库。

用户通过 Flow 构建时,会默认优先从阿里云企业私有仓库下载依赖,若无法寻找到构建依赖则会从阿里云公共代理仓库下载。

## 5.5.3. Go 镜像代理

阿里云提供了官方的 Go Module代理仓库服务,镜像源地址: https://mirrors.aliyun.com/goproxy/。 Flow Go 构建默认会通过该该代理来避免 DNS 污染导致的模块拉取缓慢或失败的问题,加速构建。

## 5.5.4. NPM 仓库

Flow 已经完成了与Packages 的 NPM 仓库的自动打通,当你在 Flow 中进行 Nodejs 构建,使用npm install指令,Flow 会优先从你 Packages 的 NPM 仓库中下载依赖文件。同时,Packages 为你自动代理 <mark>淘宝NPM镜像</mark>(NPM 镜像站为外部依赖公共服务,其服务稳定性非云效自身服务范围),您无需额外配置就可下载公共依赖文件。 注意:目前支持npm install指令从 Packages 下载依赖文件; cnpm install直接通过 淘宝NPM镜像下载资源,暂时无法通过 Packages 下载您的企业私有依赖文件。 如果你需要在构建时,下载自己的依赖文件,可以通过以下方式完成配置:

• 可以将您需要的构建依赖文件,直接上传至Packages,具体上传方式可参考仓库指南。

- 道浦 🌈 体验透镜 🛛 😏 🕐 🗶 🔢 📔 首页 > Npm > npm-registry 仓库指南 包文件 Q 成员 ② 仓库设置 新建代理仓库  $\times$ 仓库名称 * ⑦ 添加代理 基本信息 请输入仓库名称 仓库代理 仓库地址* 操作 用户名 密码 取消
- 如果您已经有了自己的私有仓库,可以将您的私有仓库添加至 Packages中的代理仓库。私有 NPM 仓库需要公网可以访问(或者 IP 白名单添加 Flow 构建集群的出口IP: Flow IP 地址)

- 如果您已有私有的 NPM 仓库,并且希望 Flow 在构建时可以直接从您的私有仓库中下载依赖文件,您可以使用企业自定义的.npmrc 文件,并将其放置在您的代码根目录下。您可以参考以下步骤生成.npmrc 文件:
- 。 执行命令 `npm config set registry YOUR_REGISTORY`, YOUR_REGISTORY为私有NPM仓库地址
- 。执行命令 `npm adduser`,交互输入访问用户私有NPM仓库的可用的用户名和密码
- 找到~/.npmrc文件,将该.npmrc文件放置到代码库根目录

## 5.6. 制品管理

## 上传到飞流的制品库

使用构建物上传步骤默认会上传到飞流的制品库。

## 上传到阿里云 OSS

用户可以将构建出的制品上传到用户的 OSS bucket中。

#### 上传到Maven私服

用户可以将制品上传到自己的 Maven 私服中。

## 5.6.1. 制品管理

## 归档至云效公共存储空间

使用构建物上传步骤默认会归档至云效公共存储空间。

#### 上传至 Packages 通用制品仓库

支持将制品上传至 Packages 通用制品仓库,对制品进行统一管理。

## 上传到阿里云 OSS

用户可以将构建出的制品上传到用户的 OSS bucket中。

## 上传至阿里云 OSS。

上传到Maven私服 用户可以将制品上传到自己的 Maven 私服中。 上传至 Maven 私服

## 5.6.2. 归档至云效公共存储空间

Flow 支持将工作空间中的文件或文件夹归档至云效公共存储空间。归档的目的主要有两个:

1. 制品归档后,会提供制品的下载地址,供后续的部署任务拉取制品并执行部署任务。

2. 制品归档后,制品文件会和流水线的运行记录进行绑定,并展示在流水线产物中。通过<mark>配置 Flow 流水线源</mark>的方式,可以实现多条流水线之间的制品传递。

#### 如何上传制品

下面以 Java 构建为示例,在飞流流水线首页点击新建流水线,并选择" Java 测试、构建"。

	女 Flow	1891 (BB)	) 🍊 442589 🕂 😫 🕐 🌔
ţ	我的流水线	选择流水线模板	× <u>新建流水线</u>
æ	<b>我参与的</b> 我的收藏 	<ul> <li>※ 快速配置</li> <li></li></ul>	
Ð	□ 流水线名称	Java         代码扫描         构建         部署	¢
	appstack;	php PHP 测试 1	* • • • •
	流水线 20	GO G	; * • •••
	代码过滤》	♣ Python 代码扫描 构建	· * • ···
	流水线 20	NET .NET Core         0014           © C++         014	• • • • • • • • • • • • • • • • • • • •
	测试离职f	🤞 Java · 测试、构建镜像 □ 和示ht≝	↓ <u></u> ★ ••••
		取消 (二)	

在编辑页面选择" Java 构建上传" 节点,在任务步骤中可以查看和配置"构建物上传"步骤,上传方式选择"归档至云效公共存储空间"

く 返回   流水线 2021-10-15	基本信息 流程画	2置 触发设置	变量和缓存	存 仅保存 保	存并运行
え の 构建		新阶段	1	编辑 ①	×
Java 代码扫描     ダ     Java 构建       Java 代码扫描     ダ     Java 构建	上传	€ 新的任务		<ul> <li>Java 构建</li> <li>构建物上传</li> <li>步骤名称</li> <li>构建物上传</li> </ul>	0
U HITS				上传方式 <ul> <li>」月档至云效公共存储空间(供后续任务拉取并部署) ⑦</li> <li>企业私有通用制品仓库(制品将上传Packages统一管)</li> <li>制品名称 ⑦</li> </ul>	) 理) ⑦
			1	Artifacts_\${PIPELINE_ID} 打包路径 ⑦ target/	(†) (†)
				ц принт Залл Вилтрурж 🕖 Нујан	

可以为制品选取一个名称,该名称在流水线后续阶段(比如部署阶段)会被引用到。

用户可以指定打包路径,指定你希望打包至制品中的文件路径,制品文件打包后用以后续任务拉取部署。该路径可以为一个文件或文件夹,系统会将其打包为一个压缩包。如 果用户想将多个文件或文件夹打包,也可以点点击"+"号添加更多的打包路径。

当流水线运行后,可以在单个流水线详情页下载构建包以及查看日志。
👬 🗧 首页 > 流水线 2021-10-15 👻	*		邀请 🌔 体验透镜 🕂 🔮 🕐 💸
<b>最近运行</b> 运行历史			<ul> <li>运行 // 编辑 // 遂请 三菜单</li> </ul>
#1 🔮 运行成功	测试	构建	
运行人 🗶 光晕	♂ Java 代码扫描	♂ Java 构建上传	
开始时间 2021-10-15 11:36:57	2         0         0         2           总数         阻塞         严重         一般	Artifacts_1474317	
持续时间 1分27秒 产物 1项	同 扫描报告 凹 日志 12秒	日志 28秒	
→ 源(1)	<ul> <li>✓ Maven 単元測试</li> <li>3 / 4 1 1 75.00%</li> <li>潮试用例 失敗 跳过 通过率</li> </ul>		
	Geomatica E ruor 2009		

### 如何使用上传制品

当你在构建中添加了"构建物上传"的步骤后,在流水线中的后续阶段任务中可以使用"构建物上传"中配置的制品。

以"主机部署"为例。

く 返回     流水线 2021-10-15	基本信息 流程配置	触发设置 变量和	缓存 仅保存	保存并运行
書略			编辑	>
Java 构建上传 🕒 😡 主机部署	+ ) (	新的任务	任务名称	
			土机能者 ✓ 部業时下裁制品	
			制品	
		1		
			制品名称.Artifacts_\${PIPELINE_ID} 请选择主机组/更多主机组请按名称搜索	~
			部署配置	
			下载路径 ②	
			/home/admin/app/package.tgz	
			执行用户 ⑦	
			root	
			您可以在部署脚本中直接使用环境变量,无需再 Base64解码处理。但是如果你的环境变量中包 符,你将无法正常使用您的环境变量 查希契	再进行 ☆特殊字 助

在制品栏可以选取前面阶段任务中的制品,并部署到指定的主机中。

## 5.6.3. 上传至 Packages 通用制品仓库

流水线软件包仓库,可以帮你统一管理企业使用 Flow 流水线构建出的软件包制品。主要功能:

- 自定义制品名称和制品版本,对企业的构建产物进行有序管理。
- 存储与制品相关的测试、构建、部署元数据,保障研发流程信息可追溯。
- 设置仓库权限,确保企业构建物安全可靠。

### 囗〉注意

制品上传至 Packages 通用制品仓库后,制品文件并未完成归档并和流水线的运行记录进行绑定。因此如果希望通过<mark>配置 Flow 流水线源</mark>的方式,实现多条流水线之间的制品 传递,需要在上传 Packages 通用制品仓库后,对制品文件进行归档操作,详情查看<mark>归档至云效公共存储空间。</mark>

#### 上传下载制品

在 Flow 流水线中,通过构建物上传,选择"企业通用私有制品仓库"。分别填写:制品名称、制品版本、打包路径。 如果你没有仓库的使用权限,将无法配置上传任务到软件包仓库,请联系仓库管理员为您配置仓库权限。 注意:通过制品名称和制品版本确定唯一的制品信息,因此同一制品的同一版本不允许重复推送。

《 返回   流水线 2021-03-10	至本信息 <b>識和</b>	<b>記言</b> 触发设置 交通和缓存	仅保存 保存并)
执行意	• E00	編編 ①	
👂 flow-example/spring-boot	· · · · · · · · · · · · · · · · · · ·	- 新版构建物	上作
⊨ master		步骤名称	
十添加代码源		807 ND 444 HD	£19
		上传方式	
		○ 云效公共	,存储空间(供后续任务拉取制品并部署)
		○ 企业私有	.通用制品仓库(制品将上传Packages统一管理)
		八 光晕	
		彩品名称	
		Artifacts	_\${PIPELINE_ID}
		制品版本	
		\${DATE1	IME}
		打包路径	
		target/	
		- 制品件	包含打包路径的目录 ⑦

#### 可通过制品页面直接下载你构建的制品产物。

### 制品部署

在部署任务节点选择构建制品,部署任务会从制品仓库下载制品并完成发布。

く 返回  流水线 2021-03-10		基本信息 流程配置 触发	议置 安量和缓存		仅保存 保存并适
	执行命令	部署	\$11.00 FZ	1818 T	
flow-avample/enring-boot			a senses	8665	
non-oxemployoping coor	y survey	y Tubu	( alkaliza	制品名称.Artifacts_\$(PIPELINE_ID)	
⊢ master				主机组	O MILE
十添加代码源				新的主机组	
				部者配置	
				下载路径 ③	
				/home/admin/app/package.tgz	
				执行用户 ③	
				root	
				② 您可以在部署脚本中直接使用环境变 解码处理。但是如果你的环境变量中 正常使用您的环境变量,直看帮助	里,无需再进行Baset 包含特殊字符,您将为
				部署脚本 ③	
				主 # 単単生を活用用単純に # 単単連載の下:先移動能 約下電量後) 単広境気能 単本(重素を代码中差別・、 、か)、大手に介用予約# ://thoughts.cl/sun.con(5) /select39546f290010xe831f /select39546f290010xe831f	ii上执行、一个商型 (在下着器径中配置 目景中、再执行启动 即示例中deploy II解释见 https horespace Z/docs

### 制品管理

进入Packages (https://packages.aliyun.com)通用制品仓库,查看并管理您通过流水线上传的制品文件。

111 2 首页 > 通用制品 > 流水线软件包	1仓库	Artifacts_1344505 版本 2021-06-07-10-17-34(latest) > 版本数 1 总下载次数	0 ×
仓库捐商 包文件列表 流水线软件包仓库 全面内可定		文件信息 元政派 	Ø
包名	最新版本	<b>最近更新</b> 推送人 推送时间	
Artifacts_1344505	2021-06-07-10-17-34	光晕(5e71d6f5u220uuu8/f91c2) 2021-06-07 10:17:54	
		ご 文仲在名       大小         12211 3.m       13.7 MB         北文作下電次数       原本描述         0       没有原本描述         ご 電鉄店本	

### 5.6.4. 上传至阿里云 OSS

Flow支持将构建工作空间中的文件或者文件夹上传至阿里云 OSS。

### 前提条件

```
1. 有一个阿里云账号,并在其中创建了一个 OSS 的 bucket。
```

2. 一般来讲,要上传的内容可以来自:

```
i. 代码源
```

ii. 构建之后的产物。这种情况,你需要先配置需要的构建步骤(比如前端静态资源构建)

### 使用 OSS 上传步骤

下面以 React 项目构建并上传至 OSS 为例进行说明。

在首页新建流水线,并选择"React 构建上传到 OSS"的模板。

☆ 我的星标	我参与的		(1) 🖘 🕒 新建流水线
② 我参与的		选择流水线模板 ×	
与 全部流水线	□ 空流水线	React 构建上传到 OSS	
	👙 Java		
(2	JS NodeJS	代码扫描	
<u> </u>	php PHP	3 构建上tfg	
	GO GO	103 Hz	
	🚭 Python	NodeJS 测试、构建、部署到主机	
	23 其他	代码扫描 的理 刻试	
		NodeJS 测试、构建	

在流水线创建向导中选择好代码源之后,进入流水线编辑页面。

源	🚯 新建源	构建上传	新的阶段编辑	
C-) group/repo			- oss ±( 2	
}∞ master	xt 代码扫描	t ➡ ♀ React 构建上借	● ● 新的任务	
	自元测试		OSS F#	
	E7 UNA INV	<u> </u>	000 TIA	
			选择授权	新建服务
			选择授权不能为空	
			Bucket区域	
			Bucket区域不能为空	
			Bucket名称	
			Bucket名称不能为空	
			Bucket目录	
			源文件目录	

为了与你的阿里云账号建立连接,点击"新建服务授权",然后点击"RAM 授权"。

R	新建源	构建上传	新的阶段	编辑	
〔−〕 group/repo }∾ master	xt 代码扫描	日 React 构建上传	<ul> <li>① 日 ① 新的任务</li> </ul>	- OSS 上传 	0 1
	自元测试		新建服务授权	× _传	
		服务膜状类型 阿里云对象存储服务 服务膜状名称 请输入服务度状名称 阿里云膜状 当前账号未进行阿里云RAM模状	点击进行 RAM授权 2	又 不能力空 区域 如不能为空 各称	(1) ♥ 新建服务授权 ✓ ✓ ✓
			取消	称不能为空 保存 目录	
				源文件目录	0

你会被要求登录阿里云账号,输入你希望使用阿里云账号进行登录,并点击通然后会自动跳转会到飞流的页面,并显示 RAM 授权的页面。

☰ (-)阿里云		Q. 搜索		费用	工单	备案	企业	支持与服务	Þ-	۵.	Ä	0	ŝ	简体中文	0
1	云资源访问授权														
	温馨提示:如需修改角色权限	请前往RAM控制台角色管理中设置	,需要注意的是,错误的配置可能导导	改Teambition无法获取到必	。要的权限。							×			
	Teambition请求获取 下方是系统创建的可供Team	<b>方问您云资源的权限</b> bition使用的角色,授权后,Teambit	ion拥有对您云资源相应的访问权限。												
	AliyunTeambition 描述: Teambition默认的 权限描述: 用于Teambi	DefaultRole 用此角色来访问您在其他云产品中的 ion服务默认角色的授权策略	服资(									~			
			同意授权	取消											
			1												

点击"同意",会跳转回到飞流的页面。填写服务授权的名称,然后点击"保存"。

	新建源	构建上传	新的阶段	编辑	
← group/repo	xt 代码扫描	・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	① 日 ④ 新的任务	→ OSS 上传 务	0 1
	单元测试		新建服务授权	×	
		服务授权类型 原里元对象在储服务		2	新建服务授权
		内主口内政行间成为			× .
		服务授权名称		区域	
		cui.liqiang 的服务连接		17/20	×.
		阿里云授权 当前账号绑定的阿里云账号为 c	ui.liqiang,且已完成RAM授权	域不能为空 名称	
					~
			ЩX	旅不能为空           第           保存	0
				海大休日三	
				源又1十日來	

继续选择"Bucket 区域"和"Bucket 名称"。

源	新建源	构建上传	新的阶段	编辑	
C-) group/repo				- ▼ OSS 上传	
}⊸ master	Script 代码扫描	i 🗄 🦻 😯 React 构建上	传 😗 🖬 💿 新的任务	步骤名称	
	.js 单元测试	◎ 并行任务		OSS 上传	
				选择授权	新建服务授
				cui.liqiang 的服务连接	~
				Bucket区域	
			(1)-		~
				Bucket名称	
			(2)-	webplus-cn-beijing-s-5d299220153de44ff0e9f1c	19 ~
				Bucket目录	(
				react-oss-test	
				源文件目录	(
				build	

填写"Bucket 目录"和"源文件目录"。"源文件目录"可以是一个文件,或者一个目录。 保存并运行流水线。

运行成功之后,你可以点击相关按钮,查看上传日志及 OSS 中的内容。

最近运行 运行历史			● 运行   ◇ 编辑   凡 成员   三菜单
#4 🔮 运行成功	测试	构建上传	
- group/repo	✓ JavaScript 代码扫描	✓ React 构建上传	
‰ master -⊳- fdea4f8	6 0 6 0 总数 阻塞 严重 一般	运行成功	
	同扫描报告 目日志 31秒	♂上传地址 □日志 1分8秒	
	⊘ Node.js 单元测试	(2) (1)	
	运行成功		
	目日志 46秒		
运行人 😡 🕷	兰力		
开始时间 2019-09-26 16:03	:03		

上传日志:

最近运行 运行	历5	React 构建上传	$\times$	编辑	<b>风 成员</b>	三菜单
#A >==========	构建	OSS 上传(4s)				
#4 VE1376640	✓ 环境分派	[executionStep begins at 2019-09-26 16:04:47.690]				
- group/repo	✓ 环境初始化(1s)	[INFO] 获取 Credential 成功 [INFO] 使用工作路径/cont/workenace/102445-crown_react-example1_LITEk				
}⊷ master	⊘ 清理工作区(3s)	[INFO]PIPELINE_ID=17253				
-o- fdea4f8	✓ 克隆代码(17s)	[INFO]PIPELINE_NAME=流水线 2019-09-26 [INFO]BUILD_NUMBER=4				
	⊘ Node.js 构建(35s)	[INFO]EMPLOYEE_ID=5cbea131f4f6250001be73b1				
	✓ OSS 上传(4s)	[INFO]WORK_SPACE=/root/workspace [INFO]PROJECT_DIR=/root/workspace/102445-group_react-example1_UTRk				
		[INFO]BUILD_JOB_ID=29330236				
		現行用尸節令 ossutil cp -r /root/workspace/102445-group_react-example1_UTRk/build_oss://webplus-cn-beijing-s-				
		5d299220153de44ff0e9f1d9/react-oss-testupdate				
		Succeed: Total num: 19, size: 519,233. OK num: 19(upload 15 files, 4 directories). 0.158951(s) elapsed				
		[SUCCESS]上传成功				
		[SUCCESS]OSS_BUCKE1_UHL=https://oss.console.aliyun.com/bucket/oss-cn-beijing/webplus-cn-beijing-s- 5d299220153de44ff0e911d9/object				
		[INFO] 执行用户命令成功				
		[executionStep ends at 2019-09-26 16:04:52.125]				
运行人						
7744 04477 000	10					

ncc	rth.	上在的内区	3.
000	Ψ.	エコマロリアリモ	£.

☰ (-)阿里云			Q 搜索	费用 工单 备	案 企业 支持与服务	d ș ă	⑦ 斎 简体	†ż 📀
对象存储	对象	存储 / w	ebplus-cn-beijing-s-5d299220153de44ff0e9f1d9 / 文件管理				② 通过 SDK 管理文件	产品动态
概览		et ap	las-cr-begrig-s-54298220153c	读写权限 私有	类型 标准存储(本地冗余)	区域 华北2(北京)	创建时间 2019-07	7-13 16:11
存储空间 + 0 11 23		概览	文件管理 基础设置 域名管理 图片处理 事件通知 函数计算	云存储网关 智能媒体	日志查询 基础数	x据 热点统计 Al	기 统计 文件访问统计	;+
Bucket 名称 Q	Ŀ	传文件	新建目录 碎片管理 授权 批量操作 > 刷新				输入文件名前缀匹配	Q
			文作名(Object Name)	文件大小	存储类型	更新时间		操作
		6	/ react-oss-test/					
			static/					删除
	<	<>	asset-manifest.json	0.801KB	标准存储	2019年9月26日 16:04	详情	更多 🗸
		•••	favicon.ico	3.779KB	标准存储	2019年9月26日 16:04	详情	更多 🗸
		\$2	index.html	1.997KB	标准存储	2019年9月26日 16:04	详情	更多 🗸
		<>	manifestjson	0.299KB	标准存储	2019年9月26日 16:04	详情	更多 🗸
		\$	precache-manifest.99a64da829a3264e6c9f4be6750f3f0d.js	0.631KB	标准存储	2019年9月26日 16:04	详情	更多 🗸
		<>	service-worker.js	1.154KB	标准存储	2019年9月26日 16:04	详情	<b>7</b>

# 5.6.5. 上传至 Maven 私服

Flow支持将构建产出物上传至用户自己的 Maven 私服中。

### 前提条件

- 1. Maven 私服需要支持公网可访问,或至少飞流的构建机可以访问你的 Maven 私服。
- 2. 若您使用的Maven私服不是云效Maven企业私库,则需要在代码库根目录放置 Maven 的配置文件 settings.xml ,里面包含了私服的认证信息。
- 3. 代码库的 pom.xml 文件中配置了正确的 Deploy 插件。

下面以Java构建为示例,在流水线首页点击新建流水线,并选择"Java测试、构建"。

☆ 我的星标	我参与的			☴↓ 🕒 新建流水线
① 我参与的		选择流水线模板	×	最近运行时间
♀ 全部流水线	5 空流水线	Neg2-0世 构建 部署	ven	2019-09-29
设置信息	👙 Java	测试	ven	2019-09-26
② 企业设置	r Node,js php PHP co Go € Python	R码扫描	ven	2019-09-21
	88 其他	例鍵	ven	2019-09-23
			ven	2019-09-11
		Java 测试、作处距明瞭	ven	2019-09-17
		代码扫描	ven	2019-09-11

在流水线编辑页面选择" Java 构建上传"节点,删除掉"构建物上传"步骤。

Į	🕀 新建源		构建	新自	的阶段 编辑		_
<ul> <li>bowenhuang/bowen</li> <li>master</li> </ul>	2	Java 代码扫描 🔹 🛨 faven 单元测试	- ゆ Java 构建		新的任务 Java 构建上 任务步骤	ę	
					<ul> <li>         → 添加步骤         → Java 构建         → 构建物上街         </li> </ul>		Ť
					任务播件 予添加播件		
					任务输出 @ [制品] de	ault	
					<b>删除任务</b> 该任务将被彻/	底删除,注意这是不可逆操作,任务下所有数据将会删	刂除

### 展开 Java 构建步骤,修改构建命令为:

1. 云效 Maven 私库 <mark>Packages</mark>

mvn -B clean deploy -Dmaven.test.skip=true -Dautoconfig.skip

### 2. 自建 Maven 私库

mvn -s settings.xml -B clean deploy -Dmaven.test.skip=true -Dautoconfig.skip

<ul> <li>bowenhuang/bowen2</li> <li>master</li> </ul>	所建築 Java 代码扫描 taven 单元测试	构建 G Java 构建上传	Strong	編辑 任务步骤 ② 添加步骤 > Java 构建 步骤名称 Java 构建 请选择Java版本 java 長	
				<pre>pukto 构建命令 1 # movem build default command 2 mon = settings.xnl = 8 clean dploy -Dmaven.test.s</pre>	() kip

运行流水线,可以查看构建日志看到构建出来的二方库是否被上传到私有仓库中。

# 5.7. 自定义镜像构建

当云效的构建环境中预置的编译环境不能满足您的要求时。您可以使用自定义构建镜像的功能来定制所需的构建环境。

### 构建镜像

#### 您可以根据你的需要来编写构建环境使用的 Dockerfile,安装软件和设置环境变量。典型示例如下:

FROM registry.cn-beijing.aliyuncs.com/rdc-builds/base:1.0
RUN cd /tmp && \
wget http://rdc-public-software.oss-cn-hangzhou.aliyuncs.com/jdk-7u80-linux-x64.tar.gz && \
tar xf jdk-7u80-linux-x64.tar.gz -C /srv/java && \
ln -s /srv/java/jdk* /srv/java/jdk
ENV JAVA_HOME=/srv/java/jdk \
PATH=\${PATH=;farH}:/srv/java/jdk/bin:/srv/java

### 在本地调试通过后,将镜像上传到阿里云或其他公网可访问的registry。

如果您不使用 registry.cn-beijing.aliyuncs.com/rdc-builds/base:1.0 作为您的基础镜像时,您需要确保您的镜像中包含 jq和bash,并且确保默认用户为root。 通过以下命令验证,如果镜像可以正常启动,则可以在云效构建环境中使用。

docker run -w /root/workspace --entrypoint=/bin/bash -it \$image

# 配置自定义镜像构建节点

在流水线编排中配置自定义镜像构建节点。

### 云效流水线 Flow 构建

< 返回 Auto-chart上传		基本信息 流程配置 触发设置	置 变量和缓存		仅保存保存并运行
職	阶段1		新阶段	选择任务组	×
flow-example/spring-boot	6 自定义铁像构建	Chart上传自定义仓库	• Mithield	企业任务组       ① 请输入任务名称         代码扫描       代码         測试       化码         测试       ② 合井代码         测试       第代码编入温中描…         特定       ② 合井代码         增优肉給       ⑨ ⑦ 翻除分支         服告書面任务执行…       ⑨ ⑦ 翻除分支         影響       ⑨ ⑦ 翻除分支         代码       ● ⑦ 翻除分支         此行命令       使用 Gat tag命令创…         空磁板       • ① 文磁板         · · · · · · · · · · · · · · · · · · ·	分支管理器 分支管理器
输入您的镜像地址和构建命令,女	11果您的镜像为非公开镜像,也可以	使用私有构建镜像。			
< 返回 Auto-chart上传		基本信息 流程配置 触发设置	重 变量和缓存		仅保存保存并运行
205	附行段1		新阶段	编辑 ⑪	×



# 5.8. 移动端构建

Flow 目前支持移动端的构建场景包括 Android 和 微信小程序。

#### Android 构建

Flow 支持 Emas Android 构建。移动研发平台(Enterprise Mobile Application Studio,简称EMAS),为企业、开发者提供一站式的应用研发管理服务,涵盖开发、测试、运 维、运营等应用全生命周期。

### 微信小程序构建发布

您的项目工程可通过集成微信小程序发布工具miniprogram-ci来实现流水线构建并发布的功能。

### 5.8.1. 移动端构建

Flow 目前支持移动端的构建场景包括 Android 和 微信小程序。

#### Android 构建

Flow 支持 Emas Android 构建。移动研发平台(Enterprise Mobile Application Studio,简称EMAS),为企业、开发者提供一站式的应用研发管理服务,涵盖开发、测试、运 维、运营等应用全生命周期。

#### Android 构建

### 微信小程序构建发布

您的项目工程可通过集成微信小程序发布工具miniprogram-ci来实现流水线构建并发布的功能。

#### 微信小程序构建发布

### 5.8.2. Android 构建

Flow 通过集成阿里云 EMAS 服务支持移动端的构建场景,目前已经支持 EMAS Android 构建。在使用 Android 构建任务前,请确保您已经开通了阿里云 EMAS 服务。阿里巴 巴应用研发平台 EMAS 是阿里云自研的云原生应用研发平台(移动App、H5应用、小程序、Web应用等),基于广泛的云原生技术(Backend as a Service、Serverless、 DevOps、低代码等),致力于为企业、开发者提供一站式的应用研发管理服务,涵盖开发、测试、运维、运营等应用全生命周期。更多 EMAS 功能请<mark>查看帮助。</mark>

### 配置 Android 构建任务

创建流水线,选择 Android 流水线模板。

我的流水线	副让出现倾权	选择流水线模板	×	et &	
<b>我参与的</b> 我的星柱	🛓 Java	↓ 移动端 · Emas Android 构建			
□ 流水线名称	Php PHP	Emas Android 构建			
Emas and	🕼 Node.js				☆ ⊛ …
_ 流水线 20:	GO GO	···· 其他·执行命令			☆ ⊛ …
□ 流水线 20:	Python	执行命令		18:06	合 🕑 …
_ 流水线 20:	NET .NET Core	··· 其他 · 函数计算 (FC) 应用发布			☆ ⊛ …
□ 流水线 20:	<ul><li>C++</li><li>移动端</li></ul>	函数计算(FC) 应			☆ ⊙ …
测试 pipeli	其他	··· 其他 - 空模板		10:35	\$ D
流水线 20:			取消 创建	23:56	☆ ⊛ …

配置 Android 构建任务,需要选择 EMAS 服务授权、产品名称、应用名称、构建流水线,其中产品名称、应用名称、构建流水线均需要在 EMAS 中进行配置;可以在 Flow 流 水线中快速配置 Emas 产品、应用。

< 返回 Emas android pipeline	基本信息 流程配到	触发设置 变量和缓存	(又)	保存并运行
移动构建		新阶段	编辑 ⑪	:
flow-example/emas-example	Emas Android构建	● 新的任务	构建集群 ⑦ 云效北京构建集群	~
⊱ master			任务步骤	
十添加代码源			选择授权	新建服务授权
			skyhook520@163.com的emas服务授权	~
			产品 ⑦	新建 Emas 产品
		(1)	emas_test_app	~
			应用 ⑦	<table-cell-rows> 新建 Emas 应用</table-cell-rows>
			emas_test_android	~
			流水线 ②	
			默认应用构建流水线	~
			任务插件	
			还加插件	

### 执行构建任务

执行流水线运行后,可通过构建详情进入 EMAS 构建任务,查看构建的具体进展。

ⅲ 扂 首页 > 1	nas android pipeline 👻 🚖	
最近运行运	历史	
REXLIGIT 运 1 ④ 运行中 行人 计始时间 *物 *项(1)	100万円 ● 修止 ● 修止 ● 修止 ● 修止 ● 修止 ● 修止 ● 修止 ● 修油 ● 使用as Android特違 ● 使用 単行中 任务状态同時 ● 取用 更多 へ ● 物能详術 回 日本	5¢ 19160

构建完成后,可在流水线详情页直接下载构建包,也可直接扫码安装。

🔢 🗾 首页 > En	mas android pipeline 👻	*
<b>最近运行</b> 运行	行历史	「聲」」
#2 🥑 运行成功		
运行人	🗶 光晕	- - A d构建
开始时间 20	021-02-24 15:07:47	app-release-unsigned.apk
持续时间	4分40秒	
产物	O项	
▶ 源(1)		

### 5.8.3. 微信小程序构建发布

您的微信小程序项目工程,可以通过集成微信小程序发布工具 miniprogram-ci来实现一键发布。 miniprogram-ci 是从 微信开发者工具 中抽离的关于小程序/小游戏项目代码的编译模块。 开发者可不打开小程序开发者工具,独立使用 miniprogram-ci 进行小程序代码的上传、预览等操作。 本文通过一个小程序的示例工程,通过集成 miniprogram-ci,结合 Flow 实现一键发布。

### 微信小程序工程集成 miniprogram-ci

示例工程集成了 miniprogram-ci, 请关注几个关键文件:

- package.json: 前端项目需要依赖 miniprogram-ci。
- private.appid.key: 小程序上传秘钥文件。
- upload.js:调用 miniprogram-ci 完成小程序上传函数。



示例工程的 package.json 文件如下:

{	
	"name": "miniprogram-preview",
	"version": "0.0.1",
	"private": true,
	"scripts": {
	"upload": "node conf/upload.js"
	},
	"dependencies": {
	"miniprogram-ci": "^1.0.94"
	},
	"wx": {
	"wxVersion": "1.1.9",
	"wxDesc": "测试一下"
	}
}	

dependencies 中引用 miniprogram-ci, script 中调用 upload.js 中的上传函数。

#### private.appid.key

小程序上传秘钥文件,可以在小程序管理后台->开发管理开发设置小程序代码上传设置中直接下载。 示例项目里的 private.appid.key 文件中,已把秘钥信息隐藏,请替换为自己的秘钥文件。

	版本管理成员管理	开发者ID		
	用户反馈	开发者ID	操作	
¢	统计	AppiD(小程序/D) wx9b042ae665586d69		
	功能			
	微信搜一搜客服	AppSecret(小程序密钥)	生成	
	订阅消息			
	以面內容接入 品牌小程序	小程序代码上传 开发者可基于配置信息调用限信开发者工具提供的代码上传模块。宣看详情		
	开发	配置信息	操作	
	开发管理 开发工具 云开发	小程序代码上传密钥	重置 ⑦	
۷	成长	IP白名单 暂无IP白名单	IP 白名单禁用后将增加代码上传风险,请谨慎操作 ① 编辑	
	违规记录 			
<u>-</u>				

### 囗 注意

在这个地方,如果您要设置上传白名单,请将 Flow 构建机的 IP 地址加入白名单中,Flow 才可以实现打包上传。Flow IP 地址

upload.js

示例工程的 upload.js 文件如下:

### 云效流水线 Flow·构建

const ci = require('miniprogram-ci')
<pre>let { wxVersion: version, wxDesc: desc } = require('/package.json').wx</pre>
if (!version) version = 'v1.0.0'
if (!desc) desc = new Date() + '上传'
const project = new ci.Project({
appid: 'wx9b042ae665586d69',
type: 'miniProgram',
<pre>projectPath: process.cwd(),</pre>
<pre>privateKeyPath: process.cwd() + '/conf/private.wx9b042ae665586d69.key',</pre>
ignores: ['node_modules/**/*'],
})
ci.upload({
project,
version,
desc,
setting: {
minify: true,
},
<pre>}).then(res =&gt; {</pre>
console.log(res)
console.log( <b>'上传成功'</b> )
<pre>}).catch(error =&gt; {</pre>
if (error.errCode == -1) {
console.log( <b>'上传成功'</b> )
}
console.log(error)
console.log( <b>'上传失败'</b> )
process.exit(-1)
})

#### 其中:

参数值	说明
appid	小程序/小游戏项目的 appid
type	项目的类型,有效值 miniProgram/miniProgramPlugin/miniGame/miniGamePlugin
projectPath	项目的路径,即 project.config.json 所在的目录
privateKeyPath	私钥文件地址
ignores	指定需要排除的规则

### Flow 完成小程序一键发布

完成项目工程配置后,您只需要在 Flow 里对您的小程序进行正常的前端工程构建编译,然后调用 miniprogram-ci 即可完成一键发布。以示例项目举例,你需要执行以下命 令:

cnpm install npm run upload

	首页 > 小程序测	试流水线 👻 🊖				邀请 🌘 体验透镜	0	99 <del>1</del>	3 ጰ
服	<b>近运行</b> 运行历史				Node.js 构建	×	编辑	鸟 成员	三菜单
#1 🖸	运行成功	构建	Node.j	; 构建(85s)		下载			
		♥申请运行环境(27s)	[20:5]	8:46] done:   8:46] done:	utils/util.js pages/index/index.is				
运行人			[20:5]	:46] proces	sing: app.wxss				
开始时间	0 2020-12	⊘ 克隆代码(3s)	[20:5]	3:46] proces 3:46] proces	sing: pages/index/index.wxss sing: pages/logs/logs.wxss				
持续时间	ŋ	⊘ 流水线缓存(2s)	[20:5]	8:48] done:	bages/index.wxss				
		Node.js 构建(85s)	[20:5]	3:48] done:	app.wxss				
产物			[20:5]	8:48] proces: 8:48] done:	sing: pages/index/index.wxml pages/index/index.wxml				
▶ 源(1	)		[20:5]	3:48] proces	sing: pages/logs/logs.wxml				
			[20:5]	8:48] done:   8:48] proces:	bages/logs/logs.wxml sing: compiling other files				
			[20:5]	8:48] done:	compiling other files				
			[20:5]	3:48] reques	t url: https://servicewechat.com/wxa/ci/upload?				
			codep [20:5]	otect=0&typ 8:49] done:	e=miniProgram&appid=wx9b042ae665586d69&version=1.1.9&desc=%e6%85%88%e8%AF%95 µpload	%E4%B8%80%E4%B8%8 }			
			[20:5]	8:49] { 8:49] cubP	ockaneTofo: [/ name: ' EUL ' cize: 2005 } ]				
			[20:5]	3:49] plug	inInfo: []				
			[20:5]	3:49] } 3:49] 上传成功					
			[20:5]	3:58] npm in	fo lifecycle miniprogram-preview@0.0.1~postupload: miniprogram-preview@0.0.1				
			[20:5]	3:58] npm in	fo ok				
			[20:5]	3:58] [SUCCE	55】步骤运行成功				
	功能								
	微信搜一搜		甲核版本						
	客服		版本号	开发者	ci机器人1	244.88			
	订阅消息		1.1.9	提交审核时间	2020-12-16 20:58:02	1年1月	ľ		
	页面内容接入		审核中	而日条计	御は一下				
	品牌小程序				//5 LL				
		_						_	
	廾友		TH 44 115 -						
	开发管理		开友版平						
	开发工具		版本号	开发者	ci机器人1	坝水市核			
	云开发		1.1.9	提交时间	2020-12-16 20:53:49	定人中议			
				项目备注	测试一下				
4	成长			A PA PA PA CAL					
	小程序评测								
	法担记录								
	1976 IG 3K								
-	推广								
	流量主								
	广告主								
-	い平								
javasč(in).;	以直								

# 6.部署 6.1. 新建部署任务

为了创建部署组,需要先在流水线中添加「主机部署」任务。用户可以通过以下方式添加部署任务。

- 使用模板新建流水线,选择包含"部署"任务的模板
- 在已有的流水线中,添加新的阶段,并选择「部署」任务

### 流水线模版创建

新建流水线时,选择对应的开发语言,可以查看当前语言下的默认流水线模版,选择带有"<mark>部署</mark>"节点的流水线模版,即可快速使用构建能力 。

☆ 我的星标	我的星标			EL 🛛 🚯 🛙	听建流水线
② 我參与的		选择流水线模板	×	最近运行时间	
⇒ 全部流水线	5 空流水线	PHP	杨超越	2019-09-24	*
设置信息	🐇 Java	PHP 测试、构建、部署到主机	杨超越	2019-09-24	*
約 正亚说重	JS NodeJS	代码扫描	刘昊然	2019-09-24	*
	GO GO	約建 卸署	吴彦祖	2019-09-23	*
	n Python		杨超越	2019-09-24	*
	品 其他	PHP 测试、构建	刘德华	2019-09-23	*
		代码扫描 构建	周杰伦	2019-09-23	*
		测试	刘昊然	2019-09-24	*
		相違	刘昊然	2019-09-24	*

### 编排流水线,添加部署任务

源	C 012 20 101 3	2017		口体部第	新的阶段
63. A			选择仕务类型	X	
l−J flow/flow-engine	9、请输入任务名称	称			● ● 新的任务
	空模板	测试			
	代码扫描 测试	Maven 単元測试           Maven 単元測试	Gradle 单元测试 Gradle 单元测试	nede NodeJS 单元测试 NodeJS 单元测试	
	测试构建	Python 单元测试 Python 单元测试	PHP Codeception 单元 PHP Codeception 单元测试	PHP Unit 单元测试 PHP Unit 单元测试	
	镜像构建	<b>Go 单元测试</b> Go 单元测试			
	代码	測试构建			
	执行命令	Java 单元测试、构建           Java单元测试、构建、构建	nede NodeJS 构建测试上传 NodeJS 单元测试、构建、	React 构建测试上传 React 构建测试并上传至 Oss	
		Python 镜像构建测试 Python 镜像构建及单元测试	Python 打包测试 Python 打包及单元测试	<b>GO 构建测试上传</b> Go 构建测试并上传	
		#1720			

# 6.2. 主机部署

Flow 提供了通用的部署能力,支持业务被部署到不同国家,不同云厂商环境还有你的私有环境的主机中。

Flow 目前支持部署以下的主机类型:

- 1. 阿里云ECS
- 2. 可以出公网的非阿里云主机
- 3. 不可访问公网的非阿里云主机(即将上线)

### 添加主机部署任务

可以通过两种形式添加部署任务:

1. 创建流水线时候,选择包含部署任务的模板
 2. 在已有流水线中添加新任务,并选择部署任务

如何添加部署任务,可查看"新建部署任务"一节

### 配置部署任务

进入部署任务,进行如下配置:

 选择制品,制品来源于上游的「构建」任务。关于如何得到构建制品,可查看"构建制品"一节
 创建或者选择「部署组」,部署组用于管理部署的资源和配置。关于如何创建部署组,可查看"主机组"一节
 通过「部署配置」,配置制品的下载路径,执行用户和具体的部署执行命令。如何进行部署配置,可查看"部署配置"一节
 配置「部署策略」,避免不稳定发布对用户的影响,保障业务交付的稳定。如何进行部署策略,可查看"部署策略"一节

### 查看部署详情

配置完部署任务后,运行流水线执行部署任务,即会将对应的制品部署到指定的部署组。在部署过程中,可以点击部署任务,获取最新的部署详情,取保部署的正确性。 如何查看部署配置,可查看"查看部署详情"一节

### 6.2.1. 主机部署

Flow 提供了通用的部署能力,支持业务被部署到不同国家,不同云厂商环境还有你的私有环境的主机中。 Flow 目前支持部署以下的主机类型: 1.阿里云ECS 2.可以出公网的非阿里云主机

### 添加主机部署任务

可以通过两种形式添加部署任务: 1. 创建流水线时候,选择包含部署任务的模板 2. 在已有流水线中添加新任务,并选择部署任务 如何添加部署任务,可查看"新建部署任务"一节

#### 配置部署任务

进入部署任务,进行如下配置: 1.选择制品,制品来源于上游的「构建」任务。 关于如何得到构建制品,可查看"构建制品"一节 2.创建或者选择「主机组」,部署组用于管理部署的资源和配置。关 于如何创建部署组,可查看"主机组"一节 3.通过「部署配置」,配置制品的下载路径,执行用户和具体的部署执行命令。 如何进行部署配置,可查看"部署配置"一节 4.配置「部署策略」,避免不稳定发布对用户的影响,保障业务交付的稳定。 如何进行配置部署策略,可查看"部署衰略"一节 5. 部署时使用环境变量参考环境变量,部署使用环境变量部分。

#### 查看部署详情

配置完部署任务后,运行流水线执行部署任务,即会将对应的制品部署到指定的部署组。 在部署过程中,可以点击部署任务,获取最新的部署详情,取保部署的正确性。 如何查看部署配置,可查看"查看部署详情"一节

### 6.2.2. 主机组

### 主机组是什么

每个 Web 应用,在集成测试的环境(通常称作日常环境)、预发的环境(称作预发环境)、对外提供服务的环境(称作正式环境)等不同的环境里运行。 对应在 Flow 中,我们把这些环境称之为「**主机组**」,也就是该应用运行在若干台机器(虚拟机/容器)。

#### 前置任务

可以通过两种方式创建主机组: 方法一:通过「**主机组管理**」添加主机组

☆ 我的星标	主机组管	理			筛选 ~	排序 ~	●新建主机组 1
◎ 我参与的		主机组名称	主机组标签	主机数量	创建人	创建时间	¢
⇒ 至部流水线	_ 3	<b>湍博测试部署组</b>	标签1	1台	3 瑞博	2020-02-24	¢ ŵ
企业管理 8 通用亦是4		新的测试docker的部		1台	<b>9</b> 柳夕	2020-02-20	C T
<ul> <li>一用支重组</li> <li>三 主机组管理</li> </ul>		罗英测试staragent被	标签1 标签2	0 台	👧 石头哥	2020-01-16	ĊŴ
② 企业设置		新的部署组	-	1台	👦 DuoDuo	2019-12-26	¢ ŵ
·		新的无主机的部署组	-	1台	😡 柳夕测	2019-12-20	¢ ŵ
		检证tty的部署组	自动化	1台	😡 柳夕测	2019-12-19	ĊŴ
			< 1	2 3 >			

### 图1. 主机组管理一新建主机组 如何使用添加部署任务*,可查阅"部署"一节*

#### 创建主机组

在部署任务中,点击「新建主机组」,进入新建主机组的流程。

- 你可以创建三种类型的主机组:

   阿里云ECS
- 部署到阿里云 ECS
- 可以出公网的非阿里云主机
   部署到公网主机
- 不可访问公网的非阿里云主机(即将上线)

### 6.2.2.1. 主机组

### 主机组是什么

每个 Web 应用,在集成测试的环境(通常称作日常环境)、预发的环境(称作预发环境)、对外提供服务的环境(称作正式环境)等不同的环境里运行。 对应在 Flow 中,我们把这些环境称之为「**主机组**」,也就是该应用运行在若干台机器(虚拟机/容器)。

### 前置任务

可以通过两种方式创建主机组: 方法一:通过「**主机组管理**」添加主机组

📁 Кж					e	÷ 0
☆ 我的星标	主机组管理			筛选 ~	排序 ~	新建主机组 1
A、我参与的 	□ 主机组名称	主机组标签	主机数量	创建人	创建时间	\$
⇒ 至即流水线	瑞博测试部署组	标签1	1台	13 瑞博	2020-02-24	C T
企业管理	新的测试docker的	ø	1台	ø 柳夕	2020-02-20	ĊŴ
<ul><li>為 通用变量组</li><li>三 主机组管理</li></ul>	罗英测试staragent	被 标签1 标签2	0 台	👧 石头哥	2020-01-16	c t
約 企业设置	新的部署组	-	1台	🤯 DuoDuo	2019-12-26	ĊŴ
	新的无主机的部署组	£ -	1台	👿 柳夕测…	2019-12-20	C T
	验证tty的部署组	自动化	1台	👿 柳夕测…	2019-12-19	ĊŴ
		<	1 2 3 >			
						þ

方法二:先在流水线中添加「**主机部署」**任务。用户可以通过以下方式添加部署任务。

- 使用模板新建流水线,选择包含"**部署**"任务的模板
- 在已有的流水线中,添加新的阶段,并选择「**主机部署」**任务

■ 流水线编排 流水线 2020-02-	-29				保存保	存并运行
▼ 源(2)	+	部署	新阶段	编辑		
…/flow-example/spring-bo	ot			任务名称		
ዀ master	63	り 王机部者 ()	G #1019	主机部署		
w-example/php-laravel-ble >> master	og			☑ 部署时下载制品		
				制品		
▼ 变量 (0)	+			制品名称.default		~
				主机组		新建主机组
習木设重, 尽击 浴加				请选择		~
				主机主机组不能为空		
				部署配置		
				下载路径 ⑦		
				/home/admin/app/package.tgz		
				执行用户 ⑦		G
<ol> <li>定时运行</li> </ol>				root		

如何使用添加部署任务*,可查阅"部署"一节* 

### 创建主机组

在部署任务中,点击「新建主机组」,进入新建主机组的流程。

- 你可以创建三种类型的主机组:
- 阿里云ECS
  - 部署到阿里云 ECS
- 可以出公网的非阿里云主机
   部署到公网主机
- 不可访问公网的非阿里云主机(即将上线)

### 6.2.2.2. 部署到阿里云 ECS

云效流水线 Flow 支持部署到阿里云 ECS 。你可以按照以下方式,将你的阿里云 ECS 加入主机组,从而让流水线能对其进行部署。

### 创建阿里云 ECS 类型主机组

### 添加方式——直接添加

1)新建主机组,选择 【主机类型】为 "阿里云ECS" 。

	\t++2-		~	
	251年.	土机突型	^	
☆ 我的星标				◇ 排序 ◇ 🕒 新建主机组
风 我参与的				创建时间
\$ 全部流水线				0010 10 01 57 <del>P</del>
				2019-12-31
企业管理				2019-12-06 🗹 🗊
& 通用变量组		(1)		
□> 标签管理	阿里云ECS	自有主机		
■ 主机组管理	阿里云提供的性能卓越、稳定可靠、弹 性扩展的 laaS 级别云计算服务	非阿里云的公网主机,包括其他公有云 主机或者可以联通公网的自有主机		
	$\ominus$	$(\mathbf{a})$		
		2		

2)选择【服务授权】和【地区】,展示【可添加的主机列表】,可【新建服务授权】。

	<	阿里	<b>≧云ECS</b> · 添加主材	l		×		
☆ 我的星标	服务授权				(1) ④ 新建	服务授权	✓ 排序 ∨ │	▶ 新建主机约
, 风, 我参与的	rdc-rdc的ecs服务授权					~	创建时间	
♀ 全部流水线	地区						2019-12-31	ľŤ
企业管理	华东 1					~	2019-12-06	ra w
& 通用变量组	可添加的主机列表					2 7		
□ 标签管理	实例ID	IP	主机名	占用情况	主机状态			
主机组管理	i-bp1ih4svu	192.168.0.68	worker-k8s	● 未占用	正常	-		
	i-bp1ih4svu	192.168.0.67	worker-k8s	● 未占用	正常	- 11		
	i-bp1c26e4j	172.16.96.219	EDAS-scale	● 未占用	正常	- 11		
	i-bp1iuki22a	192.168.0.191	EDAS-scale	● 未占用	正常	- 11		
	i-bp11kijdfc	192.168.0.190	flow-edas-t	● 未占用	正常	- 11		
						_		
	1 暂不支持跨地主机				上一步	下一步		
						_		

3)选择你想要添加至主机组的ECS主机。

	<	阿里	星云ECS · 添加主材	l		$\times$		
☆ 我的星标	服务授权				新建制	服务授权	∽ 排序 ∽ │ (	➡ 新建主机组
,以, 我参与的	rdc-rdc的ecs服务授权					~	创建时间	
♀ 全部流水线	地区						2019-12-31	C T
企业管理	华东 1					~	2019-12-06	ित गी
& 通用变量组	可添加的主机列表				Q	V 7		
□ 标签管理	_ 实例ID	IP	主机名	占用情况	主机状态			
主机组管理	i-bp1ih4svu	192.168.0.68	worker-k8s	● 未占用	正常	0		
	i-bp1ih4svu	192.168.0.67	worker-k8s	● 未占用	正常		(1)	
	i-bp1c26e4j	172.16.96.219	EDAS-scale	● 未占用	正常		$\bigcirc$	
	i-bp1iuki22a	192.168.0.191	EDAS-scale	● 未占用	正常			
	i-bp11kijdfc	192.168.0.190	flow-edas-t	● 未占用	正常			
	1 暂不支持跨地主机				上一步	下一步		

4)编辑主机信息,包括主机组名称、环境、标签;主机组成员权限的配置,创建人默认为企业拥有者;点击保存,即可完成主机组的创建。

	<		阿里云ECS·	主机信息		×		
☆ 我的星标	名称						~ 排序 ~ (	▶ 新建主机组
风 我参与的	新的主机组						创建时间	
♀ 全部流水线	流水线环境							
	请选择环境					~	2019-12-31	U W
企业管理	标签						2019-12-06	C T
。通用变量组 ————————————————————————————————————	请选择					~		
□> 标签管理							$\bigcirc$	
■」 王机组管埋	±#							
	王和	IP	主机名	占田徳沒	主机状态	過作		
	i-bp11kiidfc	192.168.0.1	flow-edas-t	• 未占用	正常	107		
	i-bp14au85	10.0.10.115	vanvu	<ul> <li>未占用</li> </ul>	正常	TIV		
			,,.					
	1 详情查看文档				上一步	保存		
								2

圖 流小线编件	墙博测试流水线 2019-11-13			味 仔		
9	◎ 新建源	部署	新阶段	編辑 ×		
				任务名称		
	Java 构		机部署 ③ 新的任务	主机部署		
				☑ 部層时下载制品		
				制品		
				制品名称.default ~		
				主机部署组 ③ 新建部署组		
•	• °			正式环境部署组demo ~		
1	◆ 「无数据源			部署配置		
				下载路径 ③		
				/home/admin/app/package.tgz		
				执行用户 ③		
				admin		
				部署脚本 ⑦		
✓ 成功 部署组正 建成功	× 式环境部署组demo 创			1 ● 部署課本信企部署組約每台机器上跌行。一个希望課本課題の下、 先常新品台(在下載器台中配置約下載器台)第匹部 中,再於行告約課本(還常在代書中做評,如示例平 ,ah)、关于这个例子的律医職務見れたtps://thought .teambition.com/shorespace/3435/486ac/640143ac/641/40cs		

#### 对部署权限的说明, 可查看"主机组成员权限"一节

#### 添加方式——通过 ECS 标签添加

同时, 主机组支持通过ECS标签添加, 如果您使用 ECS 标签对 ECS 资源进行分类管理, 可以直接在 Flow 中建立主机组和 ECS 标签的关联。

	〈 阿里云ECS·添加主机	×		
工具箱			Q 7 E↓ (	新建主相
& 通用变量组	添加方式 ● 直接添加 ● 标签方式添加 (通过ECS标签快速定位主机) 1 ⑦ 什么是ECS标	签	创建人	
□ 标签管理	服务授权 ④ 新建服务授	救	쪬 王堂刚	C 1
📃 授权管理	rdc-rdc的ecs服务授权 ~	-	🗙 光晕	C t
■ 主机组管理	地区	â	● 王堂刚	r w
※ Kubernetes集群管理	华北1 (青岛) ~	•		
💿 流水线模板管理	ECS标签键	È	🗙 光晕	C V
↩ 任务组管理	provider	÷ 1	③ 雪影↓	6 1
123 步骤管理	ECS标签值	÷	rdc-rdc	c t
8 构建集群管理	aliyunacs		🗶 光晕	C T
管理员设置	● 暂不支持跨地主机 上一步 下一步	Ê	rdc-rdc	C t

