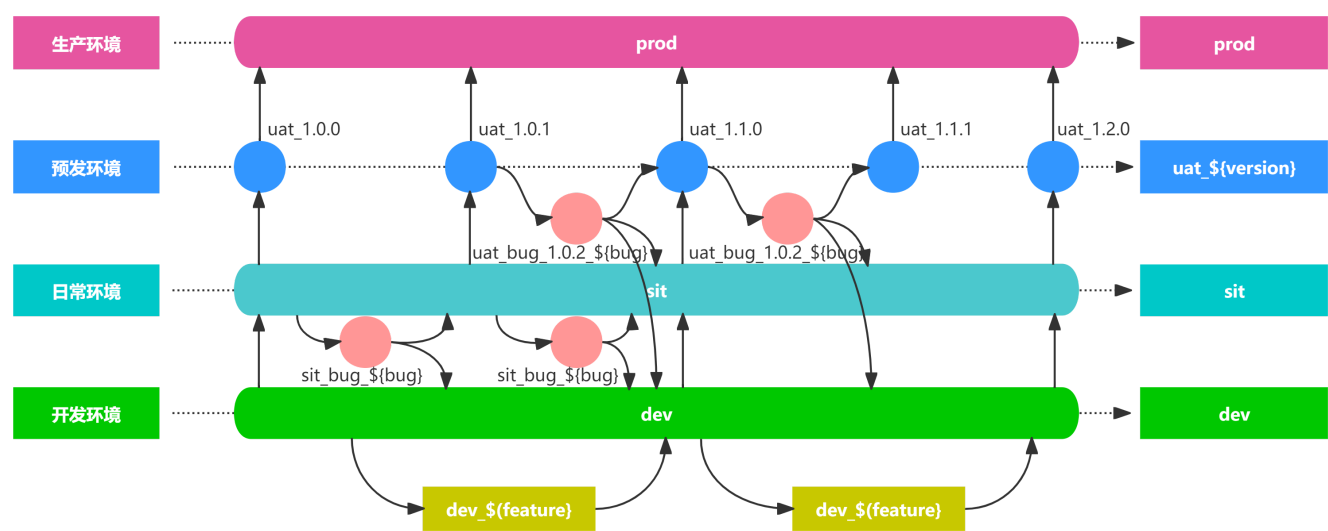


分支版本管理 V0.5



分支规约

- 1. **dev分支**: 开发分支, 该分支为变更分支或者特性分支, 用于承载具体的功能开发。
- 2. **sit分支**: 日常分支, 该分支用于存放日常开发代码和缺陷修复。
- 3. **uat分支**: 预发布分支, 该分支主要用于区分版本和缺陷修复。
- 4. **prod分支**: 生产分支, 用于存放正式发布的代码。

分支命名规约

dev_\$(feature) **功能特性开发分支**

\$(feature) 大禹编号

sit_bug_\$(bug) **缺陷修复分支**

\$(bug) 大禹编号

uat_\$(version) **预发布分支**

\$(version) 版本编号
常用的版本号为三位数版本号, 其构成如下: 主版本号.次版本号.小版本号

1. 主版本号（首位版本号）

主版本号，也叫首位版本号、顶位版本号。主版本号一般代表项目的期数与产品方向。除非项目合同改变、大规模 api 不兼容、产品方向改变、底层架构升级等情况外不轻易更新。