#### StarAgent

为保证部署通道可用性,提高部署成功率,飞流采用了云助手(https://help.aliyun.com/document_detail/64601.html)和Staragent双通道方案,云助手在阿里云ECS会默认 安装,在新增和修改部署组的时候,飞流会通过云助手自动安装StarAgent。

#### StarAgent介绍

StarAgent是阿里巴巴集团服务器运维基础设施,承载了全集团的服务器交互任务。StarAgent是打造全网通,7*24不间断提供服务,安全可靠,日均亿级调用量,99.995% 的系统成功率的命令通道。通过阿里巴巴集团安全团队对StarAgent进行了安全加固。重置帐号密钥,客户端服务端使用不同的加密方式,在服务器上执行的命令进行全量审 计等安全策略为命令通道保驾护航。

#### StarAgent基础操作

查看状态: /home/staragent/bin/staragentcll status; 启动: /home/staragent/bin/staragentcll restart; 重启: /home/staragent/bin/staragentcll restart; 卸载: 1. /home/staragent/bin/staragentctl stop;

- rm -rf /home/staragent;
- 3. rm /usr/sbin/staragent_sn

### 6.2.2.3. 部署到公网主机

通过在对应主机中添安装 Agent 的方式,云效流水线 Flow支持部署到非阿里云的公网主机(包括其他公有云主机或者可以联通公网的自有主机)。 你可以按照以下方式,将你的公网主机加入主机组,从而让 Flow 能对其进行部署。

### 创建自由主机类型主机组

1)新建主机组,选择【主机类型】为"自有主机"。

工具箱	主物伯答理《			Q 7 ≣↓	● 新建主机组
通用变量组	选择主机类型		×	间	
标签管理				-03-25 18:22:00	C t
服务连接管理				-03-07 10:44:03	िंदी भीग
主机组管理			Free		
Kubernetes集群管理				-03-07 09:41:51	C T
流水线模板管理		<b>今</b> <del>大</del> → 10	<b>存</b> 弗 休录 → 坦	-03-04 18:18:01	¢ t
任务组管理	<b>阿里云ECS</b> 阿里云提供的性能卓越、稳定可靠、弹	日 <b>日</b> 土 <b>1</b> 1 非阿里云的公网主机,包括其他公有云	<b>兄 贲14 ¹ 土 机</b> 5小时免费ECS资源,体验完整流水线	-03-04 16:35:37	C t
步骤管理	性扩展的 laaS 级别云计算服务	主机或者可以联通公网的自有主机	构建发布流程	-03-04 15:21:18	िंदी भीर
构建集群管理	() ()	Θ	()		
管理员设置		U U		-03-04 10:28:50	C V
在央边里			•	-03-03 22:35:17	C T

2) 展示【可添加的主机列表】,如果想要添加新的自有主机至【可添加的主机列表】,请复制主机添加指令。

	く自有主机・添加主机	×
☆ 我的星标	可添加的主机列表	Q、 ♀
风 我参与的	实例ID         IP         主机名         占用情况         主机状态	创建时间
€ 全部流水线	2246e420 (内网)192.1 izbp14v2sd ●未占用 正常	2020 02 24 52 197
0.11.001TE	49fe7455-8 (内网)192.1 iZ2zedyhiev ● 未占用 正常	2020*02*24
12.12官埋 8 2落田亦昌49	6c4416ab (内网)192.1 izbp19e8tq7 ● 未占用 正常	2020-02-20 🗹 🖞
63 連用支重组	6fcaa9a5-a (内网)192.1 izbp14v2sd ● 未占用 正常	2020-01-16 🗹 🖞
	a40e9e64 (内网)192.1 iZ2zedyhiev ● 未占用 正常	
② 企业设置	如何导入新机器到「可添加列表」?	) 2019-12-26 🕜 🖤
	1.清复制下列命令到需要添加的主机上,该命令运行14分39秒 后过期 2. 地名1.2公验后,新印刷会自己后入「可透如刷桌」用	. 2019-12-20 🗹 🗊
	bash < (curi http://agent-install-default.oss-ch- Install-M) http://ca.agent-install-default.oss-ch- Install-M) http://cc.agent-install-default.oss-ch- regionid-scrhaydjinuf/wokz&sign=zqi3AX1BQmDeXtbKx2F1LnewAKsD%3D&ailyunRegion=d efault8itimestamp=1683292830288 http://agent-install-default.oss-ch-	□ 复制
	上一步	下-步

3)请打开你需要添加进主机组部署的主机的命令行,并将前序操作中复制的命令,在你的自有主机上执行(请注意,主机需要能访问公网)。

install-default.oss-cn-hangzhou.aliyuncs.com/production-agent.tgz http://rdc.aliyuncs.com/api/verifySnAndRegionId?sign=3h%2FWjEIEViUXijgAjWbKqA
%3D%3D 5ep7dzeck9t4tmy7 1573994399790 default
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 5062 100 5062 0 0 142k 0:::- 145k
检查目录
目录/home/ao12或者/home/staragent已经存在于您的机器上,有可能是因为您已经安装过RDC的Agent,继续安装,会重置这些目录中的内容。请选择[Y/n]:Y
检查python
检查wget
检查zlib-dev openssl-devel bzip2-devel包是否缺失
获取SN。
SN:97a52bca-0635-44b0-a9c8-eaa0212817cc
安装SN
下载并安装Agent
2019-11-17 20:31:05 http://agent-install-default.oss-cn-hangzhou.alivuncs.com/production-agent.tgz
正在解析主机 agent-install-default.oss-cn-hangzhou.alivuncs.com (agent-install-default.oss-cn-hangzhou.alivuncs.com) 118.31.232.205
正在连接 agent-install-default.oss-cn-hangzhou, alivuncs.com (agent-install-default.oss-cn-hangzhou, alivuncs.com) [118,31,232,205]:80 已连接。
大度: 65573216 (63M) [application/octet-stream]
下在保存至于"(Inn/agent taz")
20 12 (1) (2) ( ) (1) (2) (3) (3) (3) (3) (3) (3) (3) (3) (3) (3
100%[
2019-11-17 20:31:09 (15.2 MB/s) - 已保存"/tmo/agent.tgz" [65573216/65573216])
The second s
/home/staragent/hin/agent_sh
Stanjig stanggardi proces 21508 evit successfully
Agant at numbra
ston the regrand finish
unizi agent
talistic ultr agent
/a w/Mgente
/ilom//stal agent/uli/agent.st
Another state agencies.
Agent not running.
Stop staragent tinism
Starting staragenta: 安装成功
[root@ruibo-test-1106 ~]#

4) 主机添加指令执行后,等待1-2分钟后,自有主机会导入【可添加的主机列表】,选择你想要添加至主机组的自有主机。

	<		自	有主机 · 添加主机			×		
☆ 我的星标	可添加	的主机列表				Q	7	~ 排序 ~ (	▶ 新建主机组
风 我参与的		实例ID	IP	主机名	占用情况	主机状态		创建时间	
\$ 全部流水线		2246e420	(内网)192.1	izbp14v2sd	● 未占用	正常		2010-12-21	ि∕2 भीग
		49fe7455-8	(内网)192.1	iZ2zedyhiev	● 未占用	正常		(1)	
正业官理		6c4416ab	(内网)192.1	izbp19e8tq7	● 未占用	正常		2019-12-06	C T
○ 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一		6fcaa9a5-a	(内网)192.1	izbp14v2sd	● 未占用	正常			
		a40e9e64	(内网)192.1	iZ2zedyhiev	● 未占用	正常			
	bas bas regi defa	+2分钟后,新机器会自; +2分钟后,新机器会自; h <(curl http://agent-i all.sh) 'http://rdc.aliyu onld=crhsydjinufjwok ault&timestamp=1583	中国	-hangzhou.aliyuncs.co Region? IO6zkbSGV74vPpBw% gent-install-default.oss	m/production- 3D%3D&aliyunRegio -cn-		制		
	L			_	_	上一步	—步		P

### 🥐 说明

添加、删除 agent 等操作问题可参考<mark>部署常见问题</mark>

5) 编辑主机信息,包括主机组名称、环境、标签;主机组成员权限的配置,创建人默认为企业拥有者;点击保存,即可完成主机组的创建。

		<		自有主机·当	E机信息		$\times$		
☆ 我的星标		名称						~ 排序 ~ (	新建主机组
风、我参与的		新的主机组							
➡ 全部流水线		流业经济						创建时间	
	0	请洗择环情					~	2019-12-31	C T
企业管理	U	NU 4221-0-71 476						2019-12-06	CC TH
& 通用变量组		标签							
□ 标签管理		请选择					~		
■ 主机组管理	(2)	· +							
	<u> </u>	主机							
		实例ID	IP	主机名	占用情况	主机状态	操作		
		2246e420	(内网)192.1	izbp14v2sd	● 未占用	正常	Ŵ		
		49fe7455	(内网)192.1	iZ2zedyhie	● 未占用	正常	Ŵ		
						上一步	保存		

6)至此,你可在流水线中选择使用该主机组了。

源	新建源		部署	新阶段	编辑	
					任务名称	
		Java 构建	日、今、王机部者	◎ 新的任务	主机部署	
					🛛 部署时下载制品	
					制品	
					制品名称.default	
					主机部署组	新建部
•	• ~				正式环境部署组demo	
1	◎				部署配置	
					下载路径 ⑦	
					/home/admin/app/package.tgz	
					执行用户 ③	
					admin	
					部署脚本 ⑦	
<ul> <li>成功</li> <li>部署组正 建成功</li> </ul>	× 式环境部署组demo 创				1 ■ 部署關本会在部署组約每台机程上执行。一个 先將制品包(在下電路径中配置的下電路径) 中,再执行启动關本(還常在代母中維护), .sh)。关于这个例子的详細辦罪犯 https: .teambition.com/sharespace/5d8325488cc	典型脚本逻辑如下: ) 鮮压縮 如示例中 //though c9d00143ac6ef/docs

图6.使用主机组 对部署权限的说明,可查看"流水线成员权限"一节

### 6.2.2.4. 部署到免费体验主机

Flow 为新用户提供了 5 小时的免费 ECS 主机,您可以将体验 ECS 导入 Flow 主机组,并使用体验主机组快速体验 Flow 将代码构建部署至 ECS 的 CICD 流程。

### 创建免费体验主机组

1)新建主机组,主机类型选择"免费体验主机"。

く 返回 流水线 2022-03-03		基本信息 流程配置 触发设置	变量和缓存	仅保存保存并运行
i.	^{构建} 选择主机类型	部署	编辑 ⑰	×
01fd99b0/spring-boot-2048			Ð	
<ul> <li>□ spring-boot-2048_7N90</li> <li>+添加流水线源</li> </ul>				INE_ID} ~
	同用一下の	白右士机	<b></b>	<ul> <li>● 新建主机组</li> <li>✓</li> </ul>
	阿里云提供的性能卓越、稳定可靠、弹 性扩展的 laaS 级别云计算服务	日 FF 工 176 非阿里云的公网主机,包括其他公有 主机或者可以联通公网的自有主机	无页[平动工176 云 5小时免费ECS资源,体验完整流水线 构建发布流程	请谨慎操作
	Э	Θ	Э	je.tgz
			root	
			您可以在部署脚本中」 一般四处理 但是加思想	直接使用环境变量,无需再进行Base64 你的环境恋号中句会特殊字符。你将无法

### 2) 点击"阿里云体验实验室",前往申请免费体验资源。

く 返回 流水线 2022-03-03	基本信息 流程配置 触发设置 变量和缓存		仅保存保存并运行
記原	< 免费体验主机 · 添加主机	×	×
01fd99b0/spring-boot-2048	● 如何添加免费体验主机 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	查看帮助	
‰ master	AK ID 🕐 *		
Spring-boot-2048_7N90	请粘贴AK ID到此处		
十添加流水线源	AK Secret ⑦*		INE_ID} ~
	请粘贴AK Secret到此处 ECS 实例 ID ⑦ ●		<ul> <li>● 新建主机组</li> <li>✓</li> </ul>
	请粘贴ECS 実例 ID到此处 地域 ⑦・		请谨慎操作
	上海 主机组名称* 5小时体验主机	~	je.tgz
		保存	
		」以在部署脚本中直	接使用环境变量,无需再进行Base64 的环境亦是由句今桂碑字竺 你将王注

3) 点击"开始体验",进入资源申请页面。

▶ ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	域名 Q 中国法 文档 购物车 ICP备条 控制台
首页 分享~ 学习~ 实践~ 工具&资源~ 参赛~ <mark>我的原分 100</mark>	■发文章 2 提问题 个人
《#%###      Java应用构建并部署ECS     本文格荫勘念实现研发流程的协同自动化。     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①     ①	
今天教会部: 907       ★★★★↑↑ (8.6.52)       詳細書名       建立       2         現示: 二の一品房景会在加工时间和料放,适应建造过空快播       「中面目本面」       2         定時面介           2	(律输时长(小时) 云产品资源使用时长(小时) 体验人次(次) 5 724
相关产品 ■ 云磁务器ECS	基于函数计算活键Serverless小商城
体验简介 实验报告 4223 问答 4233	《云爾生操作系统(Aubernates) 随书实验
如果你 1.使用Java进行开发	ECS服务器部署MySOL数据库 会 827 ③ 1
2. 使用Jar/War的制品形式进行交付 3. 制品最终会运行在ECS或着自有主机上	

4) 点击"创建资源",开始创建体验主机,资源创建过程需要1~3分钟,请耐心等待。



5) 完成实验资源的创建后,您可以在 云产品资源 列表查看已创建的资源信息,其中 AK ID、AK Secret、ECS 实例 ID、地域 为主机导入的关键信息。

Java应用构致	建并部署ECS 详情回		剩余体验时间: <mark>04 :</mark> 57 : 32	结束体验
体验手册	云产品资源 実務く 	> 😜		۲ <u>۶</u>
体验云账号,创建	建资源后生成 收起 /	Firefox ESR		
子用户名称:	u			<b>—</b>
子用户密码:	Aa 34lz9A 🗇	Chromium		æ
AK ID:		网页浏览器		FT
AK Secret:	(3NRPWSS1z3f 🗇			
		LX 终端		<b>O</b>
ECS服务器 ECS公网地址:	4.172.204			2
ECS登录名:	root 🗇			80
登录密码:	H 'n6J 🗇			
ECS实例ID: IP白名单·	i-uroczy Is9gq5y4svk			
地域:	华东 2 (上海) □			
	* * * * * * * * * * * * * * * *	Θ •		iii 10:16

6)返回 Flow,将体验资源的 AK ID、AK Secret、ECS 实例 ID、地域 分别填写至对应位置,保存完成主机组添加。

く 返回 流水线 2022-03-03	基本信息 流程配置 触发设置 变量和缓存	仅保存保存并运行
戋源	< 免費体验主机・添加主机 ×	: X
01fd99b0/spring-boot-2048	如何添加免费体验主机         宣看帮助           1.前往阿里云体验实验室创建体验主机ECS,需要等待1-3分钟         2.创建完毕后将AK ID、AK Secret和实例ID粘贴到下方区域完成5小时免费体验添加主机	
- l> master	AK ID ③ *	
🗅 🗅 spring-boot-2048_7N9O	LTAI 3LHvy	
+ 添加流水线源	AK Secret ⑦*	INE_ID} ~
(1)	3 bema	● 新建主机组
	ECS 实例 ID ② *	×
	i :4svk	请谨慎操作
	地域 ⑦*	
	上海	
	主机组名称。	ge.tgz
	5小时体验主机	
	保存	
		直接使用环境变量,无需再进行Base64 你的环境亦是中句今特殊字符 你将不注

7)点击保存主机组,Flow 会在体验主机上执行安装 agent ,预计会等待 15s 左右时间,请耐心等待。

### 云效流水线 Flow·部署

### 云效2020

く 返回 流水线 2022-03-04	基本信息 流程配置 触发设置 变量和缓存		仅保存保存并运行
视	< 免费体验主机·添加主机	×	×
99b0/spring-boot-amiao 🖓 🔹	<ul> <li>如何添加免费体验主机</li> <li>1.前往同盟云体验实验室创建体验主机ECS,需要等待1-3分钟</li> <li>2.创建完毕后将AK ID、AK Secret和实例ID粘贴到下方区域完成5小时免费体验添加主机</li> </ul>	查看帮助	
‰ master	AK ID 💿 *		
🗅 spring-boot-amiao_A8I7	L URP35S		
十添加流水线源	AK Secret ⑦ *		INE_ID} ~
	ECS 実例 ID ② * 请稍等,正在添加体验主机…		● 新建主机组
	i-uf€3pr6g		
	地域 ⑦ •		
	上海	~	
	主机组名称 *		ge.tgz
	5小时体验主机		
	解码	保存 强处理。但是如果你	{使用环境变量,无需再进行Base64 %的环境变量中包含特殊字符,您将无法

8) 添加成功后,即可在 Flow 中使用该主机组。

く 返回 流水线 2022-03-03	基本	医信息 流程配置 触发	设置 变量和缓	存	仅保存保存并运行
見源	构建	部署		编辑	×
01fd99b0/spring-boot-2048		<b>• •</b>	主机部	任务名称 主机部署	
- ⊫ master - ⊡ spring-boot-2048_7N90				✔ 部署时下载制品	
十添加流水线源				<b>制品</b> 制品名称.Artifacts_\${PIPELINE_ID}	~
			(1)	<b>主机组</b> 5小时体验主机	● 新建主机组
			Ŭ	部署配置	
				下载路径 ⑦ /home/admin/app/package.tgz	
				执行用户 ⑦	
				您可以在部署脚本中直接使用环境 解码处理。但是如果你的环境变量 下常使用你的环境变量。	变量,无需再进行Base64 中包含特殊字符,您将无法

### 6.2.3. 部署配置

在流水线的部署组件中,需要进行部署配置。本篇文档会给出一个示例,供你参考,如何配置。

#### 提前准备

- 1. 从示例代码库新建一个SpringBoot代码库: https://code.aliyun.com/code-template/spring-boot。其中包含启动脚本deploy.sh: https://code.aliyun.com/code-template/spring-boot/blob/master/deploy.sh。
- 2. 在部署机安装Java运行环境
- 3. 在流水线构建出制品

为了进行部署,首先需要构建出制品。Flow 中的制品是一个tgz压缩包,在构建任务中可以指定一个或者多个文件(文件夹),Flow 的制品中就会包含这些文件(文件 夹)。详情可以参考文档<del>上传到 Flow 的制品库</del>。

在本例中,需要将target/application.jar和deploy.sh两个文件打包到制品中。因此需要在构建任务中按下面的方式进行配置:

源	新建源	构建	部署	● 添加步骤	
C-J group/repo				▶ Java 构建	
}∞ master	!像构建	日 Ģ Java 构刻		主机部 - 构建物上传	
				步骤名称	
				构建物上传	
				制品名称	
				default	
				打包路径	
			(1)	target/application.jar	
			2	deploy.sh	
				任务插件	
				● 添加插件	
				任务输出	
				回 期日夕む default	

### 主机部署配置

在主机部署任务中,可以进行部署相关的配置,用于将构建产物在部署机上进行安装。

í.	• #	建源	部署	新的阶段	编辑	
		肉建 打压缩句			部署配置	
		199E, 11/238ECS			1 下载路径	0
					/home/admin/app/package.tg	t
					执行用户	
					admin	
					<b>師事</b> 則太	0
					1 # 部署版本会在部署组的等台	机器上执行。一个典型脚本逻辑如下
	未配置源				<ul> <li>1 年間希望未会在部場組約等台 ・ 先等者前品包(在下部 定目書中,再打合品單 play.ch) 未予定合期 .tambrition.com</li> </ul> <li>2 j: Lar zavf //home/admin/opp/dep1 4 = sh /home/admin/opp/dep1</li>	単書上映行。一个典型様本連續起下 通径中配置的下載通径)第匹倍割指 本 (通常在代码中様学) 如示相中de 子的 X 細胞 確见 https://help //package.tgz y.sh restart
	未配置源				<ul> <li>シークの時代の ・ 知識様子会在部業組織等合 ・ 完成時間面包(在下部 定該当年)、市民で言語 play.ab)、決予这个時 ・ tamb/time.com</li> <li>オームのの</li> <li>オームののののの</li> <li>オームのののののののののののののののののののののののののののののののののののの</li></ul>	私参上操作。一个商生課次定編集了 構設や定置的で取得会)類正常用導 本(遵定で代表中導ア、切字用導 学的注意解解型のHtgs://Melp //package_tgz y.shirestart
	未配置源				D = 10000 1 = 即乗編先会在部乗組約等台 2 完成年代一部度 play.adi、未予这个例 - transhrinum, adm 2 ま tar zavf /home/admin/agg 4 ま th /home/admin/agg 4 ま th /home/admin/agg 4 ま th /home/admin/agg	私参上除行。一个角至博次波澜近下 高校中定置前下取高台、類正指倒着 本(建定で代理修)、辺市明存他 子的22 間部電気https://help //package.tgs y.ch restart
	未記實源				D 中国のHT 1 年期構造そな在期構成時代で 二 作用構造そな在期構成時代で 二 作用構造での こ 作用構造での こ 作用構造での 2 またの力にの 日 につ zxxf / home / admin / app 4 またい / home / admin / app 5 世界 1 世界	私参上除行。一个発生度水淀細加下 病院や完成素が市新品と、描述作用当 本(違充定代表中提示)の示明や値 子の回過転用 <u>気</u> https://welp v/package tag: gr.dh restart
	未配實道				DF Blank 1 単加速でを正確単純的なら 二 定時の中、市場になっていた。 2 目前の中の日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の	私参上除行。一个角空爆水波構築下 病院や定然素で加高さい解子 病院やためます。 第二時の 子の防御新見Shttps://help //merkage tap: gr.dh.restort

#### 下载路径

部署配置中的下载路径就是构建物上传步骤产出的压缩包将要被下载到你的主机上的路径,本例中为:/home/admin/app/package.tgz

### 执行用户

填写执行部署脚本的用户,如 root,或者 admin 等。本例中为root

### 部署脚本

因为实际的部署脚本已经打包到了构建产出的压缩包中,因此部署脚本可以简单的进行解压和执行即可

mkdir -p /home/admin/application tar zxvf /home/admin/app/package.tgz -C /home/admin/application/ sh /home/admin/application/deploy.sh restart

### 🗘 注意

Flow 在部署过程中,会下发部分以 rdc_deploy_command_ 开头的部署进程,请勿在部署脚本中对此类部署进程进行操作,否则可能会导致部署失败。

### 6.2.4. 部署策略

真实的上线过程,如果采用全量发布,会给开发运维团队带来未知的风险,为了减少发布对线上业务的影响, Flow 提供了灰度发布,分批发布能力,最大限度的避免了不稳 定发布对用户的影响, 保障业务交付的稳定。

### 主机分批部署

如果你一次需要发布多台主机,Flow 支持分批发布,您可在主机部署任务中,可以指定对应的部署策略。假如你有 4 台主机,选择分 2 批发布,则每一批自动发布 2 台主 机。

### ● 暂停方式

- 第一批暂停:第一批发布完后,您可先行验证,点击继续后可执行后续发布。
- ◇ 不暂停:每一批发布完后,自动执行后续批次的发布。
- 每批暂停:每一批发布完后,都需要手动确认继续发布。

#### • 分批数量

○ 指定主机分批的数量,指定分批发布的数量,会按照相应的数量进行分批部署。



### 6.2.5. 查看部署详情

流水线配置完成,在流水线运行页面,在"主机部署"任务中,可以点击【部署详情】查看部署过程。

<b>}段2</b>		阶段3	
📀 java构建,打压缩包		🧿 主机部署	
application.tgz		1 运行中	
□ 日志	39秒	☆ 发布详情	10秒

### 部署详情

部署单会显示当前部署的整体状态和情况。

	主机部署-运行时	详情	×
3署详情			
报告 日志			
1 暂停中			终止继续下一批
部署包: 下载部署包	发布进度: 2/5	暂停方式: 第一批暂停	
主机失败率: 0.00%	发布耗时:2时40分43秒		
提交人: 💿 rdc-rdc	发布时间: 2019-09-03 16:51:45		
全部 发布中 待发布 已完	成  已失败		
批次 状态 IP	异常信息	更新时间	操作
THEFT STORE STORE			
#1 已完成 10.0.10.11	9(私网)	2019-09-03 19:32:25	查看日志
#1         已完成         10.0.10.11           #1         已完成         10.0.10.11	9(私网) 8(私网)	2019-09-03 19:32:25 2019-09-03 19:32:25	查看日志
#1         已完成         10.0.10.11           #1         已完成         10.0.10.11           #2         待发布         10.0.10.11	9 (私网) 8 (私网) 7 (私网)	2019-09-03 19:32:25 2019-09-03 19:32:25 2019-09-03 19:32:25	查看日志 查看日志
#1         已完成         10.0.10.1           #1         已完成         10.0.10.1           #2         待发布         10.0.10.1           #2         待发布         10.0.10.1	9 (私网) 8 (私网) 7 (私网) 6 (私网)	2019-09-03 19:32:25 2019-09-03 19:32:25 2019-09-03 19:32:25 2019-09-03 19:32:25	查看日志 查看日志

### 部署状态

部署中:可以执行【终止】操作,及查看日志 暂停中:可以执行【终止】,【继续下一批】操作,及查看部署日志 成功:可以查看部署日志 失败:可以查看部署日志,并在卡片视图上重新发起部署 **机器状态** 待部署:无操作 部署中:可以查看部署日志 成功:可以查看部署日志 失败:可以查看部署日志,并重试单台机器 常见部署问题,可查看"部署常见问题"一节

### 6.2.6. Docker 部署

Flow 提供了 Docker 部署能力,部署目标物可来自上游构建产物,如阿里云镜像构建,自定义镜像构建等



当然,您也可以直接使用如 docker run -d hello-world 命令来将 docker hub 中的通用镜像作为部署目标

### 添加Docker部署任务

在流水线编辑时,你可将Docker部署任务添加至您的流水线中

将代码库中的 Helm Char	t 触发 Jenkins 任务	
主机部署(无构建包)	Docker部署	
使用脚本发布到主机	主机Docker部署	
thetes YAML Kubernetes 镜像升级 升级 Kubernetes 下的已经	f Kubernetes 分批发布 Kubernetes 分批发布	
部署 EDAS ECS应用发布	EDAS Kubernetes 应用	
Helm Chart EDAS ECS应用发布	EDAS Kubernetes 应用发布	
AE) 应用	.#h	
□ □ 用友币 更新函数订算 (FC) 中因	30,	
	<ul> <li>第代局库中的 Helm Char</li> <li>主机部電(无約環句) 使用關本及希到主机</li> <li>使用關本及希到主机</li> <li>使用關本及希到主机</li> <li>新聞、Kubernetes 積優升級 升級 Kubernetes 積優升級</li> <li>新聞</li> <li>新聞</li> <li>EDAS ECS应用发布</li> <li>EDAS ECS原用发布</li> <li>自用发布</li> <li>正月发布</li> </ul>	第代周慶中的 Helm Chart       認識 建送 Jenkins 任务         記書(天内建色)       ()         使用脚本发布到主机       ()         ()       ()         ()       ()         ()       ()         ()       ()         ()       ()         ()       ()         ()       ()         ()       ()         ()       ()         ()       ()         ()       ()         ()       ()         ()       ()         ()       ()         ()       ()         ()       ()         ()       ()         ()       ()         ()       ()         ()       ()         ()       ()         ()       ()         ()       ()         ()       ()         ()       ()         ()       ()         ()       ()         ()       ()         ()       ()         ()       ()         ()       ()         ()       ()         ()       () <t< td=""></t<>

### 配置部署任务

进入部署任务,进行如下配置:

1) 创建或者选择「主机组」, 主机组用于管理部署的机器资源(用户 Docker 部署的机器您需要自行在机器上安装 Docker 软件)

关于如何创建主机组,可查看"<mark>主机组</mark>"一节

2) 配置执行用户和具体的部署执行命令 关于部署执行命令,在飞流中,一个典型的 Docker 部署的执行命令如下:

echo \$DOCKER_PASSWORD | docker login --username \$DOCKER_USERNAME --password-stdin registry.cn-hangzhou.aliyuncs.com docker run \$image

上面的 image 为您在执行命令下方的变量区域配置的上游输出产物参数,上游输出产物可以是阿里云镜像构建产物,也可以是自定义镜像构建产物等等。

### 3)配置变量

变量分为两种类型,上游输出和自定义。 上游输出产物可以是阿里云镜像构建产物,也可以是自定义镜像构建产物等等。 您也可以配置自定义变量,然后在执行命令处使用

变量会写入您的机器环境变量中,这样您可以在执行命令处直接使用,如 Smytest 这样



echo \$DOCKER_PASSWORD | docker login --username \$DOCKER_USERNAME --password-stdin registry.cn-hangzhou.aliyuncs.com (地址根据你们的项目实际情况输入)进行登录操作,此后相关命令即可对私有仓库镜像进行操作。

→ C	/1398468/edit			🖈 🕈 狐 i
应用 🕞 个人工作台 🕻 项目・首页・CODE	📸 百度一下, 你就知道 🕞 云效·阿里云一站	❶ Teambition · 团队 🕞 云	效 🥝 「飞流」产品介绍 😡 飞流 · 自动化研发	» 🕴 📋 阅读清单
返回 主机docker部署	基本信息 流	程配置 触发设置 变量和:	缓存 仅保存	存保存并运行
	部署	新阶段	编辑 ⑪	×
Java 镜像构建	多 Docker部署	●新的任务	脚本知下: ポポリーを用的Simage 是態在脚本下力的: (上幹論出度目注え) 2 echo \$DOCKER_PASSMORD   docker logint \$DOCKER_USERNAME password stdin re hangzhou.aliyuncs.com 3 docker run \$image 空量进行编码	变量处定义的变量 username egistry.cn
			交量 上游输出 ∨ image = 标签: + 添加参数	\$(DATET 🗸 🖨
5) 配置「 <del>部署策略」, 避免不稳定发布</del>	对用户的影响,保障业务交付的稳定。		部型策略	

### 6.2.7. 部署历史 和 回滚

用户可在 流水执行记录 —> 部署历史 中,查看到该流水线所有执行的部署历史记录,并可选择其中任意一条历史记录进行回滚操作。 Flow 会根据当时运行的部署脚本和构建制品重新执行部署任务,以实现回滚的效果。

 🔁 > php-	无构建部署流	水线 20 🔻 🄺						+ 99+ ?
最近运行 运	行历史 部	署历史					💿 运行	2/编辑 凡成员 三菜单
主机部署(部署	组:新的无构函	書部署的部署组) 🗸						
部署记录	状态	代码版本	构建产物	开始部署时间	部署任务	执行人	说明	操作
#190	0	master 24c9a27	-	12-29 23:06	主机部署	👿 柳夕测试1	-	回滚到
#189	0	master 24c9a27	-	12-29 22:34	主机部署	👿 柳夕测试1	-	回滚到
#188	0	master 24c9a27	-	12-29 22:03	主机部署	👿 柳夕测试1	-	回滚到
#187	0	master 24c9a27	-	12-29 21:35	主机部署	👿 柳夕测试1	-	回滚到
#186	0	master 24c9a27	-	12-29 21:04	主机部署	👿 柳夕测试1	-	回滚到
#185	ø	master 24c9a27	-	12-29 20:33	主机部署	👿 柳夕测试1	-	回滚到
#184	ø	master 24c9a27	-	12-29 20:04	主机部署	👿 柳夕测试1	-	回滚到
#183	0	master 24c9a27	-	12-29 19:33	主机部署	🗴 柳夕测试1	-	回滚到

# 6.3. Kubernetes 部署

对于 Kubernetes 用户 Flow 内置了多种发布能力,从而帮助用户快速实现 Kubernetes 应用的发布。

#### 建立集群连接

为了能够在 Flow 中完成 Kubernetes 的应用发布,用户需要首先建立 Kubernetes 集群连接,支持用户使用证书导入自建 Kubernetes 集群或者使用阿里云 RAM 授权导入 ACK 集群:

#### Kubernetes 发布方式

对于使用的原生 YAML 的用户, Flow 提供了 kubectl 发布步骤可以将代码库中的 YAML 部署到 Kubernetes 集群中,同时支持 YAML 的模板参数化,以实现更灵活的 YAML复用:

在某些场景下应用的部署由 Ops 人员完成,此时开发人员只需要负责将对应应用镜像进行升级,这个时候可以使用" Kubernetes 镜像升级"步骤对集群中已有的应用进行镜像升级,支持Deployment, StatefulSet, DaemonSet等多种应用类型:Kubernetes 镜像升级

如果用户采用了 Helm Chart 对应用镜像管理,Flow 还支持用户使用代码库中的 Chart 发布或者升级 HELM Release 实例: Helm Release 部署除了原生的发布能力支持以 外,Flow 为了让整个发布过程变的更加可控,Flow 实现了"Kubernetes 分批发布"能力:Kubernetes 分批发布

用户可以根据自己的实际情况选择适合自己的发布方式。

#### 常见问题

在使用云效Kubernetes发布时的常见问题请参考:

### 6.3.1. Kubernetes 部署

对于 Kubernetes 用户 Flow 内置了多种发布能力,从而帮助用户快速实现 Kubernetes 应用的发布。

### 建立集群连接

为了能够在 Flow 中完成 Kubernetes 的应用发布,用户需要首先建立 Kubernetes 集群连接,支持用户使用证书导入自建 Kubernetes 集群或者使用阿里云 RAM 授权导入 ACK 集群:建立 Kubernetes 服务连接

### Kubernetes 发布方式

对于使用的原生 YAML 的用户, Flow 提供了 kubectl 发布步骤可以将代码库中的 YAML 部署到 Kubernetes 集群中,同时支持 YAML 的模板参数化,以实现更灵活的 YAML 复用: Kubernetes Kubectl 发布

在某些场景下应用的部署由 Ops 人员完成,此时开发人员只需要负责将对应应用镜像进行升级,这个时候可以使用"Kubernetes 镜像升级"步骤对集群中已有的应用进行镜像升级,支持Deployment, StatefulSet, DaemonSet等多种应用类型:Kubernetes 镜像升级

如果用户采用了 Helm Chart 对应用镜像管理, Flow 还支持用户使用代码库中的 Chart 发布或者升级 HELM Release 实例:Helm Release 部署

除了原生的发布能力支持以外,Flow 为了让整个发布过程变的更加可控,Flow 实现了"Kubernetes 分批发布"能力:Kubernetes 分批发布

用户可以根据自己的实际情况选择适合自己的发布方式。

#### 常见问题

在使用云效Kubernetes发布时的常见问题请参考:Kubernetes发布常见问题

### 6.3.2. Kubernetes 服务连接

### 什么是 Kubernetes 服务连接

服务连接是 Flow 中用于与第三方服务交互的通用规范,顾名思义,Kubernetes 服务连接即用户可以通过 Kubernetes 证书文件或者是云服务授权的方式,让 Flow 可以将应 用发布到第三方的 Kubernetes 集群中。

### 阿里云容器服务 Kubernetes 集群 (ACK)

在 Kubernetes 集群管理中新建 Kubernetes 集群,如下所示:

返回		企业设置			
工具箱	Kubernetes集群管理 ②			Q 7	<b>?</b>
& 通用变量组	集群名称	集群类型	创建人	创建时间	操作
□ 标签管理	sam-integration	阿里云容器服务	🗙 光晕	2021-01-13 10:46:50	C T
🗟 授权管理	aone-build-engine-test	阿里云容器服务	rdc-rdc	2021-01-12 18:46:05	ĊŴ
⇒ 主机组管理	私有网络集群	阿里云容器服务	rdc-rdc	2021-01-12 15:12:49	ĊŴ
Kubernetes集群管理	云效	自定义集群	⑦ 雪影 3	2020-08-19 15:19:00	ĊŴ
<ul> <li>任务组管理</li> </ul>	云效	自定义集群	③ 雪影 3	2020-08-19 13:36:44	C Ŵ
123 步骤管理	test	阿里云容器服务	rdc-rdc	2020-08-13 21:22:47	C Ŵ
8 构建集群管理	kube-vela	阿里云容器服务	rdc-rdc	2020-07-27 14:56:43	C T
管理员设置	test	自定义集群	rdc-rdc	2020-06-23 14:04:51	C Ŵ
<b>爲</b> 角色设置		<	1 2 >		

选择集群类型为阿里云容器服务集群,选择授权后,在下拉框中用户可以选择当前绑定账号可以访问的所有阿里云容器服务Kubernetes集群列表。

- 同时你可设置集群的成员角色:
- 管理员:可以编辑/删除该集群。
- 成员: 在流水线编辑时, 可以使用该集群。

工具箱	新建Kubernetes集群 × Kube	♥ ● 新建Kubernetes集都
& 通用变量组	集群类型	操作
□ 标签管理	sar C-) 1-13 10:46:50	
🔁 授权管理	自定义集群 谷器服务集群 aor 1-12 18:46:05	c t
主机组管理	<ul> <li>ペ rdc-rdc的ack服务授权 &gt; (1) 添加 解绑</li> <li>私7</li> </ul>	C Ŵ
🥘 Kubernetes集群管理	ACK集群 9	
고 流水线模板管理	kube-vela	
☑ 任务组管理	云3 8-19 13:36:44	4 🗹 🖤
123 步骤管理	tes 🗶 + 3	7 🖻 🖤
8 构建集群管理	kut 取消 保存 7-27 14:56:4	3 🗹 🗇
管理员设置	test 自定义集群 🛞 rdc-rdc 2020-06-23 14:04:5	1 🗹 🖤

### 连接到自定义 Kubernetes 集群

在 Kubernetes 集群管理中新建 Kubernetes 集群,如下所示:
# 云效2020

返回		企业设置			
工具箱	Kubernetes集群管理 ⑦			9, 7	新建Kubernetes集君
& 通用变量组	集群名称	集群类型	创建人	创建时间	操作
⇒ 标签管理	sam-integration	阿里云容器服务	🗙 光晕	2021-01-13 10:46:50	ĊŴ
⊇ 授权管理	aone-build-engine-test	阿里云容器服务	rdc-rdc	2021-01-12 18:46:05	C t
	私有网络集群	阿里云容器服务	rdc-rdc	2021-01-12 15:12:49	¢ ŵ
<ul> <li>不如是们包括S条件管理</li> <li>流水线模板管理</li> </ul>	云效	自定义集群	⑦ 雪影 ■	2020-08-19 15:19:00	¢ Ŵ
🖸 任务组管理	云效	自定义集群	③ 雪影 3	2020-08-19 13:36:44	¢ Ŵ
123〕步骤管理	test	阿里云容器服务	rdc-rdc	2020-08-13 21:22:47	C Ŵ
2  构建集群管理	kube-vela	阿里云容器服务	rdc-rdc	2020-07-27 14:56:43	C Ū
管理员设置	test	自定义集群	rdc-rdc	2020-06-23 14:04:51	¢ Ŵ
0. 角色设置					

选择自定义集群,输入集群名称并提供集群配置文件,点击保存即可。在后续的发布任务中将使用该配置文件发布应用到远程集群。

同时你可设置集群的成员角色:

- 管理员:可以编辑/删除该集群。
- 成员: 在流水线编辑时, 可以使用该集群。

< 返回	企业设置			×
工具箱	新建Kubernetes集群	×	G J	● 新建Kubernetes集群
& 通用变量组	集群类型		1	操作
□ 标签管理	sar 😌 (-)		1-13 10:46:50	C Ŵ
🔁 授权管理	自定义集群 容磁服务集群 aor		1-12 18:46:05	C Ū
🗊 主机组管理	集群名称 私3		1-12 15:12:49	ि भी
🙆 Kubernetes集群管理	kube-vela			
💿 流水线模板管理	云3 集群配置文件 ⑦		8-19 15:19:00	C T
☑ 任务组管理	≂/.kuba/config		8-19 13:36:44	C Ū
123 步骤管理	tes		8-13 21:22:47	C T
2 构建集群管理	kut +		7-27 14:56:43	C T
管理员设置	tes 取消 保	存	6-23 14:04:51	C t
风 角色设置	< 1 2 →			

对于自定义 Kubernetes 集群如何获取证书文件,请查看"如何手动生成 Kubernetes Config 文件"一节

# 6.3.2.1. Kubernetes 服务连接

#### 什么是 Kubernetes 服务连接

服务连接是 Flow 中用于与第三方服务交互的通用规范,顾名思义,Kubernetes 服务连接即用户可以通过 Kubernetes 证书文件或者是云服务授权的方式,让 Flow 可以将应 用发布到第三方的 Kubernetes 集群中。

### 阿里云容器服务 Kubernetes 集群 (ACK)

在 Kubernetes 集群管理中新建 Kubernetes 集群,如下所示:

返回		企业设置			
工具箱	Kubernetes集群管理 ⑦			0° 7	● 新建Kubernetes集群
& 通用变量组	集群名称	集群类型	创建人	创建时间	操作
□ 标签管理	sam-integration	阿里云容器服务	🗙 光晕	2021-01-13 10:46:50	¢ ŵ
🔁 授权管理	aone-build-engine-test	阿里云容器服务	rdc-rdc	2021-01-12 18:46:05	ľ Ŵ
	私有网络集群	阿里云容器服务	rdc-rdc	2021-01-12 15:12:49	¢ ¢
☆ Kubernetes集群管理 ✓ 流水线模板管理	云效	自定义集群	● 雪影	2020-08-19 15:19:00	¢ ¢
🖸 任务组管理	云效	自定义集群	● 雪影	2020-08-19 13:36:44	¢ ¢
123 步骤管理	test	阿里云容器服务	rdc-rdc	2020-08-13 21:22:47	¢ †
2   构建集群管理	kube-vela	阿里云容器服务	rdc-rdc	2020-07-27 14:56:43	¢ †
管理员设置	test	自定义集群	rdc-rdc	2020-06-23 14:04:51	¢ ¢
① 角色设置					

选择集群类型为阿里云容器服务集群,选择授权后,在下拉框中用户可以选择当前绑定账号可以访问的所有阿里云容器服务Kubernetes集群列表。 同时你可设置集群的成员角色:

管理员:可以编辑/删除该集群。

• 成员:在流水线编辑时,可以使用该集群。

工具箱	新建Kubernetes集群 × Kube	🔍 🔽 <table-cell-rows> 新建Kubernetes集群</table-cell-rows>
& 通用变量组	集群类型	操作
□ 标签管理	sar 🕸 🗂	50 🗹 🗊 .
🔁 授权管理	自定义集群 容器服务集群 aor 1-12 18:46:	05 🗹 🖞
□ 主机组管理	久 rdc-rdc的ack服务授权 ~ (1) 添加 解绑	49 🗹 🐨
🛞 Kubernetes集群管理	ACK集群 😌	
💿 流水线模板管理		00 🗹 🖤
🕞 任务组管理	云5 8-19 13:36	:44 🗹 🗊
122 步骤管理	tes 2 (+) (3) 8-13 21:22	:47 🖻 🖞
2 构建集群管理	kut 取消 保存 7-27 14:56	:43 🗹 🖞
管理员设置	test 自定义集群 🕐 rdc-rdc 2020-06-23 14:04	i:51 🗹 🛈

### 连接到自定义 Kubernetes 集群

在 Kubernetes 集群管理中新建 Kubernetes 集群,如下所示:

返回		企业设置			
工具箱	Kubernetes集群管理 ⑦			9, 7	● 新建Kubernetes集群
& 通用变量组	集群名称	集群类型	创建人	创建时间	操作
□ 标签管理	sam-integration	阿里云容器服务	🗙 光晕	2021-01-13 10:46:50	ľŴ
🔁 授权管理	aone-build-engine-test	阿里云容器服务	rdc-rdc	2021-01-12 18:46:05	¢ †
E 主机组管理 Kubernetes集群管理	私有网络集群	阿里云容器服务	rdc-rdc	2021-01-12 15:12:49	ľ Ŵ
○ 流水线模板管理	云效	自定义集群	● 雪影	2020-08-19 15:19:00	ľ Ŵ
🗠 任务组管理	云效	自定义集群	◆ 雪影 🐱	2020-08-19 13:36:44	ľ Ŵ
123 步骤管理	test	阿里云容器服务	rdc-rdc	2020-08-13 21:22:47	¢ ŵ
◎ 构建集群管理	kube-vela	阿里云容器服务	rdc-rdc	2020-07-27 14:56:43	ĩ ŵ
管理员设置	test	自定义集群	rdc-rdc	2020-06-23 14:04:51	¢ ŵ
<u> 风</u> 角色设置		,			

选择自定义集群,输入集群名称并提供集群配置文件,点击保存即可。在后续的发布任务中将使用该配置文件发布应用到远程集群。

同时你可设置集群的成员角色:

- 管理员:可以编辑/删除该集群。
- 成员: 在流水线编辑时, 可以使用该集群。

< 返回	企业设置			×
工具箱	新建Kubernetes集群	×	Q 7	° ● 新建Kubernetes集群
念。通用变量组	集群关型		1	操作
□ 标签管理	Sar         Cつ           自定义集群         容器服务集群		1-13 10:46:50	C T
□ 授权管理	aor		1-12 18:46:05	ĊŴ
□ 主机组管理	集群名称 私行 kube-vela		1-12 15:12:49	C t
Subernetes集件管理 ○ 流水线模板管理	云; 集群配置文件 ⑦		8-19 15:19:00	C Û
← 任务组管理	云3 ~/.kube/config		8-19 13:36:44	C T
123 步骤管理	tes		8-13 21:22:47	C t
8 构建集群管理	kut +		7-27 14:56:43	C t
管理员设置	tes	保存	6-23 14:04:51	C Ū
<b>风 角色设置</b>	< 1 2 →			

对于自定义 Kubernetes 集群如何获取证书文件,请查看"如何手动生成 Kubernetes Config 文件"一节

# 6.3.2.2. 获取 Kubernetes 集群证书文件

### 从阿里云容器服务控制台获取 config 文件

对于使用了阿里云容器服务的用户,可以直接从阿里云控制台获取到集群的 config 文件。需要注意的是,使用证书的方式需要用户集群的 API Server 对外暴露公网访问地 址。



### 获取自建Kubernetes集群的 config 文件

用户可以直接通过master节点拷贝~/.kube/config文件即可。 如果需要全新生成config文件,可以参考<mark>"如何手动生成 Kubernetes Config</mark> 文件"一节

### 6.3.2.3. 如何手动生成 Kubernetes Config 文件

### 生成集群管理权限的 config 证书

如果需要对客户端做细粒度的权限控制,可以通过以下方式生成 config 文件。

```
curl -k https://<YOUR_API_SERVER_PUBLIC_IP>:6443
```

#### 创建 ~/.kube/config 文件,并修改文件内容

apiVersion: v1 clusters: - cluster: # needed if you get error "Unable to connect to the server: x509: certificate signed by unknown authority" insecure-skip-tls-verify: true server: https://YOUR_API_SERVER_PUBLIC_IP:6443 name: kubernetes contexts: - context: cluster: kubernetes user: kubernetes-admin name: kubernetes-admin@kubernetes current-context: kubernetes-admin@kubernetes kind: Config preferences: {} users: - name: kubernetes-admin user: client-certificate-data: LSOtLS1CRUdJTi... (base64 /etc/kubernetes/ssl/node-node1.pem) client-key-data: LSOtLS1CRUdJTiBS.. (base64 /etc/kubernetes/ssl/node-node1-key.pem) 其中 client-certificate-data 来源于:

cat /etc/kubernetes/ssl/node-node1.pem | base64 -w 0

#### client-key-data 来源于以下命令的输出:

cat /etc/kubernetes/ssl/node-node1-key.pem | base64 -w 0

#### 验证证书:

\$ kubectl	get nodes		
NAME	STATUS	AGE	VERSION
my-kube	Ready	2h	v1.6.7+coreos.0

# 6.3.3. Kubernetes Kubectl 发布

Flow 提供了通用的 Kubernetes 部署能力支持,通过内置的 Kubectl 工具将 Kubernetes YAML资源描述文件部署到任意集群中。在原生 Kubectl 的能力基础上, Flow 还内置 了动态模板的能力,从而帮助用户可以快速实现容器应用的构建与发布。

#### 添加部署任务

- 可以通过两种形式添加部署任务
- 1. 创建流水线时,选择包含 Kubernet es 发布的模板
- 2. 在流水线中添加空任务,并添加 Kubectl 发布步骤

#### 配置部署任务

- 进入 Kubectl 步骤,进行如下配置:
- 1. 选择或者新建 Kubernetes 集群连接
  - 关于如何新建 Kubernetes 集群连接可以查看"建立 Kubernetes 服务连接"一节
- 2. 输入发布的目标 Namespace
- 3. 输入当前需要部署的 YAML 文件路径
  - 关于如何在 YAML 中使用动态生成的镜像可以查看"如何在 YAML 中使用自定义变量"一节

#### 查看部署日志

在配置完成后,运行流水线执行部署任务,即会将源码中对应的 YAML 文件部署到指定的Kubernetes 集群中。 在部署过程中点击"日志",可以查看当前的部署过程以及相应记录。

如果部署失败,可以查看 "查看 Kubernetes 部署日志" 一节了解,如何查看日志以及问题排查定位

### 6.3.3.1. Kubernetes Kubectl 发布

Flow 提供了通用的 Kubernetes 部署能力支持,通过内置的 Kubectl 工具将 Kubernetes YAML资源描述文件部署到任意集群中。在原生 Kubectl 的能力基础上, Flow 还内置 了动态模板的能力,从而帮助用户可以快速实现容器应用的构建与发布。

#### 添加部署任务

可以通过两种形式添加部署任务

- 1. 创建流水线时,选择包含 Kubernetes 发布的模板
- 2. 在流水线中添加空任务,并添加 Kubectl 发布步骤

#### 配置部署任务

- 进入 Kubectl 步骤,进行如下配置:
- 1. 选择或者新建 Kubernetes 集群连接
- 关于如何新建 Kubernetes 集群连接可以查看"建立 Kubernetes 服务连接"一节
- 2. 输入发布的目标 Namespace
- 3. 输入当前需要部署的 YAML 文件路径
  - 关于如何在 YAML 中使用动态生成的镜像可以查看 "如何在 YAML 中使用自定义变量"一节

#### 查看部署日志

在配置完成后,运行流水线执行部署任务,即会将源码中对应的 YAML 文件部署到指定的Kubernetes 集群中。在部署过程中点击"日志",可以查看当前的部署过程以及相应记录。

如果部署失败,可以查看 "查看 Kubernetes 部署日志" 一节了解,如何查看日志以及问题排查定位

### 6.3.3.2. 如何在 YAML 中使用自定义变量

一般来说,在 Kubernetes 的资源描述 YAML 中我们会定义应用所有的部署资源并通过标签定义资源之间的关联关系。

Kubernetes常用资源YAML使用方式,可以参考"Kubernetes常用资源 YAML 规范参考"一节

用户可以将 Kubernetes YAML文件与源码保存在一个代码库,或者是拆分到不同的仓库中。为了能够在 YAML 中动态的使用来自前序任务的出参,如 Docker 镜像构建产生的 镜像,我们需要对 YAML文件进行模板化处理。

# 在 YAML 中定义变量

如下图所示,在 Kubernetes 资源描述 YAML 文件中,通过\${IMAGE} 占位符将 Docker 镜像 定义为一个动态参变量。

源文件(320K) 提交(5) 分支(1) 标签	合并请求 概览 +	A. 成员 ③ i
$\wp$ master $\checkmark$ $\sim$ $\sim$ +	k8s-sample / manifest / app-deployyaml	
<ul> <li>▼ k8s-sample</li> <li>▼ iii manifest</li> </ul>	更新 app-deployyaml ② 砧木 提交手 現在	8a56c42d
<ul> <li>keep</li> <li>app-deploy.vaml</li> </ul>	app-deploy.yaml	🖉 🖞 Web IDE …
Papp-svc.yaml README.md	<pre>1 epiVersion: extensions/vlbetal 2 kind: begloysmet 3 metadata: 4 labels: 5 run: app-ample 6 name: app-ample 7 spec: 8 selector: 9 matchLabels: 10 run: app-ample 11 template: 13 sentels: 14 run: app-ample 15 spec: 15 spec: 15 spec: 16 containers: 17 - image s[NIMOR] 18 imagePailPoilcy: Always 19 name: app-ample 10 ample: 11 - ample: 12 - ample: 13 - ample: 14 - ample: 15 - ample: 16 - ample: 17 - ample: 18 - ample: 19 name: app-ample 19 - ammle: 10 - ample: 11 - ample: 12 - ample: 13 - ample: 14 - ample: 15 - ample: 16 - ample: 17 - ample: 18 - ample: 19 - ammle: app-ample 19 - ammle: 10 - ample: 10 - ampl</pre>	

### 在Kubectl 步骤中传递变量给 YAML

在 Kubectl 步骤配置中,点击添加变量,并定义变量名称以及变量来源,即可发布时使用这些变量动态替换 YAML 文件中的占位符。

如下图所示*,*在变量中添加变量。

变量名需要和 YAML中定义的变量名一致,如示例中" IMAGE"。

变量值可以选择流水线前序任务的出参,如示例中选择的是前序构建任务的出参,即新构建出来的构建镜像。

■ 流小线编排	加小小35 2019-10-16				保存集合
源	新建源	新称合	新阶段	编辑	×
C 5d3fbf7c7	115200001f80397/ 镜像构建	B & Kubernetes	发布 🖬 💿 新的任务	任务步骤 ③ 添加步骤	÷
				步骤名称	W
				Kubectl发布	
				集群连接	● 新建连接
				测试集群X	~
				命名空间	0
				default	
				YAML路径	0
				manifests	
				变量	0
				IMAGE	标签.\${DATETIME} V
				十 添加參数	1)
Ø	定时运行				

# 6.3.3.3. Kubernetes常用资源 YAML 规范参考

# 工作负载

资源类型	官方文档链接
Deployments	https://kubernetes.io/docs/concepts/workloads/controllers/deployment/
StatefulSets	https://kubernetes.io/docs/concepts/workloads/controllers/statefulset/

资源类型	官方文档链接
DaemonSet	https://kubernetes.io/docs/concepts/workloads/controllers/daemonset/

# 服务发现与网络

资源类型	官方文档链接
Service	https://kubernetes.io/docs/concepts/services-networking/service/
Ingress	https://kubernetes.io/docs/concepts/services-networking/ingress/

更多参考请访问 Kubernetes 官方文档。

# 6.3.3.4. 查看 Kubernetes 部署日志

在流水线运行页面,点击任务的日志按钮即可查看 Kubernetes的 部署日志内容:

					● 运行 // 编辑 《	成员 三菜单
#5 🔮 运行成功	构建		部署			
G tyuis/shell	⊘ Docker镜像构建		✓ Kubernetes发布			
№ master -0- 75cc3d2	运行成功		运行成功			
	〇 日志	56秒	日志	3210		
运行人	计 砧木					

日志输出,如下所示,日志中会包含当前所有扫描到的 YAML 文件,kubectl apply后的标准输出,以及当前变更执行后的所有资源概要信息。

<b>最近运行</b> 运行历		Kubernetes发布 × 庸 鸟 成员 🗄
	构建	Kubectl发布(7s)
#5 🔮 延行成功	环境分派	[USER] PATH_TO_YAML=demo2
G tyuis/shell	环境初始化(1s)	[USEK] INSECURE_SKIP_ILS_VERIPY=Taise 2019/10/18 11:51:07 [INFO] save date to /root/.kube/config
‰ master	⊘ 清理工作区(9s)	
-o- 75cc3d2	☑ 克隆代码(7s)	[INFO] Kubernetes集群基本信息 Client Version: v0.0.0-master+\$Format:%h\$
	Kubectl发布(7s)	Server Version: v1.11.5
		[INFO] 准备应用Manifests定义文件
	(1)-	Generate /root/workspace/tyuis_shell_15G6/demo2/nginx.yaml to /root/workspace/tyuis_shell_15G6/.manifest/nginx.yaml
	<u> </u>	[INFO] 部署变更到Kubernetes集群
	2	service/nginx unchanged
		deployment.extensions/nginx unchanged
		[INFO] 打印当前任务执行结果
	3	service/nginx
		NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE nainx ClusterIP ***** <none> 80/TCP 8d</none>
		deployment.extensions/nginx
		NAME DESIRED CURRENT UP-TO-DATE AVAILABLE AGE
运行人		nginx 1 1 1 1 8d [INFO]执行用户命令成功
开始时间 2019		
		[executionStep ends at 2019–10–18 11:51:11.799]
运行时长	1分28秒	

# 6.3.4. Kubernetes 镜像升级

Flow 提供了 Kubernetes 镜像升级任务,支持用户对集群中的已有应用进行镜像升级。

#### 添加任务

用户可以通过添加任务选择 Kubernetes 镜像升级任务

源(1)			选择任务组		×		
…/flow-example/spring-b- imaster	企业任务组 代码扫描	Q。请输入任务名称	使用脚本发布到主机	主机Docker部署			
変量 (0)	测试 测试构建 构建	Kubernetes 发育     用于发布 Kubernetes	Kubernetes 镜像升级 升级 Kubernetes 下的	Kubernetes 分批发布 Kubernetes 分批发布	j		
智木议皇, 宗古 280月	镜像构建 工具 部署	Helm Release 部署 使用代码库中的 Helm	<b>EDAS ECS应用发布</b> EDAS ECS应用发布	<b>EDAS Kubernetes</b> EDAS Kubernetes 应用			
	代码 执行命令	Serverless(SAE) 应 Serverless(SAE) 应用	(元) 函数计算 应用发布 更新函数计算 (FC) 中				
	空模板	代码					
		合并代码 这种风险入海由继定的	松渕 Master 分支是 絵画 Master 公支星本	◆ 创建标签 使用 Git tag金会创建标			

### 部署配置

对于首次使用 Flow Kubernetes 发布相关功能的用户需要首先建立 Kubernetes 服务连接

建立连接后,选择 Kubectl 版本并设置需要发布的应用基本信息即可。需要注意的是应用的容器名需要与 Pod 中 Container 的名称一致,以确保 Pod 中存在多个 Container 时 应该升级对应的镜像:

	新阶段	新阶段	编辑 ⑪	
			2010-1-4F (LL)	
			▼ Kubectl镜像升级 ②	
+添加流水线源 🕀		新的任务	步骤名称	
			Kubectl镜像升级	
			集群连接 ②	● 新建连接
			请选择	~
			Kubectl版本 ⑦	
			v1.16.4	~
			命名空间 ?	
			default	
			Workloads 类型 ⑦	
			Deployment	~
			Workloads 名称 ⑦	
			mm data 🔿	

保存并运行流水线

							0 1015	
最近运行 运行	历史					● 运行	∅ 编辑	爲 成员 三葉
#1 🜀 运行中	⑧ 终止	执行命令						
运行人	🥙 136	阿里云镜像构建		Subernetes 镜像升级				
开始时间 202	20-03-13 15:43:50	运行成工	ъ	运行中				
持续时间	52秒	四 日志	43秒	※ 取消 □ 日志	9秒			
▶ 源(1)								

#### 点击查看日志,查看发布的详细信息:

<b>最近运行</b> 运行历.		Kubernetes 镜像升级 X	辑	鸟 成员	≡菜单
	构建	Kubecti镜像升级			
#1 🕑 141 J 782-93	♥ 申请运行环境(14s)	[15:45:04] [USER] CONTAINER_NAME=nginx (15:45:04) 2020/03/13 15:45:04 [INFO] wave data to /root/ hube/config			
运行人	⊘ 清理工作区(5s)	[15:45:04]			
开始时间 2020-	⊘ 克隆代码(8s)	[15:45:04] [INFO] Kubernetes集群基本信息 [15:45:05] Client Margion, x1 16 4			
1741431-3	<ul> <li>Kubectl镜像升级(6s)</li> </ul>	[15:45:05] Server Version: v1.11.5			
持续时间	-	[15:45:05]			
		<pre>[15:45:05] deployments, statefulsets, daemonsets</pre>			
N 307 (4)		[15:45:05] 校验Container配置			
P AB((1)		[15:45:05] 当前应用nginx中包含1个容易定义 (15:45:05, 1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100-1/100			
		[13:45:05] [INEO] 即将次时PEGISTY.CH-HANGZHOU.AIIYUNCS.COM/FACTEST/FAC-Kos-test:2020-03-13-13-43- 5.6到後上的深邃			
		[15:45:05] deployment.extensions/nginx			
		[15:45:05]			
		[15:45:05] [INFO] 部署变更到Kubernetes集群			
		[15:45:05] deployment.extensions/nginx image updated			
		[15:45:05]			
		[15:45:05] [INFO] 打印当前任务执行结果			
		[15:45:05] deployment.extensions/nginx			
		[15:45:05] NAME DESIRED CURRENT OF-TO-DATE AVAILABLE AGE			
		[15:15:05]			
		[15:45:05]			
		[15:45:05]			
		[15:45:05] [SUCCESS] 步骤运行成功			
		[2020-03-13 15:45:06] [PLUGIN] 执行成功			

# 6.3.5. Kubernetes 分批发布

Kubernetes 原生的应用升级基于终态模型,一旦发布出现问题无法暂停,只能通过YAML回滚应用。为了使 Kubernetes 应用发布过程更加可控。Flow 提供了 Kubernetes 分 批发布的能力,以让发布过程更加可控。

### 实现原理

- Kubernetes 使用 Service 在集群内暴露服务
- Service 与Deployment 的 Pod 实例之间使用 Lables 进行关联, Service 的流量会转到到关联的 Deployment 实例
- 假设 serviceA 关联了应用 app-v1 并且其 Replicas 副本数 > 1
- 分批发布开始创建 app-v2 并继承 app-v1 的所有标签,此时 serviceA 的流量会随机转发到 v1 和 v2
- 分批发布过程通过扩容 app-v2 和缩容 app-v1 实现应用的发布
- 发布过程暂停时,用户可以选择继续发布,直到 app-v2 达到预期状态
- 或者回滚移除 app-v2 并恢复 app-v1 的副本数

#### 添加分批发布任务

用户可以通过添加流水线任务并选择"Kubernetes分批发布"任务:

源(1)			选择任务组	>	<
<ul> <li>/flow-example/spring-b</li> <li>master</li> </ul>	企业任务组 代码扫描	Q。请输入任务名称 2011年			
变量 (0) 6	测试构建 构建	19-34 <b>主机部署</b> 主机部署	<b>主机部署(元构建包)</b> 使用脚本发布到主机	Docker部署     主机Docker部署     1	
智木议重, 尽击 滁川	镜像构建 工具	Kubernetes 发布 用于发布 Kubernetes Y	Kubernetes 镜像升级 升级 Kubernetes 下的	Kubernetes 分批发布 Kubernetes 分批发布	
	代码执行命令	Helm Release 部署 使用代码库中的 Helm	EDAS ECS应用发布 EDAS ECS应用发布	EDAS Kubernetes EDAS Kubernetes 应用	
	空模板	Serverless(SAE) 应 Serverless(SAE) 应用	(FC) 函数计算 应用发布 更新函数计算 (FC) 中		

### 配置发布任务

首次使用 Kubernetes 相关发布功能的用户需要首先建立 Kubernetes 服务连接。

选择发布的目标 Service 以及 Service 关联的 Deployment 中的容器名称。镜像需要由上游镜像构建步骤产生:

■ 流水线编排 流水线 2020-03-13				保存 保存开运行
▼ 源(1) +		新阶段	编辑	
…/flow-example/spring-boot			命名空间 ②	
i‰ master 2	阿里云視線构建 Rubernetes 分批发佈	G 新的任务	default	
▼ 変量 (0)			服务Service ⑦	
			nginx-svc	
暂未设置,点击 添加			关联的应用类型 ⑦	
			Deployment	
			分批策略 ②	
			分批发布第一批暂停	
			分批数 ⑦	
			2	
			容器名称 ⑦	
			nginx	
			镜像 ⑦	
			标签.\${DATETIME}	
(1) 定时运行			<ul> <li>跳过 TLS 校验 ②</li> </ul>	

**执行分批发布** 当使用分批发布第一批暂停,当第一批次发布完成后,卡片会显示暂停状态,并提示用户继续下一批次或者回滚

<b>最近运行</b> 运	行历史						• 运行	の編辑の	冬成员 三菜
#2 🖸 运行中	● 终止	构建							
运行人	🥙 136	⊘ 阿里云镜像构建		🗿 Kub	ernetes 分批发布				
开始时间 2	2020-03-13 17:58:38	运行成	功	当前发3	运行中 5暂停中,请确认后选择	恢复或者回滾			
待续时间	2分55秒	四 日志	39秒	② 取消	更多 ^	2分16秒			
r ad(1)					<ul> <li>) 回滚</li> <li>) 恢复</li> <li>□ 日志</li> </ul>				

#### 在人工验证完成后,选择继续发布完成后续批次的发布

<b>最近运行</b> 运行历史						💽 运行	の編輯の	& 成员
2 🔮 运行成功	构建							
衍人	🥙 136 💿 阿里	云镜像构建		S Kubernetes 分批发	布			
一始时间 2020-03-	3 17:58:38	运行成功		发布成工	b			
持续时间	4分31秒		2055	nginx-svc->nginx-	svc-mtkyzt			
源(1)								

# 6.3.6. Kubernetes 蓝绿发布

对于 lstio 用户,云效提供了蓝绿发布的能力支持。从而让用户可以更安全的基于 lstio 的流量管理策略实现应用的安全发布。

#### 实现原理

- 在 Kubernetes 中 Service 与多个版本的 Deployment 实例直接通过 LabelSelector 进行关联。通过 VirtualService 可以进一步控制流量与特定版本Deployment实例之间的 路由策略。
- 蓝绿发布开始,飞流会基于当前的 Deployment 实例(绿环境)创建一个新版本的应用 Deployment 实例(蓝环境)。
- 等待蓝环境创建完成后通过更新 lstio 的 DestinationRule 资源对象,生成路由表信息
- 根据用户的发布设置,修改 VirtualService 确定蓝/绿环境的流量权重。
- 用户也可以设置灰度标,确定哪些流量会直接进入到蓝环境中。
- 在人工验证完成后,完成发布将所有流量切流到蓝环境,并且将原有的绿环境实例移除。

### 准备 Istio 应用

以阿里云容器服务 Kubernetes 版为例,在创建完成集群后通过部署 Istio 组件就可以在控制台部署应用并绑定虚拟服务 (VirtualService),如下所示:

■ (-) 阿里云 账	司全部資源 🕶 🤡 全球				Q 提案文档、控制台、API、解决方案和资源	费用	工单	备案 企	业 支持	官同	E.	Ф. A	0	前休 🎙
器服务 - Kubernetes →	虚拟服务 (Virtual Service)												8185	<b>8</b> 518
概范	集群 yunxiao-research	¢ 喻名空间 demo-namespace ¢ C									3	非输入查询	58	Q
集群	名称	工作负载	服务网关	创建时间										jit j
集群	sample-svc	v1 -> sample	構定	2020-05-18 16:47:1	5								(8   112	- 更多+
节点														
存储卷														
命名空间														
授权管理														
应用														
无状态														
有状态														
守护进程集														
任务														
定时任务														
容器组														
存储声明														
发布														
工作流														
路由与负载均衡														
服务														
路由														
服务网格														
India MURI														B

这里在控制台创建了一个名为 sample-svc 的虚拟服务,并关联了一个 v1 版本的应用实例。

### 在流水线中添加蓝绿发布任务

创建流水线并完成镜像构建配置后,用户可以通过添加任务向导添加"Kubernetes 蓝绿发布"任务,如下所示:

<ul> <li>● master</li> <li>● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●</li></ul>	▼ 源(1) +	执行命令			选择任务组		×	
Numative         Numative	F/flow-example/spring-boot	P1 0 098	代码扫描	Q, 请输入任务名称				
· SE (0)       · · · · · · · · · · · · · · · · · · ·	}∾ master	The y Prime	测试构建	部署				
TA BR BR CB CB CB CB CB CB CB CB CB CB CB CB CB	· 安量 (0)		构建 镜像构建	<b>主机部署</b> 主机部署	主机部署(无构建包) 使用脚本发布到主机	Docker部署 主机Docker部署		
代码 時方命令 空暖初	暂未设置,点击 添加		工具部署	校 Kubernetes 发布 用于发布 Kubernetes Y	<b>Kubernetes 镜像升级</b> 升级 Kubernetes 下的	Kubernetes 董録发布     基于Istio进行应用的有		
Minimum       Image: EDAS Rubernetes         EDAS Rubernetes       Image: Serverfess(SAU) S.R.         Email: Edas Rubernetes S.R.       Image: Serverfess(SAU) S.R.			代码 执行命令	Kubernetes 分批发布     Kubernetes 分批发布	Helm Release 部署 使用代码库中的 Helm	EDAS ECS应用发布 EDAS ECS应用发布		
			至模数	<b>EDAS Kubernetes</b> EDAS Kubernetes 应用	Serverless(SAE) 应 Serverless(SAE) 应用	(-) 函数计算 应用发布 更新函数计算 (FC)中		
							-	
								6

在蓝绿部署配置中,配置发布的集群以及发布的目标服务即可。在发布前需要确保目标集群已经正确安装 lstio 组件,且关联的命名空间需要开启 lstio 自动注入。如下所示:

☞ 源(1) +	新阶段	新阶段 <b>编辑</b>	
…/flow-example/spring-boot		Kubect版本 ⑦	
il> master	料理 日 ダ Kubernetes 监当	¥发布 () ● 納的任务 v1.16.4	
▼ 变量 (0)		命名空间 ⑦	
		default	
暂未设置,点击 添加		服务名称 ⑦	
		sample-svc	
		容器名称 ⑦	
		sample	
		鏡像 ⑦	
		标签.\$(DATETIME)	
		灰度流量Cookie标 ⑦	
		^(.*?;)?(gray=true)(;.*)?\$	
		灰度版本初始化流量权重 ⑦	
		0%	

通过设置灰度 Cookle 标,可以让用户通过 Cookle 来进入到新版环境从而可以进行新版环境的功能验证。灰度版本初始化流量权重设置可以在新版发布完成后让部分正式流量 进入到新版环境。对于应用启动时间较长的应用可以设置启动等待时间,确保在新版应用部署完成并等待指定时间后,使新版本的流量规则生效。

### 触发蓝绿部署

通过运行流水线,用户可以触发蓝绿部署流程。

#2 🕝 运行中	⑧ 终止	执行命令		新阶段				
运行人	rdc-rdc	阿里云镜像构建		G Kub	ernetes 蓝绿发布	i		
开始时间	2020-05-18 17:21:01	运行成	功	版	运行中 \$v2020051817215	9灰度验证中		
诗续时间	9分46秒	日志	22秒	⊗ 取消	更多 ^	9分24秒		
▶ 源(1)					<ul> <li>回源</li> <li>完成</li> <li>四目志</li> </ul>			
								0

在当前状态下用户可以通过 Cookie 标进去到新版环境中进行功能验证。在验证完成后通过点击完成,一键切流到新版应用。 如果验证不通过则通过回滚取消当前发布。

最近运行	运行历史					● 运行	∅ 编辑	,尽成员 ≡	≡菜单
#2 🕑 运行成	功	执行命令		新阶段					
运行人	rdc-rdc	⊘ 阿里云镜像构建		Kubernetes 蓝绿发	之布				
开始时间	2020-05-18 17:21:01	运行成	功	发布完	成				
持续时间	13分29秒	日 日志	22秒	日志	13分7秒				
▶ 源(1)									

在发布过程中用户也可以通过阿里云容器服务 Kubernetes 查看当前的流量策略。

器服务 - Kubernetes <del>▼</del>	sample-svc 主返	回列表					刷新
任务	基本信息						
定时任务	工作负载类型:	Deployment					
容器组	创建时间:	2020-05-18 16:47:15					
存储声明	版本管理 增加灰度版	本					
发布	版本号	工作负载	实例数	灰度策略	创建时间		操作
工作流	v20200518172159	sample-svc-v20200518172159	1	查看	2020-05-18 17:22:00	管理负载   删除版本   接管所有:	た 二流量治明
各由与负载均衡	v1	sample	1	流量比例 100%	2020-05-18 16:47:15	管理负载   删除版本   接管所有:	t量   流量治明
服务 [	v20200518172159	sample-svc-v20200518172159	1	流量比例 0%	2020-05-18 17:22:00	管理负载   删除版本   接管所有》	2量   流量治理
1011							
Istio 管理							
虚拟服务							
服务网关							
inative (公測)							
组件管理							G
072 Ar 601 198							

# 6.3.7. ACK Nginx Ingress 灰度发布

灰度及蓝绿发布是为新版本创建一个与老版本完全一致的生产环境,在不影响老版本的前提下,按照一定的规则把部分流量切换到新版本,当新版本试运行一段时间没有问题 后,将用户的全量流量从老版本迁移至新版本。

其中 A/B 测试就是一种灰度发布方式,一部分用户继续使用老版本的服务,将一部分用户的流量切换到新版本,如果新版本运行稳定,则逐步将所有用户迁移到新版本。 采用 Ingress 灰度发布方式用户可以:

- 1. 控制新版本流量分配权重, 以小部分线上流量对服务进行验证
- 2. 通过 cookie 或者 header 使得部分受控用户在线上对发布进行验证
- 3. 当发布验证失败后,可以快速回滚到旧版本

### 使用前提

```
● 使用由阿里云容器服务 Kubernetes 版本提供的 Kubernetes 集群
```

### 实现原理





#### 发布过程:

1. 发布前检查: 预检查当前 Ingress 是否有且只关联了唯一的 Service 实例,且 Service 实例下有且只有唯一版本的 Deployment。

2. 生成 Canary 版本: 克隆 primary 版本的 Service 以及 Deployment 生成 canary 版本的 Service 和 Deployment , 同时修改 canary 版本 Deployment 的镜像到新版本。

3. 修改 Ingress 流量规则:根据发布配置调整 Ingress 配置,开始执行灰度。

4. 人工验证:通过 cookie 或者 header 对灰度版本进行验证,根据结果选择完成发布或者回滚。

5. 完成灰度:修改 Ingress 配置以及流量规则,下线 Primary 版本的 Service 以及 Deployment 实例。

6. 回滚发布:修改 Ingress 配置以及流量规则,下线 Canary 版本的 Service 以及 Deployment。

#### 快速开始

1. 初始化应用部署

通过以下配置快速初始化应用部署:

apiVersion: apps/v1 kind: Deployment metadata: name: old-nginx labels: run: old-nginx spec: replicas: 2 selector: matchLabels: run: old-nginx template: metadata: labels: run: old-nginx spec: containers: - image: nginx imagePullPolicy: Always name: nginx # 容器名称 ports: - containerPort: 80 protocol: TCP restartPolicy: Always ---apiVersion: v1 kind: Service metadata: name: old-nginx spec: ports: - port: 80 #**服务端口** protocol: TCP targetPort: 80 #**应用端口** selector: run: old-nginx sessionAffinity: None type: NodePort apiVersion: extensions/vlbetal kind: Ingress metadata: name: gray-release spec: rules: - host: www.example.com http: paths: # 老版本服务 - path: / backend: serviceName: old-nginx servicePort: 80

保存以上内容到, gray-release.yaml, 使用 kubectl 或者控制台完成应用初始化:

kubectl apply -f gray-release.yaml

#### 2. 配置流水线

如下所示,配置如下信息:

# 云效2020

# 云效流水线 Flow·部署

C TITE							
< 22/21 20/748 2020-12-31			84188 <b>NRE</b>	脱发说道 发展和现行			1819910217
27	新阶段	新阶段				編編 ①	×
						無群進接 ①	◎ 新建连接
74e36aff9ed/Codeup-Demo		0 H0E5				kube-vela	~
j⊨ master						Kubectl版本 ①	
+添加代码源						v1.16.4	~
						命名空间 ①	
						default	
						Ingress名称 ①	
						gray-release	
						服务第日 ①	
						80	
						19940 (A)	
						80	
						容器名称 ①	
						nginx	
						镜像 ③	14 从数据源选择
						nginx:1.19	
						灰度匹配方式 ⑦	
						cookie	
						灰度匹配名称 ⑦	
						foo	
						灰度匹配值 ①	
						bar	
						友度版本初始化流量权重 ①	
						5%	~
						自用等待时间 ⑦	
						5	

### 配置说明:

配置项	说明
命名空间	当前服务所在在 Kubernetes 集群命名空间, 示例中为 default
Ingress名称	发布的目标 Ingress 实例名称,示例中为 gray-release
服务端口	Ingress 后端 Service 实例对外暴露的端口,示例中为 80
应用端口	镜像对外暴露的端口, 示例中为 80
容器名称	应用发布时需要更新镜像的容器名,示例中为 nginx
镜像	通过前序任务镜像构建产生的镜像,或者是特定的镜像名称
灰度方式	选择使用 header 或者 cookie 方式进行灰度验证,并设置匹配的 Key/Value
灰度初始化流量权重	默认灰度版本上线后的流量权重
启动等待时间	灰度版本发布后等待该时间后再修改 Ingress 配置
下线等待时间	Ingress 配置调整后等待该时长后再移除应用实例

#### 3. 运行流水线

保存并运行流水线,当新版本发布后,流水线将任务将处于灰度发布中,等待人工验证以确定后续操作。

👬 🔁 首页 > 流水	<线 2020-12-31 ▼	*
<b>最近运行</b> 运行历	历史	
#3 🖸 运行中	● 终止	执行命令
运行人	<b>89</b> 砧木	G Kubernetes Nginx Ingress
开始时间 202	20-12-31 11:27:53	运行中
持续时间	1分24秒	
产物	0项	<ul> <li>回滾</li> </ul>
▶ 源(1)		④ 完成
-		

```
灰度中查看Kubernetes资源状态如下所示:
```

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
   annotations:
         kubectl.kubernetes.io/last-applied-configuration: |
                                                                              {"apiVersion":"extensions/v1beta1","kind":"Ingress","metadata":{"annotations":{"ngin
x.ingress.kubernetes.io/service-match":"gray-release-v20201231112858: cookie(\"foo\", /^bar$/)","nginx.ingress.kubernetes.io/service-weight":"gray-re
lease-v20201231112858: 0, old-nginx: 100"),"name":"gray-release","namespace":"default"),"spec":("rules":[("host":"www.example.com","http":{"path*:[{
    "backend":("serviceName":"old-nginx","servicePort":80},"path":"/","pathType":"ImplementationSpecific"},("backend":{"serviceName":"gray-release-v20201
231112858","servicePort":80},"path":"/","pathType":"ImplementationSpecific"}]}})
         nginx.ingress.kubernetes.io/service-match: 'gray-release-v20201231112858: cookie("foo",
                                                                                                                         /^bar$/)'
         nginx.ingress.kubernetes.io/service-weight: 'gray-release-v20201231112858: 0, old-nginx: 100'
    name: gray-release
    namespace: default
spec:
    rules:
     - host: www.example.com
        http:
            paths:
             - backend:
                 serviceName: old-nginx
                   servicePort: 80
             path: /
              pathType: ImplementationSpecific
           - backend:
                   serviceName: gray-release-v20201231112858
                   servicePort: 80
              path: /
              pathType: ImplementationSpecific
```

在灰度中云效自动创建了灰度版本的服务,命名规则未: <ingressname>-<version> 查看灰度版本的 Service 以及 Deployment 详情:

# 云效2020

apiVersion: v1 kind: Service metadata: labels: run: old-nginx version: v20201231112858 name: gray-release-v20201231112858 namespace: default spec: clusterIP: 10.0.6.88 ports: - port: 80 protocol: TCP targetPort: 80 selector: run: old-nginx version: v20201231112858 type: ClusterIP apiVersion: apps/v1 kind: Deployment metadata: generation: 1 labels: run: old-nginx version: v20201231112858 name: gray-release-v20201231112858 namespace: default spec: progressDeadlineSeconds: 600 replicas: 2 revisionHistoryLimit: 10 selector: matchLabels: run: old-nginx version: v20201231112858 strategy: rollingUpdate: maxSurge: 25% maxUnavailable: 25% type: RollingUpdate template: metadata: creationTimestamp: null labels: run: old-nginx version: v20201231112858 spec: containers: - image: nginx:latest imagePullPolicy: Always name: nginx ports: - containerPort: 80 protocol: TCP

#### 用户可以通过在浏览器设置 cookie 的后来访问新版本的应用:

document.cookie="foo=bar"

#### 4.确认发布完成

在人工验证成功后,在卡片上点击"完成"按钮,后流水线将会自动完成ingress的配置调整,以及老版本的应用下线操作:

🔛 🗾 首页 >	, 流水线 2020-12-31 ▼	*	
最近运行	运行历史		
#3 🕑 运行成功		执行命令	
运行人	😜 砧木	Vubernetes Nginx In	gress
开始时间	2020-12-31 11:27:53	发布完成	
持续时间	12分18秒	开放到gray-release-v202	12公18秒
产物	0项	E HAD	12301010
▶ 源(1)			

#### 此时查看线上应用 Ingress 信息,如下所示:

apiVersion: extensions/vlbeta1
kind: Ingress
metadata:
annotations:
kubectl.kubernetes.io/last-applied-configuration:   {"apiVersion":"extensions/v1beta1","kind":"Ingress","metadata":{"annotations":{},"name
":"gray-release", "namespace":"default"), "spec": {"rules": [{"host":"www.example.com", "http": {"paths": [{"backend": {"serviceName": "gray-release-v20201231"
112858","servicePort":80},"path":"/","pathType":"ImplementationSpecific"}]}}]}
name: gray-release
namespace: default
spec:
rules:
- host: www.example.com
http:
paths:
- backend:
serviceName: gray-release-v20201231112858
servicePort: 80
path: /
pathType: ImplementationSpecific

#### 常见发布失败的问题:

- 1. 当 Ingress 关联了多个 Service 实例时发布失败。
- 2. 当 Service 关联了多个 Deployment 实例时发布失败, Service 的 LabelSelector 需要确保与 Deployment 的 labels 保持匹配。
- 3. 线上版本镜像和当前发布镜像未变化时,发布失败。
- 4. 而发布配置中的容器名称无法匹配到容器定义时发布失败。

### 6.3.8. Helm Release 部署

Flow 提供了对于 Helm 的发布支持,用户可以将代码库中的 Chart 部署到 Kubernetes 集群。

# 🗘 注意

```
: Helm Release 发布只支持使用证书导入的 Kubernet es 集群
```

#### 添加任务

如下所在,在流水线中添加任务并选择 Helm Release 部署任务

源(1)			选择任务组		×	
/flow-example/spring-b % master	企业任务组 代码扫描	Q。请输入任务名称				
・ 変量 (0)	测试构建	<b>主机部署</b> 主机部署	主机部署(无构建包) 使用脚本发布到主机	Docker部署 主机Docker部署		
暂未设置,点击 添加	构建 镜像构建 工具	Kubernetes 发布 用于发布 Kubernetes Y	Kubernetes 镜像升级 升级 Kubernetes 下的	<b>Kubernetes 分批发布</b> Kubernetes 分批发布		
	<b>部署</b> 1 代码	Helm Release 部署 使用代码库中的 Helm	<b>EDAS ECS应用发布</b> EDAS ECS应用发布	EDAS Kubernetes EDAS Kubernetes 应用		
	执行命令 空模板	Serverless(SAE) 应 Serverless(SAE) 应用	<b>(一)</b> 函数计算 应用发布 更新函数计算 (FC) 中			
		代码			_	

### 配置发布任务

首次使用 Kubernetes 相关部署的用户需要首先使用自定义建立集群连接:建立 Kubernetes 服务连接

完成后,选择集群连接并设置 Helm chart 在代码库中的相对路径,对于需要修改发布参数的场景,用户可以在任务配置中添加变量。在发布时会默认覆盖 values.yaml 中定义 的内容

▼ 源(1) +	段1	新阶段	编辑	
/flow-example/spring-boot			任务步骤	
le master	○ 阿里云镜像构建 Helm Release部署	③ 新的任务	分添加步骤	
▼ 变量 (0)			- Helm Release	
			步骤名称	
暂未设置,点击 添加			Helm Release	
			集群连接 ⑦	● 新建连持
			k8s-management-集成测试误删	~
			命名空间 ⑦	
			default	
			本地Chart路径 ⑦	
			helm-config/app	
			Release名称 ⑦	
			test-release	
			变量 ⑦	
			十 添加参数	6

发布或者升级 Release

点击保存并运行流水线,如下所示:

<b>最近运行</b> 运行历史			● 运行 🛛 / 编辑 🛛 风 成员 🛛 三
#3 🔮 运行成功	阶段1		
运行人 🥙 13	5 🕑 Helm Release部署		
开始时间 2020-03-13 17:38:0	) 运行成功		
持续时间 571	0 日志	57秒	
▶ 源(1)			

#### 点击日志查看发布详情:

<b>最近运行</b> 运行历.		Helm Release部署 × 编	■ 久成员 三菜
	构建	Helm Release	
5 🔮 1811 AX-40		[09:38:26] 开始加载Kubernetes集群配置	
行人	⊘ 清理工作区(6s)	[09:38:26] 2020/03/13 09:38:26 [INFO] save date to /root/.kube/config [09:38:26]	
始时间 2020-	⊘ 克隆代码(6s)	[09:38:26] 加载Kubernetes集群配置成功	
A14 JI-J	Helm Release(27s)	[09:38:26]	
续时间		[09:38:26] Helm Client&Server Version: [09:38:26] Client: &version.Version(SemVer: "v2.13.1", GitCommit: "618447cbf203d147601b4b9bd7f8c37a5d [09:38:27] Server: &version.Version(SemVer: "v2.14.0", GitCommit: "05811b84a3f93603dd6c2fcfa57944dfa7	
AGK(1)		(09:38:27) [09:38:27] Install/Upgrade Chart Release [09:38:27] helm upgradeinstallnamespace=default test-release . [09:38:30] Release "test-release" does not exist. Installing it now.	
		(09:38:46) NAME: test-release (09:38:49) LAST DEFLOYED: Fri Mar 13 09:38:30 2020 (09:38:49) NAMESRACE: default (09:38:49) STATUS: DEFLOYED	
		[09:38:49] [09:38:49] meSOURCES: [09:38:49] meX vi/Deployment [09:38:49] NAME READY UP-TO-DATE AVAILABLE AGE	
		[0y:39:49]         0         48           [0y:39:49]         0         10           [0y:39:49]         0         10           [0y:39:49]         0         10	
		(09:38:49) test-release-spring-boot-sample-6b56fc46c7-	

# 6.3.9. ASM 蓝绿发布

阿里云服务网格(简称 ASM)企业版实例中,兼容 lstio,支持多个 Kubernetes 集群统一流量管理,为容器和虚拟机应用服务提供一致性的通信控制。基于服务网格最新 的 "数据面 KubeAPI 访问"能力,能够通过数据面集群的 kubeconfig 来管理 lstio 资源。和云效 Flow 集成后,可以享受 DevOps、Git Ops带 来的应用发布的便利性,还能够利 用 lstio 细粒度的流量管控能力来实现蓝绿发布,保障上线的稳定性。

### 前提条件

- 创建 lst io 版本为 1.12.4.50 或更高版本的阿里云服务网格 (ASM) 实例。
- 创建容器服务 Kubernetes (ACK) 集群。
- 将 ACK 集群添加到 ASM 实例。

#### 实践1:流水线部署应用

目标:以 Istio book-info 为例,我们将首页的标题文字进行更换。代码提交后,执行阿里云云效流水线,生成镜像,自动将应用的流量管理(VirtualService、Gateway、 DestinationRule)相关的资源创建或更新到阿里云服务网格 ASM 集群中,将应用程序编排(Deployment、Service)创建或更新到阿里云容器服务平台 ACK 集群中。 创建流水线如下。流水线源配置为 Git Hub 仓库,测试地址为https://github.com/AliyunContainerService/asm-labs,您也可以 fork 出来,然后进行个性化修改。分支选择 flow-demo。

#### 执行的步骤如下:



#### 1. 流水线流程配置

將镜像仓库地址为替换为您的地址空间: registry.cn-hangzhou.aliyuncs.com/ *{your_name_space}*/ productpage:\${DATETIME}。这里我们使用 DATETIME 作为版本号。 Dockerfile 路径为 productpage/Dockerfile。在 Kubernetes 发布的 Kubectl 发布中,设置变量 ProductpageImage 为前面配置的镜像仓库地址,在代码仓库的 YAML文件 中,productpage 的镜像被设置为了 \${ProductpageImage} 与其对应。

鼻辑 ──	×	2通4推 Ш	
任务名称	Python 构建	▼ Kubectl 发布	
代码扫描	构建集群 ②	步骤名称	
构建集群 ⑦	云效北京构建集群	〜 Kubectl 发布	
云效北京构建集群	→ 下载流水线源 ⑦	集群连接 ⑦	◎ 新建连
下我这业经演	下载全部流水线源	∽ flow-cs	~
下载流小线源 ⑦	任务步骤	Kitard Et a	
	● 添加步骤	v1.16.4	~
任务步骤	÷ 镜像构建并推送至自定义镜像仓库	•	
▶ 添加步骤	步骤名称	命名空间 ⑦	
▼ Python 代码扫描	<ul> <li>镜像构建并推送至自定义镜像仓库</li> </ul>	default	
步骤名称	德治会在批批 ①	YAML 路径 ⑦	
Python 代码扫描	registry.cn-hangzhou.aliyuncs.cc	om * ~ oductpag	
		使用Replace模式 ⑦	
请选择Python版本	选择凭据类型		
3.7	● 服务连接 ○ 用户名/密码		
□ 开启增量扫描模式 ②	选择服务连接	添加服务连接 跳过模板校 镜像仓库地址。	registry.cn-
	qifang_ly_daily@test.aliyunid.com	m的私有镜像仓库服务i ~ hangzhou.aliyu //productpag	uncs.com/a le:\${DATETIME}
排除子目录 ⑦	Dockerfile路径 ⑦	Productpage = 镜像仓库	世址.registry.cn-ha >
test/	productpage/Dockerfile	× 51	

2. 添加数据面集群并注入 Sidecar 代理

在阿里云服务网格 ASM 控制台,进入 ASM 实例后,在集群与工作负载管理中,添加希望流水线配置的 ACK 容器服务实例。并在全局命名空间中,为选定的命名空间(如 default)启用 Sidecar 自动注入。

4	服务网格 ASM /	flow-demo / Kubernete	15 W 27									
477.00 A	itā tu											1
11×01	□ 名称		版本	集群类型	地域	专有网络	安全组	城名	访问日志	状态	可规测性	
全局命名空间	flow-	demo	1.22.3-aliyun.1	Kubernetes 托 管版	INER INFAULT	vpc-bp194k	al de la factoria	cluster.local	如何开启?	<ul> <li>适行中</li> </ul>	宣看报表	
升级管理	□ 移出											
网格状态												
网格诊断												
授权信息												
件与工作负载管理 ^												
Kubernetes集群												
虚拟机												
服务条目												
$\leftarrow$	服务网格 AS	M / flow-demo / 全》	前命名空间									
格实例へ	新建	从Kubernetes集群同	步自动注入 『	同步自动注入至Kuberr	hetes集群 🤡	同步成功						
基本信息	名称	状态	标签						创建时间		自动注入	
全局命名空间	default	Active	istio-inject	ion:enabled kube	rnetes.io/metadata	a.name:default			2022年6月20日 15	:07:22	关闭 Sidecar 自动注入	•
升级管理	istio-syst	em Active	asm-syste	m:true istio-inject	ion:disabled ku	bernetes.io/metadata.name:istio-	system provide	r:asm	2022年6月20日 15	:07:35		
网格状态												
网格诊断												
授权信息												

#### 3. 创建ASM网关

为了方便的访问我们流水线部署出的应用程序,需要创建 ASM 网关来作为访问入口,端口设置为 80。

	(-) 阿里云	☆ 工作台	2 Q 搜索	
基	本信息		服务网格 ASM / a SM网关	
全	局命名空间		← 创建	
升	级管理			
网	格状态		基本信息	
网	格诊断		* 名称	
授	权信息		ingressgateway	
集群	可工作负载管理	~	* 命名空间	
Kı	lbernetes集群			
虚	拟机		* 部署集群	
BR	冬冬日		о сиеванит ат сиеванит 2377b) Х	
	作会共归		可以选择一个或多个集群,多个集群可以保证集群维度的高可用	
	作贝轼组	<	* 网关类型	
	作贝载杀日		南北向-入口	
Side	car管理(数据面)	~	负载均衡类型	
服务管	管理			
ASM	网关 ←		* 新建负载均衡 🗢 使用已有负载均衡 💿	
流量	管理	^	简约型l (slb.s1.small) 最大连接数5,000	
虚	拟服务		<ul> <li>         ・<ul> <li>             ・</li>             ・</ul></li>             ・<td></td></ul>	
目	标规则			
网	关规则			
Si	decar资源			
流	量标签			
			HTTPS V 443	

4. 执行流水线		
测试	构建	新阶段
⊘ 代码扫描	📀 Python 构建 🌗	✓ Kubernetes 发布
4         0         4         0           总数         阻塞         严重         一般	运行成功	运行成功
⑦ 扫描报告 □ 日志 22秒	日志 15分1秒	日志 28秒

### 5. 查看部署效果

在 ACK 集群中,确认镜像能够成功拉去并且启动正常,如拉取个人私有镜像失败可参考使用免密组件拉取容器镜像。

< flow-demo -	所有集群 / 集群:flow-demo / 命名	空间: default 🛫 🗲 / 无状态	
集群信息	无状态 Deployment		
▼ 节点管理	请输入搜索内容 Q		
节点池		•	
节点	□ 名称	标签 🔻	容器组数量
命名空间与配额	details-v1	app:details version:v1	1/1
▼ 工作负载			
无状态	productpage-v1	app:productpage version:v1	1/1
有状态			
守护进程集 任务	ratings-v1	(app:ratings) (version:v1)	1/1
定时任务	reviews-v1	app:reviews version:v1	1/1
容器组			
自定义资源 ▶ 网络	reviews-v2	(app:reviews) (version:v2)	1/1
▶ 配置管理			
▶ 存储	reviews-v3	(app:reviews) (version:v3)	1/1
展务网格 ASM           网格实例         0	/ flow-demo / ASM网关 使用YAML创键 C 如何通过ASM入口网关实现HTTP请求	网格内gRPC服务	
基本信息 名称	命名空间	状态 Kubernetes服务	端口映射
全局命名空间 ingressgate	way istio-system	● 创建成功 101 123 ( c1c	HTTP   80 : 80 HTTPS   443 : 443
升级管理 网格状态 网格诊断 授权信息			
集群与工作负载管理 >>			
Sidecar管理(数据面) 服务管理			
ASM网关			
复制前面创建的 ASM 网关的地址,在浏览	5器中输入 http://{ASM网关地址}/product	page,可以看到我们修改的标题已生效。	
ASM Flow D	Demo	Sign	in
	The Cor	nedy of Errors	
Summary: Wikipedia Summary: The Cornedy of Errors is one of W	Villiam Shakespeare's early plays. It is his shortest and one of his mos	t farcical comedies, with a major part of the humour coming from slapstick and mistaken identiti	ly, in addition to puns and word play.
Type: paperback Pages: 200		An extremely entertaining play by Shakespeare. The slapstick hume – Reviewer1	our is refreshing!
Publisher: PublisherA Language: English 158N-10: 123457800 158N-13: 123-1234567800		Absolutely fun and entertaining. The play lacks thematic depth whe Shakespeare. - Reviewer2	in compared to other plays by
当前显示界面固定为 reviews-v1 版本。			

### 实践2:蓝绿发布

阿里云服务网格 ASM 和云效集成,提供了蓝绿发布的能力。从而让用户可以更安全的基于 lstio 的流量管理策略实现应用的安全发布。

目标:在完成"实践1:流水线部署应用"的前提下,通过蓝绿发布,将 reviews 版本从原来的 v1 更新为 v2 版本。

#### 1. 创建新的流水线

如实践 1 配置流水线源,创建阶段 ASM 蓝绿发布。这里我们设置镜像为 reviews-v2 版本的镜像:docker.io/istio/examples-bookinfo-reviews-v2:1.16.4。灰度初始化流量为 0%,并且只有待特定 cookie 标识的用户能够访问到蓝绿版本。

	编辑 🛈
	ロカツ弾
● 新建3	Kubernetes集群连接 ⑦
	f mail
◎ 新建3	ASM集群连接
	note ism
	Kubectl版本 ②
	v1 16 4
	1110.4
	命名空间 ⑦
	default
	服务名称 ②
	reviews
	容器名称(?)
	reviews
14 直接報	镜像
nfo-reviews-v2:1.16.4	docker.io/istio/examples-bookinfo-r
	灰度流量Cookie标 ②
	^(.*?;)?(gray=true)(;.*)?\$

2. 执行

在阿里云云效 Flow 中,蓝绿发布前需要明确指明使用哪个版本的工作负载,将其他的版本的工作负载全部清空,确保流量只流入稳定版(reviews-v1)。

#### ASM 蓝绿发布

#### 解决冲突

当前服务关联了多个应用工作负载,请选择一个应用实例接管所有流量

#### 🕕 提示

执行蓝绿发布前,需要要求当前发布的服务有且只有一个唯一的应用工作负载实例,当存在多个版本的应用工作负载实例时云效将认为当前服务处于不稳定状态

命名空间	名称	容器	创建时间	操作
default	reviews-v1	𝔅 reviews	2022-06-21T12:20:30Z	接管流量
default	reviews-v2	𝔅 reviews	2022-06-21T12:20:30Z	接管流量
default	reviews-v3	⊗ reviews	2022-06-21T12:20:30Z	接管流量

ASM 蓝绿发布					×
解决冲突 当前服务关联了多个应用工作负载,	<ul> <li>请确认</li> <li>请选择一个应用实</li> </ul>	操 <b>作</b> 作将会修改istio流量规则,使用revie	ews-v1接管所有流量,并移除其它工作负载实	× 例	
提示 执行蓝绿发布前,需要要求当	前发布的服务有且		取消 确定	建立	
命名空间名和	称	容器	创建时间		操作
default rev	views-v1	𝔅 reviews	2022-06-21T12:20:30Z		接管流量
default rev	views-v2	𝗇 reviews	2022-06-21T12:20:30Z		接管流量
default rev	views-v3	𝔅 reviews	2022-06-21T12:20:30Z		接管流量

ASM 蓝绿发布

发起部署	ASM 蓝绿发布
	[10:38:3/] time: "2022-06-22102:38:3/2"
♥ 申请运行坏境(0s)	[10:38:37] name: reviews
✓ 清理工作区(0s)	[10:38:37] namespace: default
⊘ 克隆代码(4s)	[10:38:37] resourceVersion: "6206739"
▲ 法水优保方(0a)	[10:38:37] uid: 42d026a0-53ab-4db1-9923-1356c404ce73
♥ ////////////////////////////////////	[10:38:37] spec:
✓ ASM 蓝绿发布(14s)	[10:38:37] hosts:
❷ 缓存上传(1s)	[10:38:37] – reviews
(中容 627) 五	[10:38:37] http:
冲关阱次	[10:38:37] - route:
🕏 申请运行环境(0s)	[10:38:37] - destination:
✓ 清理工作区(1s)	[10:38:37] host: reviews
	[10:38:37] subset: v1
☑ 兒隆代码(5S)	[10:38:37] [INFO] 冲突解决: 移除应用实例 deployment/reviews-v2
⊘ 流水线缓存(1s)	<pre>[10:38:37] [INF0] Remove Deployments/reviews-v2</pre>
<ul> <li>ASM 蓝绿发布</li> </ul>	<pre>[10:38:37] deployment.apps "reviews-v2" deleted</pre>
● 經左上住	[10:38:38] [INFO] 冲突解决: 移除应用实例 deployment/reviews–v3
☑ 级任工15	<pre>[10:38:38] [INFO] Remove Deployments/reviews-v3</pre>
	<pre>[10:38:38] deployment.apps "reviews-v3" deleted</pre>
	[10:38:38]
	[10:38:38] [INFO] Kubernetes集群基本信息

此时,在阿里云服务网格ASM控制台查看虚拟服务 VirtualService 状态。此时状态如下:

#### O ASM 蓝绿发布

运行中

版本v20220622103841灰度验证中

⑧ 继续发布 5 回退发布 ☐ 日志 2分24秒

# 云效2020

apiVersion: networking.istio.io/v1beta1 kind: VirtualService metadata: name: reviews namespace: default spec: hosts: - reviews http: - match: - headers: cookie: regex: ^(.*?;)?(gray=true)(;.*)?\$ route: - destination: host: reviews subset: v20220622103841 - route: - destination: host: reviews subset: v1 weight: 100 - destination: host: reviews subset: v20220622103841

### 目标规则如下所示:

apiVersion: networking.istio.io/vlbetal
kind: DestinationRule
metadata:
 name: reviews
 namespace: default
spec:
 host: reviews
 subsets:
 - labels:
 version: v1
 name: v1
 - labels:
 version: v20220622103841
 name: v20220622103841

#### 3. 测试验证

因为 reviews 服务是由后端服务 product page 来调用的,不方便携带特定 token 来进行灰度测试。

修改 header 中 cookie 设置,改为使用 end-user 的登陆信息,设定只要登陆用户名称为 asm,则显示灰度版本。

虚拟	服务	VirtualServic	编辑	
创建	使用YAI	ML创建		
			0 7	{"apiversion": "networking.istic.io/vialpna3", "kind": "virtualse
	名称 ☑		2 2	generation: 0
			q	name: reviews
	bookinfo		10	namespace: default
			11	resourceVersion: '3403073'
	reviews		12	uid: b8b5cd0b-9b6e-4853-a5ae-2642d4ff5c84
			13 $\sim$	spec:
			14 $\sim$	hosts:
			15	- reviews
			16 $\sim$	http:
			- 17 $  imes$	- match:
			18 $\sim$	- headers:
			19 $\scriptstyle{\sim}$	end-user:
			20	exact: asm
			21 V	route:
			22 ~	- destination:
			23	host: reviews
			24	Subset: V20220620000000000000000000000000000000
			25 ~	- route:
			20 ~	- destination:
			27	
			20	veight: 100
			30 ~	- destination:
			31	host: reviews
			32	subset: v2022062( 1
			33	

复制前面创建的 ASM 网关的地址,在浏览器中输入 http://{ASM网关地址}/productpage,点击右上角进行登录,输入用户名称为 asm,密码任意输入。如果是未登录状 态,则显示 v1 版本的 reviews 信息。

ASM Flow Demo	asm (sign out)
The Comedy Summary: Wikipedia Summary: The Comedy of Errors is one of William Shakespeare's early plays. It is his shortest and one of his most farcice Book Details Type:	of Errors al comedies, with a major part of the humour coming from stapstick and mistaken identity, in addition to puns and word play. Book Reviews As automatic additionation by the Shalkanapage. The allocation is in additionation by the standard statement of the standard statement of the stat
paperback Pages: 200 Publisher: PublisherA English ISBN-10: 1234567890 ISBN-13: 123-1234567890	- Reviewer1 - Reviewer2 * * * * *
4. 继续发布	
在通过了灰度版本的测试之后,将全部的流量转为新版本(reviews-v2)。点击继续发布,另	已成发布。
O ASM 蓝绿发布	✓ ASM 蓝绿发布

<b>运行中</b> 版本v20220622103841灰	度验证中	<b>发布完成</b> 升级到v2022062210384
凹 日志	3分9秒	四 日志

发布完成后,未登录时也显示 reviews-v2 的版本。

3分27秒



#### 发布完成后, reviews VirtualService 的状态。

网格实例	^	应切尔尔 VirtualServic	143 + 19
基本信息		创建使用YAML创建	3冊 4耳
全局命名空间		□ 名称 ☑	1 apiVersion: networking.istio.io/v1beta1 2 kind: VirtualService
升级管理 网格状态		bookinfo	<pre>3 metadata: 4 annotations: 5 kubectl.kubernetes.io/last-applied-configuration: &gt; 6 // #aniVersion#"#actuarking istic is(v1alpha?# "kind")</pre>
网格诊断		- 删除 (0)	<pre>7 creationTimestamp: '2022-06-20T08:19:10Z' 8 generation: 10 9 name: reviews</pre>
授权信息 集群与工作负载管理	~		10 namespace: default 11 resourceVersion: '3410556' 12 uid: b8b5cd0b-9b6e-4853-a5ae-2642d4ff5c84 13 spec:
Sidecar管理(数据面)	~		14 hosts: 15 - reviews 16 http:
服务管理 ASM网关		<	17 - route: 18 - destination: 19 host: reviews 20
流量管理	^		21
<b>虚拟服务</b> 目标规则			

#### 总结

在 Kubernetes 中 Service 与多个版本的 Deployment 实例通过 Label 进行关联。蓝绿发布开始,阿里云 Flow 会基于当前的 Deployment 实例创建一个新版本的应用 Deployment 实例。等待蓝环境创建完成后通过更新 阿里云服务网格 ASM 的 VirtualService 和 DestinationRule 资源对象,进一步控制流量与特定版本 Deployment 实例之间 的路由策略。

根据发布设置,可以修改 VirtualService 确定蓝/绿环境的流量权重。用户也可以设置灰度标,确定哪些流量会直接进入到蓝环境中。在人工验证完成后,完成发布将所有流量 切流到蓝环境,并且将原有的绿环境实例移除。

作为业内首个全托管 lstio 兼容的阿里云服务网格产品 ASM,一开始从架构上就保持了与社区、业界趋势的一致性,控制平面的组件托管在阿里云侧,与数据面侧的用户集群 独立。ASM 产品是基于社区 lstio 定制实现的,在托管的控制面侧提供了用于支撑精细化的流量管理和安全管理的组件能力。通过托管模式,解耦了 lstio 组件与所管理的 K8S 集群的生命周期管理,使得架构更加灵活,提升了系统的可伸缩性。从 2022 年 4 月 1 日起,阿里云服务网格 ASM 正式推出商业化版本,提供了更丰富的能力、更大的规模支 持及更完善的技术保障,更好地满足客户的不同需求场景,详情可见产品介绍: https://www.aliyun.com/product/servicemesh。

# 6.3.10. Kubernetes发布常见问题

#### 发布使用私有镜像

如果发布到 Kubernetes 集群的镜像是私有的用户需要在 Kubernetes 集群中进行配置已支持集群能够拉取私有镜像。阿里云Kubernetes用户请参考:

阿里云帮助中心-阿里云,领先的云计算服务提供商

其它用户请参考Kubernetes官方文档进行配置

https://kubernetes.io/docs/tasks/configure-pod-container/pull-image-private-registry/

#### 使用证书创建集群时跳过TLS校验

当使用证书中声明了insecure-skip-tls-verify时,如下所示:

insecure-skip-tls-verify: true

#### 需要用户在部署配置中手动勾选"跳过TLS校验",否则发布任务失败并提示:

error: specifying a root certificates file with the insecure flag is not allowed

#### Kubectl YAML发布状态

云效中提供了使用下发Kubernetes YAML文件到集群中的能力支撑,云效已是否成功下发YAML作为任务执行成功失败条件,但是不校验具体的资源状态,比如通过 Deployment创建出的Pod状态等。

# 6.3.11. Kubernetes发布和运维职责分离

在实际的发布中由于一些安全和管理因素分离开发人员和运维人员的职责。本文将介绍在云效 Kubernetes 发布中如何分离开发和运维人员。

#### 基于代码库分离发布和运维权限

当用户采用 Kubernetes YAML方式部署应用时,应用的所有配置实际由 YAML进行描述其中包含了应用的所有运行时配置信息。 在这种模式下为了分离发布和运维的权限, 需要将应用代码库和 Kubernetes YAML分离到不同的代码仓库中并通过对代码库的权限控制确保只有运维人员能够修改 YAML文件的内容:

● 应用代码库:该代码库中主要包含应用的源码以及用于构建镜像的 Dockerfile 文件。并且主要由开发人员负责维护

● 部署配置库:该代码库中主要包含 Kubernetes 发布所需的所有 YAML 源文件(或模板文件)。并且主要由运维人员负责维护

在一条流水线中分别添加应用代码库和部署配置库作为流水线的输入源。在前序的任务中基于应用代码库完成应用代码的编译构建等工作,并生成发布所需的 Docker 镜像, 在部署任务中管理部署配置库中的 YAML,并将前序任务生成的镜像作为发布的入参完成最终的发布动作。

该模式通过代码库让应用开发人员专注于应用的开发与发布,而无需关心应用的部署配置。当需要进行配置更新时,由运维人员对 YAML进行更新即可。

# 6.3.12. 部署历史 和 回滚

用户可在流水线执行记录 -> 部署历史中查看该流水线所有执行的部署历史记录,并可选择其中一条历史记录进行回滚操作。

最近运行 运行历!	史部署历史							③ 运行	③回滚 🦉 编辑 🔍 成员 🗉
Kubernetes 发布	×								
部署记录	状态	代码版本		部署包	开始部署时间	部署任务	执行人	说明	操作
#39	0	master	2>	-	07-14 15:26	Kubernetes 发布	rdc-rdc	回滾至部署记录#37	日志 田源
#38	٥	master	2>		07-14 15:20	Kubernetes 发布	rdc-rdc	回滾至部署记录#36	日志田泉
#37	٥	master	2>		07-13 20:16	Kubernetes 发布	rdc-rdc		日志田泉
#36	0	master	2>	-	07-13 19:58	Kubernetes 发布	rdc-rdc	-	日志 回滚
#3	0	master	2>		06-23 11:51	Kubernetes 发布	rdc-rdc		日志田液
#2	0	master	2>		06-23 11:49	Kubernetes 发布	rdc-rdc	-	日志 回流
#1	0	master	2.>	÷	06-23 11:37	Kubernetes 发布	rdc-rdc		日志 回滚
									6

# 6.4. EDAS 应用发布

Flow 支持将构建出来的制品部署到 <mark>EDAS(企业级分布式应用服务</mark>)。企业级分布式应用服务 EDAS(Enterprise Distributed Application Service)是一个应用托管和微服务管 理的 PaaS 平台,提供应用开发、部署、监控、运维等全栈式解决方案,同时支持 Spring Cloud、Apache Dubbo(以下简称 Dubbo)等微服务运行环境,助力您的各类应用 轻松上云。EDAS支持多种集群类型,本文讲述如何对 ECS 集群的 EDAS 应用进行部署。

### EDAS 应用

你需要有一个 ECS 集群类型的 EDAS 应用。如下图:

应用信息			
ID	950abdc0-ab72-47ba-bc01-6f7d19218a68	命名空间	cn-hangzhou:FlowDev
集群类型	ECS集群	集群名称	FlowDevCluster
应用运行环境	标准Java应用运行环境	状态	运行中 2 / 共 2
部署包类型	FATJAR	部署包	默认分组: application.jar
负责人	rdc-rdc 🖌	应用描述	使用标准Java环境的Spring boot应用示例 🖌
应用创建时间	2020-01-15 10:52:46	最后变更时间	2020-03-17 21:41:38
应用设置 基本	信息 挂载脚本		
应用端口	8080	Tomcat Context	ROOT 编辑
负载均衡 (私网)	添加	负载均衡(公网)	添加
健康检查	靖口健康检查被禁用 修改 健康检查URL被禁用 修改	殘參 MVL	N/A 编辑

该应用包含两台 ECS。

# 部署目标物

创建一条空流水线,选择示范例代码源:

源	构建	新阶段	添加流水线源				×
			代码源	选择代码源			
I YANYII CAANAY	Java 19/#_LTV		Jenkins Flow流水线	示例代码源 阿	<b>回</b> 可里云Code	<b>巴</b> 自建Gitlab	<b>6</b> 码云
				<b>区</b> 云效Codeup	<b>③</b> 用Git	Github	\$
				示例代码库由 配置和执行	由飞流提供,用	于新用户快速尝试	【流水线的
				代码类型 O JAVA O G O PHP	Go 🔿 Node	aJS O Python	
				代码仓库 ⑦ https://code.ally			
				默认分支 ⑦			

对预置的第一个任务进行编辑:

> 這水线编排 流水线 2020-03-17				保存保存并运行
☞ 源(1) +	阶段1	新阶段		编辑
$\not\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!\!$	日 · · · · · · · · · · · · · · · · · · ·	● 新約任务	(1)	任务名称 构雄并上传
▼ 変量 (0)			Ŭ	任务步骤
暂未设置,点击 添加			2	▶ )Java 构建
			(3)	▶ 构建物上传(EDAS/SAE使用)
				11.79 ABI+
				任务输出 ④ 制品名称.default
				<b>删除任务</b> 该任务将被彻底删除、注意这是不可逆操作、任务下所有数据将会删除
				HI 10

#### 注意,构建物上传的步骤需要选择标注有" EDAS 使用"步骤。

#### 添加 EDAS 部署任务

点击"新的任务"添加部署到 EDAS 应用的任务:

< 返回 流水线 2020-07-30		基本信息 流程配置 触发设置 变量和缓存		仅保存保存并运行
源	构建	新阶段	选择任务组	>
十添加代码源		③ 新的任务	<b>代码扫描</b> Q。请输入任务名称	
			測试 測试构建 Docker部署 主机Docker部書	Kubernetes 发布 用于发布 Kuberne
			构建 镜像构建 获 Kubernetes 镜	Kubernetes 蓋 基于Istio进行应用
			上具 部署 代码	Helm Release 使用代码库中的 H
			执行命令 空模板 DAS ECS应用 EDAS ECS应用	EDAS Kuberne EDAS Kubernetes
			Serverless(SA Serverless(SAE)	Cつ 资源编排服务(R 资源编排服务(ROS
			<b>()</b> 函数计算 应用 更新函数计算 (FC	
			代码	

#### 配置 EDAS 部署任务

接下来配置部署任务。

首先点击新建服务授权,根据跳转完成阿里云 RAM 授权,授权信息会自动回填到任务配置中,然后依次选择到前面准备好的 EDAS 应用,并对发布策略进行相应配置:

Important       Important	源(1)	+	阶段1	新阶段	新阶段		编辑
C 2	/flow-example/spring- i~ master	-boot			◎ 新的任务		任务名称 EDAS ECS 应用发布
	• 安量 (0) 智未说置,点击 激加	8 +				(1)→	任务参加 辺F提校
							田川 spring-boot-sample 牧行名
1919 - ジェカズ - 国の限い							#)加 http://default 夏希批次 1歳 分批等待时间
							1分钟 分批方式 自动确认

这里选择的策略是分两批发布,分批方式为手动确认。该 EDAS 应用有两台 ECS,因此实际的发布过程为先发布一台,等待手动确认后,再进行第二台的发布。

### 运行流水线

配置好之后,触发流水线。

构建完成之后,进入部署节点。由于配置了分两批,手动确认,所以在部署完第一台之后,会暂停,点击"更多"查看可进行的操作:



- 继续第二台机器的发布
- 跳转到 EDAS 控制台查看更多信息

点击继续"恢复变更"之后,可以看到飞流继续完成了本次部署。

最近运行 运行历史			⊙ 运行 ℓ 編
<ul> <li>运行成功</li> </ul>	龄段1	8FB) 19	
EL G #			
除时间 2020-03-17 22:21:2	● 何用升上119	♥ EDAS ECS Ⅲ用我们	
#时间 6分4	default	运行成功	
2011 U. 1	[] 日志 1分9秒		
源(1)			

# 6.5. EDAS Kubernetes 应用发布

Flow 支持将构建出来的制品部署到 EDAS(企业级分布式应用服务)。企业级分布式应用服务 EDAS(Enterprise Distributed Application Service)是一个应用托管和微服务管 理的 PaaS 平台,提供应用开发、部署、监控、运维等全栈式解决方案,同时支持 Spring Cloud、Apache Dubbo(以下简称 Dubbo )等微服务运行环境,助力您的各类应用 轻松上云。EDAS支持多种集群类型,本文讲述如何对 Kubernetes 集群的 EDAS 应用进行部署。

### EDAS 应用

你需要有一个 Kubernetes 集群类型的 EDAS 应用。如下图:

<		📾 edas-k8s-	-test		部署应用 应用扩	缩 删除应
基本信息						
变更记录		基本信息实例	间部署信息			
应用事件		应用信息				
日志管理	$\sim$	ID	666666666666666666666666666666666666666	命名空间	cn-beijing	
应用监控	$\sim$	集群类型	容器服务K8s集群	应用所在集群	k8s-management	
間连路级	~	应用运行环境	标准Java应用运行环境	运行实例数	当前2个实例/目标2个实例	02-19-10-44-0
PLANEP+ NA		应用创建时间	2020-03-18 19:28:05	最后变更时间	2020-03-18 19:45:11	-03-10-19-44-0
通知报警	~	负责人	rdc-rdc	负责人邮箱	test@mail.com	
服务列表		K8S命名空间	default			
K8s Service 管环	89					
		应用设置				
		负载均衡 (私网)	添加	負载均衡 (公网)	1.1.1.1:8081 TCP 编辑   删除	
		规格	CPU: 1 核 内存: 1000MB 变更规格			

该应用包含两个 pod。

### 部署目标物

创建一条空流水线,选择示范例代码源:
#### 云效2020

< 返回		基本信息 流程配置 触发设置 变量和缓存			仅保存	保存并运行
源	构建	新阶段	添加流水线源			×
十添加代码源	0 及 Java 构建上传	◎ 新的任务	代码源选择代码》	Ł		
			Jenkins Flow流水线	回源 阿里云Code	O 自建Gitlab	6 码云
			C 云效Coc	eup 通用Git	Github	\$
			• 代码类	示例代码库由飞流提供, 尼置和执行 型	用于新用户快速尝	试流水线的
			44 O PP 代码仓 https	VA Go Nor IP 车 ⑦ //code.aliyun.com/flow	deJS O Pytho	-boot.git
			默认分	支 ②		

对预置第一个任务进行编辑:

< 返回 流水线 2020-07-30		基本信息 流程配置 触发	发设置 变量和缓存			<b>仅保存</b> 保存并运行
28	构建	新阶段			编辑 ①	
flow-example/spring-boot	•         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •         •	◎ 新的任务			任务步骤	
🌾 master					添加步骤	
十添加代码源					<ul> <li>镜像构建并推送至阿里云镜像仓库</li> </ul>	e
					步骤名称	
					DURINE/TIEZEPTEADUR CH	
					选择授权	新建服务授权
					rdc-rdc的docker_register_aliyun服务	授权 ~
					区域	
					华东1 (杭州)	~
					仓库 ②	11 直接输入
				(2)→	registry.cn-hangzhou.aliyuncs.com/a	aone-build/luoying ~
				$\smile$	标签(?)	
				$(3) \rightarrow$	\${DATETIME}	
				$\mathbf{O}$		
					Dockerfile路径 ⑦	
					bookenne	

#### 添加 EDAS 部署任务

点击"新的任务"添加部署到 EDAS 应用的任务:

〈 返回		基本信息 流程配置 触发设置 变量和缓存		仅保存	保存并运行
源	构建	新阶段	选择任务组		×
Flow-example/spring-boot	● 孕 阿里云镜像构建	③ 新的任务	代码扫描 Q。请输入任务	名称	
⊳ master			测试 <b>王机</b> 测试构建		E机部署(无构 使用脚本发布到主机
十添加代码源			构建 镜像构建	<b>ker部署</b> Kar部署 Kar部署 Kar部署 Kar部署 Kar部署 Kar部署 Kar部署 Karana Kar	<b>(ubernetes 发布</b> 目于发布 Kuberne
			工具 部署 米ub 升级	Kubernetes	<b>(ubernetes 蓋</b> 『于Istio进行应用
			执行命令 空模板	ernetes 分 rnetes 分批	<b>leim Release</b> f用代码库中的 H
			EDA EDAS	S ECS应用 S ECS应用发布 E	DAS Kuberne DAS Kubernetes
			Servi	erless(SA erless(SAE)	<b>资源编排服务(R</b> 资源编排服务(ROS
			<b>(-) 函数</b> 更新i	<b>计算 应用</b> 函数计算 (FC	

#### 配置 EDAS 部署任务

接下来配置部署任务。

首先点击新建服务授权,根据跳转完成阿里云 RAM 授权,授权信息会自动回填到任务配置中,然后依次选择到前面准备好的 EDAS 应用,并对发布策略进行相应配置:

					保存 保存并运行 X
▼ 源(1) +	新阶段		新阶段	编辑	×
…/flow-example/spring-boot				<b>社务亚臻</b> 选择	新建限を掲載
‰ master	构建 🖬 🦻	EDAS Kubernetes 应用发布	1 新的任务	rdc-rdc的edas服务授权	
▼ 变量 (0)				区域	
				华北2(北京)	~
暂未设置,点击 添加			-	应用	
			2	edas-k8s-tesst	~
				镜像	
			(3)	标签.\${DATETIME}	~
				发布批次	
				2批	~
				分批等待时间	
				127 11	~
			4	分批方式 ▶ 手动确认	6
① 定时运行			•		

这里选择的策略是分两批发布,分批方式为手动确认。该 EDAS 应用有两个 pod,因此实际的发布过程为先发布一个,等待手动确认后,再进行第二个的发布。

#### 运行流水线

#### 配置好之后,触发流水线。

构建完成之后,进入部署节点。由于配置了分两批,手动确认,所以在部署完第一台之后,会暂停,点击"更多"查看可进行的操作:



你可以选择

- 终止本次变更
- 继续第二个 pod 的发布
- 跳转到 EDAS 控制台查看更多信息

点击继续"恢复变更"之后,可以看到飞流继续完成了本次部署。

🔁 > 3	欹水线 2020-03-18 ▼								•	99+ ⑦
最近运行	运行历史							💽 运行	⊘编辑 ,	咚 成员 三支
#3 🕑 运行成	功	构建		新阶段	新阶段					
运行人	R RDC支持	✓ 阿里云镜像构建		SEDAS Kul	pernetes 应用	发布				
开始时间	2020-03-18 19:44:09	运行成功			运行成功					
持续时间	5分28秒	〇 日志	40秒	∂ 发布单详情	四 日志	4分48秒				
▶ 源(1)										
										G

# 6.6. SAE 应用发布

Flow 提供了 <mark>SAE</mark> 部署能力,Serverless 应用引擎 SAE(Serverless App Engine)是面向应用的 Serverless PaaS 平台,能够帮助 PaaS 层用户免运维 laaS、按需使用、按量计 费,做到低门槛微服务应用上云。相对于其他 Serverless 产品,它抽象了应用的概念,并提供了一整套微服务解决方案,支持 Spring Cloud、Dubbo、HSF 等主流的微服务开 发框架,实现了 Serverless 架构和微服务架构的完美结合。

#### 部署目标物

部署目标物可来自上游构建产物,支持两个方式:构建(Java构建等)、镜像构建(阿里云镜像构建等)。

▼ 源(0) +	构建			选择任务类型		×	
55 + 20 m		Q。请输入任务名称	8				
智木说道, 烏古 2020	Ja - Q - Ell	代码扫描	Python 构建测试上传 Python 构建测试上传	-GO 构建测试上传 Go 构建测试并上传			
▼ 変量 (0) 🔗 +		测试测试和理	构建				
暂未设置。点击 添加		物建物建	Java 构建 Java 构建, 压缩包上传, 用	发布 Maven 二方库 Maven 构建,并上传到 Mav	Python 构建 Python 构建上传到仓库		
		工具	- <b>CO</b> 构建 Go 构建	nede Node.js 构建 Node.js 构建	PHP 构建 PHP 构建上传到仓库		
		代码	镜像构建				
		执行命令 空模板	<b>戸里云镜像构建</b> ACR 阿里云镜像构建	→ 阿里云镜像企业版 ARC EE 构建镜像并上传	➡ 自定义镜像构建 自定义镜像构建	(2)	
			ТД				
			AIFa AIta	Helm Chart 上传自定义 将代码库中的 Helm Chart	Jenkins 任务 触发 Jenkins 任务	_	

#### 🗘 注意

:如果您使用的是 Java 构建,构建完成后,请使用 构建物上传(EDAS/SAE使用)步骤。

< 返回 流水线 2020-11-24		基本信息 流程配置 触发设置 变量和缓存		8 项任务未配置完	仅保存 保存并运行
ā.	构建	部署	新阶	编辑 谊	
1. 25 40 /0 10 25				任务名称	
- 38/00TUP388		G 及 Serverless(SAE) 应用发布 G	C	Java 构建上传	
				构建集群 ②	
				云效北京构建集群	~
				任务步骤	
				③ 添加步骤	
				› Java 构建	0
				→ 构建物上传(EDAS/SAE使用)	c
				步骤名称	
				构建物上传(EDAS/SAE使用)	
			(1)	制品名称 ⑦	
				default	
				上传文件 ⑦	
				target/application.jar	
				江友评价	
				任务插件	

#### 添加SAE部署任务

在流水线编辑时,你可将SAE部署任务添加至您的流水线中。

▼ 遯(0) +	构建			选择任务类型		×	
新生设委 古主 現如		Q。请输入任务名称					
HE AC INC. IN . AN CALL AND M	ini 9 Ja	代码扫描	AIta AIta	Helm Chart 上传自定义 将代码库中的 Helm Chart	Jenkins 任务 触发 Jenkins 任务		
▼ 变量 (0) · · · · · · · · · · · · · · · · · · ·		测试测试构建	部署				
暂未设置,点击 添加		构建	<b>主机部署</b> 主机部署	主机部署(无构建包) 使用脚本发布到主机	Docker部署 主机Docker部署		
		镜像构建 工具	成 Kubernetes 发布	Kubernetes 镜像升级	Kubernetes 分批发布		
		部署	用于发布 Kubernetes YAML	升級 Kubernetes 下的已有	Kubernetes 分批发布		
		代码 执行命令	Helm Release 部署 使用代码库中的 Helm Chart	EDAS ECS应用发布 EDAS ECS应用发布	EDAS Kubernetes 应用… EDAS Kubernetes 应用发布		
		空模板	Serverless(SAE) 应用 Serverless(SAE) 应用发布				
			代码				
			A.0.0.0		Annua		

#### 配置SAE部署任务

1)需要先在阿里云SAE控制台添加应用,支持程序包和镜像两种方式。

2)部署配置,支持灰度发布和分批发布。	
27 即看此旦,又将从皮及仰袖力加及仰。	

▼ 遯(0) +	构题	构建	新阶段	编辑
				任务名称
智木设置,点击 源加	G G Java Kjilt	G Gerverless(SAE) 应用发布 ①	◎ 新的任务	Serverless(SAE) 应用发布
▼ ^{空景} (0) - 2 →				任务步骤
				选择授权 ③ 新建服务器
暂未设置。点击 添加				rdc-rdc的sae服务授权
				区域
				华东1 (杭州)
				SAE应用
				sae-jar
				約38,25%
				制品名称.default
				友仲東略
				20 38-90 HP
				分批方式
				自动确认
				发布批次
				2批
				分批等待时间
(页) 定时运行				不等待

#### 发布 ACR 企业版镜像

目前 Flow 支持将 ACR EE (企业版) 镜像发布至 SAE,可以在流水线中构建 Docker 镜像并推送至 ACR EE (企业版)中,并将其发布至 SAE,如下图所示:

く 返回 SAE 支持 ACR EE	基本信息 流程配置 触发设置	变量和组	夏存 仅保存 保存并近	运行
	新阶段		编辑 ⑪	×
			命名空间	
镜像构建并推送至阿里云镜像仓库企业版	Serverless(SAE) 应用发布		liuxi-test	~
			SAE应用 ⑦ 📵	
			php-test-liuxi	~
			构建产物	
			镜像构建并推送至阿里云镜像仓库企业版.镜像VPC地址	~
			☑ 使用阿里云镜像仓库企业版	
		1	镜像仓库企业版实例ID ②	
			cri-9kl26z	
			发布策略	
			分批发布	~
			分批方式	
			自动确认	~

注意:在 SAE 发布节点配置中,需填写镜像仓库企业版实例 ID,你可通过 ACR 控制合 找到你的 ACR EE (企业版)的实例 ID,如下图所示:

		1001		7月11年20-30 半示1	(1)(7)(7) 关闭及音 10/(EAX 运行/公司	↓ 1211
既览		实例信息		⊘ 标准版	重要公告	
全库管理	^	实例ID cri-9kl26zh /w (1	付费类型 包年包日	实例规格 标 <b>准版</b>	♥【升级通知】容器镜像服务企业版将	= 2021
镜像仓库		4444t	网期日期	计设计版字件	年 9 月 23 日 00:00 - 02:00 进行张 州地域实例升级。升级期间可能出现知	家口, 杭 豆暂推送
命名空间		华东1(杭州)	2021-10-21【剩 26 天】	500节点	或拉取失败,客户端可以尝试重试。给	合您带来
代码源 NEW					的个使敬请谅解,有任何问题可工单助 们。	大杀戎
访问控制		合废信自				
域名管理		已年后志				
版本管理 NEW		仓库	4/3000 OSS B	ucket I26z y		
elm Chart	~	命名空间	1/25 OSS #		头例设置	
《例管理	~		1/25 177.23	加密保护中 8 GB	公开匿名拉取 🕜	$\bigcirc$
发管理	~					
容可信	~				组件设置	
		基本功能			Charts 🕜	
			♠ 白完议域名 NEW	⊘ 按票加裁提效 NEW		
		▲ 王球问罗加速				

# 6.7. 阿里云函数计算 FC 发布

Flow 提供了<mark>阿里云函数计算(FC)</mark>发布能力,支持对已有的函数进行发布。支持使用 OSS 中保存的函数 zip 包发布到阿里云函数计算。函数计算(Function Compute)是一 个事件驱动的全托管 Serverless 计算服务。您无需管理服务器等基础设置,只需编写代码并上传。函数计算会为您准备好计算资源,并以弹性、可靠的方式运行您的代码。

#### 添加函数发布任务

用户可以在已有流水线中添加新任务,并选择阿里云函数计算发布任务:

遼(0) +				选择任务类型	×	
		Q。请输入任务名称	8			
暂未设置,点击 添加	◎ 新的任务					
		代码扫描	部署			
变量 (0) 🔗 🕂		测试 测试构建	主机部署	■ 主机部署(无构建包) 使用脚本发布到主机	Docker部署 主机Docker部署	
暂未设置。点击 激加		构建 镜像构建	Kubernetes 发布 用于发布 Kubernetes YAML	Kubernetes 镜像升级 升级 Kubernetes 下的已有	Kubernetes 分批发布 Kubernetes 分批发布	
		工具	Helm Release 部署 使用代码库中的 Heim Chart	EDAS ECS应用发布 EDAS ECS应用发布	OF EDAS Kubernetes 应用 EDAS Kubernetes 应用发布	
		代码 执行命令	Serverless(SAE) 应用… Serverless(SAE) 应用发布	國数计算 应用发布     更新造数计算 (FC)中函数	1	
		空模板	代码			
			合并代码 将代码输入源中指定的代码		<b>创建标签</b> 使用 Git tag命令创建标签。	

#### 授权 Flow 访问用户函数计算服务

首次使用 FC 发布的用户需要首先授权 Flow 访问用户的函数计算服务:

圖 流水线编排 函数计算				保存保存并运行
▼ 源(0) +	新阶段	8580 FR	1930	
100			任务名称	
智木说置。 息古 2870	G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G     G	◎ 新的任务	函数计算(fc) 应用发布	
▼ 変量(0) 22 +			任务步骤	
			● 添加步骤	
暂未设置,点击 添加			~ 阿里函数计算部署	
			步骤名称	
			阿里函数计算部署	
			选择授权 (1)	◎ 新建服
			区域	
			服务名	
			函数名	
			OSS Bucket	
			OSS Object Name 🕥	
位 定时运行				

#### 部署配置

完成授权后,选择函数所在区域(Region),并选择需要发布的服务和函数。并指定函数所在的 OSS Bucket 以及函数 zip 包的 Object Name。

图 流水线编排 函数计算				保存
▼ 源(0) +	新阶段	新阶段	1611E	
			V ANDER W	
暫未设置,点击 添加	□ · · · · · · · · · · · · · · · · · · ·	◎ 新的任务	~ 阿里函数计算部制	6
▼ 変量 (0)			步骤名称 阿里函数计算部	6
			选择授权	◎ 新建施
暂未设置,点击 激加			rde-rdc的tc服务	受权
			医城	
			华东1 (杭州)	
			服务名	
			hello-world-Gre	tingService-B5AFDD68EAF3
			函数名	
			1 flow-fc-release	
			OSS Bucket	
			liuxi-tb-test-oss	
			OSS Object Name	0
			fc-demo/test-6	1.zip
			C.8.16.1	
			□ 求加强件 ● 添加强件	
			删除任务	

#### 运行并发布到函数计算

保存并运行流水线,在流水线最新运行也查看运行状态,发布完成后点击函数控制台连接可以跳转到函数计算服务。

> 函数计算 * 音				o 🖉 🕐
<b>最近运行</b> 运行历史			⊙ 运行	✓ 編辑 八、成员 三百
#1 🔮 运行成功	新阶段			
运行人 🥶 136	☑ 函数计算(fc) 应用发布			
开始时间 2020-03-12 21:13:16	运行成功			
持续时间 33秒	⑦ 函数控制台 □目志 33秒			
				G

## 6.8. 资源编排服务 ROS 发布

<mark>阿里云资源编排服务(Resource Orchestration Service 简称 ROS)</mark>是一种简单易用的云计算资源自动化部署服务。用户可以通过使用 Json/Yaml 格式的模版描述多个云计算资 源(如ECS、RDS、SLB)的配置、依赖关系等,并自动完成所有云资源在多个不同地域以及多个账户中的部署和配置,实现基础设施即代码(Infrastructure as Code)。 通过使用资源编排 ROS 和 Flow ,您可以做到持续交付、自动构建、测试创建更新 ROS 模板,然后再将这些更改应用到生产资源栈。此发布过程可让您快速可靠地对 Aliyun 基础设施进行更改。

例如,您可以创建一个工作流程,此工作流程在您将更新的模板提交到代码存储库时会自动构建测试资源栈。在资源编排 ROS 构建测试资源栈后,您可以测试它,然后决定是 否将更改推送至生产资源栈。

#### 实例:为测试和生产资源栈构建流水线

如果您需要建立一个发布流程:您提交一个资源编排 ROS 模板,然后资源编排使用该模板自动构建测试资源栈。在您审查测试资源栈后,可预览您的更改将如何修改生产资源 栈,然后选择是否实施更改。要完成此工作流程,您可使用资源编排构建测试资源栈、创建更改集,然后执行更改集。但是,在每个操作中,您都需要手动与资源编排进行交 互。在此实例中,我们将会构建一个流水线来实现上述多个操作的自动化,从而帮助您使用资源编排资源栈实现持续交付工作流程。

#### 实例简述

在资源栈中为示例 WordPress 站点构建一个流水线。此流水线分为三个阶段。每个阶段均必须包含至少一个操作,此操作是流水线将在您的构件 (您的输入)上执行的任务。阶 段组织流水线中的操作。流水线必须完成阶段中的所有操作,然后阶段才会处理新构件,例如,您提交新输入以重新运行流水线。 在此实例过程中,您将获得执行以下工作流程的流水线:

#### 第一阶段

从存储库中检索源构件 (资源编排 ROS 模板及其配置文件)。

您将准备一个包含示例 WordPress 模板的构件并将该构件上传到代码仓库。

1、先从资源编排获取示例模板"基于 ECS 和 RDS 创建 WordPress 环境"

	上海) ▼			Q 接张文档、控制台、API、解决方案和资源	週用 工单	骨窦 企业	支持 盲网	D Ó	`₩ ®	简体 🌔
资源编排 ROS	资源编排ROS客户支持打打群: 11783495									×
院迎页	普页 / 模板示例									
可视化编辑器 [3]	模板示例							_		
资源线	云计算基础服务 弹性计算 网络 容職	数据库 存储 安全 开发运维 wir	ndows应用服务				WordPre			Q
治源线组 NCW	应用场员 网站 大数据 人工智能									
资源类型										
構板へ	(I)	Ŵ	®							
我的模板	w	w	w							
模板示例	创建一个WordPress可弹性伸缩的集群	创建wordpress编群并配置VPC的SNAT 功能访问internet	基于ECS和RDS创建WordPress环境。 基于ECS和RDS创建WordPress环境。							
解決方変中() 1001	包含RDS、SLB、ECS及NATGateway。	在VPC网络环境下创建wordpress集群,并配置 VPC的SNAT功能防闭internet。								
日初代認識のいて	4	LI 6679 00 9700 0719 00 00 00 00 00 00 00 00 00 00 00 00 00								
	弹性计算 网站	网站 网络 page.solu	网站 page.solutions							
	查看洋街 创建资源线	查看洋博 创建资源线	查看洋街 创建资源线							
										*

#### 2、上传到代码仓库

#### 第二阶段

流水线将创建测试资源栈,然后等待审批。

```
1. 点击右上角创建流水线,选择其他,点击空白模板。
```

		选择流水线模板	$\times$		
☆ 我的星标		Python · 镜像构建,发布到Kubernetes集群/阿里云容器服务	*	S & Et	分析建流水线
▲ 我参与的		镜像构建 部署		最近法行时间	
♀ 全部流水线	👙 Java			40004213 49145	~
	php PHP				₫ ☆
企业管理					₫ ☆
A。通用变量组	🕼 Node.js	执行命令			
□ 标签管理	GO GO			2020-04-28	₫ ☆
□ 主机组管理	Python	其他·空模板			
☑ 流水线模板管理	—————————————————————————————————————	空模板			
☑ 任务组管理					
□ 步骤管理		••• 其他 · 镜像构建,发布到Kubernetes集群/阿里云容器服务			
		10/46-51708			
⑥ 企业设置		DREASE 20-982			
			T		
		I	以消 创建		

2. 添加代码源

く 返回 流水线 2020-06-17		基本信息	流程配置 触发设置 变量	晶和缓存			仅保存	保存并运行
源	阶段1			添加流水线源				×
+ %10/689)	<b>9</b>		<ul> <li>新的任务</li> </ul>	代码源 Jenkins	送择代码源 示例代码源	③       阿里云Code       通用Git       通用Git       Kolker nd simperation       快 ③	C 自建Gitlab Github	G 码云
				添加				

#### 3. 添加流水线步骤

く 返回 流水线 2020-06-17	基本信息	流程配置 触发设置 变量和缓存		仅保存	保存并运行
源	阶段1	新阶段	编辑 ①		×
G thread th	WordPressTest	<ul> <li>新的社会</li> </ul>	任务名称 WordPressTest 构建集群 ⑦ 国内集群 任务步骤		~
			<ul> <li>添加少報</li> <li>静态扫描</li> <li>測试</li> <li>代码</li> <li>覆盖率</li> <li>内建</li> <li>双右</li> <li>万载</li> <li>上传</li> <li>自定义少骤</li> </ul>	阿里函数计算部署 Helm Release Kubect 发布 Kubect 機像升级 ROS 发布	

4. 编辑 ROS 发布步骤,编辑完后,并点右上角仅保存

#### 云效2020

〈 返回	基本信息 流程配置 触发设置 变量和缓存	仅保存 保存并运行
源 阶段1	新阶段	编辑 位
WordPressTes     waster	a #69915#	- ROS 发布 步骤名称 ROS 发布
+ #####################################		
		阿里茲 Region ⑦
		ROS 扮演角色 ⑦ TestTeamRole
		Stack名称 ⑦ WordPressTest
		stack交件略径 ⑦ Test/WordPress.yml
		ABatatal ⑦ 10
		☑ 忽略环境未发生变化的错误 ⑦

5. 创建评审,新建任务,选择工具,人口卡点



6. 编辑人工卡点

#### 云效流水线 Flow·部署

く 返回  流水线 2020-06-17		基本信息 流程配置 触发设置 变过	量和缓存		仅保存保存并运行
	阶段1	新阶段	新阶段	编辑 並	×
Hilliam HC Citry vincepage     Hilliam HC Citry vincepage     Hilliam HC Citry	6 WordPresTest	•         •	• #991.9	任务名称 代码评审 验证者类型 人及 验证人 任务插件 ③ 添加新祥	~ ~
					Ø

7. 点击保存并运行,等待审批

* * * <b>*</b>				
运行历史				
⑧ 终止	阶段1		新阶段	
🕒 milet 🖲 🗤	WordPressTest		● 代码评审	
100 - 10 P C 16	运行成功		等待中	
30秒	2 日志	27段	<ul> <li>验证不通过      <ul> <li>验证通过</li> </ul> </li> </ul>	280
	▲ ● ● ● ◆ ◆ 运行历史 ● ● 傑止 ● ● 傑止 ● ● 『 ■ 『 ■ 『 ■ 『 ■ 『 ■ 『 ■ 『 ■ 『 ■ 『 ■	■ ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	▲ ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	■ 計画 ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・

```
在审查测试资源栈后,可选择继续使用原来的流水线或创建并提交另一个构件以进行更改。
如果您批准通过,然后流水线将继续下一阶段。
如果您批准不通过,可以返回第一阶段,修改模板或配置文件,重新执行流水线,再次审批。
```

#### 第三阶段

流水线将针对生产资源栈创建更改集,然后等待审批。 **情况一:** 在初始运行中,您如果没有生产资源栈。更改集显示资源编排 ROS 将创建的所有资源。如果您批准,该阶段将执行更改集并构建您的生产资源栈。 **情况二:** 在初始运行中,您有生产资源栈。更改集显示资源编排 ROS 将更新的所有修改过配置的资源。如果您批准,该阶段将执行更改集并更新您的生产资源栈。

✓ 注意
 : 需要使用的模板可以从资资源编排 ROS 控制台的模板示例中获取"基于 ECS 和 RDS 创建 WordPress 环境"

# 6.9. 无构建部署

如果部署时不需要拉取包,而只是通过git更新代码,可以使用我们无构建部署的能力。

#### 添加获取 Git 版本

首先给流水线添加一个代码源,新增一个空的阶段,在阶段中添加一个获取 Git 版本的步骤。并设置制品名称。

■ 流水线编排 流水线 2020-05-27			保存 保存并运行
▼ 遼(1) ⊣	- 阶段1	新阶段	编辑
w-example/php-laravel-blog	日日日	6 新約任务	任务名称
¦∾ master			空任务
▼ 变量 (0)			任务步骤
			添加步骤
暂未设置,点击 添加			~ 获取 Git 版本
			步骤名称
		1	获取 Git 版本
			制品名称 ⑦
			defalut
			任奏碼件
			◎ 添加插件
			住务输出 @ 制品名称.defalut
₽ 触发设置			<b>删除证穷</b> 该任务将被彻底删除,注意这是不可逆操作,任务下所有数据将会删除

#### 部署配置

在后续部署步骤,例如主机部署任务中,选择对应制品,即可在部署脚本中使用对应 Git 环境变量。

- GIT_REPO: Git 地址
- GIT_BRANCH: Git 分支
- COMMIT_ID: 提交版本号

echo \${GIT_REPO} echo \${GIT_BRANCH} echo \${COMMIT_ID} git clone \${GIT_REPO} -b \${GIT_BRANCH}

〈 返回	基本信息 流程配置 触发设置 变量	和缓存	仅保存 保存并运行
<b>第</b> 行的 经	部·考	新阶段	编辑 ① ×
flow-example/spring-boot	●         主机部署	③ 新的任务	制品名称.defalut
+添加代码源			主机组 ● 新建主机组 孝玄 ~ ~ · · · · · · · · · · · · · · · · ·
			部署配置
			下载路径 ⑦ /home/admin/app/package.tgz
			执行用户 ⑦ root
			部署砌本 ⑦ ² echo \${0T_REP0} ² echo \${0T_REP0}
		2	3 echo \${COMMIT_ID} 4 git clone \${GIT_REPO} -b \${GIT_BRANCH}
			- 李贵洪行检码

部署脚本可以直接使用这几个变量完成拉取代码操作。服务器上需要自行解决拉取代码的认证问题,比如可以将服务器上 SSH 公钥配置到代码库中。

# 7.质量检测 7.1.代码扫描

#### 7.1.1. 代码扫描能力

Flow 为主流语言提供了开箱即用的代码扫描能力,为用户提供了快速检测代码质量的能力。

#### 支持语言和规则

目前飞流支持扫描的语言和规则,如下表所示:

语言	扫描能力	规则说明
Java	阿里巴巴规约扫描	根据阿里巴巴多年经验提炼的总共7大类Java代码规则,包括:1)OOP规 约 2)并发处理 3)控制语句4)命名规约 5)常量定义 6)注释规范 7)其他。如果未做扫描配置,默认会对所有规则进行扫描,如需配置,可 参考更多扫描配置一节
Java	Findbugs安全扫描	Findbugs默认扫描规则
JavaScript	ESlint	Eslint默认扫描规则
Nodejs	ESlint TSlint	Eslint默认扫描规则
Python	Flake8	PhpMetric默认规则
Php	PhpMetric	PhpMetric默认规则
C++	CppCheck	Cppcheck默认规则
Golang	Golangci-lint	Golangci-lint默认规则

#### 扫描场景支持

- 代码全量扫描
- 代码增量扫描
  - 全量扫描和增量扫描
- 特定的代码目录扫描指定或排除目录扫描
- 指定规则集扫描 指定扫描规则集

#### 7.1.2. 使用代码扫描

Flow 提供了开箱即用的代码扫描能力,在流水线中添加「代码扫描」任务,即可启用。用户可以通过以下方式添加代码扫描任务。

- 使用模板新建流水线,选择包含「**代码扫描」**任务的模板
- 在已有的流水线中,添加新的阶段,并选择「**代码扫描」**任务

#### 流水线模版创建

新建流水线时,选择对应的开发语言,可以查看当前语言下的默认流水线模版,选择带有"代码扫描"节点的流水线模版,即可快速使用代码扫描能力 。

☆ 我的星标	我的星标			EL 🛛 🖸	新建流
② 我參与的		洗择流水线模板	×	最近运行时间	
♀ 全部流水线		YE314 (107) 200 D4 DV	杨超越	2019-09-24	
设置信息	Java PH	P 测试、构建、部署到主机	杨超越	2019-09-24	
② 企业设置	JS NodeJS	代码扫描	刘吴然	2019-09-24	
	php PHP GO GO	部署	吴彦祖	2019-09-23	*
	Python	251 124	杨超越	2019-09-24	*
	吕 其他 PHI	P 测试、构建	刘德华	2019-09-23	
		代码扫描 构建	周杰伦	2019-09-23	*
	C	测试	刘昊然	2019-09-24	*

#### 编排流水线,添加代码扫描任务

在流水线编排中,可以通过添加任务,在任务类型分类"代码扫描"的任务,可以快速的添加代码扫描任务。

đ.	C *****	3862-#	****	口谷如果	新的阶段
(-) flow/flow-engine			远挥仕穷失望	~	
% master	9、请输入任务名	称			➡ ⑤ 新的任务
	空模板	测试			
(1)	代码扫描	Maven 単元測试 Maven 単元測试	Gradle 单元测试 Gradle 单元测试	NodeJS 单元测试	
	测试		Citate #7583.m	100000 = 7089 M	
	测试构建	Python 单元测试 Python 单元测试	PHP Codeception 单元 PHP Codeception 单元测试	PHP Unit 单元测试 PHP Unit 单元测试	
	构建				
	镜像构建	Go 单元测试 Go 单元测试			
	部署				1
	代码	測试构建			
	执行命令	Java 単元測试、构建 Java单元測试、构建、 Ava单元測试、构建、构建	nede NodeJS 构建测试上传 NodeJS 单元测试、构建、	nede React 构建测试上传 React 构建测试并上传至 Oss	
		Python 镜像构建测试           Python 镜像构建及单元测试	Python 打包测试 Python 打包及单元测试	<b>Go 构建测试上传</b> Go 构建测试 <b>上</b> 传	
		#512#			

## 7.1.3. 全量扫描和增量扫描

Flow 支持代码全量扫描和增量扫描,可以根据需要,选择合适的扫描方式进行使用。
1) 代码规约全量扫描,通过飞流的扫描任务对指定的代码工程进行编码规约全量检测。
优点: 支持跨文件引用,代码扫描全面。
缺点: 但扫描速度较慢,问题量会比较多,难以修复
2) 代码规约增量扫描,是基于代码的一次 push,自动获取 diff 内容,对 diff 文件用编码规约规则进行扫描,并过滤出此次提交产生 diff 规约问题功能。
优点: 只扫描 diff 文件,扫描速度很快,增量问题直接关联到人,能有效防止代码提交引进新问题数。
缺点: 因只扫描 diff 文件,不能发现跨文件引用出现的规约问题。

## 7.1.4. P3C代码扫描

Java 规约扫描是阿里巴巴多年经验提炼的 Java规范- 包含7大类 Java 代码规则:

- 命名规约
- 注释规范
- 集合
- OOP 规约
- 并发处理
- 控制语句
- 其他。如果未做扫描配置,默认会对所有规则进行扫描,也可以在规约扫描组件配置。

#### 7.1.4.1. P3C代码扫描

Java 规约扫描是阿里巴巴多年经验提炼的 Java规范- 包含7大类 Java 代码规则

- 命名规约
- 注释规范
- 集合
- OOP 规约
- 。 并发处理
- ∘ 控制语句
- 其他。如果未做扫描配置,默认会对所有规则进行扫描,也可以在规约扫描组件配置。

#### 7.1.4.2. 指定扫描规则集

#### 配置指定规则集

配置规则集如下图所示:

- 1) 在流水线编辑状态,点击相应的规约扫描卡片
- 2) 在展开的卡片抽屉上,点击任务列表 如图中: Java 代码扫描
- 3) 在规约集合处, 配置相应的规则集

编辑流水线		返回详情页 保存 道行 く 更多く	*壮穷石林-
			Java代码扫描
入源	● 阶段 (1) ④ 100% ∈	阶段:代码扫描 🖹 ×	构建步骤:
Cつ 101942-group_java 同里云Code		*阶段名称:	*步骤类型
	代發目編	. 代码扫描 4/20	Java代码规约扫描
	atEI弱升avaL	*流转配置: 2	*步骤名称
		自动流转 点击	step1-Java代码规约扫描 160
	0 NH / 0	任务列表:	1001年今 (1)
	Javanyae, 11/2/08/20	Java代码扫描	ali-commont yml ali-consurrant yml ali-
	nin p	<ul> <li>新建任务</li> </ul>	(3) constant.xml.ali=exception.xml.ali=
	Maven单元测试		flowcontrol.xml.ali-naming.xml.ali-oop.xml.ali-
			子目录:⑦
			语输入
			排除子目录: ②
			test/

#### 7.1.4.3. 指定或排除目录扫描

配置目录扫描:

- 1) 在流水线编辑状态,点击相应的规约扫描卡片,
- 2) 在展开的卡片抽屉上, 点击任务列表 如图中: Java 代码扫描
- 3) 子目录: 配置扫描的相应目录; 排除子目录: 配置排除在扫描范围内的相应目录。

编辑流水线		返回详情页 保存 运行 V 更多 V	*注穷石标-
e) 28		80.60- 4PE 121218 C V	Java代码扫描
豊くな		WINC TOWARDS	构建步骤:
C-D 101942-group_java		*阶段名称:	*开闢 凿积
		代码扫描 4/20	Java代码规约扫描
	代码扫描 🖉		
	Java代码扫描	<ul> <li>流转配置:</li> <li>2</li> </ul>	*步骤名称
		自动流转	step1-Java代码规约扫描 16/2
	· 〇 · · · · · · · · · · · · · · · · · ·	Т	
	inva传统,打压缩快	任务列表:	*规则集合: ⑦
		Java代码扫描	all assument und all annu ment und all
	潮ば ター・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・		constant.xml.ali-exception.xml.ali-
	Maxan 第一部注	④ 新建任务	flowcontrol.xml.ali-naming.xml.ali-oop.xml.ali-
			子目录: ②
			VIII VIII
			前潮入
			(3)
			● 押除于目录: ②
			test/
(0, strp+1=4=			

#### 7.1.5. 使用自定义代码扫描规则

#### 7.1.5.1. 使用自定义代码扫描规则

目前 Flow 已经内置了支持 Java,JavaScript,Python,Go,TypeScript,PHP 等语言的代码质量扫描能力,并提供了默认的代码扫描规则。对于需要对扫描规则进行调整的 用户可以使用自定义代码扫描规则。

目前支持自定义代码扫描规则如下:

- ESLint 自定义规则
- TypeScript自定义规则
- Java P3C自定义规则

#### 7.1.5.2. ESLint自定义规则

#### 添加代码扫描任务

添加流水线任务并选择JavaScript代码扫描,如下所示:

1	0 -	代码扫描		选择任务类型	×
huaihu/ts-lint-san	nple	0-9-T	Q。请输入任务名称 SI		
<b>变量(0)</b> 智未设置。点击	<b>۵ -</b>		空模板 代码目鏡 测试 测试内镜 构镜 镜像构镜 工具 (代码 执行命令	State       State       State       Mark       Mark   <	PHP Metrics III/2E315 PHP Metrics III/2E315
				潮ば <u>4</u> Maven 単元期は <u>4</u> Maren 単元期は <u>5</u> Gradie 単元期は <u>5</u> Gradie 単元用は	nede Node js 奉元期任 Node A # C # H
					C C

#### 使用自定义规则

默认情况下Flow使用内置的eslint检查规则,如果需要使用自定义规则,需要在任务配置中勾选"使用本地ESLint配置",勾选后需要确保项目根路径下包含.eslintrc.*文件即 可。

加水热调拌 流水熱	2019-11-28			保存 研究开
1)	0 -	代码扫描	新阶段	编辑
huaihu/ts-lint-sam	ple	Fill Charles (P12) 1214	CO. MENTER	任务名称
		an 4	WIBYLT3	JavaScript 代码扫描
变量(0)	0 -			任务步骤
暂未设置,点击	oto:			活力的分類
and the second second				- JavaScript 代码扫描
				步骤名称 JavaScript 伊丽扫描
				运行本地 ESLInt 配置所需依赖 ③
				排除子目录 ⑦ test/
				node_modules/ vendor/
				红线信息
				十 添加红线
				任务播件
A street 20				● 添加播件
(1) 定时运行				

#### 更多规则请参考ESLint官方文档

#### 7.1.5.3. TypeScript自定义规则

#### 添加TS代码扫描任务

添加任务并选择TypeScript代码扫描任务,如下所示:

1)	0 -	代码扫描		1	选择任务类型	×	<
🕒 huaihu/ts-lint-sam	ple		Q。请输入任务名称				
∲∽ master		₽ ¢ Ja	空模板	呈目的过旁,用户"可以承担			
時变量(0)	0 -	- & Ja	代码扫描	代码扫描			
暂未设置,点击	满力口	· \$	测试	Java 代码扫描 同里巴巴 Java 代码规约扫描	Java 安全扫描 Java 安全扫描	PHP Metrics 静态扫描 PHP Metrics 静态扫描	
		ې PH چ Jav	构建 镜像构建	nede JavaScript 代码扫描 JavaScript 代码扫描	Python 代码扫描 Python 代码扫描	Go 代码扫描 Go 代码扫描	
		· •	IR 1	TS TSLint 代码扫描 TypeScript 代码扫描			
		\$ TS	代码	测试			
			执行命令	Maven 单元测试 Maven 单元测试	Gradie 单元测试 Gradie 单元测试	nede Node.js 单元测试 Node.js 单元测试	
				Python 单元测试 Python 单元测试	PHP Codeception 单元 PHP Codeception 单元责试	PHP Unit 单元测试 PHP Unit 单元测试	
(6) 定时运行							

#### 使用自定义规则

默认情况下,Flow会使用TSLint默认的代码扫描规则。为了使用自定义的代码扫描规则用户需要在项目根路径下提供tslint.json文件。 初次使用可以使用如下命令生成该文件:

tslint --init

保存并运行流水线,TSLint任务会自动检查代码项目根路径下是否包含该文件,如果包含则使用用户定义的tslint.json文件。

高水线编排 流水线	2019-11-28			1 保存并运行
(1)	0 -	代码扫描	题FIRF FQ	1211
huaihu/ts-lint-sam	ple	日	3推 ③ 新約任务	任务名称
(O)	0.1			TSLInt代码扫描
交重(0)	0.			任务步骤
85 m 1/1 80 Jr m	NE da			● 添加步骤
而水说直, 凤田	292.0U			~ TypeScript代码扫描
				步骤名称
				TypeScript代码扫描
				扫描原码路径 ①
				src/**/*.ts
				捕鲸子目录 ①
				test/
				红线信息
				十添加红线
				任务播件
				● 添加插件
				MRF6
(6) 定时运行				·····································

#### 更多规则可以参考TSLint 官方文档

7.1.5.4. Java P3C自定义规则

#### 自定义规则集

用户可以从Github 获取默认的规则集合文件,并提交到代码仓库中。添加 Java 代码扫描如下所示:

(1)	0 -	构建			选择任务类型		×	
@820f5/pub-access			Q。请输入任务名	5称				
}∽ master	分脱发	<b>0</b> - 9	阿里空模板	空模板				
局变量(0)	0 -		代码扫描 测试	+ 空白任务 安白的任务,用户可以添加				
暂未设置,点击 添加	1		测试构建	代码扫描				
			构建 1	Java 代码扫描 同里巴巴 Java 代码成约扫描	Java 安全扫描 Java 安全扫描	PHP Metrics 静态扫描 PHP Metrics 静态扫描		
			工具	nade JavaScript 代码扫描 JavaScript 代码扫描	Python 代码扫描 Python 代码扫描	<b>Go 代码扫描</b> Go 代码扫描		
			代码 执行命令	TSLint 代码扫描 TypeScript 代码扫描				
				测试				
				Maven 单元测试 Maven 単元測试	Gradle 单元测试 Gradle 单元测试	nede Node.js 单元测试 Node.is 单元测试		
								6
								<b>P</b>

在任务配置中,勾选"使用自定义规则目录"选项,并制定自定义规则文件所在目录,如下图所示:

■ 流水线编排 流水线 2019-12-16					保存 保存并述
(1)	构建	新阶段		编辑	
⊕ …820f5/pub-access     与 触发	B B Java (PRI-12)	0 MM/48		任务名称	
}∾ master				Java 代码扫描	
局交量(0)	•			任务步骤	
				添加步骤	
暂未设置,点击 激加				- Java 代码规约扫描	
				步骤名称	
				Java 代码规约扫描	
			(1)→	▶ ☑ 使用自定义规则目录 ⑦	
				自定义规则目录 ⑦	
			(2)	src/resources/rules	
			<u> </u>	子目录 ⑦	
				排除子目录 ⑦	
				test/	
				红线信息	
				十 激加红线	
				任务播件	6
16 定时运行					

保存并运行流水线即可使用代码库中定义的自定义规则集。

## 7.2. 单元测试

Flow 为主流语言提供了开箱即用的单元测试能力,可以方便的对代码进行单元检测。

#### 支持语言

- 目前飞流支持单元测试的语言如下表所示:
- Java Maven
- Java Gradle
- Nodejs
- Python
- PHP Codeception
- PHP Unittest
- Golang

#### 如何使用单元测试

在飞流中可以通过流水线任务快速的使用单元测试。

#### 新建流水线时,通过流水线模版创建单元测试任务

新建流水线时,选择对应的开发语言,可以查看当前语言下的默认流水线模版,选择带有"测试"节点的流水线模版,即可快速使用单元测试能力

☆ 我的星标	我的星标			EL 🛛 🔿 1	折建流
② 我参与的		法指义卡布森抗	~ 1	最近运行时间	
\$ 全部流水线		人口が中川レイトの公式後位又	杨超越	2019-09-24	
设置信息	与 空流水线	PHP 测试、构建、邮署到主机	1名 北卫主党	2019-09-24	
② 企业设置	JS NodeJS	(+ID12)#	190 8.2 826	2010 00 24	
	PHP PHP	10001111111111111111111111111111111111	刘昊然	2019-09-24	*
	co Go	测试	吴彦祖	2019-09-23	1
	Python		杨超越	2019-09-24	
	20 Mile	ドドア週間は、特別運	刘德华	2019-09-23	
		代码扫描	周杰伦	2019-09-23	
		" 测试		2010 00 24	
			刘昊然	2019-09-24	*
		< 1 2 >			

#### 编辑流水线时,通过任务模版创建单元测试任务

在流水线编排中,可以通过添加任务,选择分类在"测试"下的任务,对应使用的语言来创建单元测试。

	C 20 20 20 30	1421 <del>9</del>	hze 选择任务类型		新的阶段
C→ flow/flow-engine % master	9、请输入任务名称	ф.			● 新約任务
	空模板	测试			
	代码扫描	Maven 单元测试 Maven 单元测试	Gradle 单元测试 Gradle 单元测试	nede NodeJS 单元测试 NodeJS 单元测试	
	测试				
	测试构建	Python 单元测试 Python 单元测试	PHP Codeception 单元 PHP Codeception 单元测试	PHP Unit 单元测试 PHP Unit 单元测试	
	构建 镜像构建 部署	<b>GO 单元测试</b> Go 单元测试			
	代码	测试构建			
	执行命令	<b>Java 单元激试、构建</b> Java单元激试、构建、构建	nede NodeJS 构建测试上传 NodeJS 单元测试、构建、	React 构建测试上传 React 构建测试并上传至 Oss	
		Python 镜像构建测试 Python 镜像构建及单元测试	Python 打包测试 Python 打包及单元测试	<b>Go 构建测试上传</b> Go 构建测试并上传	
		#172 <b>0</b>			

## 7.3. 质量红线

质量红线是 Flow 流水线提供的质量卡点能力, 用于标准化质量标准,当阶段中存在质量项尚未达标的情况下,阻止发布流程进入到下一阶段(环节)。

#### 配置质量红线

质量红线当前支持在测试相关的任务中进行设置,包含:

- 代码扫描能力
- 单元测试

在流水线编辑状态,点击测试相关的任务,点击"添加红线",即可出现红线信息,用于指定当前测试任务必须达到的标准。

以扫描任务为例,可以设定红线如下图:

- Blocker的问题为0
- Major的问题为0

#### • Critial的问题为0

	●新建源	测试	构建		编辑				
					CY 38 T	0 FUN			
l⊷ master	e-racuitest/new	₽ Ģ Java代码		Java构建,打压缩包	Java	安全扫描			
		G Maven # 7			构建命	1 <b>\</b>			0
					3 4 5 7	<pre>www.B.clean.compile_Dmc Dautoconfig.skip # gradle build default cc # ./gradlew build</pre>	wen.test.skip=tr	ue	
					Tota	「问题数」、	小于等于 ~	0	•
					且	Blocker 问题数 ~	小于等于 ~	0	•
					E	Blocker 问题数 ~ Critical 问题数 ~	小于等于 ~ 小于等于 ~	0	•
					E E	Blocker 问题数 ~ Critical 问题数 ~ Major 问题数 ~	小于等于 ~       小于等于 ~       小于等于 ~       小于等于 ~	0	
					E E	Blocker 问题数 ~ Critical 问题数 ~ Major 问题数 ~ Total 问题数 ~	小于等于        小于等于        小于等于        小于等于        小于等于	0	

#### 质量红线生效

在流水线运行状态,会根据对应的质量红线对测试任务进行判断,是否能够通过红线,如果未通过红线,对应的任务将失败。

- aonecloud-pre-rdcuitest/new-	② Java代码扫描 ① 《 1/2 ▶	🕗 Java构建,打压缩包
% master	Java代码规约扫描         Total 问题数大于0         Major 问题数大于0         4         0         0         4           統计效素         紅丝未通过         紅丝未通过         总数         班書         严重         一級	等待中
	○ 重试 更多 ~ 39秒	0秒
	<ul> <li>● Maven単元測试 ① &lt;1/3 →</li> <li>2/3 1 0 67%</li> <li>測试用列 失敗 跳过 通过率</li> <li>② 重试 更多 ~ 1932秒</li> </ul>	

#### 质量红线跳过

考虑在一些特殊的情况下,未通过质量红线的流程也需要继续往下执行,飞流也提供了"跳过"的能力,可以由管理员将红线跳过。



# 8.变量和缓存

# 8.1. 环境变量

定义环境变量是实现流水线过程定制化的一种常见方法,可以在执行过程的任何阶段使用这些变量。

注意:环境变量不可使用横杠符号 -

#### 环境变量的来源

环境变量可以来自以下几种途径:

#### 1. 内置环境变量

环境变量名	说明
PIPELINE_ID	流水线 ID
BUILD_NUMBER	流水线的运行编号,从1开始,按自然数自增
PIPELINE_NAME	流水线名称,比如"前端项目发布"
PROJECT_DIR	运行命令的工作目录,比如" /root/workspace/1084-abc_docker-08191_b0wE"
DATETIME	当前时间戳,比如 2017-06-22-23-26-33
TIMESTAMP	当前时间戳,比如 1581581273232

#### 如果流水线配置了一个代码源时,则会有以下内置环境变量:

环境变量名	说明
CI_COMMIT_REF_NAME	代码库的分支名或者 Tag 名(根据用户运行时选择),比如 master or V1.0
CI_COMMIT_TITLE	最后一次提交的提交信息
CI_COMMIT_SHA	最后一次提交的代码版本的 commit ID: 如2bfb63d779e3648c91950f82d374a25784cdabaf
CI_COMMIT_ID	最后一次提交的代码版本的 8 位 commit ID
CI_SOURCE_NAME	代码源名称

#### 当您的流水线配置了多个代码源时,不同代码源对应的以上环境变量可以通过变量名+数字后缀(第n个代码源)

环境变量名	说明
CI_COMMIT_REF_NAME_n	代码库的分支名或者 Tag 名(根据用户运行时选择),比如 master or V1.0
CI_COMMIT_TITLE_n	最后一次提交的提交信息
CI_COMMIT_SHA_n	最后一次提交的代码版本的 commit ID: 如2bfb63d779e3648c91950f82d374a25784cdabaf
CI_COMMIT_ID_n	最后一次提交的代码版本的 8 位 commit ID
CI_SOURCE_NAME_n	代码源名称

#### 环境变量和代码源对照关系如下图所示:

く 返回 新增环境变量测试	基本信息 流程配置	触发设置 变量和缓存	仅保存保存并运行
流水线源	执行命令	构建	新阶段
<ul> <li>○ …0001fd99b0/Codeup-Demo</li> <li>☆ master</li> <li>○ Codeup-Demo_q8QA</li> <li>⑦ flow-example/spring-boot</li> <li>☆ master</li> <li>○ spring-boot_BLqy</li> <li>+ 添加流水线源</li> </ul>		今       Node.js 构建         今       Node.js 构建	新的任务

#### 2. 流水线运行参数

在流水线编辑页面,可以定义流水线的环境变量,在流水线运行时,可以将环境变量执行过程的任何阶段使用这些变量。 在添加全局变量的时候,选择"运行时设置"选项,可以在流水线运行时进行参数动态配置

新規2       AB       新払値       私用模式       通行曲         AB       第       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1       1	(0)			配置全局变量			×		
新決设置,点击 添加 1 1 test >>> test 2 1 test 2 1 test 2 2 1 1 test 2 2 1 1 test 2 2 1 1 test 2 2 1 1 test 2 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2		参数	默认值		私密模式 ①	运行时设置 🕕	操作		
est >> test1 >> test1 ) ) (文 在)	暂未设置,点击 添加	test	>> test				<b>Ū</b>		
設置(2) ・ 近行値 est >> test   取消 父々	Ψ	test1	>> test1				Û		
P数 >> 运行值 est >> test 取消 保存 est1 >> test1	局变量(2)	● 添加变量					- 1		
est >> test 2011 (% ??	參数 >> 运行值								
est1 >> test1	test >> test					取消	保存		
	test1 >> test1						-		

§(O)	0 -	阶段1		运行参	<b>》</b> 数配置	(1)	×		
暂未设置,点击 添加		<b>B</b> — Ş	参数 testEnvir >>	运行值 testValue		Ļ			
6局变量(2)	0 -		<ul> <li>编辑模式下可修改</li> </ul>			取消	运行		
参数 >> 运行値 testEnvir >> testValue									
ab-password >>									

#### 3. 私密配置项

在应用构建中,通常会需要一些配置项,类似于

1. 依赖系统的 URL

2. 数据库链接用户名密码

配置项会涉及一些私密信息,不适合公开给他人查看。飞流提供了私密模式的全局变量,可以用于保存私密配置。

在流水线编辑页面,可以定义流水线的环境变量,在新建变量时,选择将变量设置成为"私密模式",对应的变量可以作为私密配置项。
对于"私密模式"的全局参数,对应的参数值将会加密不对外显示,并且在对应的执行日志中,也不会显示。

Ē(O)	o	配置全	局变量	×	
	参数	默认值	私密模式 🛈	运行时设置 ① 操作	
暂未设置,点击 添加	testEnvir	>> testValue		□ Ŵ	
	db-password	>>		<b>•</b>	
局变量(2)	③ ③ 添加变量				
参数 >> 运行值					
testEnvir >> testValue				取消保存	
db-password >> ******					

#### 使用环境变量

在你需要使用环境变量的地方通过 \${name} 引用您需要的环境变量。如下图,将配置文件 a.conf 中的 key 的参数值,从"123"更改为变量名为 abc 的环境变量。

2010 人工下品测试		基本信息 流程配置	触发设置 变量和缓存		仅保存保存并
	执行命令		新阶段	编辑 宣	
	<ul> <li>φ ΔΙτά</li> </ul>	执行命令	● 新的任务	内建集群 ① 云文北京构建集群 ④ 万文北京构建集群 ④ 第加少策 ④ 予加少策 ● 少暖名称 泉行命令 ● 現行命令 ● 見行命令 ● 単低名称 泉行命令 ● 単位の令 ● 単位の合 ● 泉行命令 ● ● 泉行命令 ● ● ● 泉 日 ● ● ● 泉 日 ● ● ● ● ● ● ● ● ● ● ● ● ●	/key=\${abc}/" a.conf
				> 构建物上传	

#### 部署组使用环境变量

Flow 经过优化后,新的主机部署不会再对环境变量进行默认编码,如果你的环境变量中不包含特殊字符(空格、引号、分号等),您可以在部署脚本中通过 \${key} 使用环 境变量,无需再进行 Base 64 解码处理。但是如果你的环境变量中包含特殊字符,您将无法正常使用您的环境变量,可参考以下文档进行处理:主机部署—环境变量包含特殊字符

#### 镜像构建参数使用环境变量

当你在镜像构建时,需要将流水线的环境变量作为构建参数时,需要在镜像构建配置的构建参数中,将您的环境变量赋给您的构建参数,如下图所示:

构建	新阶段	编辑 ①
		区域
c0e0001fd99b0/spring-boot O Docker 镜像构建	③ 新的任务	华东1(杭州) ~
ŀ~ master		仓库 ⑦ 😏 11 直接输入
十添加代码源		registry.cn-hangzhou.aliyuncs.com/cpj-space/test
		标签 ⑦
		\$(DATETIME)
		Dockerfile路径 ⑦
		Dockerfile
		Contant Bath
		□ 不使用缓存 ⑦
	C	构建参数 ⑦
	C	自定义 ~ argName = \${abc}
		十 添加参数

在 Dockerfile 中,通过 ARG argName 完成变量引用,如下图所示:

源文件(728.01K) 提交 62 分支	② 标签 合并请求 安全 报表 ⁹ 概览 十	凡 成员 ⑥ i
master $\lor$ Q ₆ +	spring-boot / Dockerfile	⑥ 克隆/下载
spring-boot	更新 Dockerfile	5e33c73d
app-configs	光晕 提交于 9 秒前	
configs		
🕨 🚞 infra	Dockerfile	🖉 🐨 Web IDE …
Src	1 FROM java:8	
⇔ .env	3 ARG argName	
.gitignore	5 RUN echo sargName	
I Dockerfile	6 7 ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom","-jar","/app.jar"]	
Dockerfile-backup		
Dockerfile2		
README.md		
deploy.sh		
c) pointximi		
		6

#### 配置文件中的参数使用环境变量

如果您的配置文件中某个参数,需要通过环境变量进行修改,例如:代码库配置文件 a.conf 中有参数 username ,希望用流水线的环境变量进行替换。可修改配置文件 a.conf 如下,其中将 abc 的参数值配置在流水线的环境变量中:

		A. 成员 @ 设
master $\lor$ Q ₆ +	conf_test / a.conf	② 克隆/下载
conf_test	更新 a.conf	0.0000
tt a.conf	✗ 光暈 提交于现在	00282019
	a.conf	2 1 Web IDE ***
	<pre>3 username = \${abc}</pre>	
	4	
		9

在流水线中选择以下工具,并填写配置文件的源文件路径;目标文件路径非必填,如果填写了目标文件路径,则会生成一个新的配置文件,不填写目标文件路径则直接修改源 文件。

返回 修改配置文件测试	1	基本信息 流程書	2置 触发设置	变量和缓存			仅保存保存并认
					选择任务组		
e17c0e0001fd99b0/conf_test	◎ 新的任务				企业任务组	Q。请输入任务名称	
▶ master					代码扫描	PHP 构建后, 阿里	(1)
十添加代码源					测试构建	工具	载换文件内协环境
I MARINE VEDAR					构建	人工卡点	督换文件中的环境 替换文件中的环境
					镜像构建	人工卡点	<b>一</b> 替换文件中的环境.
					工具	Helm Chart 上 将代码库中的 Hel	Jenkins 任务 触发 Jenkins 任务
					代码	部署	
					执行命令 空模板	<b>主机部署</b> 主机部署	<b>主机部署(无构</b> 使用脚本发布到主体
						Docker部署 主机Docker部署	<b>Kubernetes 发</b> 用于发布 Kuberne
						<b>Kubernetes 镜…</b> 升级 Kubernetes …	<b>Kubernetes 蓝</b> . 基于Istio进行应用。
返回 修改配置文件测试		<b>基本信息 流躍</b> 調	電行機 電影	空量和缓存			仪保存 保存出
返回	新幹段 ク	基本信息 流程前 新阶段	<b>2重</b> 触发设置	交量和緩存		编辑 ①	仅保存 保存并近
返回	新阶段 2	基本信息 流程 新阶段	2章 触发设置	支量和纖存		<b>编辑</b> ① - 替换文件中的环境变量	仅保存 保存并运
返回 修改配置文件测试 e17c0e0001fd99b0/conf_test	新阶级 2	基本信息 流程 新阶段	2章 触发设置	变量和顺存		<ul> <li>編輯 查</li> <li>· 替换文件中的环境变量</li> <li>步调名称</li> </ul>	仅保存 保存并近
返回 修改配置文件测试 e17c0e0001fd99b0/conf_test ( 下 master	新於段 🖉 Ģ • • • • • • • • • • • • • • • • • • •	基本信息 流程道 新阶段	2章 触发设置 新的任务	支量和緩存		編輯 ①           • 替换文件中的环境变量           步骤名称           替换文件中的环境变量	仅保存 保容并近
返回 修改配量文件测试 e17c0e0001fd99b0/conf_test - ▶ master +添加代码源	新阶段 2 第一 第 第 第 一 数 技文件中的环境空量 ● 単 行任务	基本信息 <b>法程</b> 制 新新设	2重 触发设置 新的任务	变量和暖存	(1)	<ul> <li>編輯 ①</li> <li>         · 督換文件中的环境变量         · 步骤名称         · 督換文件中的环境变量         · 资次件器         · 资次件器         ⑦         · ②         ·         ·</li></ul>	仅保存 保存并近
返回 修改配置文件测试 e17c0e0001fd99b0/conf_test ト master + 添加代码源	新阶段 2	基本信息 <b>流程</b> 函 新附段	2重 触发设置 新的任务	变量和维存	1	<ul> <li>編輯 ①</li> <li>              登換文件中的环境变量 步環名称      </li> <li>             登換文件中的环境变量         </li> <li>             歌文件語径 ①         </li> <li>             a.conf         </li> </ul>	仅保存 保存并近
返回 修改配置文件源试 e17c0e0001fd99b0/conf_test ト master + 版加代码源	新阶段 2	基本信息 <b>流程</b> 新尚段	2重 触发设置 新的任务	变量和缓存	1	編輯 ①           • 替换文件中的环境变量           步振名称           替换文件中的环境变量           源文件器径 ⑦           a.conf           目标文件器径 ⑦           b.conf	仅保存 保存并近
返回 修改配置文件測试 …e17c0e0001td99b0/conf_test ト master +添加代码源	新阶段 2 多 ●   ●   ●   ●   ●   ●   ●   ●   ●   ●	5本信息 流程 新約段 ●	2重 触发设置 新的任务	变量和爆存	1	<ul> <li>編輯 ①</li> <li>              ・</li></ul>	仅保存 保存并近
返回 修改配置文件测试 e17c0e0001fd99b0/conf_test ▶ master +- 添加代码源	新阶段 2	5本信息 <b>流程</b> 新前段	2重 触发设置 新的任务	安量和維存	1	<ul> <li>編編 ①</li> <li>              ◆ 替換文件中的环境変量 歩環名際 登換文件中的环境変量 変文件器径             ①</li></ul>	(文保存) (文保存)
返回 修改配置文件源试 e17c0e0001fd99b0/conf_test ト master + 版加代码源	新阶段 2	基本信息 流程 新治役	2重 触发说画 新的任务	变重和缓存	1	編輯 ①           • 替换文件中的环境变量           步振名称           潜決文件中的环境变量           源文件器径 ⑦           a.conf           目标文件器径 ⑦           b.conf           b.conf           步速名称           少成分表示           少波名称           力波名称	( <b>仅保存</b> ) 保存并近
返回 侍改配置文件测试 e17c0e0001fd99b0/conf_test ト master + 添加代码源	新阶段 2 5 • 数技文件中的环境变量 • • 并行任务	5本信息 流程 新商役	2重 触发设置 新的任务	变量和爆存	1	編集 ①	(又保存) 保存并近 (又保存) 保存并近

查看修改效果如下, a.conf 中的 username 被修改为 b.conf 中的 my_name_is_hanmeimei 。注意:由于 Flow 相同任务共享工作区,因此修改配置文件变量步骤需要和您后续使用配置文件的步骤放在同一任务节点中。

<b>最近运行</b> 运行历史		替换文件中的环境变量	×	编辑	鸟 成员	i ≡3
会 法行成Th	构建	执行命令(1s)	下载			
V 221310240	❷ 申请运行环境(25s)	[executionStep begins at [2020-11-20 11:06:51]] [2020-11-20 11:06:51] [INFO] [PUICIN] 开始地行	1			
۶. A		[INFO] PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin				
中前 2020 1	⊘ 克隆代码(3s)	[11:06:52] [SUCCESS] 获取 Credential 成功				
19 10J 2020-1	☑ 流水线缓存(2s)	[11:06:52] [INFO] 使用工作邮控/FOOL/WORKspace/Seawe/FCe1/Coedool103900_Cont_test_ImFt				
时间	◎ 替换文件中的环境变	[11:06:52] [INFO] 系统变量				
	执行命令(1s)	[11:06:52] [INFO] PIPELINE_ID=1184638 [11:06:52] [INFO] PIPELINE NAME=修改配置文件测试				
	<ul><li>(13)</li><li>(13)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li><li>(15)</li></ul>	[11:06:52] [INFO] BUILD_NUMBER=1				
原(1)	w(1+⊥1+(25)	[11:06:52] [INFO] EMPLOYEE_ID=5e71d6f503283833284f91c2 [11:06:52] [INFO] WORK SPACE=/root/workspace				
		<pre>[11:06:52] [INF0] PROJECT_DIR=/root/workspace/Sea4e7fce17c0e0001fd99b0_conf_test_1Mrt</pre>				
安重(1)		[11:06:52] [INFO] PLUGIN_DIR=/root/plugins				
		[11:06:52] [INFO] 准备环境执行上下文准备				
	$\sim$	[11:06:52] [INFO] 执行用户命令				
	(1)	[11:06:52] [H:(Tap V) + Cat a.cont [11:06:52]				
	$\smile$	[11:06:52]				
		[11:06:52] username = \${abc} [11:06:52]				
		[11:06:52] [执行命令] + cat b.conf				
	$\bigcirc$	[11:06:52]				
		[11:06:52] username = my_name_is_hanmeimei				
		[11:06:52]				
		[11:00:52] [SULLESS] 步骤险门队均				

# 8.2. 运行选择变量

Flow 环境变量支持运行选择变量,进入流水线编辑页 -> 变量和缓存,可添加运行选择变量。运行选择变量需设置参数名和选项值,每个变量有一个默认选项值。

	安量	添加	四变量	×		
<b>又里</b> 通用变量组	通过近参数					
缓存	字符3 key					Q. 6 新建变量
	选项		默认值	操作		
	Value1			Ŵ	运行时设置(?)	操作
	value2			Ŵ		
	添加选项					
	运行注			70 284 297 40		Q. 🕒 新建变量
	\$¥	1	1	蚁 湘 ◎ 加		操作
			没有数据			

当你手动运行流水线时,运行选择变量支持下拉选择,无需手动输入。

		运行配直	×		
变量					
通用变量组	代码源	分支/标签			
缓存	m/flow-example/spring-boot	master			Q. 🕒 新建变量
	道田市長	18		行时设置⑦	操作
	进行文里				
	key	value1			
	(1)	value1			0. 6 新建变量
			_	,	
	参数	默认值	选项		操作
	key	value1	value1,value2		01

图2.运行流水线选择变量选项

# 8.3. 通用变量组

通用变量组是企业统一管控的通用环境变量,用企业统一管理,流水线可添加关联 创建变量组之前,先确认自己所属角色是否拥有"变量组创建权限"。 如图所示,点击通用变量组->新建变量组->输入变量组名称和参数键值对->创建。

<ul><li>☆ 我的星标</li><li>② 我参与的</li></ul>		
♀ 全部流水线		
◎ 设置		
企业设置		
(1) 通用变量组		
	with	
	暂无通用变量组 请试试解读通用变量组	

	新建通用变量组	×	
☆ 我的星标			
② 我参与的	<b>容量组</b> 名		
♀ 全部流水线	测试变量组		
② 设置			
企业设置	说明 用工业试动运用亦是4		
通用变量组	用」兩曲的通用支重组		
	参数 默认值	私密模式 ① 操作	
	VAR >>	<b>(</b> )	
	● 添加变量		
		3	
		Y III	
		取消创建	

注意:同一企业中通用变量组不得重名,同一个通用变量组中不能存在名称相同的变量。

#### 在流水线关联变量组

进入流水线编辑页面,点击如图所示位置,选择变量组,点击关联。

x(0)		通用变量组 ×
警末设置、点击 洗加 <u>XX2用交量相</u> 全局交量(0)	私由 源加	新元夫联連用変量組 用数価額的交量価額系有増点 支援
	点击 添加	

注意:

变量优先级:组件输出变量 > 运行变量 > 流水线全局变量 >变量组参数
 同一流水线绑定不同变量组中有变量同名的,后添加的变量组名称生效。

#### 修改变量组

修改变量组之前,先确认自己所属角色是否拥有"变量组管理权限"。

☆ 我的星标	通用变量组					新建变量组
② 我参与的	名称	关联流水线	1	修改人	修改时间	编辑变量组
♀ 全部流水线	测试变量组	测试流水线	đ	<b>***</b> 金镗	2019-12-31	
③ 设置					$\odot$	
企业设置						
通用变量组						
						6
						•

#### 删除变量组

删除变量组之前,先确认自己所属角色是否拥有"变量组管理权限"。

公 459-313     名称     关联流火线     修改人     修改付け     創除交量総       合 全部淡水线     第1支量組     潮试流水线     単 全望     2019-12-31     1     ●       企业设置     通用安量組	☆ 我的星标		通用变量组				新建变量组
□ 王即派小线 ② 设置 企业设置 通用变量组	(2) 我参与的	P	名称	关联流水线	修改人	修改时间	删除变量
③ 设置         企业设置           通用变量组	⇒ 全部流水3	ł	测试变量组	测试流水线	())) ())	全镗 2019-12-3	1 (1)
企业(2重 通用变量组	② 设置						
通用变量组	企业设置						
	通用变量组	1					

## 8.4. 缓存设置

#### 缓存目录

由于 Flow 流水线执行时,每个任务节点都会创建新的构建环境,构建完后会销毁,为解决流水线运行过程中,构建依赖反复下载的问题,提升整体构建效率,Flow 提供了自 定义缓存功能。用户可以自定义自己的缓存目录,实现流水线任务多次运行的缓存共享。

比如在 Nodejs 构建任务执行 npm install 会在项目中生成 ./node_modules , 而缓存在 root/.npm 中。可以将 root/.npm 保存起来方便下次加速下载。

同时,若出现缓存导致的构建失败,您可以对缓存进行清空重置。

Flow 的缓存默认保存 365 天,若无使用则会自动清除。

您可通过 流水线编辑 -> 变量与缓存 -> 缓存,进行缓存目录设置。

く 返回  流水线 2020-06-11		基本信息 流程配置 触发设置 变量和缓存		仅保存	<b>R存并运行</b>
安量 通用变量组	<b>缓存</b> Flow 会将工作区缓存目录下的构	建依赖进行缓存,以提高整体构建速度, <mark>查看文档</mark>			
缓存	缓存目录			😌 添加缓存目录 📩 清理缓存	
	缓存目录	描述	是否开启	操作	
	/root/.m2	maven依赖缓存		₩	
	/root/.gradle	gradle依赖缓存		₩	
	/root/.npm	npm依赖全局缓存		Ŵ	
	target			Ψ.	

目前 Flow 暂不支持流水线同一次运行过程中,任务之间的缓存共享。Flow 的缓存能力主要解决的是任务级别多次运行之间的下载依赖问题。Flow 默认会缓存以下目录,以 下目录均为 Flow 构建环境中默认指定的缓存目录:

管理工具	缓存目录
Maven	/root/.m2
Gradle	/root/.gradle
NPM	/root/.npm
yarn	/root/.yarn
go mod	/go/pkg/mod
其他缓存	/root/.cache

支持设置自定义缓存目录,自定义缓存目录需满足以下规则:

• 不允许填写 "/"、"/root"、"/root/workspace"以及 /root/workspace/ 以下的目录

● 不允许包含 "..",比如 "/root/abc/.."

• 必须为一个合法的文件目录,例如以下目录为非合法文件目录 "/root /%&df af"

若需要将构建过程中工作区 /root/workspace 下某个目录或文件进行缓存,可以将其复制到/root 下其他目录并将其设置为缓存目录。示例如下:

く 返回 自定义缓存流水线		基本信息	<b>秔程配置</b> 触发设置 变量和	口缓存	仅保存	保存并运行
线源	执行命令		新阶段	编辑 亚		×
十 添加流水线源	<b>G B</b>	执行命令	新的任务	* 执行命令		
				「加加」		
				_{少骤石} 称 执行命令		
				执行命令 1 cd /root/cpj	_	2
			C	2 ls		
			(1	4 cd \${PROJECT_DIR 5 echo "aaa" > a.t		
				<pre>6 7 cp \${PROJECT_DIR</pre>		
				环境变量 (?) + 添加		
				任务插件		
				● 添加插件		
〈 返回 自定义缓存流水线		基本信息 汾	荒程配置 触发设置 <b>变量和</b>	1)缓存	仅保存	保存并运行
〈 返回 自定义缓存流水线		基本信息 消	17程配置 触发设置 <b>变量和</b>	四缓存	仅保存	保存并运行
<ul> <li>《 返回 自定义缓存流水线</li> <li>变量</li> <li>通用亦量41</li> </ul>		基本信息 <b>援存</b> Flow 会将不同任务工作区约	就程配置 触发设置 <b>变量和</b> 要存目录下的构建依赖分别进行	<b>口缓存</b> 缓存,下次该任务执行时可快;	<b>仅保存</b> 速下载缓存,以提高整体构	保存并运行
《 返回 自定义缓存流水线 变量 通用变量组 4月5		基本信息 3 援 <b>存</b> Flow 会将不同任务工作区组 建速度, 查看文档	就程配置 触发设置 <b>变量和</b> 要在目录下的构建依赖分别进行	<b>口缓存</b> 缓存,下次该任务执行时可快;	仅保存 速下载缓存,以提高整体构	保存并运行
< 返回 自定义缓存流水线 变量 通用变量组 <b>缓存</b>		基本信息 3 援存 Flow 会将不同任务工作区组 建速度,查看文档 缓存目录	荒程配置 触发设置 变量和	<b>口缓存</b> 缓存,下次该任务执行时可快 <b>④</b> 添灯	仅保存 速下载缓存,以提高整体构 加缓存目录 <b>清理缓存</b>	保存并运行
<ul> <li>2 返回</li> <li>自定义缓存流水线</li> <li>变量</li> <li>通用变量组</li> <li>缓存</li> </ul>		基本信息 3 援存 Flow 会将不同任务工作区组 建速度,查看文档 缓存目录 缓存目录	就程配置 触发设置 变量和     反存目录下的构建依赖分别进行     描述     描述	<b>□緩存</b> 缓存,下次该任务执行时可快) ● 添 是否开启	仅保存 速下载缓存,以提高整体构 加缓存目录 合 清理缓存 操作	保存并运行
《 返回 自定义缓存流水线       安量       通用交量组 <b>缓存</b>		基本信息 3 援存 Flow 会将不同任务工作区线 建速度,查看文档 缓存目录 	就程配置 触发设置 变量和 更存目录下的构建依赖分别进行 描述	<b>□媛存</b> 缓存,下次该任务执行时可快) ● 添 是否开启	仅保存       速下载缓存,以提高整体构       加缓存目录 合清理缓存       操作       ①	保存并运行
<ul> <li>&lt; 返回 自定义缓存流水线</li> <li>交量</li> <li>通用变量组</li> <li>缓存</li> </ul>		基本信息 9 援存 Flow 会将不同任务工作区组 建速度,查看文档 缓存目录 (root/m2 /root/gradle/caches	旅程配置 触发设置 <b>变量和</b> 要存目录下的构建依赖分别进行 描述	<b>□缓存</b> 缓存,下次该任务执行时可快 ● 添 是否开启 ●	仅保存       速下载缓存,以提高整体构       加缓存目录 合清理缓存       操作       位	保存并运行
<ul> <li>( 返回 ) 自定义缓存流水线</li> <li>变量</li> <li>通用变量组</li> <li><b>缓存</b></li> </ul>		基本信息 第 <b>援存</b> Flow 会将不同任务工作区纸 建速度,查看文档 缓存目录 缓存目录 /root/.gradle/caches /root/.ppm	底程配置 触发设置 变量和 更存目录下的构建依赖分别进行 描述	<b>1)緩存</b> 缓存,下次该任务执行时可快 ● 添 是否开启 ● ① ● ① ● ①	(欠保存)       速下载缓存,以提高整体构       加缓存目录 會清理缓存       操作       位       位       位       位	保存并运行
《 返回 自定义缓存流水线       交量       通用交量组 <b>緩存</b>		基本信息 3 基本信息 3	□ 2017年1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1月1日日本1日日本	□緩存 缓存,下次该任务执行时可快力 ● 添加 是否开启 ● ① ● ① ● ① ● ① ● ① ● ① ● ①	仅保存       速下载缓存,以提高整体构       加缓存目录 查 清理缓存       操作       並       並       並       並       近       近       近       近	保存并运行
《 返回 自定义缓存流水线 变量 通用变量组 缓存		基本信息 が 基本信息 が また。 また。 また。 また。 基本信息 が また。 また。 また。 また。 また。 また。 また。 また。	就程配置 触发设置 变量和 要存目录下的构建依赖分别进行 描述	<b>□媛存</b> 缓存,下次该任务执行时可快	使 取 観 使 存 、 以 提 高 整 体 构 加 観 存 目 家 合 清理 3 健 存 通 し 低 価 し 低 低 低 低 低 低 低 低 低	保存并运行
<ul> <li>く 返回</li> <li>自定义缓存流水线</li> <li>変量</li> <li>通用変量组</li> <li>4 緩存</li> </ul>		基本信息 が 基本信息 が まま度、童看文档 量速度、童看文档 「coot/.m2 1000000000000000000000000000000000000	就程配置 触发设置 变量和     模存目录下的构建依赖分别进行     描述     描述     目示     日本     日本	20緩存 編集 の 20緩存 編集 の 20緩 ク 20 20 20 20 20 20 20 20 20 20 20 20 20	(欠保存)       速下载缓存,以提高整体内       加缓存目录 合清理缓存       操作       ①       ①       ①       ①       ①       ①       ①       ①       ①       ①       ①       ①       ①       ①       ①       ①       ①       ①       ①       ①       ①       ①       ①       ①       ①       ①       ①       ①       ①       ①       ①       ①       ①	保存并运行
<ul> <li>く 返回</li> <li>自定义缓存流水线</li> <li>変量</li> <li>通用交量组</li> <li>援存</li> </ul>	1	基本信息 7 基本信息 7 Flow 会将不同任务工作区组 建速度, 查看文档 建速度, 查看文档 (root/.m2 (root/.gradle/caches 14 (root/.gradle/caches 14 (ro	□ 2011 100 100 100 100 100 100 100 100 10	2 (2 の) (2 0 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0 0) (2 0	で         な         の         彼         存         日         、         、         、	保存并运行
《 返回 自定义缓存流水线       安量       通用交量组       銀存	1	基本信息 が 基本信息 が まなにのでのでのでは、 してののでは、 基本信息 が またのでのでは、 またのでのでは、 またのでのでは、 してのので、 またのでので、 またのでので、 またのでので、 またのでので、 またのでので、 またのでので、 またのでので、 またのでので、 またのでので、 またのでので、 またのでので、 またのでので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 またので、 までので、 までので、 までので、 まのので、 まのので、 まのので、 まのので、 まのので、 まのので、 まのので、 まのので、 まのので、 まのので、 まのので、 まのので、 まのので、 まのので、 まのので、 まのので、 まのので、 まのので、 まのので、 まのので、 まのので、 まのので、 まのので、 まのので、 まのので、 まのので、 まのので、 まのので、 まのので、 まのので、 まのので、 まのので、 まのので、 まのので、 まのので、 まのので、 まので、 まので、 まのので、 まので、 まので、 まので、 まので、 まので、 まのでので、 まのでので、 まので、 まのでので、 まので、 まのでので、 まので、 まので、 まので、 まので、 まので、 まので、 まので、 まので、 まので、 まので、 まので、 まので、 まので、 まので、 まので、 まので、 まので、 まので、 まので、 まので、 まので、 まので、 まので	就程配置 触发设置 变量和 要存目录下的构建依赖分别进行 描述	■ 螺存,下次该任务执行时可快) ② 源7 ② ほのでの ② ① ③ ① ③ ① ③ ① ③ ① ③ ① ③ ① ③ ① ③	使下载缓存,以提高整体构 加緩存目录 合清理缓存 操作 ・ ・ ・ ・ ・ 、 、 、 、 、 、 、 、 、 、 、 、 、	保存并运行
く返回       自定义缓存流水线         変量       通用变量组         緩存	1	選本信息 が 選本信息 が	战灾设置 变量和     受量和     受量和     受付日录下的构建依赖分别进行     描述     描述     日示     日示    日示     日示     日示     日示     日示	Q緩存 緩存,下次该任务执行时可快) ② 添 是否开启 ○ ① ○ ① ○ ① ○ ① ○ ① ○ ① ○ ① ○ ①	速下载緩存,以提高整体物 加緩存目录 高清理緩存 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一	保存并运行

部 🌹 首	i页 > 自定义缓存流水线 ▼		🕖 🕂 🖉 🕐 🚺
最近运:	执行命令		× 请 ≡菜
2 🕑 运行	构建	执行命令(2s)	下载
	✓ 申请运行环境(3s)	[executionStep begins at 2022-04-18 10:59:44]	
也发信息		<pre>[INF0] PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin</pre>	
	☑ 流水线缓存(1s)	[10:59:44] [SUCCESS] 获取 Credential 成功	
F始时间	● 执行命令(2c)	[10:59:44] [WARNING] 未指定工作路径,使用默认路径/root/workspace/code	
	• (7(1) 4) (23)	[10:59:44] [INFO] <u></u> 775 (10-1731311)	
持续时间	绥仔上传(2S)	[10:59:44] [INFO] PIPELINE NAME=自定义缓存流水线	
- 41-		[10:59:44] [INFO] BUILD_NUMBER=2	
「物		[10:59:44] [INF0] EMPLOYEE_ID=5e71d6f503283833284f91c2	
		<pre>[10:59:44] [INF0] WORK_SPACE=/root/workspace</pre>	
		<pre>[10:59:44] [INF0] PROJECT_DIR=/root/workspace/code</pre>	
		[10:59:44] [INFO] PLUGIN_DIR=/root/plugins	
		[10:59:44] [INF0] BUILD_JUB_ID=53832096 [10:50:44] [INF0] 堆名环境地德上国文堆名	
		[10·59·44] [INFO] 准备环境为[]上下文准备 [10·59·44] [INFO] 执行田户命令	
		<pre>[10:59:44] [User Command] + cd /root/cpj</pre>	
	(	1) [10:59:44] [User Command] + ls	
		[10:59:44] a.txt	
		<pre>[10:59:44] [User Command] + cd /root/workspace/code</pre>	
		[10:59:44] [User Command] + echo aaa	
		<pre>[10:59:44] [User Command] + cp /root/workspace/code/a.txt /root/cpj [do co_uil [cueses]]</pre>	
		[10:59:44] [SUCCESS] 步骤运行成功	

```
NPM 构建时,使用以下命令安装依赖包,优先从本地缓存获取依赖包:
```

```
# 构建中优先使用缓存中
```

npm install --prefer-offline --no-audit

yarn 构建时,使用以下命令安装依赖包,优先从本地缓存获取依赖包:

# **构建中优先使用缓存中** yarn install --prefer-offline

#### 私有构建集群缓存配置

当你使用公共构建集群时,所有的构建依赖均会缓存在云端,每次构建时需要从云端拉取缓存。

当你使用私有构建集群时,您可以选择:

1本地缓存,此时所有的依赖文件会存储在私有构建集群上,直接使用本地缓存,无需再从云端拉取。此功能可以解决私有构建机和云端因为网络问题导致的构建缓慢。
 2 云端缓存,每次构建会下载云端的构建缓存文件。主要解决构建集群中存在多个构建机时,每次构建任务可能被调度到不同的构建机,通过云端缓存可以重复利用缓存的依赖文件。

、返回    流水线 2022-01-04	基本信息 流程配	置 触发设置 变量和缓存		仅保存	
变量通用变量组	<b>缓存</b> Flow 会将不同任务工作区缓存目 建速度, 查看文档	录下的构建依赖分别进行缓存,	下次该任务执行时可快速下载缓存	,以提高整体构	
缓仔	缓存目录 ● 添加缓存目录 壹 清理缓存				
	缓存目录	描述	是否开启	操作	
	/root/.m2			Ŵ	
	/root/.gradle/caches			Ŵ	
	/root/.npm			Ŵ	
	/root/.yarn			Ŵ	
	/go/pkg/mod			Ŵ	
	/root/.cache			Ŵ	
	<ul> <li>私有构建集群缓存配置</li> <li>本地缓存(所有依赖文件均约)</li> <li>云端缓存(每次构建会下载云)</li> </ul>	会存储在私有构建集群上) ( 5端的构建缓存)	1		
# 9.通知 9.1. 使用 Webhook 插件发送通知

用户可以在流水线中配置 Webhook 通知插件,为指定地址推送流水线阶段任务运行信息。

### 在流水线中配置 Webhook 通知插件

编辑流水线的任务,在任务插件中选择 Webhook 通知,并填写 Webhook 地址和选择运行时机。如下图所示。注意 Webhook 地址必须公网可访问。

			1末 1	序 保存并运行 X
▼ 源(1)	+ 构建	新阶段	编辑	×
S //law exemple/enving he			→ Java 构建	
> master	日 Java 构建上传	◎ 新的任务	> 构建物上传	0
▼ 空景(1)			任务插件	
			중加插件	
参数 >> 运行值			~ Webhook通知插件	0
abc >> 123			源性实际	
			Webhook通知播件	
			TO OUT IN OUT OWN AND IT	
			webhook地址	
			运行时机	
			失敗 ×	~
			任务输出	
			@ 制品名称.default\${BUILD_NUMBER}	
			删除任务	9
₽ 触发设置			该任务将被彻底删除,注意这是不可逆操作,任务下所	「有数据将会删除
ງ Webhook 请求一律为 F	rost 请求,任务状态信息的	Payload 示例如下。		
) Webhook 请求一律为 F ": 'task', n": 'status', : { elineId": 183, elineName": 'test pipe	Yost 请求,任务状态信息的 line',	Payload 示例如下。		
9 Webhook请求一律为 F ": 'task', n": 'status', : { elineId": 183, elineName": 'test pipe geName": ' <b>构建</b> ',	Yost 请求,任务状态信息的 line',	Payload 示例如下。		
) Webhook 请求一律为 F ": 'task', n": 'status', : { elineId": 183, elineName": 'test pipe geName": '拘建, kName": 'java构建,	fost 请求,任务状态信息的 line',	Payload 示例如下。		
<pre>9 Webhook 请求一律为 F ": 'task', n": 'status', : { elineId": 183, elineName": 'test pipe geName": 'pava构建', ldNumber": 19, </pre>	fost 请求,任务状态信息的 line',	Payload 示例如下。		
) Webhook 请求一律为 F ": 'task', n": 'status', : { elineId": 183, elineName": 'test pipe geName": 'd建', kMame": 'java构建', ldNumber": 19, tusCode": 'jacf成功'	Yost 请求,任务状态信息的 line', \ UNKOWN, RUNNING, WAITI	Payload 示例如下。 NG, CANCELLING, CANCELED, FIN	ISH, SUCCESS, FAIL, SKIP	
<pre>) Webhook 请求一律为 F ": 'task', n": 'status', : { elineId": 183, elineName": 'test pipe geName": 'java构建, ldNumber": 19, tusCode": 'SUCCESS', \ tusName": '运行成功, elineUrl": 'https://rd</pre>	Post 请求, 任务状态信息的 line', \ UNKOMN, RUNNING, WAITI c.alivun.com/ec/pipeline	Payload 示例如下。 NG, CANCELLING, CANCELED, FIN ss/156539?build=19',	ISH, SUCCESS, FAIL, SKIP	
<pre>J Webhook 请求一律为 F ": 'task', n": 'status', : { elineId": 183, elineName": 'test pipe geName": 'java构建', ldNumber": 19, tusCode": 'SUCCESS', \ tusName": '运行成功', elineUrl": 'https://rd sage": '[test pipeline</pre>	Yost 请求, 任务状态信息的 line', \ UNKOWN, RUNNING, WAITI c.aliyun.com/ec/pipeline ]流水线阶段 构建]任务[java	Payload 示例如下。 NG, CANCELLING, CANCELED, FIN ms/156539?build=19', 构建]运行成功'	ISH, SUCCESS, FAIL, SKIP	
) Webhook 请求一律为 P ": 'task', n': 'status', : { elineId": 183, elineName": 'test pipe geName": 'java构建', ldNumber": 19, tusCode": 'SUCCESS', \ tusName": '运行成功', elineUrl": 'https://rd sage": '[test pipeline	Post 请求, 任务状态信息的 line', \ UNKOWN, RUNNING, WAITI c.aliyun.com/ec/pipeline ] <b>流水线阶段[构建]任务</b> [java	Payload 示例如下。 NG, CANCELLING, CANCELED, FIN us/156539?build=19', 构建]运行成功'	ISH, SUCCESS, FAIL, SKIP	

"commitId": 'master', "privousCommitId": 'ddddd', } ], "globalParams": [

```
globalrarams : [
    {"key": 'test', "value": 'test1'},
    {"key": 'test2', "value": 'test2'}
]
}
```

## 9.2. 钉钉机器人发送群消息

用户可以在流水线任务中配置钉钉群通知插件,为指定钉钉群推送流水线运行信息。

## 添加钉钉群机器人

通过"群设置"->"添加机器人"->选择自定义 webhook 机器人。

消息推送:	开启	
Webhook:	https://oapi.dingtalk.com/robot/send?a 复制	重置
	* 请保管好此 Webhook 地址,不要公布在外部网站上,泄露4 使用 Webhook 地址,向钉钉群推送消息 查看文档	有安全风险
安全设置 🕝 说明文档	☑ 自定义关键词	
安全设置 🥝 说明文档	✓ 自定义关键词 流水线	
安全设置 🛛 说明文档	<ul> <li>✓ 自定义关键词</li> <li>流水线</li> <li>④ 添加(最多添加 10 个)</li> </ul>	
* 安全设置 @ 说明文档	<ul> <li>✓ 自定义关键词</li> <li>流水线</li> <li>◆ 添加(最多添加 10 个)</li> <li>✓ 加签</li> </ul>	

#### 可参考以下文档:

https://developers.dingtalk.com/document/robots/custom-robot-access/ 复制 webhook 地址, 并配置到流水线钉钉插件中。

#### 囗 注意

- 钉钉群机器人必须开启安全设置:
  - 若开启"自定义关键词",请填写"流水线"。
  - 若开启"加签",请在记录保存加签密匙,并配置到流水线的钉钉插件中。

## 流水线任务配置钉钉群通知插件

在流水线任务节点中,选择添加钉钉通知插件。

く 返回 预发通知插件	基本信息 流程配	置 触发设置 变量和级	爱存 仅保存 保存并运行
流水线源	执行命令	新阶段	编辑 ŵ
flow-example/spring-boot		● 新的红泉	● 添加插件
		WINJIT23	→ 钉钉机器人通知插件
>			2 如何配置钉钉通知消息, 查看帮助
			插件名称
十 添加流水线源			钉钉机器人通知插件
			webhook地址
			https://oapi.dingtalk.com/robot/send?access_token=2745
			加签秘钥 ②
			SECa5a625d1edfe622b29ad744bf4848d02b4e65fc95;
			运行状态 ⑦
			成功 失敗 🗸
			通知内容
			运行状态 流水线名称 流水线环境 流水线标签 >

- 1. 复制填写钉钉机器人 webhook 地址。
- 2. 若钉钉机器人开启了"加签",复制填写加签密匙。
- 3. 选择运行状态,当该流水线的任务进入选择状态后,会触发消息通知。
- 4. 选择通知内容,所选的通知内容会自动添加至推送消息中。
- 5. 支持填写自定义内容,支持引用环境变量,如\${Cl_COMMIT_TITLE},支持钉钉Markdown语法,查看帮助
- 6. 支持@手机号,多个手机号(钉钉手机号)之间以,分隔,支持@all(输入all即可)。

### 触发流水线运行

触发流水线运行,当流水线任务进入通知插件配置的运行状态时,触发消息通知。

offet I	
@所有人	
流水线: 预发通知插件	
流水线坏境: 日常坏境	
流水线标签:标签1	CC 141 18 km ch sta
流水线备注:	所远遇知内谷
执行人:光晕	
流水线阶段:执行命令	
流水线任务:执行命令	
运行状态: 👗 运行失败	
哈哈哈	
commit信息: blade_198	6(aliyun_057966) add
commit信息: blade_198 spring-boot.release file.	6(aliyun_057966) add 自定义内容
commit信息: blade_198 spring-boot.release file. cpj:????	6(aliyun_057966) add 自定义内容

# 9.3. 企业微信机器人发送群消息

用户可以在流水线任务中配置企业微信群通知插件,为指定企业微信群推送流水线运行信息。

### 添加企业微信群机器人

通过"群设置"->"添加群机器人"->"新建机器人"。

M     D  )T		×
:D]	已添加 lalaa,配置Webhook地址后可推送消息到群 Webhook地址: https://qyaein.qc,.:omin/webhook/send?key= 8d9deb64-47: 复制地址 配置说明	

复制 webhook 地址,并配置到流水线企业微信插件中。

### 流水线任务配置企业微信群通知插件

在流水线任务节点中,选择添加钉钉通知插件。

く 返回 预发通知插件	基本信息 流程配。	置 触发设置 变量和线	缓存 仅保存 保存并运行
流水线源	执行命令 🧷	新阶段	编辑 ⑪ (2011)
⑦ flow-example/spring-boot № master ① flow-example_spring-bo ↓添加请水线漂	少         执行命令           ●         并行任务	新的任务	插件名称          企业微信群通知         webhook地址         https://qyapi.weixin.qq.com/cgi-bin/webhook/send?key=c         运行状态 ⑦         成功         通知内容         运行状态 氮水线标签 氮水线标签 氮水线备注         自定义内容 ⑦         哈哈哈 commit值息: \${Cl_COMMIT_TITLE} 脑斑斑 cpi:3{cpi}         weize_id_@ize20

- 1. 复制填写企业微信机器人 webhook 地址。
- 2. 选择运行状态,当该流水线的任务进入选择状态后,会触发消息通知。
- 3. 选择通知内容,所选的通知内容会自动添加至推送消息中。
- 4. 支持填写自定义内容,支持引用环境变量,如\${CI_COMMIT_TITLE},支持钉钉Markdown语法,查看帮助
- 5. 支持@user_id,多个手机号(企业微信 user_id)之间以,分隔,支持@all(输入all即可)。

#### 🗘 注意

```
通过企业微信管理后台 -> 通讯录获取用户 user_id
```

○,企业微信					API文档   联系客服   退出
首页	通讯录	应用管理	客户联系	管理工具	我的企业
Q. 搜索成员、部门	+ 成员详情				
向团队	: 《返回	编辑 置顶 禁用 激	除		
	X	₩5: Chen. ung.u	)		
	手机:	15 ····································			
	座机:	未设置			
	邮箱:	未设置			
	微信:	C74			
	地址:	未设置			
	部门:	1			
	职务:	未设置			
	身份:	普通成员			
	微信插件:	未关注			
	对外信息				

## 触发流水线运行

触发流水线运行,当流水线任务进入通知插件配置的运行状态时,触发消息通知。

flow测试机器人BOT

Flow 流水线消息通知	
@all	
流水线: 预发通知插件	
流水线环境:	
流水线标签:	所洗诵知内容
流水线备注:	
执行人: 光晕	
<b>流水线阶段</b> :执行命令	
<b>流水线任务</b> :执行命令	
运行状态: 🗸 运行成功	
哈哈哈	自定义内容
anna the blade 1006 (aliana	057066) add agring beat release file
commiti言思: blade_1986(allyun	_057900) add spring-boot.release file.
啦啦啦	

# 9.4. 飞书机器人发送群消息

用户可以在流水线任务中配置飞书群通知插件,为指定飞书群推送流水线运行信息。

添加飞书群机器人

通过"群设置"->"群机器人"->"添加机器人"->选择自定义机器人。

<		×
第二步: 配置 w	rebhook	2
复制以下地址,在	外部系统中完成 webhook 设置	Ш
webhook 地址	https://open.feishu.cn/open-apis/bot/v2/hook/b01e7a 复制	
	请保管好此 webhook 地址。不要公布在 Github、博客等可公开查阅的 网站上。地址泄露后可能被恶意调用发送垃圾信息	
安全设置	□ 自定义关键词 ⑦	Ш
	□ IP 白名单 ⑦	
	□ 签名校验 ⑦	
设罟	□ 仅群主和添加者可编辑. 移除此机器人	
	完成	

可参考以下文档:

https://open.feishu.cn/document/ukTMukTMukTM/ucTM5YjL3ETO24yNxkjN?lang=zh-CN

复制 webhook 地址,并配置到流水线飞书插件中。

#### 🗘 注意

- 飞书群机器人可以选择是否开启安全设置:
  - 若开启"自定义关键词",请填写"流水线"。
  - 若开启"加签",请在记录保存加签秘匙,并配置到流水线的飞书插件中。

## 流水线任务配置飞书群通知插件

### 在流水线任务节点中,选择添加飞书群通知插件。

(法源 执行命令 新阶段 編4 ① 、 (本) fow-example/spring-boot	く 返回   预发通知插件	基本信息 流程配	置 触发设置 变	量和缓存		仅保存保存并	运行
10w-example/spring-boot       今       执行命令       新的任务         ト master       1       通知合意       道智報助         市场加流水线源       「市場通知法会社会社」       1         中添加流水线源       1       1         1       1       1         1       1       1         1       1       1         1       1       1         1       1       1         1       1       1         1       1       1         1       1       1         1       1       1         1       1       1         1       1       1         1       1       1         1       1       1         1       1       1         1       1       1         1       1       1         1       1       1         1       1       1         1       1       1         1       1       1         1       1       1         1       1       1         1       1       1         1       1	(线源 执行命令		新阶段	绢	輪輯 ⑪		×
	flow-example/spring-boot ト master ☐ flow-example_spring-bo 十添加流水线派		新的任务	1	<ul> <li>▼ 北邦通知</li> <li>③ 如何配置飞书通知消息,查看帮助</li> <li>酒牛名称</li> <li>飞书群通知</li> <li>webhook地址</li> <li>https://open.feishu.cn/open-apis/bu</li> <li>加签秘钥 ⑦</li> <li>若飞书机器人安全设置开启加签,请</li> <li>运行状态 ⑦</li> <li>成功 失敗</li> <li>通知内容</li> <li>运行状态 氮水线系统 氮水线系统</li> </ul>	h bt/v2/hook/1b1586c8 填写加蓝秘钥	

- 1. 复制填写飞书机器人 webhook 地址。
- 2. 若飞书机器人开启了"加签",复制填写加签秘匙。
- 3. 选择运行状态,当该流水线的任务进入选择状态后,会触发消息通知。
- 4. 选择通知内容, 所选的通知内容会自动添加至推送消息中。
- 5. 支持填写自定义内容,支持引用环境变量,如\${CI_COMMIT_TITLE}。
- 6. 支持@user_id, 多个user_id (飞书user_id) 之间以,分隔,支持@all(输入all即可)。

#### 🗘 注意

通过飞书管理后台 -> 成员与部门获取用户 user_id

>				0	1 防鹏杰的团队 创建人
l→	组织架构 / 成员与部门				
â	成员 部门				
8	Q 搜索成员和部门 ←	■ 111111111111111111111111111111111111		成员详情	〈上一个 下一个〉   >
1	陈 师 防团队 :	帐号状态 全部     ∨		Y y==>e_r-	
88	As 测试部门	团队中尚有1人未使用飞书 第	送提醒   导出	姓名	
\$	+ 新建部门 28 管理部门	□ 姓名	帐号状态	yanmei	
$\bigtriangledown$		yanmei	⊘ 正常	用户 ID ⑦ 40a23 1	
ф		1190001453	<ul> <li>正常</li> </ul>	手机号码	
ō		(1) (1) (1) (1) (1) (1) (1) (1) (1) (1)	◎ 未激活 发送提醒	+86 136075	
0				部门	
$\leq$				性别	
<b>•</b> 0				保密	
0		○ 仅展示部门的直属成员		回 操作离职	… 查看详情 编辑

## 触发流水线运行

-

触发流水线运行,当流水线任务进入通知插件配置的运行状态时,触发消息通知。

日正义	《机器人 机器人 單 迪过Webnook時日定又服务的消息推达主 6节
Flo	w 流水线消息通知 所有人
流7.	K线: 预发通知插件(https://
流7.	K线环境:日常环境
流7.	K线标签:标签1
流7.	大线备注: 所选通知内容
执行	テ人: 光晕
流7.	K线阶段:执行命令
流7.	K线任务:执行命令
运行	う状态: 🗙 运行失败
哈哈	合哈
con	nmit信息: blade_1986(aliyun_057966) add spring-boot.release file.
啦呐	拉啦 自定义内容
cpj	?????
cpi	123:dddddddddd

# 10.企业管理 10.1. 权限管理

## 10.1.1. 企业权限与角色管理

## 企业权限

企业管理员或企业拥有者可以点击「企业设置」->「角色设置」中查看和设置每个角色的具体权限,每个企业也可以根据需要,自行配置和增加相应的角色和权限。

工具箱	角色设置		新建角色
通用变量组			
示签管理		置理员	→ 成员 默认角色 ヘ
服务连接管理			权限设置
主机组管理	▲ 外部成员 ✓	abc	→ 新建角色 修改角色权限
Kubernetes集群管理			□ 删除角色
<b></b>			流水线管理
壬务组管理			<ul> <li>创建流水线</li> <li>所有管理权限</li> <li>查看所有流水线</li> </ul>
步骤管理			
勾建集群管理			✔ 创建部署组 □ 所有管理权限
			■ 服务连接管理
<b>管理员设置</b>			保存
角色设置			

### 主要是针对企业的权限进行约束。权限表如下:

权限点	子权限点	说明		
	新建角色			
权限限制	修改角色权限	在企业设置中对角色进行管理		
	删除角色			
	创建流水线			
流水线管理	所有管理权限	对流水线的操作权限		
	查看所有流水线			
十	创建主机组	거 : 1 / 4 / 슈 1용 // 보기 阳		
工作也目述	所有管理权限	AJ <u>エ つし-21</u> D 31来   F 1ス 42		
服友授权等进	创建服务授权	对际田子的呢条语权(加 FCC OSC笔)的場件权限		
	所有管理权限	אאריז אנט (ארכט, או אריז אנט אינאראמנים אבייינא		
	创建流水线模板			
流水线模板管理	所有管理权限	对 <mark>流水线模板管理</mark> 的操作权限		

权限点	子权限点	说明	
亦是但範囲	创建变量组	动逐甲亦是约约姆作切阻	
又里地目还	所有管理权限	スプレイン 単立 ロンボート 大阪	
行这个米德国	创建标签分类		
你业儿子自注	所有管理权限	או ארו ו אננטיי נו יערוו גע	
4.7次 節 頂	创建标签	对卡尔茨价语。作物,	
你业目还	所有管理权限	אואגרין אנני ענוגע	
步骤管理	所有管理权限	对 <mark>步骤管理</mark> 得操作权限	
红冬得鲸油	创建任务组管理	对任务组管理得操作权限	
江方地目述	所有管理权限		
約 2季 年 聖 餘 2用	创建构建集群管理	과 상 2분 수도 환성 이 응고 사는 17 18	
19姓禾叶目廷	所有管理权限	ער אדי דין אדע אין אדער איז אין אדער איז אין איז אין איז אין איז	
企业公钥管理	所有管理权限	配置代码源时,可以使用或重置企业公钥	
企业Maven配置管理	所有管理权限	在企业设置配置自定义 Maven 配置	

## 10.1.2. 流水线成员权限

## 流水线成员主要分为三类角色:

- 1. 拥有者
- 2. 管理员
- 3. 成员

最近运行 运行历史			() 运行	〒 1 ☆编辑→ 凡成员 … 菜
#1 😵 运行失败	黔段1	阶段2	成员 shilei leilei shi@gmail.com	◆ 添加 所有权限
aonecloud-pre-rdcuitest/new- k master -0- commit	<ul> <li>Java代码扫描</li> <li>G 运行失敗,请查看日志!</li> </ul>	<ul> <li>java构建,打压组</li> <li>与 等待中</li> </ul>	都夕测试1 iuxi.rf@alibabaine.com	→ 运行权限 所有权限
	□ 日志 ④重试 20秒			运行权限 > 查看
	<ul> <li>Maven単元測试</li> <li>              G行失敗,请宣看日志!      </li> </ul>			移除 移交
	□日志 ④重试 18秒			
运行人 💮 shilei				
运行时间 2019-08-30 17:54:25				
持续时间 20秒				

具体权限表如下:

	拥有者权限	所有权限	运行权限	查看权限
查看流水线	<b>√</b> ₽	✔ ₽	✔ ₽	<b>√</b> ₽
编辑流水线	<b>√</b> ₽	✔ ₽	×	×
删除流水线	<b>✓</b> ₽	<b>√</b> ₽	×	×
运行流水线	<b>✓</b> T	✔ ₽	✔ ₽	×
添加成员	<ul><li>✓ Ţ</li></ul>	<b>√</b> ₽	×	×

## 10.1.3. 主机组成员权限

主机组主要分为两类角色:

1. 管理员

2. 使用者

部署管理员或企业管理员可以在「主机组管理」->「编辑主机组」->「成员权限」中添加相应的成员,并且为相应的成员指定对应的角色。

		4	编辑主机组 — — — — — — — — — — — — — — — — — — —	
☆ 我的星标	基础资料	0 查找成员	A 法的成员	排序 🗸 🕒 新建主机组
A、我参与的	主机列表		添加成员 ※加成员	创建时间
♀ 全部流水线	成员权限	<u> 勝杰</u> 陈鹏杰		2010 12 21 52 197
A. II. Marm		程號 程潇	Q。请输入关键字 已选 0	2019-12-31
企业管理			□ = = 心 木兰心	2019-12-06 🗹 🖞
◎ 通用受重组			加久 柳夕	
■ 土机珇管堆			□ 🍑 金镗	
			🗆 🥰 刘玄	
			- In TE	
			确定	
			取消 保存	

### 具体角色权限表如下:

	管理员	使用者
使用主机组	<b>√</b> ₽	<b>√</b> ₽
编辑主机组	<b>√</b> ₽	×
删除主机组	<b>√</b> ₽	×
添加成员	<b>√</b> ₽	×

## 10.1.4. 流水线分组权限

流水线分组成员主要分为三类角色:

```
1. 所有权限
```

2. 运行权限

#### 3. 查看权限

☆ 我的流水线	۲	分组1		80	新建流水线	o° ≞† ,	
全部流水线		二 流水线名称	成员列表	Q、 <table-cell-rows> 新增成员 —</table-cell-rows>	运行人	创建人	最近运行时间3
认线分组	+	流水线1	测试部门	默认为 查看 ~		🗙 光晕	
分组1			光晕 成员隶属于: 测试部门	所有权限 ~			
分组2			smytest 成员隶属于: 测试部门	查看 >			
① 未分组流水线							

#### 具体权限表如下:

	所有权限	运行权限	查看权限
查看分组下流水线	<b>√</b> ₽	<b>✓</b> ₽	<b>√</b> ₽
编辑分组下流水线	<b>√</b> ₽	×	×
删除分组下流水线	<b>√</b> ₽	×	×
运行分组下流水线	<b>√</b> ₽	<ul> <li>✓ T</li> </ul>	×
添加分组下流水线的成员	<b>√</b> ₽	×	×
添加分组的成员	<b>√</b> ₽	×	×

部分权限规则:

- 若分组上设置了某个成员的权限,而分组下的流水线上未设置该成员权限,该成员对流水线的权限继承于分组。
- 若分组上设置了某个成员的权限,分组下的流水线上也设置该成员权限,该成员对流水线的权限 根据分组权限和流水线权限 的大小,取最大权限范围。
- 对流水线拥有所有权限,才可移动该流水线至某个分组。
- 拥有流水线分组的所有权限,可以添加分组成员/在分组下创建流水线/将流水线移动至该分组。

## 10.2. 流水线模板管理

企业成员在创建流水线时,可以选择流水线模板。Flow 提供了一些预置的流水线模板,帮助你快速的创建自己的流水线;同时,Flow 支持创建企业模板,用户可个性化定制 属于自己企业的流水线模板,方便企业内部成员创建流水线。

		选择流水线模板	×	
☆ 我的星标	■ 企业模板			₩序 ∨ 🛛 勤建流水
▲ 找参与的		III 企业模板	+	- 最近运行Ⅰ 〔 ³ 〕
⇒ 王部流水线	🎍 Java	① 企业模板 · 测试企业模板		- 🖸 🕁
企业管理	php PHP			2020-0 5%
& 通用变量组	Node.js	Jenkins 任务		2020-0
□ 标签管理	GO GO			2019-12 📝 🚖
□ 主机组管理	Python	🚽 Java · 构建、部署到阿里云ECS/目有主机		2019-12 🗹 🌟
☑ 流水线模板管理	其他	构建 部署		
⊖ 任务组管理				2019-12 📝 🚖
123 步骤管理		🛓 Java ·镜像构建,发布到Kubernetes集群/阿里云容器服务		2019-12 🛛 📌
A. 6.11.10.00		鏡像构建 部署		
193 企业设置				
			取消创建	

## 企业流水线模板

## 创建企业流水线模板

用户可通过两种方法点击创建企业流水线模板。

方法一: 创建流水线选择流水线模板时, 可快速创建企业流水线模板。

		选择流水线模板	$\times$		
☆ 我的星标				非序 ~ 🕒 新建流水线	
▲ 我参与的		III 企业模板	+	最近运行	
♀ 全部流水线	👙 Java				
	php PHP	💿 企业模板 · 测试企业模板		- U V	
企业管理	Node is	Jenkins 任务		2020-0 📝 🚖	
& 通用变量组	W Node.js			2010 15 52	
□ 标签管理	GO GO	👙 Java · 构建、部署到阿里云ECS/自有主机		2019-12	
曰 主机组管理	Python	1016 MILE		2019-12 🛛 🛧	
🗇 流水线模板管理	… 其他	何進即省		2010-11 52 +	
☑ 任务组管理				2013-12	
123 步骤管理		Java · 镜像构建,发布到Kubernetes集群/阿里云容器服务		2019-12 📝 🌟	
		镜像构建 部署		-	
② 企业设置					
		Ę	(消)创建		
					D

方法二:在企业管理 -> 流水线模板管理 中,创建企业模板。

···· 🖻 飞流		🕑   😳 📀 📾
☆ 我的星标	流水线模板管理	● 新建企业模板
戌、我参与的 ↓ 全部流水线	武 企业模板 Java PHP Node.js Go Python 其他	
企业管理	② 企业模板 · 测试企业模板	
& 通用变量组	Jenkins 任务	
□ 标签管理		
<ul> <li>三」王机组管埋</li> <li>流水线模板管理</li> </ul>		
☑ 任务组管理		
123 步骤管理		
③ 企业设置		

方法三:将已有流水线保存为流水线模板。

<b>最近运行</b> 运行历史			● 运行
#1 📀 运行成功	测试	构建	1
运行人 🛛 🔊 陈鹏杰	✓ Java 代码扫描	✓ Java 构建上传	□ 复制流水线
开始时间 2019-12-27 15:17:08	2 0 0 2 总数 阻塞 严重 一般	default	☆ 删除
持续时间 3分33秒	同 扫描报告 [] 日志 20秒	日志 1分31秒	
▶ 源(1)			
	🕑 Maven 单元测试		
	1/1 0 0 100% 测试用例 失敗 跳过 通过率		
	同 測试报告 [] 日志 2分2秒		

## 编排企业流水线模板

点击创建企业模板后,可对模板进行编排后保存成企业流水线模板。

■ 菜水线模拟       新自菜火蒸水线模拟       学学课版       ×         ● 菜(0)       → +       ● 新的任务         ● 文星 (0)       → +       ● 新的任务		
<ul> <li>第(0)</li> <li>第該理, 点击 類加</li> <li>交量(0)</li> <li>()</li> <li>()</li></ul>	■ 流水线模板 新自定义流水线模板	保存模板 ×
<ul> <li>● 交量 (0)</li></ul>	▼ 源(0) +	
◆ 安量(0)	暂未设置,点击 添加	● 新的任务
「「「「」「」「」「」「」「」」「」」「」」「」」「」」	▼ 変量 (0)	
	暂未设置,点击 添加	
Þ		
0		
		<b>a</b>

## 管理流水线模板

可在 企业管理 -> 流水线模板管理 中管理 Flow 提供的预置模板 和 用户创建的企业模板,流水线模板可关闭,关闭后企业成员在创建流水线时,将无法继续使用该模板。

🔁 飞流		🛨   🕮 💿 🕻
合 我的星标	流水线模板管理	新建企业模板
风 我参与的	<b>診 企业模板</b> Java PHP Node.js Go Python 其他	
♀ 全部流水线		关闭流水线模板
企业管理	💿 企业模板,测试企业模板	
。通用变量组	Jenkins 任务	
□ 标签管理		
□ 主机组管理		
👿 流水线模板管理		
☑ 任务组管理		
123 步骤管理		
◎ 企业设置		
		G

## 10.3. 任务组管理

企业成员在创建流水线的过程中,对流水线任务节点进行编排时,可以选择任务组。Flow 提供了一些预置的任务组,帮助你快速的创建自己的流水线;同时,Flow 支持创建 企业任务组,用户可个性化定制属于自己企业的任务组,方便企业内部成员创建流水线。

▼ 源(0)			选择任务类型		×
暂未设置,点	<b>企业任务组</b> 代码扫描	Q。请输入任务名称			
▼ 变量 (0)	测试 测试构建 1	企业任务组 ■●● 測试任务组 潮试任务组描述			
暂未设置,点	镜像构建	代码扫描			
	工具部署	Java 代码扫描 阿里巴巴 Java 代码规	Java 安全扫描 Java 安全扫描	PHP Metrics 静态 PHP Metrics 静态扫描	_
	代码 (2) 执行命令	nede JavaScript 代码扫描 JavaScript 代码扫描	<b>Python 代码扫描</b> Python 代码扫描	Python依赖安全扫描 扫描Python依赖文件,	
	呈模倣	Python 代码安全扫描 使用Bandit扫描代码安	<b>Go 代码扫描</b> Go 代码扫描	TSLint 代码扫描 TypeScript 代码扫描	

## 企业任务组

## 创建企业任务组

用户可通过 企业管理 -> 任务组管理 创建企业任务组。

📁 飞流		+ 99+ 0 🛤
☆ 我的星标	任务组管理	新建企业任务组
A. 我参与的	■ 企业任务组 代码扫描 测试 构建测试 构建 镜像构建 工具 部署 代码 执行命令	空模板
♀ 全部流水线		
企业管理		
為 通用变量组	测试任务组	
□ 标签管理	测试任务组播述	
□ 主机组管理		
🗇 流水线模板管理		
🍛 任务组管理		
123 步骤管理		
◎ 企业设置		
		Ø

## 编排企业任务组

第一步:选择已有任务组创建新的企业任务组,点击下一步。

		过	择新的企业仕务组类型		×	
☆ 我的星标	代码扫描	○ 連続 ) /7 タ ク 約			<b>新建</b> 企	业任务组
风 我参与的	测试	~ 请າ和八世另右你			_	
♀ 全部流2	测试构建	代码			£	
	+h2#	06 合并代码	会 检测 Master 分支导	创建标签		
企业管理	何知	将代码输入源中指定的	检测 Master 分支是否	使用 Git tag命令创建标		
& 通用变 <b>i</b>	镜像构建					
□ 标签管理	工具	执行命令				
	部署	执行命令	自定义镜像执行命令			
	代码	● 用于执行命令	使用用户自定义镜像执			
☑ 流水线私	执行命令					
→ 任务组制	空模板	空模板				
四 步骤管理		+ 空白任务 空白的任务。用户可以…	+ 空白任务 (不可编排 空白任务 (不可编排)			
③ 企业设计					- 18	
				-	-	
					-	

第二步,编排任务组内步骤,设置步骤内参数后,保存任务组。

	配置新的企业任务组	×	
☆ 我的星标			新建企业任务组
风、我参与的	→ 执行命令	•	
5 全部流水线	步骤名称	代码 执行命令	空模板
	执行命令		
企业管理	执行命令		
。通用变量组	1 # input your command here 2 echo hello,world!		
□ 标签管理	3		
□ 主机组管理			
🖸 流水线模板管理			
➡ 任务组管理	变量		
四 步骤管理	十 添加參数		
	计包标件		
◎ 企业设置	11797/011+		
		上一步 保存	

## 管理任务组

可在 企业管理 -> 任务组管理 中管理 Flow 提供的任务组 和 用户创建的企业任务组,任务组可关闭,关闭后企业成员在编排流水线时,将无法继续使用该任务组。

☆ 数段框   △ 数分库机   △ 含物水母   ▲ 含用空量   ▲ 含用空量   ● 介盤管置   ● 介盤管置   ● 介盤管置   ● 介盤管置   ● 介盤管置			+ 997 3 🛤
A. Stepsin   C. Stepsin-Ka     Currenter     A. Bantoz Bata   B. Asternation   B. Asternation <t< th=""><th>☆ 我的星标</th><th>任务组管理</th><th>新建企业任务组</th></t<>	☆ 我的星标	任务组管理	新建企业任务组
· 全部流水线       文川在务组         · 企业管理       · · · · · · · · · · · · · · · · · · ·	. ²⁴ . 我参与的	<b>譯 企业任务组</b> 代码扫描 测试 构建规试 构建 镜像构建 工具 部署 代码 执行命令	空模板
企业管理       ▲       通用交量组       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●	♀ 全部流水线	关闭任务组 1	
<ul> <li>▲ 通用变量组</li> <li>测试任务组 测试任务组用递</li> <li>副试任务组用递</li> <li>副试任务组用递</li> <li>① 主机组管理</li> <li>② 流水线模板管理</li> <li>④ 任务组管理</li> <li>③ 步骤管理</li> <li>③ 企业设置</li> </ul>	企业管理		
<ul> <li>「林盤管理</li> <li>○ 主机组管理</li> <li>○ 流水线模板管理</li> <li>○ <b>任务组管理</b></li> <li>③ 步骤管理</li> <li>③ 企业设置</li> </ul>	. 通用变量组	测试任务组	
<ul> <li>□ 主机组管理</li> <li>□ 流水线模板管理</li> <li>□ 伊鴉管理</li> <li>③ 企业设置</li> </ul>	□ 标签管理	测试任务组播还	
<ul> <li>○ 流水线模板管理</li> <li>▲ 任务相管理</li> <li>□ 步骤管理</li> <li>③ 企业设置</li> </ul>	○ 主机组管理		
<ul> <li>▲ 任务组管理</li> <li>&gt; 步骤管理</li> <li>② 企业设置</li> </ul>	☑ 流水线模板管理		
<ul><li>(9) 企业设置</li></ul>	⊶ 任务组管理		
③ 企业设置	四 步骤管理		
	② 企业设置		

## 10.4. 任务组批量升级

过支持任务组升级功能,可以帮助企业用户对流水线进行统一修改和升级。

## 配置任务组批量升级

通过企业设置->任务组管理->新建企业任务组,在创建新建任务组时,选择开启"支持批量升级"。

	■ 配置新的企业任务组	×	
工具箱	任务:		新建企业任务:
為 通用变量组	任务名称 譯 企: Java 构建	执行命令 空移	板
□ 标签管理	任务组描述		
🔄 授权管理	任务组描述		
□ 主机组管理	Jax 支持批量升级 ⑦	主机部署 企业主机部署任	防组
※ Kubernetes集群管理			
🕞 流水线模板管理	构建集群 ⑦		
⊶ 任务组管理	云效北京构建集群	•	
103 步骤管理	任务步骤		
3 构建集群管理	● Java 构建		
管理员设置		上一步保存	

**流水线使用任务组** 在配置流水线过程中,选择已经创建的任务组。

く 返回 任务組	批量升级流水线	基本信息 流程配	触发设置 变量和缓存 仅保存 保存并	运行
	新阶段	部署	选择任务组	
lava 构建	• • • AI+s	5. 0 9	<b>企业任务组</b> Q, 请输入任务名称	
			代码扫描 企业任务组	
	<b>0</b> #11:	19	割试 内建 力 3/3 3/3 3/3 3/3 5/3 3/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5/3 5	房组
			测试构建 <b>主机部署</b> 使像构建 企业主机部署任务组	
			正確 代码扫描	
			代码 其行命令 Java 安全扫描 PHP Metrics 副 Java 安全扫描 PHP Metrics 副	<b>静</b> 态
			空模版 nede JavaScript 代 JavaScript 代码扫描 Python 代码扫描	3描
			<ul> <li>源码温洞检测</li> <li>Python依赖安</li> </ul>	!

#### 支持批量升级,流水线在使用任务组时无法任意更改任务组配置。

〈 返回 任务组批量升级流水线		基本信息 <b>流程配置</b>	触发设置 变量和缓	存	仅保存保存并运行
新阶段		部署 🧷		编辑 ⑪	
				任务名称	
va 构建 🕒 🦻	人工卡点	• •	主机部署		
			● 并行任务	该任务组支持批量升级,流水线在 改任务组配置	使用任务组时无法任意更
				☑ 部署时下载制品	
				制品	
				制品名称.default	~
				主机组	🕒 新建主机组
				测试主机组	~
				部署配置	
				下载路径 ⑦	
				/home/admin/app/package.tgz	
				执行用户 ⑦	

## 升级任务组

进入任务组管理,编辑修改任务组后,会对关联流水线中的任务节点进行升级。

< 返回	企业设置	×
工具箱 ふ 通用支量组 ・ 标签管理 - 振签管理	任务: ● 升级提示 × × 更改任务组会升级1条流水线,磷认继续? 下载路径 下载路径 ○ 方式路径 ○ 方式路径 ○ 方式 ()	<ul> <li>新建企业任务组</li> </ul>
<ul> <li>主机组管理</li> <li>Kubernetes集群管理</li> <li>流水线模板管理</li> </ul>	其们用户     ①       Jan     root       金ュ主机部署       企业主机部署       企业主机部署       企业主机部署任务组	
<ul> <li>● 任务组管理</li> <li>□ 步骤管理</li> <li>□ 构建集群管理</li> </ul>	部著脚本 ②	
管理员设置 风, <b>角色设置</b>	制除保存	

## 升级规则

支持批量升级的任务组,在流水线使用时部分配置支持自定义修改,此部分配置项修改任务组配置时不会升级:

### • 主机部署 — 制品/主机组

〈 返回 任务组批量升级流水线	基本信息 流程配置 触发设置	发量和缓存 仅保存 保存并运行
<b>示阶段</b> 部署	新行阶段	编辑 ①
	±机如果 ● 新約任务	任务名称
Y AIRM Y	TAUDA NIBALA	主机部署
		该任务组支持批量升级,流水线在使用任务组时无法任意更 改任务组配置
		✓ 部署时下载制品
		制品
		制品名称.default ~
		1 主机组 ④ 新建主机组
		测试主机组 ~
		部署配置
		下载路径 ②
		/home/admin/app/package.tgz
		执行用户 ③

## 变量/构建参数等

、返回 任务组批量升	级流水线		I 受重和暖仔	仅保存保存并运行
	新阶段	部署	编辑 ①	
Java 构建	<ul> <li>日本</li> <li></li></ul>	• • ±	- 执行命令	0
	A #544		步骤名称	
	\$ W11m 4			
			执行命令 1 # input your command h	ere.
			3	
			空量 自定义 ∨ abc	= 123
			1 上游输出 ~ def	= 制品名称.def ~
			十 添加参数	
返回 任务组批量升	十级流水线	基本信息 <b>流程配置</b> 触发设1	任务播件 置 交量和缓存	仅保存保存并远
返回 任务组批量升	<b>升级流水线</b> 新阶段	基本信息 <b>流程配置</b> 触发设1 部署	任务插件       重 支量和缓存       编辑 ①	仅保存 保存并运
返回 任务组批量升	升级流水线	基本信息 <b>流程配置</b> 触发设1 部署	任务插件 重 交量和缓存 編辑 ①	仅保存 保存并运
返回 任务组批量升 Java 构建	1-级流水线 新阶段 ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	基本信息 <b>流程配置</b> 触发设 部署 ● <b>♀</b> ま	任务插件 重 交量和缓存 編辑 ① E-机部書 全库 ② ③ hangzhou.aliyuncs.co	仅保存 保存并运 1 直接输入 n/cpj-space/cpj-repository ~
返回 任务组批量并 Java 构建	→提流水线 新阶段 ● タ 人工卡点 ● タ の里云镜像构建	基本信息 <b>流程配置</b> 触发设1 部署	任务插件 重 交量和磁存 編辑 ① 合库 ② ③ hangzhou.aliyuncs.com 标签 ③	仅保存 保存并运 1 直接输入 n/cpj-space/cpj-repository 〜
返回 任务组批量升 Java 构建	H级流水线 新阶段 タ 人工卡点 タ 阿里云镜像构建	基本信息 <b>流程配置</b> 触发说 部署 <b>● 今</b> 主	任务攝件 重 交量和緩存 編編 ① 全库 ② ③ hangzhou.aliyuncs.com 标签 ③ S(CATETIME)	仅保存 保存并运 1↓ 直接输入 n/cpj-space/cpj-repository ~
返回 任务组批量并 Java 构建		基本信息 流程配置 触发说 部署 • 9 主	任务攝件 重 支量和緩存 転期 ① 全庫 ⑦ ② hangzhou.aliyuncs.com 标签 ⑦ S(DATETIME) Dockerfile路径 ⑦	仅保存 保存井运 1 直接输入 n/cpj-space/cpj-repository 〜
返回 任务组批量升 Java 构建	+銀流水线 新阶段	基本信息 流程配置 触发说 部署 第 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日 日	任务攝件 重 交量和緩存 年机部署 を応 ② ③ hangzhou.aliyuncs.co 标签 ③ S(DATETIME) Dockerfile路径 ③ Dockerfile	仅保存 保存并运 1 直接输入 n/cpj-space/cpj-repository ~
返回 任务组批量升 Java 构建	H 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	基本信息 <b>流程配置</b> 触发设1 部署 • • • • ±	任务攝件 重 支量和選存 編編 ① 全席 ② ③ hangzhou.aliyuncs.com 标签 ⑦ S(DATETIME) Dockerfile路径 ③ Dockerfile	仅保存 保存并运 1 直接输入 n/cpj-space/cpj-repository >
返回 任务组批量并 Java 构建	H銀遼水线 新阶段 タ 人工卡点 タ 阿里云镜像构建		任务攝件 重 交量和緩存 編報 ① 合库 ② ③ hangzhou.aliyuncs.com 标签 ③ S(DATETIME) Dockerfile路径 ③ Dockerfile路径 ③	仅保存 保存并运 1 直接输入 n/cpj-space/cpj-repository >
返回 任务组批量升 Java 构建	H 型流水线 新阶段 ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	基本信息 <b>流程配置</b> 触发说 部署 <b>● 9</b> <u>ま</u>	任务攝件 重 交量和選存 編輯 ① E-MI部書 た席 ⑦ @ hangzhou.aliyuncs.co 称签 ⑦ S(DATETIME) Dockerfile路径 ⑦ Dockerfile ので ContextPath ⑦ 	仅保存 保存并运 1 直接输入 n/cpj-space/cpj-repository ~
返回 任务组批量并 Java 构建	H銀遼水线 新阶段		任务攝件 重 交量和進存 編報 ① 合库 ⑦ ④ hangzhou.aliyuncs.com 标签 ⑦ S(DATETIME) Dockerfile路径 ⑦ Dockerfile路径 ⑦ □ □ □ 一 で使用振存 ⑦ 构建参数 ⑦	仅保存 保存并运 11 直接输入 n/cpj-space/cpj-repository ✓
返回 任务组批量升 Java 构建	H 型流水线 新阶段 ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	Arte Are Area	任务攝件 重 交量和運存 編輯 ① 住机部書 全席 ⑦ ⑦ hangzhou.aliyuncs.com 标签 ⑦ S(DATETIME) Dockerfile路径 ⑦ Dockerfile路径 ⑧	仅保存 保存并运 1↓ 直接输入 n/cpj-space/cpj-repository ~

# 10.5. 步骤管理

步骤作为 Flow 最底层的核心能力,所有的流水线都是通过步骤的编排组合而成的。用户在编排流水线的时候,可以选择步骤并设置步骤参数。

<ul> <li>※ 源(0) +</li> <li>部署</li> <li>新設段</li> <li>編載</li> <li>任务名称</li> <li>任务名称</li> <li>「日本の本の本の本の本の本の本の本の本の本の本の本の本の本の本の本の本の本の本の</li></ul>	
Kubernetes & th	
<ul> <li>交量(0) </li> <li>(2) +     <li>(3) 添加步骤     </li> </li></ul>	
	1
Line Kubecti 发布 集群连接 ① ●	新建连接
無算连根不能为空 Kubect 版本 ① V184	~

#### 管理步骤

可在 企业管理 -> 步骤管理 中管理 Flow 提供的步骤 和 用户创建的企业步骤,步骤可关闭,关闭后企业成员在编排流水线时,将无法继续使用该步骤。

··· [•] 📁 飞流		💿 😳 🧑
☆ 我的星标	步骤管理	查看文档进行新建 ④
A 我参与的	■ 企业步骤 静态扫描 测试 代码 覆盖率 构建 部署	发布 默认 下载 上传
♀ 全部流水线	关闭步骤	
企业管理	<b>自定义步骤</b> 由金键提供 ① ① ① ① ① ① ① ① ① ① ① ① ① ① ① ① ① ① ①	SonarQube(预发) 由rdc-rdc提供
& 通用变量组		
□ 标签管理	自定义步骤3 <b>Wayne发布-预发</b> 中全環境性 中rde_rde提供	
🗊 主机组管理	IT THE RA	
☑ 流水线模板管理		
☑ 任务组管理		
123 步骤管理		
⑥ 企业设置		

# 10.6. 授权管理

通过统一授权管理,可以管理企业成员的阿里云服务和代码仓库的权限。

返回		企业设置		
工具箱	授权管理 ②			
🚴 通用变量组	阿里云账号名	拥有者	创建时间	操作
□ 标签管理	)@163.com	🗶 光晕	2021-01-15 09:59	ľŤ
🔁 授权管理	31	0 wts1231231	2021-01-04 11:53	ľ Ŵ
□ 主机组管理	21@126.com	1延枚	2020-12-24 16:57	ľ Ý
❀ Kubernetes集群管理	1-inner.com	🕼 企业成员名称codecenter	2020-12-18 11:34	ľ Ŵ
<ul> <li>○ 流水线模板管理</li> <li>○ 任务组管理</li> </ul>	10.000	hey_ing12345	2020-12-04 11:46	ľ Ŵ
123 步骤管理	Idemo	🔵 testzjldemo	2020-09-16 10:26	C V
2 构建集群管理	mc	yioren@163.com	2020-09-01 16:14	C V
管理员设置	*********	C ZhenRan	2020-08-26 16:53	c t
		< 1 2 3	>	

## 阿里云服务授权

可以设置阿里云服务的授权范围和授权服务

- 授权范围
- 。 企业内均可使用:企业内的成员在编排流水线时,均可使用您授权的服务
- 仅本账号使用:只有你的账号可以使用自己授权的服务
- 授权服务
- ◎ 目前支持授权服务包括: OOS/ACR/SAE/EDAS/FC/ACK/ECS/ROS/EMA

工具箱		阿里云权网	<b>浸设置</b>	×	
<ul> <li>▲ 通用变量组</li> <li>□ 标签管理</li> <li>□ 振び等理</li> </ul>	<b>阿里云权限设置</b> 代码库设置	<b>阿里云账号</b> (若需要添加其他阿里云账号 	授权,将其添加至本企业,并完成授权。查看 <b>帮助文档</b> ) 佛田		操作
<ul> <li>☑ 其机组管理</li> <li>❀ Kubernetes集群管理</li> </ul>		<ul> <li>● 正型(4)(1)(2)(H) ● 1(2)(2)(2)(2)</li> <li>● 1(2)(2)(2)(2)(2)(2)</li> <li>● 1(2)(2)(2)(2)(2)(2)(2)</li> <li>● 1(2)(2)(2)(2)(2)(2)(2)(2)(2)</li> <li>● 1(2)(2)(2)(2)(2)(2)(2)(2)(2)(2)(2)(2)(2)(</li></ul>	JECH1		
<ul> <li>示水线模板管理</li> <li>任务组管理</li> </ul>		<ul> <li>✓ 対象存储(OSS)</li> <li>✓ Serverless应用引擎(SAE)</li> </ul>	<ul><li>✓ 容器镜像服务(ACR)</li><li>✓ 企业级分布式应用(EDAS)</li></ul>		
<ul> <li>应 步骤管理</li> <li>18 构建集群管理</li> </ul>		<ul><li>✓ 阿里云函数计算(FC)</li><li>✓ 云服务器(ECS)</li></ul>	<ul> <li>容器服务Kubernetes(ACK)</li> <li>资源编排服务(ROS)</li> </ul>		C U C U
管理员设置	L	✓ 移动研发平台(EMAS)	77724		C T

## 代码库授权

第三方代码仓库授权,包括:

- 阿里云 Code
- 码云
- Git hub

		正並改畫		
工具箱		代码库设置	×	
為 通用变量组	阿里云权限设置	受权范围		操作
□ 标签管理	代码库设置	<ul> <li>仅本账号使用</li> <li>已授权服务</li> </ul>		C 4
2 授权管理		🕞 阿里云Code	● 添加授权	C T
② 主机组管理		账号: ky ol @yeah.net	解释绑	C t
※ Kubernetes集群管理		6 码云	● 添加授权	िंदी भीत
고 流水线模板管理		•		
☑ 任务组管理		Github 账号: skyhook-cpj	(+) 添加授权 解绑	C Ū
123 步骤管理				ĊŴ
8 构建集群管理	yi)163.com	ioren@163.com	2020-09-01 16:14	C 1
	1300000081	C ZhenRan	2020-08-26 16:53	C 1
<u>官</u> 埋页设置				

## 10.7. 服务连接管理

Flow 授权管理全面升级为服务连接管理,升级完成后为企业提供以下特性:

1支持配置管理各种阿里云服务(ECS/ACR/ACK等)、代码服务(Codeup/Github/Gitlab等)、其他服务(Jenkins/私有 Docker 仓库)的服务连接。

2 所有服务连接支持设置使用范围:所有人可见、仅自己可见、指定成员可见。企业成员可以使用别人的服务连接创建流水线调用外部服务。

3 成员离职等原因导致的服务连接失效问题,可通过配置服务连接中的服务授权/证书完成修改,无需重复修改流水线中的配置。

进入企业设置 -> 服务连接管理,可以查看管理企业内所有的服务连接。

工具箱	服务连接管理 ②				Q 7	奇 新建服务连接
通用变量组	服务连接名	服务连接 ID	类型	拥有者	创建时间	操作
标签管理	ナ 23的E itlab服务…	3fw7 79ew	自建Gitlab	🗶 陈鹏杰	2021-10-20 15:04	C t
服务连接管理		w3ah,y4hp	Gitlab	🗶 陈鹏杰	2021-10-20 14:53	C t
土机组官理 Kubernetes集群管理	疗 ■ 23的移动研发平台(	. hk	移动研发平台(EMAS)	🗶 陈鹏杰	2021-10-20 14:40	C T
流水线模板管理	, Bitbucket服务连接	zpzymw6	Bitbucket	🗙 陈鹏杰	2021-10-14 19:09	ĊŤ
任务组管理	re c的Jenkins服务连接	° Tr7kqpks7	Jenkins	rdc-rdc	2021-10-14 17:21	ĊŤ
_{罗號官理} 构建集群管理	nc的云效Packages	x1ce( ahe77b6	云效Packages	rdc-rdc	2021-10-14 17:11	¢ t
	> I码云服务连接	qnr6ł neyi9m76	码云	🗶 陈鹏杰	2021-10-13 17:04	ĊŤ
管理员设置	ri c的码云服务连接	3ws1ahxmy9m6	码云	rdc-rdc	2021-10-13 17:03	C Ŵ
Maven设置						

创建服务连接,选择具体连接类型,填写服务连接名,创建或选择服务授权/证书,并设置使用范围。

#### 阿里云服务

目前支持的阿里云服务包括:

- 云服务器(ECS)
- 容器镜像服务(ACR)
- 容器服务 Kubernetes(ACK)

- 对象存储(<mark>OSS</mark>)
- 企业级分布式应用(EDAS)
- Serverless应用引擎(SAE)
- 资源编排服务(ROS)
- 阿里云函数计算(FC)
- 移动研发平台(EMAS)
- 云效 Packages

		< 新建服务连接	×		
工具箱	服务连接	x4-142→1×171	_	0° 7 (	) 新建服务连持
通用变量组	服务连接名	た友安室 云服务器(ECS)	~	创建时间	操作
示签管理	光晕123的	服务连接名		2021-10-20 15:04	ľ Ú
服务连接管理	光晕123的	□■ □ 的云服务器(ECS)服务连接	19/64	2021-10-20 14:53	
E机组管理		服务连接ID			
ubernetes集群管理	光晕123的	e9y5gnx t1		2021-10-20 14:40	C Ŵ
п水线模板管理	光晕的Bitt	服务授权/证书		2021-10-14 19:09	ľ Ŵ
E务组管理	rdc-rdc的	skyho 520 com的RAM授权 v	+ 新建	2021-10-14 17:21	ĊŴ
>骤管理 ▶ ▶ ▶ ₽ ■ ▶ ₽ ₽ ■ ▶ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ ₽ </td <td>rdc-rdc的</td> <td>○ 公开:所有人可见 ○ 私密:仅自己可见 ○ 自定义:指定人员可见</td> <td>- 1</td> <td>2021-10-14 17:11</td> <td>C T</td>	rdc-rdc的	○ 公开:所有人可见 ○ 私密:仅自己可见 ○ 自定义:指定人员可见	- 1	2021-10-14 17:11	C T
	光晕的码z		- 1	2021-10-13 17:04	C V
管理员设置	rdc-rdc的		_	2021-10-13 17:03	C Ŵ
角色设置		上一步 创	建		

## 代码服务

目前支持的代码服务包括:

- 阿里云 Code
- 码云
- Git hub
- 云效 Codeup
- 自建 Git lab
- 通用 Git
- Git lab
- Bit bucket

		< 新建服务连接	×		
工具箱	服务连接			0 7	● 新建服务连持
通用变量组	服务连接领	连接类型 Github	~	创建时间	操作
示签管理	光晕123的	服务连接名		2021-10-20 15:04	ľŤ
服务连接管理	光晕123的	的Github服务连接	16/64	2021-10-20 14:53	CA M
主机组管理		服务连接ID			
Kubernetes集群管理	光晕123的	urh say99w		2021-10-20 14:40	ľŤ
<b>流水线模板管理</b>	光晕的Bitk	服务授权/证书		2021-10-14 19:09	ľŤ
壬务组管理	rdc-rdc的	skyhook-cp)的Github授权 使用范围	✓ + 新建	2021-10-14 17:21	C t
步骤管理	rdc-rdc的	○ 公开:所有人可见 ○ 私密:仅自己可见 • 自定义:指定人员可	见	2021-10-14 17:11	ra w
9建集群管理	光晕的码z	🛞 邀请		2021-10-13 17:04	
管理员设置	rde_rde%			2021-10-12 17:02	
角色设置	ruc-ruces	十一步	创建	2021-10-13 17:03	Ľ W

## 其他服务

## 目前支持的其他服务包括:

- Jenkins
- 私有镜像仓库

		< 新建服务连接	×		
工具箱	服务连接	No.44 (2011)	_	0, 7 (	新建服务连持
通用变量组	服务连接名	<b>注接</b> 突望 私有镜像仓库	~	创建时间	操作
标签管理	光晕123的	服务连接名		2021-10-20 15:04	ľŤ
服务连接管理	光晕123的	光晕123的私有镜像仓库服务连接	16/64	2021-10-20 14:53	
主机组管理		服务连接ID			
Kubernetes集群管理	光晕123的	xrykorwhxkzsscuz		2021-10-20 14:40	ľŤ
流水线模板管理	光晕的Bitt	服务授权/证书		2021-10-14 19:09	ľŤ
任务组管理	rdc-rdc的	光晕123的镜像仓库证书 ~ / / / / / / / / / / / / / / / / / /	+ 新建	2021-10-14 17:21	ĊŴ
步骤管理 构建集群管理	rdc-rdc的	<ul> <li>公开:所有人可见</li> <li>私密:仅自己可见</li> <li>自定义:指定人员可见</li> </ul>	- 1	2021-10-14 17:11	ĊŴ
	光晕的码z		- 1	2021-10-13 17:04	C t
管理员设置	rdc-rdc的			2021-10-13 17:03	ĆŴ
角色设置		上一步	创建		

# 11.自定义步骤 CLI 工具 11.1. 自定义步骤 Flow CLI 子命令: step

Flow 是「流水线」飞流推出的一款命令行工具,用以管理飞流中各种资源,子命令 step 可以让用户在飞流流水线中运行自定义镜像及命令达到扩展流水线能力的效果。 Flow 中一个流水线中运行的镜像被称为一个"步骤",步骤是有一段描述 yaml 和一个 docker 镜像组成的一个最小运行单元,yaml 描述决定步骤运行时的入参,docker 镜像 决定步骤运行的环境;这篇文档将引导用户在 Flow 上创建一个自定义步骤。

### 下载安装

操作	系统	下载地址
MacOS	32位	https://rdc-plugin-storage.oss-cn-beijing.aliyuncs.com/flow-cli/0.0.6/aliyun/flow- step-cli_darwin_386
MaLUS	64 位	https://rdc-plugin-storage.oss-cn-beijing.aliyuncs.com/flow-cli/0.0.6/aliyun/flow- step-cli_darwin_amd64
Wiedowe	32位	https://rdc-plugin-storage.oss-cn-beijing.aliyuncs.com/flow-cli/0.0.6/aliyun/flow- step-cli_windows_386.exe
WINDOWS	64 位	https://rdc-plugin-storage.oss-cn-beijing.aliyuncs.com/flow-cli/0.0.6/aliyun/flow- step-cli_windows_amd64.exe
	32位	https://rdc-plugin-storage.oss-cn-beijing.aliyuncs.com/flow-cli/0.0.6/aliyun/flow- step-cli_linux_386
Linux	64 位	https://rdc-plugin-storage.oss-cn-beijing.aliyuncs.com/flow-cli/0.0.6/aliyun/flow- step-cli_linux_amd64
	ARM	https://rdc-plugin-storage.oss-cn-beijing.aliyuncs.com/flow-cli/0.0.6/aliyun/flow- step-cli_linux_arm

#### 以 Mac 为例:

\$ curl https://rdc-plugin-storage.oss-cn-beijing.aliyuncs.com/flow-cli/0.0.6/aliyun/flow-step-cli_darwin_amd64 -o /usr/local/bin/flow && chmod +x /us
r/local/bin/flow

#### \$ flow -v # 输出版本信息说明安装成功

#### 登录 Flow

\$ flow login

# 登录

浏览器会打开进入 Flow 登录页面,如果已经登录会提示 success,登录成功后命令行会提示选择账号下企业,输入企业对应的数字进入企业。

→ step login
Hi 金镗! You've successfully authenticated.
Please select an organization to continue:
[0]
[2]
[3] 金镗的企业
[4] 的企业
[5] \
[6]
[7]
[8] 27 企业
[9] 上企业
[10]
[11] 企业
[12] 二 袮 ムキ └ 告 方
3
Yor're now at organization: 金镗的企业

#### 新建步骤

```
$ flow step init <your-step-sign> # 新建步骤
```

将 <your-step-sign> 更换为你自己的步骤标识,注意: step 标识为整个云效唯一,若被其他步骤占用则无法创建成功。

### 这个命令会在本地创建如下文件:

aml

• entry.sh: 步骤执行的基础入口脚本, 非特殊情况无需修改。

- step.yaml:用于定义 step.执行所需的输入参数,包括 step 的运行的镜像环境地址,以及在流水线配置时该步骤所需的配置参数。
- step.sh:步骤运行时执行的脚本文件,你可以在该脚本内获取流水线的环境变量、step.yaml中配置的参数,并定义你想要执行脚本逻辑。

• Dockerfile:用于构建步骤镜像的 Dockerfile文件,当你进行步骤推送时,会根据此文件构建你的步骤环境镜像,并推送到 step.yaml 中定义的镜像地址中。

当你完成自定义步骤配置推送后,流水线执行你的自定义步骤时,会先从你的环境镜像地址拉取镜像,并在该环境中执行你所定义的脚本文件。

对初始化文件进行简单修改,下面通过一个示例详细讲解3个关键文件的具体作用。

1. step.yaml 里定义了步骤有关的描述:

name: 自定义步骤	# 步骤名称
sign: <your-step-sign></your-step-sign>	# 步骤唯一标识
description: 我的第一个步骤	# 步骤描述
image: <docker-image-url></docker-image-url>	# 步骤运行环境镜像
items:	# 入参表单
- label: 执行命令	
name: command	
type: shell	
value:	
# input your command	here
echo hello,world!	
- label: 参数 abc	
name: abc	
type: input	

其中 image 为该步骤环境镜像的地址,请填写您自己的镜像仓库地址。当你进行步骤推送时,会根据 Dockerfile 进行镜像构建并推送至该地址。

#### ↓ 注意

请确保该镜像地址为公网不通过认证可以公开访问的地址,以确保构建机可以正常拉取。

items 定义了流水线编辑页面中该供用户配置的输入表单,支持多种前端组件,如输入框 input, 下拉款 dropdown, 复选框 checkbox 等;示例中 type 类型 shell 为一个脚本输 入框。更多语法见步骤 yaml 描述语言。示例中定义了两个 item,分别为类型为 shell 的 command、类型为 input 的 abc。

2. step.sh 既步骤执行的命令, step.sh 在流水线中运行时可以以环境变量的方式获取到来自流水线上下文的系统变量,以及用户通过编辑页面、运行传入的方式写入的入参, 脚本中列举了一部分可供使用的环境变量:

#!/bin/sh	
set -e	# 脚本遇到错误退出
# 系统提供参数,从流水线上下文获取	
echo [INFO] PIPELINE_ID=\$PIPELINE_ID	# <b>流水线</b> ID
echo [INFO] PIPELINE_NAME=\$PIPELINE_NAME	# 流水线名称
echo [INFO] BUILD_NUMBER=\$BUILD_NUMBER	# 流水线运行实例编号
echo [INFO] EMPLOYEE_ID=\$EMPLOYEE_ID	# <b>触发流水线用户</b> ID
echo [INFO] WORK_SPACE=\$WORK_SPACE	# /root/workspace容器中目录
echo [INFO] PROJECT_DIR=\$PROJECT_DIR	# 代码库根路径,默认为/root/workspace/code
echo [INFO] PLUGIN_DIR=\$PLUGIN_DIR	# 插件路径,默认为/root/workspace/plugins
echo [INFO] BUILD_JOB_ID=\$BUILD_JOB_ID	# build-service 任务ID
cd \$PROJECT_DIR	# 进入代码库所在路径
sh -ex \$WORK_SPACE/user_command.sh	# 执行 command 脚本
echo \${abc}	# <b>输出</b> abc

示例中通过 sh -ex \$WORK_SPACE/user_command.sh 直接执行在步骤中配置的指令。另外,将我们在 step.yaml 中配置的参数 abc 进行了输出。

#### ? 说明

user_command.sh为 Flow 的一个默认规则,会将 step.yaml 中, name为 command 的 shell 脚本转化为 user_command.sh。当然,你也可以定义其他的 shell 脚本,通 过环境变量来获取脚本指令。

3. Dockerfile 中包含了步骤运行时所需要的环境,目前步骤所需的基础镜像必须是公网不通过认证可拉取的。

FROM registry.cn-beijing.aliyuncs.com/rdc-builds/base:1.0
MAINTAINER xxx <xxx@alibaba-inc.com>
COPY .step/*.sh /root/
COPY <your-step-sign>/step.sh /root/step.sh
RUN chmod +x /root/*.sh
ENTRYPOINT [ "/root/entry.sh"]

Dockerfile 本质是提供你执行 step.sh 的镜像环境,你可以:

- 直接在 Dockerfile 中安装你所需的依赖。
- 或修改基础镜像地址,请确保基础镜像中已经<del>安装了</del>jq。

#### 上线步骤

\$ flow step publish <your-step-sign> # 发布镜像

这个命令会执行两个动作:

1. 将本地的 Dockerfile 构建成为 step.yaml 中的 docker 镜像并 push 到远端仓库。请确保本地已经安装了 Docker。

2. 将 step.yaml 描述的步骤原信息 publish 到云效的服务器。

使用 flow step ls 可以查看已经发布的步骤

→ step step ls				
SIGN	NAME	CREATE TIME	MODIFIED TIME	DESCRIPTION
test-step	自定义步骤	2019-12-30 19:56:52	2019-12-30 19:56:52	我的第一个步骤

#### 使用步骤

进入 Flow 流水线页面,从空模板创建流水线,点击阶段 ->添加步骤->自定义步骤,找到刚才创建的步骤。



步骤添加完成之后,可以对步骤的内容进行配置,点击保存并运行

氘水线源	新阶段	新阶段	编辑 亚	
十添加流水线源	♀ 空任务	新的任务	<ul> <li>自定义步骤</li> <li>步骤名称</li> </ul>	
			自定义步骤	
		1	执行命令 1 # %nput your command here 2 echo hello,world! 3	e ²
			参数 abc 123	
			任务插件 ● 添加插件	

可以看到步骤在流水线中运行起来,并打印出 hello world! 和 123。

🌈 💷	贝 〉 流水线 2022-05-27 ▼	*	邀请 🕂 🖆 🧷 🕺
最近运	空任务		★ 请 三菜单
#7 🕑 运行	构建	自定义步骤(1s)	下载
触发信息 开始时间 持续时间 产物	<ul> <li>申请运行环境(Os)</li> <li>清理工作区(1s)</li> <li>流水线缓存(1s)</li> <li>自定义步骤(1s)</li> <li>缓存上传(1s)</li> </ul>	<pre>[executionStep begins at 2022-06-01 14:53:32] [14:53:33] [SUCCESS] 获取 Credential 成功 [14:53:33] [INFO] 共指定工作経合、使用默认路径/root/workspace/code [14:53:33] [INFO] PIPELINE_ID=1792020 [14:53:33] [INFO] PIPELINE_NAME=添水线 2022-05-27 [14:53:33] [INFO] BUILD_NUMBER=7 [14:53:33] [INFO] BUILD_NUMBER=7 [14:53:33] [INFO] HMPOIYEE_ID=5e71d6f503283833284f91c2 [14:53:33] [INFO] PROJECT_DIR=/root/workspace/code [14:53:33] [INFO] PROJECT_DIR=/root/workspace/code [14:53:33] [INFO] BUILD_JOB.ID=61526508 [14:53:33] [INFO] BUILD_JOB.ID=61526508 [14:53:33] [IMFO] BUILD_JOB.ID=61526508 [14:53:33] [INFO] BUILD_JOB.ID=61526508 [14:53:33] [SUCCESS] 步骤运行成功</pre>	

## 更新步骤

若你需要更新步骤,可以修改 step.yaml、step.sh、Dockerfile 文件后,重新执行 flow step publish <your-step-sign>,Flow CI 会重新进行镜像构建和步骤上传。 若你进入流水线编辑页后,发现步骤参数配置内容未发生变化,可以尝试刷新缓存。

## 其他操作

teambitiondeMa NAME:	cBook-Pro:~ teambition\$ flow step help
Flow step -	A new cli application
USAGE: Flow step [0	lobal options] command [command options] [arguments]
v0.0.6	
COMMANDS:	
init, i	Create a directory of step project
publish, p	Publish step to your enterprise on flow
list, ls	List all steps
get	Show step description
remove, rm	Remove step
run	Run step local
valid, val	Run step local
public	Make step public
version	Get step version list
search	Get step from marketplace
add	Add step from marketplace
help, h	Shows a list of commands or help for one command
GLOBAL OPTIONS	
help, -h	show help (default: false)
version	-v print the version (default: false)

# 11.2. 步骤 yaml 描述语言

ltem 属性

属性名称	含义	是否必填	默认值	哪些 Type的item 可以使用
name	作为该 item 的唯一标识,用于 step.sh 中环境变量的 key	是	无	全部
type	item的不同类型决定了其展现形式。 • input • password • textarea • shell • select • checkbox	是	无	全部
rules	<pre>该item的校验规则,比如 rules:     -required # 表示该字段是必须的     -length: # 校验输入长度     min: 4     max: 10     -regex: xxxxxxx # 校验正则</pre>	ጽ	无	input/password
label	该 item 在表单中显示的标题	是	无	全部
datasource	数据源。 静态数据源的 value 是数组: datasource: - label: 选项1 value: value1 - label: 选项2 value: value2	否	无	select

## 云效2020

属性名称	含义	是否必填	默认值	哪些 Type的item 可以使用
showCondition	<pre>这 item 的显示条件, 比如: - name: abc label: 参数 abc type: select datasource:     label: 选项1     value: 123     label: 选项2     value: 456 - name: edf label: 参数 edf type: input showCondition: \${abc} == "123"  jx示例标识, 只有参数 abc 为选项 1 时, 才展示参数 edf 的配置选项 </pre>	否	无	所有
value	默认值	否	无	input
description	在item的右边显示一个问号,显示提示信息。		无	所有

#### ltem Type列表

- input
- 普通的输入框
- password
- 一个界面现实为星号的输入框
- textarea
- 一个普通文本域
- shell
- 一个对 bash 语法展示高亮的文本域
- select
  - 下拉框
- checkbox
  - 一个checkbox,选中时,值为true,未选中时,值为false

不同类型 item 示例:

编辑 ①	×
input 参数	
123	
password 参数	
textarea 参数	
123	
aball 40 Pt	л
1 123	Ľ
calact 裁判	
选项1	~
✓ checkboy 参数	
CITEGRIDA & XX	
	编辑 ① input 参致 123 password 参数 ・・・ textarea 参数 123 shell 参数 123 shell 参数 3 223 shell 参数 23 checkbox 参数

## 11.3. 步骤 Yaml 描述语言场景补充

### 1、使用 java构建或者docker构建等前置任务的输出

在step.yaml中items 下添加如下内容

label: 外部输入
 name: INPUT
 type: upstream_component_output
 autoComplete: true
 input_type: artifacts
 description: 前序输入

### type upstream_component_output 表示使用前面任务的产出物,支持构建出的产物以及构建出的镜像

#### 目前支持构建的产物,例如java构建的包数据如下

{"ARTIFACT_URL":"https://rdc-build.aliyuncs.com/aone2/build-service/api/v2/ossproxy/download?ns=execution-component&bucketName=execution-component-rd c&fileId=aone2/2000860/1607602660253/default.tgz&fileName=default.tgz&md5Sign=02d27f4c5b81fa50c8ea1ad4676e4512","ARTIFACT_NAME":"default.tgz", "ARTIF ACT_MD5":"369fbdeb12dbb1182f3ed56c5717aa31", "WEB_DOWNLOAD_URL":"execution-component&bucketName=execution-component-rdc&fileId=aone2%2F2000860%2F1607 602660253%2Fdefault.tgz&artifactName=default.tgz"}

#### 镜像构建的产物数据如下

{"DOCKER_URL":"registry.cn-hongkong.aliyuncs.com/xiaoxuan/java-ceshi-spring-boot-demo:2020-12-10-20-38-01","ARTIFACT_URL":"https://rdc-build.aliyuncs .com/aone2/build-service/api/v2/ossproxy/download?ns=execution-componentsbucketName=execution-component-rdcsfileId=aone2/2002689/1607603999672/applic ation_docker_url_config.desc6fileName=application_docker_url_config.desc6md5Sign=4abee7adfc770e4dd943cb7302180d0e","ARTIFACT_MD5":"bf531410bfcf36c42b 1735bb8c927edd")

在step.sh可以通过\${INPUT}获取到以上数据进行处理。

### 2、自定义步骤的产物输出,方便后续步骤使用

step.yaml中加上以下内容

```
- label: 制品名称

name: ARTIFACT

value: default

export_to_output:

type: artifact

description: 自定义产出物名称,定义后在部署组件使用

alias: 制品

dynamicName: default

type: input

description: 制品名称为当前构建产物的名称。用于区分多次构建产生的不同产出物,定义后在部署组件使用。例如"default"

step.sh 中加上以下修改
```

```
ARTIFACT 名称要和 step.yaml中定义的名字一样
ARTIFACT 后面输出的数据,就可以在后面的步骤中获取到
```

# 将输出写入params

echo "\$stepIdentifier.ARTIFACT={\"name\":\"testData\"}" >>\$WORK_SPACE/params

## 11.4. Flow cli使用常见问题

### 1、运行环境

需要安装Docker、Go语言环境,请根据对应系统进行安装。

```
2、 windows 系统双击运行,闪退
```

flow cli不支持双击运行,请采用命令行方式运行。

3、windows 系统运行,报 missing Location in call to Time.In

panic: time: missing Location in call to Time.In	
goroutine i [running]:	
time.Time.In()	
/usr/local/Cellar/go/1.13.4/libexec/src/time/time.go:1126	
<pre>gitlab.alibaba-inc.com/aone/flow-step-cli/util.beijingTime(0×c00018aab0, 0×a, 0×c00018a</pre>	ab0, 0×d)
/Users/lxliuxuan/go/src/gitlab.alibaba-inc.com/aone/flow-step-cli/util/time.go:	18 +0×152
gitlab.alibaba-inc.com/aone/flow-step-cli/util.TimeFormat(0×1739b645658, 0×2, 0×2)	
/Users/lxliuxuan/go/src/gitlab.alibaba-inc.com/aone/flow-step-cli/util/time.go:	9 +0×5d
gitlab.alibaba-inc.com/aone/flow-step-cli/step.List(0×c00004b140, 0×0, 0×0)	
/Users/lxliuxuan/go/src/gitlab.alibaba-inc.com/aone/flow-step-cli/step/list.go:	27 +0×2f6
github.com/urfave/cli.(*Command).Run(0×c00009cea0, 0×c00004b080, 0×0, 0×0)	
/Users/lxliuxuan/go/src/github.com/urfave/cli/command.go:163 +0×4c0	
github.com/urfave/cli.(*App).RunAsSubcommand(0×c000035380, 0×c00004ae40, 0×0, 0×0)	
/Users/lxliuxuan/go/src/github.com/urfave/cli/app.go:427 +0×9e9	
github.com/urfave/cli.(*Command).startApp(0×c00009cb40, 0×c000004ae40, 0×c0000dbca0, 0×4	0a4e1)
/Users/lxliuxuan/go/src/github.com/urfave/cli/command.go:278 +0×693	
github.com/urfave/cli.(*Command).Run(0*c00009cb40, 0*c00004ae40, 0*0, 0*0)	
/Users/lxliuxuan/go/src/github.com/urfave/cli/command.go:94 +0x9f2	
github.com/urfave/cli.(*App).RunContext(0×c000035200, 0×a27300, 0×c00001e238, 0×c00004a	080. 0×3. 0×4. 0×0. 0×0)
/Users/lxliuxuan/go/src/github.com/urfave/cli/app.go:306 +0×797	
github.com/urfave/cli(*Aon).Run()	
///sers/lxliuxuap/op/src/github.com/urfave/cli/app.go:217	
main main()	
/// core/lyliuyuap/oo/src/gitlab alibaba-inc com/aono/flow-stop-cli/main go:111 +	Axacıl
/ Users/ childhuan/ gu/ sic/ giclab.acibaba-inc.com/ aune/ row-scep-cci/ main.gu.iii +	Urac4
安装Go语言环境即可解决上述问题。	

#### 4、想自定义端口

无法支持,目前登录回调地址为127.0.0.180端口,不支持自定义端口。

#### 5、开发环境80端口被占用或者无浏览器,如何使用Flow cli。

目前Flow cli 是依赖~/.flow.json 文件,来进行鉴权,用户可以自己生成该文件,实现登录。文件内容如下:

{"userToken":{"access_token":"ACCESS_TOKEN","token_type":"Bearer","user_id":"USER_ID"},"organization": {"error":"","message":"","id":"","createdAt":"","updatedAt":"","creatord":"","name":"","logo":"","logo":"","category":"","description":"","website":"","background":"","co

请将ACCESS_TOKEN 以及USER_ID替换成用户自己的值,这两个值的获取方式如下:

访问<mark>云效流水线</mark>,登录之后,右键查看网页源代码,源代码中的tbUserld即为USER_ID,accessToken即为需要的ACCESS_TOKEN。

将上述内容替换后,写入到~/.flow.json即可。

### 6、运行报错,且没有日志

构建	自定义步骤(2s)	下载
✓ 申请运行环境(30s)	没有日志	
✔ 清理工作区(1s)		
」 ✔ 流水线缓存(1s)		
● 缓存上传		

如果出现上面错误,是由于步骤对应的镜像,不是公开Pull权限,云效没有权限pull镜像导致无法运行,请将step.yaml中定义的镜像改为公开pull权限,同时为了数据安全, 请不要再镜像中存有敏感数据。

## 11.5. 创建一个带红线功能的测试步骤

作为自定义步骤的进阶,以 Sonar 为例创建一个用于测试/扫描的步骤,并开放给所有企业。

#### 准备

- 一个公网可访问的 Sonar 服务。
- 一个用于测试的代码库,以 maven 项目为例,确保 pom 文件中引入了 sonar 插件:

```
<br/>
<build>
<pluginManagement>
<plugins>
<plugin>
<plugin>
<plugin>
<plugin>

<artifactId>sonar-maven-plugin</artifactId>
<version>3.4.0.905</version>
</plugin>
</plugins>
</pluginas
</pluginas
</pluginas</pluginas</pre>
```

### 本地开发

参照自定义步骤 Flow CLI 子命令: step搭建好开发环境,初始化一个步骤:

```
flow step init sonar-qube
```

sona-qube 目录下会有三个文件

```
      drwxr-xr-x
      5 lxliuxuan
      staff
      160 Dec 10 19:36 .

      drwxr-xr-x
      20 lxliuxuan
      staff
      640 Dec 10 19:36 ..

      -rw-r--r--
      1 lxliuxuan
      staff
      217 Dec 10 19:36 Dockerfile

      -rw-r--r--
      1 lxliuxuan
      staff
      720 Dec 10 19:36 step.sh

      -rw-r--r--
      1 lxliuxuan
      staff
      285 Dec 10 19:36 step.yaml
```

Dockerfile 用于构建步骤镜像,Dockerfile 中的基础镜像为云效提供的基础镜像,里面已经内置了一些软件,尽量不要更改基础镜像,如需更改基础镜像,请确保新的基础镜 像中安装了jq.

step.sh 为在步骤中执行的脚本

step.yaml 用户描述步骤信息

#### 前端

编辑 sonar-qube 文件夹下面的 step.yaml :

#### 注意: image 需要改成用户自己的镜像地址,且必须为公开pull权限

```
name: SonarQube(预发)
sign: sonar-qube-staging
description: SonarQube用于测试公开步骤
category:静态扫描
image: registry.cn-beijing.aliyuncs.com/build-steps/sonar-qube-staging:1.0
items:
 - label: Sonar服务器地址
   name: STEP SONAR HOST
   type: input
  - label: 红线信息
   name: CHECK_REDLINES
   type: addable_group
   rules:
     - require: false
   add button:
     type: icon
     icon: plus
     text: 增加红线
    tip:
      icon: question-circle
       description: 红线校验失败任务将失败
    template:
     items:
       - name: redline
        label: 红线
        position: flat
         type: custom_redline_dropdown
         datamap: '[{"key": "Total","type": "LE"},("key": "Blocker","type": "LE"},("key": "Critical","type": "LE"},{"key": "Major","type": "LE"}]'
         rules:
          - require: false
```

name 用于定义步骤名称

 sign 步骤唯一标识,全局唯一

 category 步骤分类

 image 步骤镜像地址, image 需要改成用户自己的镜像地址,且必须为公开pull权限

 items: 步骤语言描述 请参考步骤语言Yaml描述。

 redline: 红线信息

 datamap:

 type 枚举字段 LE: 实际值小于等于期望值; GE: 实际值大于等于期望值; EQ 实际值等于期望值

key 红线待校验字段 以下key 支持显示中文描述,自定义key则直接显示key。

key(不区分大小写)	中文描述		
passedrate	测试通过率		
total	Total 问题数		
blocker	Blocker 问题数		
high	High 问题数		
medium	Medium 问题数		
critical	Critical 问题数		
major	Major 问题数		
violation	Violation 问题数		
information	Information 问题数		
warning	Warning 问题数		
error	Error 问题数		
linecoveragerate	行覆盖率		
branchcoveragerate	分支覆盖率		
methodcoveragerate	方法覆盖率		
classcoveragerate	类覆盖率		
instructioncoveragerate	指令覆盖率		

#### 校验 Yaml,在 sonar-qube 文件夹下执行:

flow step valid

#### 没有任何输出表示校验通过。

### 后端

编辑 Dockerfile :

FROM registry.cn-beijing.aliyuncs.com/rdc-builds/oracle-jdk:1.8
MAINTAINER jintang <yidong.dyd@alibaba-inc.com>
COPY .step/*.sh /root/COPY sonar-qube/step.sh /root/step.sh
RUN chmod +x /root/*.sh
ENTRYPOINT [ "/root/entry.sh"]

#### 请确保基础镜像中已经<del>安装jq</del>。

编辑 sonar-qube 文件夹下面的 step.sh :

#!/bin/bash set -e

export SONAR_INFO=\$WORK_SPACE/sonar_info.json

#### # 系统提供参数,从流水线上下文获取

echo	[INFO]	PIPELINE_ID=\$PIPELINE_ID	#	流水线ID
echo	[INFO]	PIPELINE_NAME=\$PIPELINE_NAME	#	流水线名称
echo	[INFO]	BUILD_NUMBER=\$BUILD_NUMBER	#	流水线运行实例编号
echo	[INFO]	EMPLOYEE_ID=\$EMPLOYEE_ID	#	<b>触发流水线用户</b> ID
echo	[INFO]	WORK_SPACE=\$WORK_SPACE	#	/root/workspace容器中目录
echo	[INFO]	PROJECT_DIR=\$PROJECT_DIR	#	代码库根路径,默认为/root/workspace/code
echo	[INFO]	PLUGIN_DIR=\$PLUGIN_DIR	#	插件路径,默认为/root/workspace/plugins
echo	[INFO]	BUILD_JOB_ID=\$BUILD_JOB_ID	#	build-service <b>任务</b> ID

cd \$PROJECT_DIR

mvn install -DskipTests
mvn sonar:sonar -Dsonar.host.url=\$STEP_SONAR_HOST

STEP_PROJECT_GROUP_ID=`mvn -q -Dexec.executable=echo -Dexec.args='\${project.groupId}' --non-recursive exec:exec 2>/dev/null`
STEP_PROJECT_ARTIFACT_ID=`mvn -q -Dexec.executable=echo -Dexec.args='\${project.artifactId}' --non-recursive exec:exec 2>/dev/null`

curl \$STEP_SONAR_HOST/api/measures/search/?projectKeys\=\$STEP_PROJECT_GROUP_ID%3A\$STEP_PROJECT_ARTIFACT_ID\&metricKeys\=alert_status%2Cbugs%2Creliabi lity_rating%2Cvulnerabilities%2Csecurity_rating%2Ccode_smells%2Csqale_rating%2Cduplicated_lines_density%2Ccoverage%2Cncloc%2Cncloc_language_distribut ion > \$SONAR_INFO

STEP_SONAR_BUGS=`jq -r '.measures[] | select(.metric == "bugs") | .value' \$SONAR_INFO`
STEP_SONAR_VULNERABILITIES=`jq -r '.measures[] | select(.metric == "vulnerabilities") | .value' \$SONAR_INFO`
STEP_SONAA_SMELLS=`jq -r '.measures[] | select(.metric == "code_smells") | .value' \$SONAR_INFO`
STEP_SONAA_COVERAGE=`jq -r '.measures[] | select(.metric == "coverage") | .value' \$SONAA_INFO`

redline Bugs:缺陷:\$STEP_SONAR_BUGS:Error Vulnerabilities:漏洞:\$STEP_SONAR_VULNERABILITIES:Warning Smells:坏味道:\$STEP_SONAR_SMELLS:Warning Coverage:覆盖 率:\$STEP_SONAR_COVERAGE:Default Report:\$STEP_SONAR_HOST/dashboard?id=\$STEP_PROJECT_GROUP_ID%3A\$STEP_PROJECT_ARTIFACT_ID

- set -e 表示运行过程中遇到任何错误步骤即错误退出,随机流水线也会中断运行
- cd \$PROJECT_DIR 表示进入代码库所在路径
- mvn inst all -DskipTests 安装 maven 依赖
- mvn sonar:sonar -Dsonar.host.url=\$STEP_SONAR_HOST 执行 Sonar 扫描
- redline ...

redline 这个方法主要作用是将输入结构简单化,并会自动解析红线。以空格为分隔,每段字符包含<key>:<name>:<value>:<style>,报告格式为: Report:<url>。 key 要和step.yaml中红线定义的key一致

Style 展示的样式 Error: 红色,Warning: 橙色,Default: 灰

影响前端显示,最终会在日志里形成下列输出:



[stepIdentifier]	STAT_INFO_TITLE: Test
[stepIdentifier]	STAT_NAME_Bugs: 缺陷
[stepIdentifier]	STAT_VALUE_Bugs: 0 (1)
[stepIdentifier]	STAT_STYLE_Bugs: Error
[stepIdentifier]	STAT_NAME_Vulnerabilities: 漏洞
[stepIdentifier]	STAT_VALUE_Vulnerabilities: 0
[stepIdentifier]	STAT_STYLE_Vulnerabilities: Warning
[stepIdentifier]	STAT_NAME_Smells: 坏味道
[stepIdentifier]	STAT_VALUE_Smells: 20
[stepIdentifier]	STAT_STYLE_Smells: Warning
[stepIdentifier]	STAT_NAME_Coverage: 覆盖率
[stepIdentifier]	STAT_VALUE_Coverage: 0.0
[stepIdentifier]	STAT_STYLE_Coverage: Default
[stepIdentifier]	STAT_URLREPORT: http://47.89.50.196:9000/dashboard?id=com.adu%3Aspring-boot-hello-wor

左侧每一个元素对应右侧三行输出,分别为:

• STATE NAME xxx: 字段名称
• STATE_VALUE_XXX: 字段值

• STATE_STYLE_xxx: 字段样式(Error: 红色,Warning: 橙色,Default: 灰色)

STATE_URL___REPORT 为报告链接,在前端显示为报告按钮,点击调整到值对应的 Url。

如果想上传本地文件生成报告链接,可以在自定义步骤后追加"报告上传"步骤。

#### 本地运行

步骤的本质是一个容器镜像,因此可以在本地运行,flow cli帮助在本地模拟一个和流水线上运行相似的环境,包括挂载代码路径、缓存以及一些流水线上下文环境变量。在 sonar-qube 文件夹下执行:

flow step run

输入 STEP_SONAR_HOST , CHECK_REDLINES 和 workspace 三个变量,步骤即可在本地运行。

#### CHECK_REDLINES 举例:

[{\"identifier\":\"10_1581409320500_10_1581421770772_CHECK_REDLINES_0\",\"key\":\"Bugs\",\"type\":\"LE\",\"threshold\":0},{\"identifier\":\"10_158 1409320500_10_1581421770772_CHECK_REDLINES_1\",\"key\":\"Vulnerabilities\",\"type\":\"GE\",\"threshold\":2}]Bash

identifier为步骤唯一标识,可以是随机字符串, key 为待校验字段,type 为枚举字段,分别标识:LE: 实际值小于等于期望值;GE: 实际值大于等于期望值;EQ 实际值等于期 望值;thredshold 为期望值。

你可以通过日志输出观察运行情况以及在 \$WORK_SPACE/params 里是否得到预期的返回。

- 这个命令会执行 docker build, docker run 两个命令以在本地运行步骤。
- 输入的参数会换存到 sonar-qube/params.env 文件内,无需每次输出。
- 多次运行 maven 依赖会缓存到本地 ~/.m2 。
- 运行时输入的 workspace 对应到本地的一个绝对路径,会被 mount 到容器中 /root/workspace/code 路径下,这个路径在流水线运行时为可变路径,可以在 step.sh 中使用 \$PROJECT_DIR 变量获得

#### 企业内使用

在 sonar-qube 文件夹下执行:

flow step publish

该命令会执行 docker build, docker push 两个命令以发布步骤镜像,同时将步骤信息(yaml)同步到企业下的步骤列表。执行以下命令查看列表

flow step ls

同时进入企业流水线编辑页面,可以在 添加步骤->自定义步骤 中看到 SonarQube 这个步骤。

▼ 源(1) +	阶段1	新阶段	编辑		
21/spring-boot-hello-world			任务名称		
⊱ master	日 タ Sonar扫描	◎ 新的任务	Sonar扫描		
▼ 变量 (0)			任务步骤		
			分 添加步骤		
暂未设置,点击 添加			静态扫描		
			测试		
			代码		
			復盃平		
			部署		
			发布	自定义步骤	
			默认	自定义步骤	§下所有数据将会删除
			下载	SonarQube	<b>←</b> (1)
			上传	自定义步骤	
			自定义步骤	Wayne友仲-预友	

这个步骤可以和系统提供的其他步骤一样使用:

ATT人 金鑵 F始时间 2020-02-10 20:10:38	● ② 空任务
F始时间 2020-02-10 20:10:38	
持续时间 1分58秒	0         0         20         0.0           缺陷         漏洞         坏味道         覆盖率
	同报告 凹日志 1分58秒
<ul> <li>&gt; ∭(1)</li> <li></li></ul>	

## 跨企业共享步骤

## 开放步骤给所有 Flow 的企业,在 sonar-cube 所在目录下执行:

flow step public sonar-qubeflow step ls

SIGN	NAME	STATUS	CREATE TIME
sonar-qube	SonarQube	1 PROCESSING	2020-02-10 15:32:00
second	自定义步骤	PRIVATE	2020-02-09 22:11:38
test-step	自定义步骤	PRIVATE	2019-12-30 19:56:52
jintang-test	自定义步骤	REJECTED	2020-01-17 16:38:53
first	自定义步骤	PRIVATE	2019-12-31 19:46:39
wayne-staging	Wayne发布-预发	PRIVATE	2019-11-28 14:51:11
• etem flow stop le			
STEP FLOW STEP IS	ΝΔΜΕ	STATUS	CREATE TIME
sonar-aube	SonarOube	2 PRIVATE	2020-02-10 15:32:00
second	自定义步骤	PRIVATE	2020-02-09 22:11:38
test-step	自定义步骤	PRIVATE	2019-12-30 19:56:52
jintang-test	自定义步骤	REJECTED	2020-01-17 16:38:53
first	自定义步骤	PRIVATE	2019-12-31 19:46:39
wayne-staging	Wayne发布-预发	PRIVATE	2019-11-28 14:51:11
→ step flow step se	arch		
SIGN	NAME	STATUS	CREATE TIME
sonar-qube-103530	SonarQube	3 PUBLIC	2020-02-10 15:32:00
second-103530	自定义步骤	PUBLIC	2020-02-10 09:56:23
test-step	自定义步骤	PUBLIC	2019-12-30 19:56:52
first	自定义步骤	PUBLTC	2019-12-31 19:46:39

可以看到 sonar-cube 步骤的状态变为了 PORCESSING ,这个时候需要等待云效的工作人员进行人工审核,一段时间后步骤状态再次变为 PRIVATE ,使用命令 flow step search 可以查看到一个 sonar-qube-<regionId> 的步骤说明步骤被成功公开,如果审核不通过,会在云效页面右上角的通知栏里告知审核未通过原因。

步骤公开成功后得到两个步骤,一个为企业内可见的自定义步骤,sign为 sonar-qube ,一个为公开步骤,sign为 sonar-qube-<regionId> ,如果想要更新线上步骤,需 要再执行一次 flow step public ,所有步骤历史版本都会保留,通过 flow step version <sign> 可以查看步骤版本, flow step get <sign> -v <version> 可以查看 该步骤历史版本详细信息。

sonar-qube 显示在步骤编辑页中 自定义步骤栏 , sonar-qube-<regionId> 显示类别位置由 step.yaml 中的 category 字段决定:

▼ 源(1) +	阶段1	新阶段	编辑		
21/spring-boot-hello-world			任务名称		
}∞ master	日 ゆ Sonar扫描	◎ 新的任务	Sonar扫描		
▼ 全局变量(0) 🔗 +			任务步骤		
			● 添加步骤		
暂未设置,点击 添加			静态扫描	Java PMD 静态代码扫描-预发	
			测试	Python 代码安全扫描	
			代码	Python 代码安全扫描(预发)	
			覆盖率	Python 依赖安全扫描	
			何建	Python依赖安全扫描 (预发)	
			发布	Python 代码扫描	
			默认	TypeScript代码扫描(预发)	<b>š将会删</b> ]
			下载	SonarQube	-
			上传		
			自定义步骤		C

# 12.常见问题 12.1.构建问题

# 12.1.1. Java 构建失败—构建依赖缺失

## 错误说明

Flow 在进行 Java 构建时需要下载您的项目的构建依赖文件,Flow 会去<mark>阿里云 Maven 公库</mark>,下载公共的构建依赖,但是如果您的项目需要部分自己开发的依赖文件,Flow 会 下载失败并报错,典型的报错信息如下:

<b>最近运行</b> 运行历史		Java 构建上传	× )编辑	具 成员	1 三菜单
#3 👩 运行失败	构建	Java 构建(23s)	下载		
	申请运行环境(35s)	test/4.3.13.RELEASE/spring-test-4.3.13.RELEASE.jar (602 kB at 398 kB/s)	(h		
國行人	⊘ 清理工作区(1s)	core/1.3/hamcrest-core-1.3.jar (45 kB at 29 kB/s)			
F始时间 2020-10	⊘ 克隆代码(3s)	[16:50:40] [INFO] Downloaded from mirror: https://maven.aliyun.com/nexus/content/groups/public/org/springfra core/4.3.13.RELEASE/spring-core-4.3.13.RELEASE.jar (1.1 MB at 722 kB/s)	amew		
\$4# <del>8.1</del> 00	⊘ 流水线缓存(1s)	[16:50:40] [INFO] Downloading from rdc-releases: https://packages.aliyun.com/maven/repository/2001041-releases	se-		
239643101	🔇 Java 构建(23s)	JP4Muh/a/b/c/cpjcpj/1.0/cpjcpj-1.0.jar [16:50:40] [INFO]			
= 物	● 构建物上传	(16:50:40) [INFO] BUILD FAILURE			
> <u></u> (1)		[16:58:40] [INFO] Total time: 19.718 s [16:58:40] [INFO] Finished at: 2020-10-12715:58:40+08:00 [16:58:40] [INFO] Finished at: 2020-10-12715:58:40+08:00 [16:58:40] [INFO] (16:58:40] [INFOR] Failed to execute goal on project con-example:spring-boot-hello-world:jar:0.0.1- SMAPSHOT: Could not find artifact a.b.c:cpjcpj:jar:1.0 in mirror (https://maven.aliyun.com/nexus/content/gro > [Help 1] (16:58:40] [INFOR] To see the full stack trace of the errors, re-run Maven with the -e switch. (16:58:40] [INFOR] For more information about the errors and possible solutions, please read the following a [16:58:40] [INFOR] For more information about the errors and possible solutions, please read the following a [16:58:40] [INFOR] BUILD ERROR (16:58:40] [INFOR] BUILE FROM (16:58:40] [	pups. arti		

为了让 Flow 可以正常下载您的构建依赖文件并完成构建,您可以通过以下 3 种方式完成配置:

- 1. 将缺失的依赖包上传至云效制品仓库 packages
- 2. 将原有依赖包仓库添加至云效制品仓库 packages 的代理仓库
- 3. 使用企业自定义的 settings.xml 文件

#### 方法一:将依赖包上传至云效制品仓库 packages

如果用户项目代码库的根目录没有 Maven settings.xml 且没有选择企业自定义 Maven 配置, Flow 构建时会为用户自动生成一个 settings.xml 文件,并且配置连云效制品仓库 packages 和阿里云公共代理仓库。因此您可以将自己需要的依赖文件上传至 云效制品仓库 packages,具体上传操作步骤可登录参考制品仓库中的仓库指南。

## 方法二: 将原有依赖包仓库添加至云效制品仓库 packages 的代理仓库

如果您已有私有的 Maven 仓库,又不想将仓库内的依赖包文件手动上传至 <mark>云效制品仓库 packages,</mark>您可以将原有的依赖包仓库添加至云效制品仓库 packages 的代理仓库。 此时,Flow 在构建时就会从您设置的代理仓库中去拉取依赖文件。

您可通过云效制品仓库的仓库设置 -> 代理设置,完成代理仓库设置。建议您在生产库-release 和非生产库-snapshot 的代理仓库中均添加您原有的依赖包仓库。

首次上传	绑定代理仓库	×	<b>只成员</b> ② 仓库;
基本信息	代理 代理仓库央型* 当即 ○ 列星云公共仓库 ○ 私有仓库		
代理设置	仓库名* 调输入仓库名		
	仓库地址 * 调码入仓库地址		
	用户名		
	###U/10/10 密码		
		8	
	RCR	绑定	

# 方法三:使用企业自定义的 settings.xml 文件

如果您已有私有的 Maven 仓库,并且希望 Flow 在构建时可以直接从您的私有仓库中下载依赖文件,您可以使用企业自定义的 settings.xml 文件。当您启用企业自定义 Maven 配置时,Flow 按照您的 settings.xml 配置去进行 Java 构建。

		企业	设置	×			
☆ 我的星标	₹ ,风,角色设置	企业Maven设置			9, 5	? EU   (	)新建流水线
A 我参与的	B Maven设置	<ul> <li>云效默认 Maven I</li> </ul>			运行人	最近运行的	
🛱 全部流水线		云XLIKIA配重芸術総合 理仓库 maven.aliyun packages.aliyun.cor	a定生成 setting.xml 文件,默认: .com, 并自动引入您的云效 Maver 1	E按阿里云公共代 n 私库	▲ 陈融木	2020-05	同众
企业管理	-	<ul> <li>自定义 Maven 配置</li> </ul>	Ē		•		
. 通用变量组		若您需要引入其他 Ma 配置文件: O setting	wen 私库, 您可以上传自己的 M: g.xml	aven 配置文件。	🛞 陈鹏杰	2020-09	C ★
<ul> <li>□ 标签管理</li> </ul>		重新上传			🛞 陈鹏杰	2020-09	C ★
🗊 主机组管理					▲ 陈鹏杰	2020-09	同众
🗇 流水线模板管理	-				•		
☑ 任务组管理					(文) 陈鹏杰	2020-09	
回 步骤管理					(文) 陈鹏杰	2020-09	64
3 构建集群管理	java 1	× .	#1 - 📀	构建	▲ 陈鹏杰	2020-05	₫ ☆
愈 企业设置	□ 流水线 2020-08-3	1 -	#2 - 🕑	构建	🛞 陈鹏杰	2020-08	C &
						_	_
			< 1 2	3 4 >			

您可以在 Flow 的企业设置 -> Maven 设置中选择自定义 Maven 配置,并上传 settings.xml 文件,该企业内所有的 Java 构建均使用此配置文件。

## ♫ 注意

请确保 Flow 可以通过公网正常访问您的私有仓库,若您的私有仓库有白名单限制配置,请查看以下文档完成并将 Flow IP 地址配置到您的白名单中。

# 12.1.2. Java 构建缓慢—未设置有效缓存目录

## 问题说明

由于 Flow 流水线执行时,每个任务节点都会创建新的构建环境,构建完后会销毁,为解决流水线运行过程中,构建依赖反复下载的问题,提升整体构建效率,Flow 提供了自 定义缓存功能。用户可以自定义自己的缓存目录,实现流水线任务多次运行的缓存共享。

如果您未设置正确的缓存目录,会导致您每次进行 Java 构建时,即使您的依赖文件未发生任何变化,都会下载您项目所有的构建依赖。典型现象如下:

#### • 即使运行多次后,依然需要耗时很久

运行记录	状态	代码信息	运行详情	触发方式	运行人	持续时间	开始时间	操作
#3	0	master 4dd617e	构建	手动触发	🗶 光晕	47秒	10-22 12:01	立石
#2	0	master 4dd617e	构建	手动触发	🗶 光晕	47秒	10-22 11:57	宣看
#1	0	master 4dd617e	构建	手动触发	🗶 光晕	47秒	10-22 11:57	查看

• 即使运行多次后,依然要下载大量的构建依赖



## 加速方案

如果您的情况是以下任意一种:

- 没有使用企业自定义的 settings.xml (代码库根目录 or 企业自定义 Maven 配置)
- 使用了企业自定义的 settings.xml (代码库根目录 or 企业自定义 Maven 配置),但 settings.xml 中不包含自定义缓存目录节点 <localReposit ory>
  </localReposit ory>

```
请在 流水线编排 -> 变量与缓存 中添加 Maven 缓存目录 /root/.m2 。
```

```
♥ 注意
缓存目录填写正确且开关开启。
```

回 Java构建—有缓存	基本	信息 流程配置 触发设置 <b>变量和缓存</b>		仅保存	保存
安量通用变量组	缓存 Flow 会将不同任务工作区缓存目录下的	的构建依赖分别进行缓存,下次改任务执行时可!	央速下载缓存,以提高整体构建速度,	查看文档	
缓存	缓存目录			③ 添加缓存目录 📩 清理缓存	
	缓存目录	描述	是否开启	操作	
	/root/.m2	maven依赖缓存		①	
	/root/.gradle/caches	gradle依赖缓存		Ŵ	
	/root/.npm	npm依赖全局缓存		丗	
	/root/.yarn	yarn依赖全局缓存		Û	
	/go/pkg/mod	go mod缓存		Û	
	/root/.cache	其它缓存		Ť	

## 提速效果

增加缓存目录的流水线,只会在第一次构建时下载您所需的依赖文件,后续构建会直接从缓存中加载依赖。 项目的依赖文件越多,缓存的加速效果越明显。

进行记录         秋志         代田道息         通行详细         触发方式         运行人         排版时间         开始时间         排作           #3         •         master 4dd617e ⁴ / ₄ 手动触发         デ光牵         4359         10-22 12:01         重着           #2         •         master 4dd617e ⁴ / ₄ 手动触发         デ光 ⁴ 3959         10-22 11:57         重着           #1         •         master 4dd617e ⁴ / ₄ 手动触发         デ光 ⁴ 549         10-22 11:55         重着	操作 宣者 宣者
#3     master 4dd617e     物提     手动触发     光景     43秒     10-22 12:01     宣有       #2     master 4dd617e     物提     手动触发     第光第     39秒     10-22 11:57     宣有       #1     master 4dd617e     物提     手动触发     第光第     54秒     10-22 11:55     宣有	<b>立石</b> 立石 立石
#2         master 4dd617e         Nit         手动触发         第         39秒         10-22 11:57         重有           #1         ●         master 4dd617e         Nit         手动触发         第         54秒         10-22 11:55         重有	宣者
#1 master 4dd617e 构進 予劫触发 愛光愛 54秒 10-22 11:55 重着	宣看

# 12.1.3. Java 构建缓慢—删除缓存文件

## 问题说明

由于 Flow 流水线执行时,每个任务节点都会创建新的构建环境,构建完后会销毁,为解决流水线运行过程中,构建依赖反复下载的问题,提升整体构建效率,Flow 提供了自 定义缓存功能。用户可以自定义自己的缓存目录,实现流水线任务多次运行的缓存共享

当您在构建指令中使用 Im -rf 删除了部分缓存文件,会导致您每次构建过程中都会反复下载依赖文件,从而导致构建缓慢。

# 加速方案

如果您是担心依赖的 snapshot 包更新后,流水线无法及时拉取,可以修改您的构建指令如下:

mvn -B clean package -Dmaven.test.skip=true -Dautoconfig.skip -U

在构建指令中添加 -- 的作用:

- 对于 release 版本,缓存中已经存在,则不会重复下载
- 对于 snapshots 版本,不管缓存中是否存在,都会强制刷新,但是刷新并不意味着把jar重新下载一遍。只下载几个比较小的文件,通过这几个小文件确定本地和远程仓库的版本是否一致,再决定是否下载

# 12.1.4. Java 构建缓慢—自定义 Maven 配置

### 问题说明

由于 Flow 流水线执行时,每个任务节点都会创建新的构建环境,构建完后会销毁,为解决流水线运行过程中,构建依赖反复下载的问题,提升整体构建效率,Flow 提供了自 定义缓存功能。用户可以自定义自己的缓存目录,实现流水线任务多次运行的缓存共享。

当您使用了企业自定义的 settings.xml (代码库根目录 or 企业自定义 Maven 配置),且在 settings.xml 中 <localRepository></localRepository>自定义了缓存目录。如果您未将该目录添加至 Flow 流水线的缓存目录,则会导致您每次进行 Java 构建时,即使您的依赖文件未发生任何变化,都会下载您项目所有的构建依赖。

#### 加速方案

↓ 注意

您可以通过以下两种方案:

● 删除您企业自定义的 settings.xml (代码库根目录 or 企业自定义 Maven 配置)中 <localRepository></localRepository>节点,使用默认缓存目录,确保流水线编排-

```
> 变量与缓存 中添加 Maven 缓存目录 /root/.m2 。
```

#### 缓存目录填写正确且开关开启。 く返回 Java构建—有缓存 基本信息 流程配置 触发设置 变量和缓存 仅保存 保存并运行 变量 缓存 ow 会将不同任务工作区缀存目录下的构建依赖分别进行缓存,下次改任务执行时可快速下载缓存,以提高整体构建速度, 查看文档 通用变量组 缓存目录 添加缓存目录 📩 清理缓存 缓存 缓存目录 是否开启 操作 描述 /root/.m2 maven依赖缓存 /root/.gradle/caches gradle依赖缓存 ŵ /root/.npm npm依赖全局缓存 Ŵ /root/.yarn yarn依赖全局缓存 Ŵ /go/pkg/mod go mod缓存 Ŵ 其它缓存 /root/.cache Ŵ

• 将您企业自定义的 settings.xml (代码库根目录 or 企业自定义 Maven 配置) 中 <localRepository></localRepository> 节点的目录,添加到 流水线编排 -> 变量与缓存 的缓存目录中。请注意:缓存目录填写正确且开关开启。

# 12.1.5. Java 构建缓慢—下载海外依赖

## 问题说明

当您使用云效进行 Flow 进行 Java 构建时,Flow 支持通过三种方式设置 settings.xml 完成 Maven 配置:

• 代码库根目录 settings.xml 文件

- Flow 企业设置 -> Maven 设置中上传 settings.xml 文件
- 云效默认的 Maven 配置 settings.xml 文件
- 其中,第三种方式,云效默认 Maven 配置会帮你自动配置连接阿里云企业私有仓库和阿里云公共代理仓库。

若您通过前两种方式使用了企业自定义的 settings 文件(代码库根目录 or 企业自定义 Maven 配置),且 settings.xml 中未配置国内的代理仓库,会导致您所有公共的构 建依赖都从境外 Maven 中央库进行下载,从而导致构建缓慢。典型现象如下:

最近运行 运行历史	Java 构建上传	×	编辑	风 成员	≡ #
推定進行 進行初交 4 ④ 运行成功 4 ④ 运行成功 4 ① 运行成功 4 ① 运行成功 4 ① 运行成功 4 ① 运行成功 4 ① 正行成功 4 ① 正行成功 4 ① 正行成功 4 ① 正行成功 4 ① 正行成功 5 減代码(23) 5 減代码(23) ① 減水4億行(23) ① 加水 約((243s) ① 小陸物上传(3s) ④ 概存上传(2s) 2	Java Rig world [17:06:36] [INF0] Not copying test resources [17:06:36] [INF0] maven-compiler-plugin:3.1:testCompile (default-testCompile) @ spring-boot-h [17:06:36] [INF0] maven-surefire-plugin:3.1:testCompile (default-test) @ spring-boot-hello-world [17:06:36] [INF0] maven-surefire-plugin:2.18.1:test (default-test) @ spring-boot-hello-world [17:06:36] [INF0] maven-surefire-plugin:2.18.1:test (default-jar) @ spring-boot-hello-world [17:06:36] [INF0] maven-surefire-plugin:2.18.1:test (default-jar) @ spring-boot-hello-world [17:06:36] [INF0] maven-surefire-plugin:2.18.1:test (default-jar) @ spring-boot-hello-world [17:06:36] [INF0] maven-jar-plugin:2.5.9.RELEASE:repackage (default) @ spring-boot- hello-world [17:06:36] [INF0] spring-boot-maven-plugin:1.5.9.RELEASE:repackage (default) @ spring-boot-hello- hoot-loader-tools/1.5.9.RELEASE/spring-boot-loader-tools-1.5.9.RELEASE.pam (3.8 kB at 7 kB/s) [17:06:36] [INF0] Downloaded from central: https://maven.aliyun.com/nexus/content/groups/public/toot- hoot-loader-tools/1.5.9.RELEASE/spring-boot-loader-tools-1.5.9.RELEASE.jar (152 kB at 1.2 MB/s) [17:06:37] [INF0] Downloaded from central: https://maven.aliyun.com/nexus/content/groups/public/toot- hoot-loader-tools/1.5.9.RELEASE/spring-boot-loader-tools-1.5.9.RELEASE.jar (152 kB at 1.2 MB/s) [17:06:37] [INF0] Downloaded from central: https://maven.aliyun.com/nexus/content/groups/public/toot- hoot-loader-tools/1.5.9.RELEASE/spring-boot-loader-tools-1.5.9.RELEASE.jar (152 kB at 1.2 MB/s) [17:06:37] [INF0]	تی ello-world jar ello-world org/springframe org/springframe org/springframe	BN/3R	<i>,</i> м. т.	

#### 加速方案

优化您的企业 settings 文件(代码库根目录 or 企业自定义 Maven 配置),添加阿里云公共代理仓库,如下:

```
<mirrors>
<mirrors>
<id>mirror</id>
</mirrorOf>
<mirrorOf>central,jcenter</mirrorOf>
<mirrorOf>eurl>https://maven.aliyun.com/nexus/content/groups/public</url>
</mirror>
</mirror>
```

### 提速效果

#### 从阿里云公共代理仓库下载公共依赖,更快更稳定。



# 12.1.6. 镜像构建缓慢—从海外下载基础镜像

## 问题说明

当您使用云效进行镜像构建时,如果你的 Dockerfile 中,指定下载 dockerhub 的镜像,比如:

- From Nginx
- From java:8
- FROM openjdk:8-jdk-alpine

Flow 在构建时,就会从 Dockerhub 拉取镜像,由于海外网络问题,会导致整体构建速度极其缓慢。典型现象如下:



### 加速方案

按照以下步骤,修改您的 Dockerfile:

1. 将境外镜像在 pull 到本地。

docker pull openjdk:8-jdk-alpine

2. 将基础镜像 push 到阿里云镜像仓库(cr.console.aliyun.com)的国内 region(比如北京、上海等)。

docker tag openjdk:8-jdk-alpine registry.cn-beijing.aliyuncs.com/yournamespace/openjdk:8-jdk-alpine

docker push registry.cn-beijing.aliyuncs.com/yournamespace/openjdk:8-jdk-alpine

- 3. 修改你的 dockerfile 中 FROM,从你自己的镜像仓库下载镜像。
- From registry.cn-beijing.aliyuncs.com/yournamespace/openjdk:8-jdk-alpine

# 12.1.7. 构建物上传失败

#### 问题说明

当您的代码经过 Flow 构建后,生成了您的构建产物(例如:jar 包),需要先将构建物进行上传,后续部署任务(例如:主机部署 ECS)才可以获取构建物的下载地址并执行 部署。

在构建物上传的步骤中,您可以为制品选取一个名称,该名称在流水线后续阶段(比如部署阶段)会被引用到。此外,用户可以指定打包路径,该路径可以为一个文件或文件 夹,系统会将其打包为一个压缩包。如果用户想将多个文件或文件夹打包,也可以点点击"+"号添加更多的打包路径。

流水线编排 流水线 2	019-09-29					保存保存并运行	行
	● 新建源		构建	新的阶段	编辑		
bowenhuang/bowen2					Java 构建上传		
i⊷ master		lava 代码扫描	及 Java 构建上传	□ □ 新的任务	任务步骤		
	1	aven 单元测试	● 并行任务		分 添加步骤		
					▶ Java 构建		Ŵ
					→ 构建物上传		Ŵ
					步骤名称		
					构建初上传		
					制品名称		0
					default	(3	9
					打包路径		0
				2	target/	(†)	Θ
					任务插件		
					添加插件		
① 定时运行							

#### 当您的打包路径可能填写错误,会导致您的构建物上传失败,典型的报错信息如下:



典型错误场景: Java 项目中包含了多个微服务, Flow 的构建物上传步骤的打包路径中需要填写构建物的子目录。如下图,微服务 spring-boot-mp-demo 的构建物子目录在 spring-boot-mp-demo/target/目录下,需要将打包路径修改为 spring-boot-mp-demo/target/。您可以通过 Java 构建步骤的日志中,确认你需要上传的具体路径。

#3 ● 通行失数 #3 ● 通信大統	13         通行失敗         构建           四行人         0         0           五行人         0         0           市地时间         2020-11         0           市地时间         2020-11         0           市地时间         2020-11         0	Java 构 申请运行环境(25s) [10:37 清理工作区(1s) [10:37 克隆代码(3s) [10:37 克隆代码(3s) [10:37	i∰(22s) *:14] [INFO] *:14] [INFO] maven-compiler-plugin:3.8.1:testCompile (default-testCompile) @ spring-boot-mp-demo *:14] [INFO] Not compiling test sources *:14] [INFO]	下载	
<ul> <li>● 申谐治行环境(253)</li> <li>● 申谐治行环境(253)</li> <li>● 「第三行序気(15)</li> <li>● 「第三行音(15)</li> <li>● 「第三行号(15)</li> <li>● 「第三行号(15)</li></ul>	E行人 9 5 T始时间 2020-11 9 5	申请运行环境(25s) [10:37 [10:37 青理工作区(1s) [10:37 克隆代码(3s) [10:37 [10:37]	7:14] [INFO] 7:14] [INFO] maven-compiler-plugin:3.8.1:testCompile (default-testCompile) @ spring-boot-mp-demo 7:14] [INFO] Not compiling test sources 7:14] [INFO]		
第項工作文(1) 第項工作文(1) [NF0] maven-compiler-fulgin:3.8.1:testCompile (default-testCompile) @ spring-boot-mp-demo [10:37:14] [NF0] [NF0] maven-surefire-plugin:3.8.1:testCompile (default-testCompile) @ spring-boot-mp-demo [10:37:14] [NF0] [NF0] maven-surefire-plugin:2.22.2:test (default-test) @ spring-boot-mp-demo [10:37:14] [NF0] [NF0] maven-surefire-plugin:3.1.2:jar (default-jar) @ spring-boot-mp-demo [10:37:14] [NF0] [NF0] maven-surefire-plugin:3.1.2:jar (default-jar) @ spring-boot-mp-demo [10:37:14] [NF0] [NF0] maven-are-jurgin:3.1.2:jar (default-jar) @ spring-boot-mp-demo [10:37:14] [NF0] [NF0] maven-are-plugin:3.1.2:jar (default-jar) @ spring-boot-mp-demo [10:37:14] [NF0] [NF0] maven-are-plugin:3.1.2:jar (default-jar) @ spring-boot-mp-demo [10:37:14] [NF0] [NF0] maven-are-plugin:3.1.2:jar (default-jar) @ spring-boot-mp-demo [10:37:15] [NF0] Doublading from mirror: https://maven.aliyun.com/nexus/content/groups/public/org/springframe boot-loader-tools/2.2.7.RELEASE/spring-boot-laader-tools-2.2.7.RELEASE.pom [10:37:15] [NF0] Downloading from mirror: https://maven.aliyun.com/nexus/content/groups/public/org/springframe boot-loader-tools/2.2.7.RELEASE/spring-boot-loader-tools-2.2.7.RELEASE.jar [10:37:15] [NF0] Do	行人 O 5 f始时间 2020-11 O 5 f续时间	青理工作区(1s)     [10:37]       克隆代码(3s)     [10:37]       東水地域方(2a)     [10:37]	7:14] [JNFO] maven-compiler-plugin:3.8.1:testCompile (default-testCompile) @ spring-boot-mp-demd 7:14] [INFO] Not compiling test sources 7:14] [INFO]		
<ul> <li>202011 ● 克賀代利(3)</li> <li>202011 ● 克賀代利(3)</li> <li>202011 ● 克賀代利(3)</li> <li>202011 ● 克賀代利(3)</li> <li>202011 ● 白菜(1)</li> <li>202011</li></ul>	始时间 2020-11 0 1	克隆代码(3s) [10:33 [10:33 [10:33	14] [INFO] NOT COmpiling lest sources		
<pre>Main 2020-11 * 1.0410*103********************************</pre>	始时间 2020-11- 23 2020-11- 20 5 20 5 20 5 20 5 20 5 20 5 20 5 20	在水线频布(20)			
「加入性酸(r43)     「18:37:14] [INFO] Tests are skipped.     「18:37:14] [INFO]     「18:37:14] [INFO]     「18:37:14] [INFO]     「18:37:14] [INFO]     「18:37:14] [INFO] naven-jar_plugin:3.12;jar (default-jar) @ spring-boot-mp-demo     [18:37:14] [INFO] Building jar: /root/workspace/Sea4e7fce17cee0001fd99B@_my-java-demo_37Bp/pring-boot-mp-     iemo/target/spring-boot-mp-demo-0.6.1-SNAPSHOT.jar     [18:37:15] [INFO]     [18:37:15] [INFO]     [18:37:15] [INFO] spring-boot-maven-plugin:2.2.7.RELEASE:repackage (repackage) @ spring-boot-mp-demo     [18:37:15] [INFO] spring-boot-maven-plugin:2.2.7.RELEASE:repackage (repackage) @ spring-boot-mp-demo     [18:37:15] [INFO] Downloadd from mirror: https://maven.aliyun.com/nexus/content/groups/public/org/springframev     boot-loader-tools/2.2.7.RELEASE/spring-boot-loader-tools-2.2.7.RELEASE.jar     [18:37:15] [INFO] Downloadd from mirror: https://maven.aliyun.com/nexus/content/groups/public/org/springframev     boot-loader-tools/2.2.7.RELEASE.jar     [18:37:15] [INFO] Downloadd from mirror: https://maven.aliyun.com/nexus/content/groups/	续时间	AA 102 M T. M. C. M. C. LO. 1	14] [INF0] maven-surefire-plugin:2.22.2:test (default-test) @ spring-boot-mp-demo		
Image: Control (1997)       Image: Control (1997)         Image: Control (1997)       Image: Control (1997) <td>NCH 3 [m]</td> <td>ルハエス (10:37 [10:37</td> <td>(:14] [INFO] Tests are skipped.</td> <td></td> <td></td>	NCH 3 [m]	ルハエス (10:37 [10:37	(:14] [INFO] Tests are skipped.		
<ul> <li>● 物理地上作(is)         <ul> <li>● 物理地上作(is)             <ul> <li>● 物理地上作(is)                 <ul> <li>● 物理地上作(is)                      <ul></ul></li></ul></li></ul></li></ul></li></ul>	Ø.	Java 构建(22s) [10:3]	/:14] [INFO] /:14] [INFO] mayee_jar_p]ugin:3 1 2:jar (default_jar) @ spring_boot_mp_demo		
Image: Section 2016       Image: Section 2016         Image: Section 2016       Image: Section 2016 <td>物 😣 🕯</td> <td>构建物上传(1s) [10:37</td> <td>11] [INFO] Building jar: /root/workspace/5ea4e7fce17c0e0001fd99b0_my-java-demo_37Bp/spring-boot-mp</td> <td>-</td> <td></td>	物 😣 🕯	构建物上传(1s) [10:37	11] [INFO] Building jar: /root/workspace/5ea4e7fce17c0e0001fd99b0_my-java-demo_37Bp/spring-boot-mp	-	
<pre>18(1) [10:37:15] [INFO] [10:37:15] [INFO] [10:37:15] [INFO] Downloading from mirror: https://maven.aliyun.com/nexus/content/groups/public/org/springframe boot-loader-tools/2.2.7.RELEASE/Syring-boot-loader-tools-2.2.7.RELEASE.pom (2.5 KB at 13 kB/s) [10:37:15] [INFO] Downloading from mirror: https://maven.aliyun.com/nexus/content/groups/public/org/springframe boot-loader-tools/2.2.7.RELEASE/Syring-boot-loader-tools-2.2.7.RELEASE.pom (2.5 KB at 13 kB/s) [10:37:15] [INFO] Downloading from mirror: https://maven.aliyun.com/nexus/content/groups/public/org/springframe boot-loader-tools/2.2.7.RELEASE/spring-boot-loader-tools-2.7.RELEASE.pom (2.5 KB at 13 kB/s) [10:37:15] [INFO] Downloading from mirror: https://maven.aliyun.com/nexus/content/groups/public/org/springframe boot-loader-tools/2.2.7.RELEASE/spring-boot-loader-tools-2.7.RELEASE.jar [10:37:15] [INFO] Downloading from mirror: https://maven.aliyun.com/nexus/content/groups/public/org/springframe boot-loader-tools/2.2.7.RELEASE/spring-boot-loader-tools-2.7.RELEASE.jar [10:37:15] [INFO] Downloading from mirror: https://maven.aliyun.com/nexus/content/groups/public/org/springframe boot-loader-tools/2.2.7.RELEASE/spring-boot-loader-tools-2.7.RELEASE.jar [10:37:15] [INFO] Replacing main artifact with repackaged archive [10:37:15] [INFO] BUILD SUCCESS [10:37:15] [INFO] BUILD SUCCESS [10:37:15] [INFO] HUILD SUCCESS [10:37:15] [INFO] HUILD SUCCESS [10:37:15] [INFO] HUILD SUCCESS [10:37:15] [INFO] Finished at: 2202-11-09T10:37:15+08:00</pre>	0	要存上传 demo/1	target/spring-boot-mp-demo-0.0.1-SNAPSHOT.jar		
<pre>[18:37:15] [INFO] spring-boot-maven-puign:2.2.7.RELEASE:repackage (repackage) &amp; spring-boot-mg-demo [18:37:15] [INFO] Downloading from mirror: https://maven.aliyun.com/nexus/content/groups/public/org/springframew boot-loader-tools/2.2.7.RELEASE/spring-boot-loader-tools-2.2.7.RELEASE.pom [18:37:15] [INFO] Downloading from mirror: https://maven.aliyun.com/nexus/content/groups/public/org/springframew boot-loader-tools/2.2.7.RELEASE/spring-boot-loader-tools-2.2.7.RELEASE.pom [18:37:15] [INFO] Downloading from mirror: https://maven.aliyun.com/nexus/content/groups/public/org/springframew boot-loader-tools/2.2.7.RELEASE/spring-boot-loader-tools-2.2.7.RELEASE.pom [2.5 kB at 13 kB/s] [18:37:15] [INFO] Downloading from mirror: https://maven.aliyun.com/nexus/content/groups/public/org/springframew boot-loader-tools/2.2.7.RELEASE/spring-boot-loader-tools-2.2.7.RELEASE.jar ( 18:37:15] [INFO] Downloaded from mirror: https://maven.aliyun.com/nexus/content/groups/public/org/springframew boot-loader-tools/2.2.7.RELEASE/spring-boot-loader-tools-2.2.7.RELEASE.jar ( 18:37:15] [INFO] Downloaded from mirror: https://maven.aliyun.com/nexus/content/groups/public/org/springframew boot-loader-tools/2.2.7.RELEASE/spring-boot-loader-tools-2.2.7.RELEASE.jar ( 18:37:15] [INFO] Downloaded from mirror: https://maven.aliyun.com/nexus/content/groups/public/org/springframew boot-loader-tools/2.2.7.RELEASE/spring-boot-loader-tools-2.2.7.RELEASE.jar ( 18:37:15] [INFO] Meplacing main artifact with repackaged archive [18:37:15] [INFO] Multi Success [18:37:15] [INFO] Multi Success [18:37:15] [INFO] Total time: 11-970 s [18:37:15] [INFO] Total time: 11-970 s [18:37:15] [INFO] Multi Advance and advance andvance and advance and advance and advance and advance</pre>	• 源(1)	[10:37	:15] [INFO]		
<pre>tite:j:line() Goundong Trigm Interfactory Interfactory Content of Synthesis Content of S</pre>		[10:3]	/:15] [INFO] spring-boot-maven-plugin:2.2.7.RELEASE:repackage (repackage) @ spring-boot-mp-demo 2.15] [INFO] Downloading from mirror: https://mayon.aljvun.com/nevus/content/groups/owblis/org/corig	of ramo	
<pre>[10:37:15] [INFO] Downloaded from mirror: https://maven.aliyun.com/nexus/content/groups/public/org/springframew boot-loader-tools/2.2.7.RELEASE/spring-boot-loader-tools-2.2.7.RELEASE.pom (2.5 kB at 13 kB/s) [10:37:15] [INFO] Downloading from mirror: https://maven.aliyun.com/nexus/content/groups/public/org/springframew boot-loader-tools/2.2.7.RELEASE/spring-boot-loader-tools-2.2.7.RELEASE.jar [10:37:15] [INFO] Downloading from mirror: https://maven.aliyun.com/nexus/content/groups/public/org/springframew boot-loader-tools/2.2.7.RELEASE/spring-boot-loader-tools-2.2.7.RELEASE.jar [10:37:15] [INFO] Downloading from mirror: https://maven.aliyun.com/nexus/content/groups/public/org/springframew boot-loader-tools/2.2.7.RELEASE/spring-boot-loader-tools-2.2.7.RELEASE.jar [10:37:15] [INFO] Replacing main artifact with repackaged archive [10:37:15] [INFO] BuillD SUCCESS [10:37:15] [INFO] BuillD SUCCESS [10:37:15] [INFO] Fotal time: 11.970 s [10:37:15] [INFO] Finished at: 2020-11-09T10:37:15+08:00</pre>		boot-1	loader-tools/2.2.7.RELEASE/spring-boot-loader-tools-2.2.7.RELEASE.pom	girame	
boot-Loader-tools/2.2.7.RELEASE/spring-boot-Loader-tools-2.7.7.RELEASE.pom (2.5 KB at 13 kB/s)         [10:37:15] [INFO] Downloading from mirror: https://maven.aliyun.com/nexus/content/groups/public/org/springframe         boot-Loader-tools/2.2.7.RELEASE/spring-boot-Loader-tools-2.7.7.RELEASE.jar         [10:37:15] [INFO] Downloaded from mirror: https://maven.aliyun.com/nexus/content/groups/public/org/springframe         boot-Loader-tools/2.2.7.RELEASE/Syring-boot-Loader-tools-2.7.7.RELEASE.jar         [10:37:15] [INFO] Downloaded from mirror: https://maven.aliyun.com/nexus/content/groups/public/org/springframe         boot-Loader-tools/2.2.7.RELEASE/Syring-boot-Loader-tools-2.7.7.RELEASE.jar         [10:37:15] [INFO] Replacing main artifact with repackaged archive         [10:37:15] [INFO]		[10:37	7:15] [INFO] Downloaded from mirror: https://maven.aliyun.com/nexus/content/groups/public/org/spring	framew	
[10:37:15] [INFO] Downloading from mirror: https://maven.aliyun.com/nexus/content/groups/public/org/springframe         bot-Loader-tools/2.2.7.RELEASE/Syring-boot-Loader-tools-2.2.7.RELASE.jar         [10:37:15] [INFO] Downloaded from mirror: https://maven.aliyun.com/nexus/content/groups/public/org/springframe         bot-Loader-tools/2.2.7.RELEASE/Syring-boot-Loader-tools-2.2.7.RELASE.jar         [10:37:15] [INFO] Replacing main artifact with repackaged archive         [10:37:15] [INFO]		boot-1	loader-tools/2.2.7.RELEASE/spring-boot-loader-tools-2.2.7.RELEASE.pom (2.5 kB at 13 kB/s)		
boot-loader-tools/2.2.7.RELEASE/spring-boot-loader-tools-2.2.7.RELEASE.jar [10:37:15] [INFO] Downloaded from mirror: https://maven.aliyun.com/nexus/content/groups/public/org/springframew boot-loader-tools/2.2.7.RELEASE/spring-boot-loader-tools-2.2.7.RELEASE.jar (153 kB at 1.5 MB/s) [10:37:15] [INFO] Replacing main artifact with repackaged archive [10:37:15] [INFO] BUILD SUCCESS [10:37:15] [INFO] HUILD SUCCESS [10:37:15] [INFO] Total time: 11.970 s [10:37:15] [INFO] Finished at: 2820-11-09T10:37:15+08:00		[10:37	:15] [INFO] Downloading from mirror: https://maven.aliyun.com/nexus/content/groups/public/org/sprin	gframe	
[18:37:15] [UNFO] Downloaded from mirror: https://www.nalyou.com/neus/content/groups/public/org/springframew         boot-loader-tools/2.7.2.7.RELEASE.Sysring-boot-loader-tools-2.7.7.RELEASE.jar (153 kB at 1.5 MB/s)         [10:37:15] [UNFO]		boot-1	oader-tools/2.2.7.RELEASE/spring-boot-loader-tools-2.2.7.RELEASE.jar		
0001-100067-10015/2.2.7.NELEXS:SpT:37-0001-00067-10015-2.2.7.NELEXS:Jar (153 KB at 1.5 MD/S) [10:37:15] [INFO]		[10:37	15] [INFO] Downloaded from mirror: https://maven.aliyun.com/nexus/content/groups/public/org/spring	framew	
[10:37:15] [INFO] MEPICALING MULT AT LIAC WITH PERAKAGEO ATTIVE [10:37:15] [INFO]		D00T-	Loader-tools/2.2./.KELEASE/Spring-Doot-Loader-tools/2.2./.KELEASE.Jar (153 KB at 1.5 MB/s)		
[10:37:15] [INFO] BUILD SUCCESS [10:37:15] [INFO]		[10:3]	11) [INFO] HEROLAND HEIN ATTITATT WITH TEPAKAGEU ATTHIVE		
[10:37:15] [DHF0]		[10:37	113 [INFO] BUILD SUCCESS		
[10:37:15] [INFO] Total time: 11.970 s [10:37:15] [UNFO] Finished at: 2020-11-09710:37:15+08:00		[10:37	(INFO)		
[10:37:15] [INFO] Finished at: 2020-11-09T10:37:15+08:00		[10:37	:15] [INF0] Total time: 11.970 s		
		[10:37	:15] [INFO] Finished at: 2020-11-09T10:37:15+08:00		
[10:37:15] [INFO] Final Memory: 39M/441M		[10:37	:15] [INF0] Final Memory: 39M/441M		
[10:37:15] [INFO]					

当然,如果您需要上传多个微服务的构建物,也可添加多个打包路径,分别填写他们对应的子目录。

# 12.1.8. 本地调试优化

云效为用户提供的一定额度的免费构建环境资源,由于构建环境为系统自动生成分配,构建完成就会随之销毁。针对可能是由于构建环境导致的构建失败问题,云效提供了本 地调试优化工具。

该工具的工作原理为,在你的本机环境中,拉取云效流水线环境镜像,模拟与 Flow 上相同的容器环境,并直接在本机中进行命令调试。因此,使用该工具的前提需要在本地 安装 Docker。

# 步骤一:复制本地调试指令并执行

部分流水线任务执行失败后,可以查看本地调试工具。复制本体调试指令,并在本地执行。

<b>最近运行</b> 运行历史		☑ 编辑	● 运行	2 邀请	■菜
#3 😵 运行失败	Java 构建上传	×			
触发信息 🕺 陈鹏杰・页面手动船	本地调试工具				
开始时间 2022-07-12 14:56	复制以下命令,并在本地执行可在本地模拟与 Flow 上完全相同的构建环境,你可在本地进行调试排查具 体构建失败的原因,查看帮助	◎ 复制			
寺续时间 1.	docker volume createname works ename cache-				
午物	215101696&&docker runrm 5101696:/root/workspace -v cache- 215101696:/root/.m2 -v c 5101696:/ -v cache-215101696:// -v cach 215101696:/root/.m1 -v 5101696 -v cache-215101696:/ iod	he-			
▶ 源(1)	registry.cn-beijing.aliyuncs.com/build-stę text:1.4& :ker r -rm -v workspace- 215101696:/root/workspa / cac 2010/101696:/root/ root/ root/ 215101696:/root/ root/ cache 2151015696:/root/ root/	·v			
	cache-21510165 o/pkg/mod -e L="h ncs.com/execution- component/g initin am? msld 433 18-	# - V			
	bd1b37ae6c oker a69f 15ee e91≲ tam; 57610351121" registry.cn-beijing.aliyuncs.com/buil workspace.215101696./rc 'vorks +-215101 ^*/root/.m2 -v cache	v			
	215101696:/root/.gradle	che-			
	215101696/root/cache - , the-2 yd -e TASK_UF/ https://devope uncs /getRunTimeP/ paramsli 7.86de7-e0a6 38-9				
	21b4511e2f2c&token=5a 3352 faf9! 3b21 amp=165761 1124" regis beijing.aliyuncs.c⊂ uild- s/cac .205	try.cn-			
	215101696:/root kspa / cac 151/	N .			

由于第一次模拟环境,需要拉取构建环境镜像,可能会耗费较长的时间。

💿 😑 💼 docker-test — root@8c8459df45d2: ~/workspace/spring-boot_JgAC — com.doc
086f882ca53c: Pull complete
Digest: sha256:05ccb6a2ebfcb3e9eec2fd71505d3a7f81ec4f318bf2f08865c6b903b4625455
Status: Downloaded newer image for registry.cn-beijing.aliyuncs.com/build-steps/
clean-context:1.4
[14:37:01] [INFO] 开始清理工作区
[14:37:01] [INFO] 清理完成!
[14:37:01] [SUCCESS] 执行命令成功
Unable to find image 'registry.cn-beijing.aliyuncs.com/build-steps/clone-repo-li
st:2.2448-fed8849f' locally
2.2448-fed8849f: Pulling from build-steps/clone-repo-list
556ab54f6580: Pull complete
6ca8fd6ef32a: Pull complete
4d17f384d66a: Pull complete
26925a969bc7: Pull complete
d9b6054fade6: Pull complete
5bb88d94de42: Pull complete
5ea48da0f0e6: Pull complete
ef58178bca26: Pull complete
490ec50e9131: Pull complete
39c111b35152: Pull complete
9ca7d95245f2: Pull complete
4db8525ba71f: Pull complete
ddc6ddf5bc38: Pull complete
54fd1a965563: Pull complete

### 步骤二:本地调试

本地模拟环境,会在你失败的流水线步骤中停止,你可以查看所有的环境情况,并调试解决问题。如以下示例,流水线任务在"Java 构建"步骤执行失败,本地调试工具会进 入该步骤构建环境。



# 12.1.9. 自有构建集群缓存清理

当你使用自有构建集群,在运行一段时间之后,会产生较多的镜像缓存与依赖缓存需要进行清理。

- 我们提供的脚本会对以下内容做清理
- 镜像缓存
- Untaged的镜像,如采用相同的标签进行镜像构建会产生。
- 自定义环境构建过程中产生的中间镜像,即名称为custom-*的镜像
- 一些不再会被使用的步骤镜像

#### • 目录缓存

- 超过30天未运行的过的流水线工作区缓存
- 超过7天其他本地缓存

#### 清理命令如下:

docker run -v /var/run/docker.sock:/var/run/docker.sock -v /root/yunxiao/build:/root/yunxiao/build registry.cn-beijing.aliyuncs.com/build-steps/cache-cleaner:0.0.1

如果需要定期进行缓存,建议将以上命令配置到cronjob中。

# 12.1.10. Composer 版本升级

#### 云效默认 PHP 环境中,提供的 Composer 为 1.8.6。

您可通过添加以下指令将默认环境中的 Composer 版本快速升级至 2.1.5。

wget https://rdc-public-software.oss-cn-hangzhou.aliyuncs.com/composer/2.1.5/composer.phar -0 /usr/local/bin/composer



# 12.1.11. 流水线运行过程中定义环境变量

定义<mark>环境变量</mark>是实现流水线过程定制化的一种常见方法,可以在执行过程的任何阶段使用这些变量。

在流水线配置页定义的环境变量为预置的固定环境变量,在流水线实际生产使用过程中,也会出现通过流水线运行时的输出来定义新的环境变量,并将其传递给后续的步骤或 任务使用的场景。对应以下两种场景:

- 任务内环境变量传递:步骤一生成自定义环境 -> 步骤二使用
- 任务间环境变量传递:任务一生成环境变量->任务二使用

### 任务内环境变量传递

这种场景主要是单个任务节点内的环境变量共享,比如步骤一生成变量 USER_abc=123,步骤二通过 \${USER_abc} 引用。 您可以在前序步骤中通过将环境变量输出到 .env 文件中完成环境变量注入。注意: .env 文件中的环境变量需要以 USER_开头。

く 返回 环境变量测试流水线	基本信息 流程配置	触发设置 变量和线	愛存	仅保存保存并运行
源命令	?	新阶段	编辑 壶	×
十添加代码源	3 任务-	● 新的任务	步骤一:执行命令	
			执行命令 1 # input your command here 2 echo 'USER_abc=123' > .env	μ ^N
			变量 十 添加参数	
			→ 步骤二:执行命令	٥
			步骤名称 步骤二:执行命令	
			执行命令	Ľ ⁷
			2 echo \${USER_abc} 3	

## 任务间环境变量传递

这种场景主要是流水线内多个任务节点内的环境变量共享,比如任务一生成变量 USER_abc=123,任务二通过 \${USER_abc} 引用。

- 1. 您可以在任务一的步骤中通过将环境变量输出到 .env 文件中完成环境变量注入。注意: .env 文件中的环境变量需要以 USER_ 开头。
- 2. 在任务一中添加步骤"设置变量",将环境变量设置为流水线级别的环境变量。
- 3. 在任务二中直接使用该环境变量。



# 12.2. 部署问题

# 12.2.1. 主机部署—环境变量包含特殊字符

## 错误说明

Flow 经过优化后,新的主机部署不会再对环境变量进行默认编码,如果你的环境变量中不包含特殊字符(空格、引号、分号等),您可以在部署脚本中通过 \${key} 使用环 境变量,无需再进行 Base 64 解码处理。 但是如果你的环境变量中包含特殊字符,您将无法正常使用您的环境变量,典型错误如下,环境变量。由e 中包含特殊字符空格,导致环境变量打印出错:

最近运行 运行历史	主机部署	×	⊘编辑 ●	) = 莱
	Bit B La      Consert _LOTAD      Conser			

## 解决方案

因为您的环境变量中包含了特殊字符,所以需要通过以下操作对流水线进行配置,才可正常使用环境变量

- 1. 主机部署任务中,勾选"变量进行编码"。
- 2. 在部署脚本中,把您需要使用的所有环境变量进行 Base 64 解码。比如要使用 PIPELINE_ID 的环境变量,在部署脚本最前面添加一行

```
export PIPELINE_ID=$(echo $PIPELINE_ID | base64 -d)
```

## 示例:

〈 返回 预发base64测试流水线 基本信息 流程配置 触发设置 变量和缓存 仅保存 构建 部署 编辑 🗇 执行用户 ⑦ flow-example/spring-boot ⑤ 新的任务 root 🦳 🎠 master 部署脚本 ⑦ 十添加代码源 2 64 d ☑ 变量进行编码 部署策略 暂停方式 第一批暂停 分批数量 2 任务插件 〇 添加插件

<b>最近运行</b> 运行历史	主机部署	×	∂ 编辑	8	≡菜
6 🕑 运行成功	部長神 数量日空				
行人 始助詞 2020-12- 域助詞 物 * <b>遼(1)</b> ・ 安麗(1)	Export Part Jeon - DVT 10:07     export Part INE NAME easkESY+RYmF2ZTY05/WL 6k+V5rW85rC057q/     export Part INMBERAIge=     export Statistic Part Part Part Part Part Part Part Part				
	+ export C_COMUS_SHAA=Mm2NHAM IROVINYGW2D220D27D2/2DBM2E2Y2M1722yMm22M1E4NQ== + C_COMUMT_SHAA=Mm2NHTMWTRIOWFYGw2D22ND2j2DBIM2E2Y2M1Y2QyMm22MTE4NQ== + export machine_group_id=MTE5MDA= + export fx0iHE_PPELINE_INST_UD=MJE3N2Y3MW== + EXDINE_PPELINE_INST_UD=MJE3N2Y3MW== 				
	exit 0 exit 0 ] DeployCommand execution completed				

# 12.3. 管理功能问题

# 12.3.1. 授权管理--添加多个阿里云账号的授权服务

当你的企业中包含多个阿里云账号时,而每个账号都有阿里云的服务资源,云效支持将多个账号的阿里云资源添加至同一个企业。 比如,您的企业有账号 A 和账号 B,两个账号下都有 ECS 资源,您可通过以下方式,将 A 和 B 的 ECS 都加入您的企业,并通过云效流水线进行发布: •通过企业管理后台 -> 成员 -> 添加成员,把账号 A 和 B 都加入到您的企业。

· 基础配置	新加入		邀请您的团队成员加入企业	×			点击查看
基本信息 <b>2、成员配置</b>	成员	成员邀请链接 添加阿里云RAM用户	如何同步钉钉企业组织和成员?宣誓详情		步钉钉企业组织和成员?	十添加成	1页 🛛 企业成员
成员							
组织架构	企业成员		*				· 一 · · · · · · · · · · · · · · · · · ·
信息隐藏							U WHAT SAAR
成员信息配置					角	色了	操作
会 权限管理			邀请链接		拥	有者	
◎ 项目配置			https://account-devops.aliyun.com/invite-from-				
☞ 应用管理			sig.,-32c0 Jad )e		Ê	理员 ~	
○ 安全管理					成	员 ~	
					成	员 ~	
		<ul> <li>链接有效期至 2020</li> </ul>	12-17 18:09	刷新链接复制链接	管	理员 ~	
		<b>雨知</b> 未设置却箱	测试	DevOps —始武 ~	管	理员 ~	
		张莉0123	客户成功、测试、子部门1	DevOps 一站式 ~	1	理员 ~	

• 账号 A 和 B 分别登录云效,并在 服务连接管理 中分别添加 A 和 B 的 ECS 服务连接。

• 授权范围设置为所有人可见,或者设置为指定成员可见。

〈 企业设置					×
工具箱	服务连接管理 ②		_	Q P	新建服务连接
通用变量组	服务连接名	〈 新建服务连接	×	创建时间	操作
标签管理		连接类型		2022-01-17 13:56:43	r t
服务连接管理		云服务器(ECS)	~		
主机组管理	用Git服务计	服务连接名		2021-12-23 21:44:00	C T
Kubernetes集群管理	用Git服务)	]云服务器(ECS)服务连接	17/64	2021-12-23 21:43:59	ľ t
流水线模板管理	的云效Codeu	服务连接 ID X3I uqu, yu,Zt	16/16	2021-11-22 10:52:25	C T
任务组管理	hahaha	服务授权/证书	1. 20572	2021-11-22 10:22:32	c t
构建集群管理	с	使用范围	1 771 822	2021-11-22 10:12:38	C T
	b	<ul> <li>公开:所有人可见</li> <li>私密:仅自己可见</li> <li>自定义:指定人员可见</li> </ul>		2021-11-22 10:12:12	¢ ¢
管理员设置	а			2021-11-22 09:51:59	c t
角色设置 Maven设置		上一步	刘建		
				1 2 3	4 5 >

## • 服务连接创建完成后,在您的企业下,可通过切换服务连接,选择不同账号下的 ECS。

< 企业设置						×
工具箱	主机组管理 ②	< 阿里云ECS · 添加主机	×		ςγ	=↓ ◎ 新建主机组
<ul> <li>工具箱</li> <li>通用变量组</li> <li>标签管理</li> <li>服务连接管理</li> <li>主机组管理</li> <li>还此级带管理</li> <li>试。然线频管理</li> <li>行务组管理</li> <li>行务组管理</li> <li>行资编管理</li> <li>行渡管理</li> <li>行复集管管理</li> <li>行复集管管理</li> </ul>	<b>主机组管理 ③</b> <ul> <li>主机组名称</li> <li>新的主机组</li> <li>新的主机组</li> <li>新的主机组</li> <li>新的主机组</li> <li>新的主机组</li> <li>潮试主机组</li> </ul>	《 阿里云ECS - 添加主机 添加方式 ● 直接添加 ● 标签方式添加 (通过ECS标签快速定位主机) 服务连接 ● 63.com的分型服务器(ECS)服务连接 ● 63.com的分型服务器(ECS)服务连接 ● 可添加的主机列表 ●	× ③ 什么是ECS标签 ④ 添加服务连接 查看 编辑 查看 编辑	·顯杰 ·顯杰 ·顯杰	<ul> <li>Q ●</li> <li>●</li> <li>●<!--</th--><th>E ● 新建主机组       ③       ③       ④       ④       ④       ①       ①       ①       ①       ①       ①       ①</th></li></ul>	E ● 新建主机组       ③       ③       ④       ④       ④       ①       ①       ①       ①       ①       ①       ①
角色设置 Maven设置		请先选择地域 ● 暂不支持购地主机	上一歩 下一步			¢ 1 )

♫ 注意

:如果您的账号为 RAM 子账号,可能会出现授权失败的情况,请联系 RAM 主账号,为您的子账号添加 AliyunRAMFullAccess 权限。

# 云效2020

☰ (-) 阿里云					Q 搜索文档、控制台、API、解决	表方案和资源 费用	工単	备案 企)	业 支持	官网	<u>&gt;_</u>	Ō. Ĕ	1 (1)	简体	0
RAM 访问控制	RAM 访问控制 / 授	树又			添加权限										×
概览	授权				单次授权最多支持5条策略,5	如需绑定更多策略,请分多	次进行。								
人员管理へ	新増授权	用户名 🗸 请输入			* 授权范围										
用户组	权限应用范围	被授权主体	主体类型	权限策略名称	<ul> <li>云账号全部资源</li> </ul>										
用户	全局	AliyunCodePipelineDefaultRo le@role.1218858987274713.	RAM角	AliyunCodeP	· 指定资源组 请选择或输入资源组名称进行搜索										<i>y</i>
设直 SSO 管理		onaliyunservice.com	E		* 被授权主体										
权限管理へ	全局	ultRole@role.1218858987274 713.onaliyunservice.com	RAM角 色	AliyunContai	test@1218858987274713.ona	aliyun.com 🗙									
授权	全局	AliyunCSClusterRole@role.12 18858987274713.onaliyunse rvice.com	RAM角 色	AliyunCSClu	<ul> <li>选择权限</li> <li>系統策略</li> <li>自定义策略</li> </ul>	+ 新建权限策略				E	选择 (1)			清空	
权限策略管理 RAM 角色管理	全局	AliyunCSDefaultRole@role.12 18858987274713.onaliyunse	RAM角 色	AliyunCSDef	ram				8	Aliy	/unRAMFu	IIIAccess		×	
OAuth 应用管理(公测中)		rvice.com			权限策略名称 AlivunRAMFullAccess	管理访问控制 (RAM	的权限。	の管理用户以	及授权的						
	全局	sRole@role.12188589872747 13.onaliyunservice.com	RAM角 色	AliyunCSMar	AliyunRAMReadOnlyAccess	只读访问访问控制(F	IAM) 的权	限,即查看用	户、组以…						
	全局	AliyunCSServerlessKubernet esRole@role.1218858987274 713.onaliyunservice.com	RAM角 色	AliyunCSSer	AliyunECDRamUserAccess	RAM用户管理弹性云	桌面(ECD)的	的权限。							
	全局	AliyunEDASDefaultRole@rol e.1218858987274713.onaliyu nservice.com	RAM角 色	AliyunEDASR											
	全局	AliyunMPCADefaultRole@rol e.1218858987274713.onaliyu	RAM角 色	AliyunMPCAI	确定 取消										

# 12.3.2. 授权管理—RAM 账号授权失败

如果您的账号为 RAM 子账号,可能会出现授权失败的情况,请联系 RAM 主账号,为您的子账号添加 AliyunRAMFullAccess 权限。

					Q 搜索文档、控制台、API、解决	央方案和资源 费 <b>用</b>	工単 备案	企业	支持	官网 区	а ф <b>.</b>	₩ 0	简体	0
RAM 访问控制	RAM 访问控制 / 损				添加权限									×
概览	授权				单次授权最多支持5条策略,5	如需绑定更多策略,请分多	次进行。							
人员管理    ^	新增授权	用户名 🗸 请输入			* 授权范围									
用户组	权限应用范围	被授权主体	主体类型	权限策略名称	<ul> <li>云账号全部资源</li> <li>集字资源组</li> </ul>									
用户设置	全局	AliyunCodePipelineDefaultRo le@role.1218858987274713. onaliyunservice.com	RAM角 色	AliyunCodeP	请选择或输入资源组名称进行搜索								`	<i>y</i>
SSO 管理 权限管理 へ	全局	AliyunContainerRegistryDefa ultRole@role.1218858987274 713.onaliyunservice.com	RAM角 色	AliyunContai	* 被授权主体 test@1218858987274713.ona	aliyun.com 🗙								
授权	全局	AliyunCSClusterRole@role.12 18858987274713.onaliyunse rvice.com	RAM角 色	AliyunCSClu:	<ul> <li>选择权限</li> <li>系統策略</li> <li>自定义策略</li> </ul>	十 新建权限策略				已选择	₿ (1)		清空	
化脱汞酚香油 RAM 角色管理	全局	AliyunCSDefaultRole@role.12 18858987274713.onaliyunse rvice.com	RAM角 色	AliyunCSDef	ram 权限策略名称	备注			ø	Aliyunl	RAMFullAcce	55	×	
OAuth 应用管理(公测中)	全局	AliyunCSManagedKubernete sRole@role.12188589872747 13.onaliyunservice.com	RAM角 色	AliyunCSMar	AliyunRAMFullAccess AliyunRAMReadOnlyAccess	管理访问控制(RAM) 只读访问访问控制(F	的权限,即管理 IAM)的权限,即	用户以及授权 查看用户、组	R的	]				E?
	全局	AliyunCSServerlessKubernet esRole@role.1218858987274 713.onaliyunservice.com	RAM角 色	AliyunCSSer	AliyunECDRamUserAccess	RAM用户管理弹性云	桌面(ECD)的权限	•						
	全局	AliyunEDASDefaultRole@rol e.1218858987274713.onaliyu nservice.com	RAM角 色	AliyunEDASR										
	全局	AliyunMPCADefaultRole@rol e.1218858987274713.onaliyu	RAM角 色	AliyunMPCAI	稿定 取消									

# 12.3.3. 通过钉钉进行人工卡点审批

在 Flow 流水线中,你可以通过增加人工卡点组件,实现发布前的审批确认。 目前 Flow 的人工卡点组件已经打通了钉钉消息,你可以直接在钉钉客户端上操作流水线审批任务。

步骤一: 绑定钉钉企业和个人信息

企业绑定:企业绑定-完成组织架构和成员同步

个人绑定:个人绑定-加入企业并接收钉钉消息

# 步骤二:流水线添加人工卡点并执行流水线

在流水线编排中,添加人工卡点组件。

<b></b>	新阶段	选择任务组		
+添加流水线源	今         人工卡点           ● 并行任务	企业任务组 代码扫描 测试 构建 测试构建 部署 工具 代码 执行命令 空模板	请输入任务名称         ▲ppStack题署         工具         ①         【         人工卡点         人工卡点         人工卡点         化码作的日本上         将代码库中的 Hel         代码         〇〇         角井代码         第代码输入逐中描	替換文件中的环 替换文件中的环境                 登録                 登録                 登録                 登録                 登録                 登録                 受支管理器                 分支管理器                 分支管理器                 分支管理器                 分支管理器

选择具体的验证方式、验证者类型和验证人,保存触发流水线运行。

		+/合自 法101100 8+10330 00	鼻初硬左	
<ul> <li>区回 流水线 2022-06-21</li> <li>3.3.3.3.3.3.3.3.3.3.3.3.3.3.3.3.3.3.3.</li></ul>			里仙波仔 <b>/全報</b> ①7	
<23, 113.	利用日本		湖田 2日	~
十添加流水线源		新約任务	任务名称	
1 Januar July 2 Sounds		WIN SIL25	人工卡点	
			验证者方式	
			或签 (一名审批人同意或拒绝即可)	~
			验证者类型	
			成员 (选择企业成员作为审核人员)	~
			验证人	
			陈鹏杰 smytest	
			任务插件	
			● 添加插件	
#4 2 等待中 ⑧ 终	新阶段			
触发信息 ▲ 陈鹏木,页面毛动				
开始时间 2022-06-23 10:25	7:24 WHA			
	*24	E		
持续时间	<ul> <li>6秒</li> <li>⑥ 验证不通过</li> <li>⑥ 验证通过</li> </ul>	5秒		
产物	0项			
钉钉审批消息,进行验证操作。				
10:27				
2 审批				
流水线人工卡点审批				
流水线名称: 流水线 2022	2-06-21			
任务名称:人工卡点				
流水线运行记录:#4				
审批状态:待审批				
待审批人: 陈鹏杰, smyte	st			

查看详情

验证不通过 验证通过

# 12.4. Flow IP 地址

当您的代码仓库开启白名单功能时,需要在白名单中分别设置 Flow 的 IP 地址 和 Flow 的构建机 IP 地址。

Flow 的 IP 地址:设置完成后,才可在流水线编排中设置对应代码源。

Flow 的国内/中国(香港)构建集群 IP 地址:设置完成后, Flow的构建机才可完成对应仓库的代码克隆。

Flow IP 地址	112.126.70.240、123.56.255.38、47.94.150.88
Flow 国内构建集群 IP 地址	47.93.89.246、47.94.150.17
Flow 中国(香港)构建集群 IP 地址	47.57.70.87、47.242.65.197、47.90.29.115

# 12.5. 部署常见问题

# 阿里云ECS部署失败解决方案

### 首先需要确认信息

### 查看云助手版本: aliyun-service -v

```
如果是1.00.1404之前的版本,请升级云助手
wget https://aliyun-client-assist.oss-accelerate.aliyuncs.com/linux/aliyun_assist_latest.rpm
rpm -ivh --force aliyun_assist_latest.rpm
```

#### 1. 部署详情显示失败, 如何排查?

在部署详情中,显示具体的部署错误码,不同的错误码对应的错误说明和错误排查如下:

User.NoPermission

用户没有调用某个API的权限。请检查部署组的服务链接是否正常。

#### 2.获取部署日志失败

- 参照停止或启动云助手客户端启动云助手服务
- 3.机器状态离线 (deploy channel error)
- 阿里云ECS参照升级或禁止升级云助手客户端先升级云助手,再参照停止或启动云助手客户端启动云助手服务
- 私有主机参照添加Agent失败FAQ解决
- 4.部署命令在机器上执行成功,但Flow上部署失败
- 添加与命令相关的环境变量,例如: "source /root/.bash_profile;source /etc/profile;"。
- 部署相关脚本尽量使用绝对路径,例如"/home/admin/app/deploy.sh",不要使用"./deploy.sh"。
- kill 命令中类似:"ps -ef | grep at hens | grep -v grep" 需要政为:"ps -ef | grep at hens | grep -v grep | grep -v rdc_deploy_command"。
- 5.主机磁盘满导致部署失败
- 登录你的主机执行 df -hl 查看磁盘剩余空间,若磁盘空间已满,请清理磁盘空间。

### 自有主机部署失败解决方案

参见:添加Agent失败FAQ

# 12.6. 错误: "超时取消"问题排查

在流水线先执行中,遇到超时终止的问题,用户可以按照文档进行相应的排查。

	> 超时kill的流水线 201	e.	
最近	运行 运行历史		
87 🧧 运行失败		除段1 ② 超时kill的任务空任务 运行失意	анян Э
	<b>ジョンです</b> 智光数解測	C en C He	181978
运行人 开始时间	🧐 柳夕 2019-10-21 21:04:11		
运行时长	1时1公8秒		

Java 构建 jar 包下载卡住

日志中出现 jar 包下载然后不继续往下执行。出现这种情况是由于 jar 包下载失败,首先判断使用 maven 源是否为国内镜像源,推荐使用 aliyun 提供的<mark>mave镜像仓库</mark> 进行依 赖下载加速。

#### NodeJS 构建 Npm 包下载卡住

日志中 npm install 后不继续往下执行,可能是由于使用海外 npm 镜像源,使用 cnpm 替换 npm 可以切换到使用 淘宝 NPM 镜像。

#### Golang 构建依赖下载卡住

日志中出现依赖下载后不继续执行,可能是由于使用了海外的 go 镜像源。阿里云提供了官方的 Go Module <mark>代理仓库服务</mark>,来避免模块拉取缓慢或失败的问题,加速构建。

#### Docker 构建出现 FROM:... 卡住

这种情况是由于 docker 构建的基础镜像下载不下来,可以把 FROM 的镜像 pull 到本地,再 push 到阿里云镜像仓库 加速镜像构建。

#### 其他依赖如 tar 包无法下载

可以考虑将依赖上传到 阿里云OSS,再生成外部下载链接下载。

#### 其他问题

检查是否在脚本中执行了耗时超过一小时的操作。我们建议一个步骤的功能尽可能简单,如果有耗时过长的步骤导致任务超时,请将其拆分为多个步骤且分别在不同任务中进 行。

# 12.7. 构建常见错误排查

## 1. Maven 依赖下载报错,提示:"上传的包有问题或者找不到"

这种情况通常是用户使用二方库依赖导致,排查顺序:首先确认下载依赖使用的私有仓库地址(默认的 settings.xml 文件在 ~/.m2/settings.xml 下),再登录到对应仓库查看 该依赖是否存在。

如果依赖存在且使用的是云效提供的私有仓库(repo.rdc.aliyun.com),可以寻求云效答疑帮助。

#### 2. Go 镜像源访问报错,提示: "unrecognized import path "

这种情况通常是引文海外 Go 镜像源存在访问不稳定的情况。建议将镜像源改为使用阿里云提供的 Go 镜像源: https://mirrors.aliyun.com/goproxy/

#### 3. 代码下载报错,提示: "Clone Failed"

这种情况通常是触发流水线的没有代码的访问权限,或者输入源中配置的分支不存在,请检查相关配置。

## 4. npm 构建报错,提示: "找不到 module"

这种情况通常是没有执行 cnpm inst all , 请检查相关任务的执行命令中是否有 cnpm inst all , 没有请加上。

## 5. 构建报错,提示:"上传文件路径不存在"

请检查本地执行构建的构建产物路径和流水线中配置的产物上传路径是否一致。

#### 6. 日志显示: pom.xml 文件不存在

请检查代码是否克隆成功,如果克隆成功,请确认代码库根目录存在 pom.xml 文件。

#### 7. 创建Docker build 提示没有权限

请检查 Dockerfile 中的 FROM 镜像是否为公共镜像。

### 8.Docker build 的时候找不到 COPY 的文件

请检查 Dockerfile 中使用到的文件在构建后是确实存在的,且 Dockerfile 写的相对 context path 的路径配置正确。

#### 9.Docker build 的时候克隆代码失败

请检查 Dockerfile 中克隆的代码库是否为公共代码仓库,如果不是需要在 Dockerfile 或基础镜像中对私有代码仓库进行授权。

## 10.上传文件的时候试图使用正则匹配,例如: */target/*.jar

上传文件路径不支持正则匹配,需使用 target/或者 target/app.jar 这种格式。

## 11.如何在镜像构建中获取 Java 构建的构建产物

Flow 流水线内每个任务节点之间工作区相互独立,需要将 Java 构建步骤和镜像构建步骤放在同一个任务节点,可选择一下任务组。

# 云效流水线 Flow·常见问题

< 返回 oss测试		基本信息 流程配置	触发设置 变量和缓存			仅保存保存并运行
源	执行命令	新阶段		选择任务组		×
+添加代码源	4 执行命令	節約任务	)	代码扫描 测试 测试均建 构建 镜像构建 工具 影署 代码 执行命令 空模板	調論入任务名称         阿里云镜像构建         ACR 阿里云镜像构建         副定义镜像构建         副定义镜像构建         Python 镜像构建         Python 桃屋后,         Co 镜像构建         Go 构建后, 阿里         PhP 镜像构建         PHP 機變為美         PHP 機變為美         PLT卡点	ア里云镜像企业版         ARC EE 构建镜像-J         Java 核爆角建         Java 核進后、阿里         NET Core镜像         .NET Core镜像,         Node.js 核爆角建         Node.js 核凝后,         Node.js 核凝后,         Mode.js 核凝后,         R代码库中的 Hel

# 云效2020

# 13.迁移 13.1.从CodePipeline迁移

## 开始迁移

## 创建云效企业

访问<mark>云效</mark>平台,并创建一个企业

导入流水线

访问<mark>迁移页面</mark>



点击"开始导入"



绑定阿里云账号,并加载任务

		迁移CodePip	eline项目到Flow			
	<ul> <li>—</li> </ul>		2		3	
	导入方式	Ę	入配置		导入结果	
I 😒	在使用: cui.liqiang	@gmail.c 🗸			绑定新账号	解绑
CodeF	Pipeline任务列表:			上一步	加载任务	开始迁移
	任务名称	编号				备注
•	job1	11111-35e9-4191-8e9f-b3	a53ea35e9b			
	job2	22222-8bd1-4ca2-8051-c	cb36bca0d2c			

选择您需要的项目,进行批量迁移

<ul> <li>——</li> </ul>		<ul> <li>—</li> </ul>	<u> </u>
导入方式	Ę	异入配置	导入结果
当前任务状态			
job1			✓ 查看 1
job2			✓ 查看
上一步			
历史任务状态状态: 2			
任务名称	状态	信息	

导入成功后,您可以查看生成的飞流流水线。同时也可以看到该阿里云账号之前运行过的历史任务。

#### 对流水线进行微调

本节介绍对生成的流水线可能进行的微调 **代码源** Flow 支持多种<del>流水线源</del>,迁移的任务会默认生成"Git"类型的代码源,您可以根据需要进行切换。如果你选择继续使用"Git"类型的代码源,则需要重新创建代码证书

编辑代码源	×
选择代码源	
代码仓库 ⑦ https://code.aliyun.com/group/repo.git	
分支 master	
证书	<ol> <li>● 新建证书</li> </ol>
代码提交触发	
删除	取消 确定

点击新建证书

	新建证书	×	×
选择代码源	名称		
	阿里云Code的证书		
Git	用户名		
	myusername		
代码仓库 https://c	密码/令牌		
	•••••	Ø	
分支	可见范围		
master	○ 仅创建者可见 ○ 全局可见		
证书			新建证书
	H	<b>双</b> 消 保存	~
代码提交触发	È.		
删除		取消	确定

如果分支中包含了通配符,需要去掉。 OSS上传下载 需要重新配置与相应阿里云账号的服务授权,并选取bucket

I 流水线编排 testjob2			保存	穿 保存并运行 X
<ul> <li>第</li> <li>77997-group</li> <li>▶ master</li> </ul>	新新校	前的任务	编辑 步骤名称 oss_upload_step Bucket区域 华北2(北京) 选择授权不做为空 Bucket名称不能为空 Bucket名称不能为空 Bucket名称不能为空 Bucket名称不能为空 Bucket名称不能为空 G 第文件目录 ⑦ 第文件目录 ⑦ 第次件目录不能为空 G 第次件目录不能为空 F 6 第次件目录不能为空 第次件目录 ⑦ 第次件目录 ④ 第次件目录 ④ 第次件目录 ④	× • • • • • • • • • • • • • • • • • • •

#### 完整配置见:上传至阿里云 OSS

#### Docker构建

docker构建步骤会被翻译为"Docker构建并推送到自定义镜像仓库"步骤,你也重新配置构建阿里云容器镜像

77997-group master ゆ 构建任务 ① ● 新的任务 ・ custom_docker_build 步骤名称			▶新建源 阶段	新阶段	编辑	
custom_docker_build  使像仓库地址 ③  registry.cn-beijing.aliyuncs.com/ns/repo	,	▶ 77997-group ⊨ master	₿ ● ● ● ● ● ● ● ● ● ● ● ● ●	新的任务	- custom_docker_build 步環名称 custom_docker_build 镜像仓库地址 ① registry.cn-beijing.aliyuncs.com/ns/repo 用户名 username 窓码  Dockerfile路径 ① Dockerfile ContextPath ①	đ

### ECS部署

ECS部署的步骤会被翻译成为一个单独的任务,在这个任务中,你需要首先新建一个主机组,然后填写相应的配置。详见<del>主机组</del>。 Flow 会将你指定的制品文件(比如target/demo.war)打成一个tgz包。因此需要调整部署脚本,先进行解压。

■ 流水线编排	ecs				保	存 保存并运行 X
源	新建源 介段			新阶段	编辑	×
CodePipelin	ne/java-demo	477年77月7日		A #44/7.4	任务名称	
‰ */master	4	构建任务	deploy-ecs-tb	1 新的社务	deploy-ecs-tb	
					制品	
					制品名称.default	~
					主机部署组	1 💿 新建部署组
					请选择	~
					主机部署组不能为空	
					部署配置	
					下载路径 ⑦	
					下载路径,如/home/admin/app/package.tgz	
					下戦時位不能为空 执行用户 ⑦	
					执行用户,如root	
					执行用户个能为空 部署脚本 ⑦	
					1	-
Q X	定时运行					

# **14.计费规则** 14.1. 并发数和构建时长

当您的企业在使用Flow的过程中,发现流水线无法正常运行的情况,可能是您企业的并发数或者构建时长达到了企业套餐上限。

### 计费规则

流水线是由多个任务组成,而一个任务则是由多个步骤组成

- 如下图所示,该流水线包含两个任务:
- Java 构建上传
- 主机部署
- 而"Java 构建上传"任务包含两个步骤:
- Java 构建
- 构建物上传

			编辑 ⑰	×
			任条名称	
〈 返回		基本信息 <b>流程配置</b> 触发设置 变量和缓存	Java 构建上传	
源	构建	部署 	构建集群 ⑦	
7 flow-example/spring-boot			云效北京构建集群	~
⊳ master			任务步骤	
十添加代码源			● 添加步骤	
			→ Java 构建	0
			> 构建物上传	0

Flow 根据 **任务** 的运行状态来统计企业并发数和构建时长,但并不是所有的流水线任务均会耗费您的企业额度。

只有当你使用云效提供的构建资源(<mark>北京和中国(香港)构建集群</mark>),且你的流水线任务中包含以下类型的步骤,当其运行时才会被计入并发任务数和构建时长:

- 各类语言的构建步骤
- 各类语言的单元测试步骤
- 各类语言的代码扫描步骤
- 执行命令步骤

• 企业自定义步骤 (Flow CLI 子命令:step)

因此,上图中的流水线只有"Java 构建上传"任务会消耗您的企业并发任务数和构建时长。

但是,当你使用自有构建集群(自有构建集群)时,无论你使用任何步骤均不会消耗您的企业额度。

## 什么是并发数

并发数指的是,企业内所有流水线正在同时运行的任务。当企业的并发数达到上限时,需要等待前序任务运行完成或者取消,超出额度的任务会处于等待状态。

111 🗾 首页	> 2分钟任务 ▼ 🚖		
最近运行	运行历史 运行 #30		
#30 🥑 等待中	⑧ 终止	阶段1	
运行人	💷 昌意	🥑 2分钟	
开始时间	2021-06-22 11:41:29	<b>等待中</b> 任务中包含计费步行	<b>3</b> 靡,排队中
持续时间 产物	39秒	② 取消	3910
▶ 源(1)			

Flow提供了一定额度的免费并发额度。当然,也可以通过升级套餐的方式,提升企业的并发额度。

## 什么是构建时长

构建时长指的是,企业内所有任务运行时长的总和。当企业的构建时长达到上限时,新的流水线任务将无法正常运行。

💠 🗾 首页	〉 流水线 2021-06-15 ▼	*	
最近运行	运行历史		
#1 🙁 运行失败		执行命令	
运行人	21 rdc-rdc	🔇 执行命令	
开始时间	2021-06-15 16:42:10	运行失	敗
持续时间	10秒	· ○ 重试	县黄町式政制
产物	0项		
< ▶ 源(1)			

Flow每个月都提供了一定额度的免费构建时长。可以通过升级套餐的方式,提升企业的构建时长。

# 如何计算当月最大构建时长和并发任务数

根据企业套餐类型、套餐购买类型、是否购买资源包等确定企业当月使用的上限:

- 若企业是基础版:
- 若未购买基础资源包:最大构建时长 1800 分钟/月;最大并行任务数 3。
- ◇ 若已购买基础资源包:最大构建时长 = 基础资源包购买数量 * 10000 分钟/月;最大并行任务数弹性伸缩。
- 基础版资源、基础资源包,最大构建时长不可叠加、最大并行任务数不可叠加。
- 若企业是高级版:
- 若购买的是"30 人套餐":最大构建时长 10000 分钟/月;最大并行任务数弹性伸缩。

- 若购买的是"100 人套餐":最大构建时长不限;最大并行任务数弹性伸缩。
- 若购买的是"自定义人数套餐": 最大构建时长不限; 最大并行任务数弹性伸缩。
- ◎ 若已购买基础资源包:最大构建时长 = 基础资源包购买数量 * 10000 分钟/月;最大并行任务数弹性伸缩。
- 若同时购买以上套餐、基础资源包,最大构建时长可叠加。

# ? 说明

基础资源包、高级版订单到期后,当月最大构建时长和并发任务数按未到期的订单重新计算。如:企业是基础版,2021.7.25购买了1个基础资源包,订单到期时间:2022.7.24,则从2021.7.25购买开始可以使用的当月最大构建时长是10000分钟/月,订单到期后重新计算,即:2022.7.24可以使用的当月最大构建时长是10000分钟/月,2022.7.25可以使用的当月最大构建时长是1800分钟/月。若2022.7.25已使用构建时长超过1800分钟,则新的流水线任务在当月将无法正常运行。

## ? 说明

如何查看企业套餐类型、未到期订单:企业管理员访问云效,在企业管理后台-基础信息-套餐信息中可以查看。

#### 更多信息请参考云效 2020 资费

# 15.最佳实践 15.1. 构建加速

在持续集成的代码构建中的耗时通常集中在2个部分,首先是从外部获取构建所需的依赖以及代码本身的编译行为。 通过使用Flow的自定义缓存能够我们有效降低获取依赖时 的耗时。本文将介绍如何加速常见的编译工具的构建效率。

#### Yarn构建加速

使用国内registry和mirror

使用国内registry仓库避免由于海外网络访问导致的依赖下载慢的问题

yarn config set registry https://npmmirror.com

## 对于构建中由于Building fresh packages导致构建慢的情况需要指定特定外部依赖的国内镜像源:

yarn config set sass_binary_site "https://npmmirror.com/mirrors/node-sass/"

#### 除了通过yarn config set指定依赖下载路径以外,还可以通过.yarnrc进行配置。在项目根路径中创建.yarnrc并配置以下内容:

registry "https://registry.npmmirror.com"
sass_binary_site "https://npmmirror.com/mirrors/node-sass/"
phantomjs_cdnurl "https://cdn.npmmirror.com/binaries/phantomjs"
electron_mirror "https://cdn.npmmirror.com/binaries/electron/"
sqlite3_binary_host_mirror "https://foxgis.oss-cn-shanghai.aliyuncs.com/"
chromedriver_cdnurl "https://cdn.npmmirror.com/binaries/chromedriver"

#### 配置yarn全局缓存

设置缓存目录,并在流水线自定义缓存中添加缓存目录/root/.yarn配置

# 设置全局缓存

yarn config set cache-folder ~/.yarn # 构建中优先使用缓存中 yarn install --prefer-offline

通过以上命令设置缓存目录,并确保下载依赖过程中优先使用本地缓存,可以最大限度减少由于外部网络导致的依赖下载慢的问题。

#### Npm构建加速

#### 使用国内registry和mirror

#### ? 说明

或者直接使用cnpm

#### 对了使用npm的用户可以创建.npmrc并设置以下内容:

registry="https://registry.npmmirror.com"
sass_binary_site="https://npmmirror.com/mirrors/node-sass/"
phantomjs_cdnurl="https://cdn.npmmirror.com/binaries/phantomjs"
electron_mirror="https://cdn.npmmirror.com/binaries/electron/"
sqlite3_binary_host_mirror="https://foxgis.oss-cn-shanghai.aliyuncs.com/"
chromedriver_cdnurl="https://cdn.npmirror.com/binaries/chromedriver"

#### 构建中使用全局缓存

构建时,使用一下命令设置npm的全局缓存路径,并在流水线自定义缓存配置中添加/root/.npm缓存路径,并使用以下命令安装依赖包,优先从本地缓存获取依赖包:

npm config set cache ~/.npm
npm install --prefer-offline --no-audit

#### Golang构建加速

#### 使用go proxy

#### 使用Goproxy从国内下载外部依赖

export GOPROXY=https://goproxy.cn

#### 添加go mod缓存

持久化Go Mod缓存,Flow中Go构建环境的默认缓存目录是/go/pkg/mod。用户需要在流水线自定义缓存中添加该缓存路径即可。 在构建过程中将会优先使用/go/pkg/mod 中的本地缓存,从而减少由于外部网络请求导致的构建慢的问题。

#### Python构建加速

使用pypi镜像

#### 使用阿里云pypi镜像

pip install -r requirements.txt -i https://mirrors.aliyun.com/pypi/simple/

#### 添加pip依赖缓存

缓存pip本地依赖,流水线自定义缓存配置中添加缓存目录/root/.cache/pip,确保pip install时优先从版本缓存中获取依赖包

#### Dockerhub镜像下载慢

如果你的Dockerfile中FROM了dockerhub的镜像,比如

```
FROM nginx:1.19.1
```

由于跨境网络问题,这种Dockerfile的构建会不稳定或者比较慢,解决该问题,有两种方式:

1. 使用北京和中国(中国(香港))构建集群中的中国(中国(香港))构建集群。

2. 将境外镜像在pull到本地,然后push到阿里云镜像仓库(cr.console.aliyun.com)的国内region(比如北京,上海等),然后修改你的dockerfile中FROM。比如:

```
docker pull nginx:1.19.1
docker tag nginx:1.19.1
registry.cn-beijing.aliyuncs.com/yournamespace/nginx:1.19.1
docker push registry.cn-beijing.aliyuncs.com/yournamespace/nginx:1.19.1
```