另外，项目未正式发布、未正式孵化、未正式上线，则首位版本号为 0，一期发布，则为 1，二期发布则为 2。

2. 次版本号（迭代号）

次版本号，也叫迭代号，一般代表某个迭代发布的功能集合（一个迭代发布会包含若干个功能更新）。如：

- 1.1.0：第一期项目第一迭代发布版本
- 1.2.0：第一期第二迭代发布版本
- 1.18.0：第一期第十八个迭代发布版本。

3. 小版本号

小版本号，是为了某些小功能的临时上线，热修复的临时上线设置的小迭代，通常不包含大的功能性更新，常常是围绕某个功能点进行升级或者某个 bug 的修复进行上线。

uat_bug_\${version}_\${bug} 缺陷修复分支

`${version}` 版本编号

`${bug}` 大禹编号

新功能特性开发流程

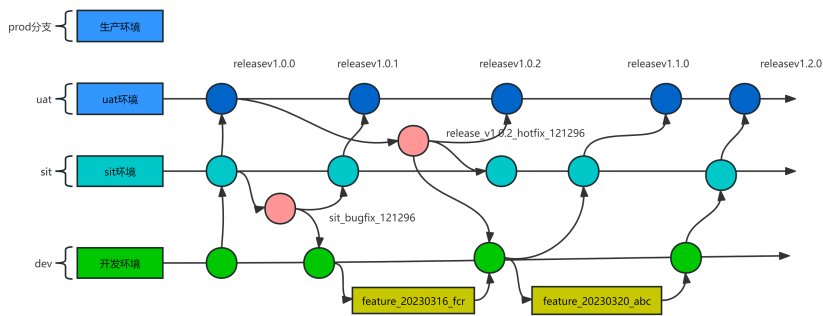
1. 根据新功能特性工时、人员、上线时间安排研发计划，并从 dev 支基础上创建 dev_\${feature} 分支（角色：研发组长、敏捷教练）
2. 基于新建分支开发新功能（角色：开发人员）
3. 开发人员完成新功能并自测通过后，提交代码合并请求（角色：开发人员）
4. 根据代码合并请求和研发计划，在代码检查后合并代码到 dev 分支（角色：研发组长）
5. 据研发计划从 dev 分支合并代码到 sit 分支，并发布日常环境进入测试流程（角色：研发组长、测试组长）
6. 日常环境测试题通过后，从 uat 上一分支拉取新的分支并按照命名规约创建新 uat_\${version} 分支，并将 sit 代码合并到 uat 最新分支，按计划发布预发布环境进行测试（角色：研发组长、测试组长）
7. 测试题通过后，将代码合并到 prod 分支，并据研发计划发布生产环境（角色：研发组长、实施组长）

缺陷修复流程

1. 生产环境缺陷修复需要根据现场要求、升级窗口期、Bug 的严重性等安排计划以及预发布版本 uat_\${version}（角色：研发组长、实施组长）

2. 开发人员根据 Bug 和命名规范创建 `sit_bug_${bug}` / `uat_bug_${version}_${bug}` 分支 (角色: 开发人员)
3. 修复预发布环境和生产环境的 Bug , 预先规划新的 `uat_${version}` 分支和研发计划 (角色: 研发组长)
4. 开人员修改 Bug 完成并自测通过后, 提交代码合并请求 (角色: 开发人员)
 - i. 如果是日常环境 Bug , 除 `sit` 分支合并请求外, 还需要`dev`分支合并请求
 - ii. 如果是预发布/生产环境 Bug , 除 `uat_${version}` 分支合并请求外, 还需要`sit`分支合并请求、`dev`分支合并请求
 - iii. 合并请求标题使用缺陷修复分支名称
5. 根据代码合并请求和研发计划, 在代码检查后合并代码到 `sit` / `uat_${version}` 分支 (角色: 研发组长)
6. 发布到日常环境/预发布环境通知相关测试人员进行验证 (角色: 研发组长、测试组长)
7. 测试通过后, 根据现场要求和升级窗口期, 发布到生产环境环境测试。 (角色: 研发组长、实施组长)

第二套版本管理方案



分支命名规范

1.release分支命名规则: `release_version`
例如: `release_v1.0.2`

2.release bug分支命名规则: `release_version_hotfix_xxxxxx` (大离编号)
例如: `release_v1.0.2_hotfix_121296` (大离编号)

3.sit_bug分支命名规则: `sit_bugfix_xxxxxx`
例如: `sit_bugfix_121296` (大离编号)

一、修改生产BUG流程: (前序需求确认Bug问题不在讨论范围)

1. 开发人员根据BUG和命名规范 创建hotfix或Bugfix分支 (角色: 开发人员)
2. 从release分支拉代码创建hotfix分支, 从Sit分支拉代码创建Bugfix分支 (角色: 开发人员)
3. 开人员修改Bug完成并自测通过后, 通知相关测试人员进行验证 (角色: 开发人员)
5. 测试验证无误, 测试负责人合并代码到release分支上 (角色: 测试人员)
6. hotfix根据现场要求和升级窗口期, 发布到release环境测试。 (角色: 实施经理)
7. 测试题通过后, 对已完成Bug进行升级 (角色: 实施经理)

二、开发新功能流程: (前序需求设计不在讨论范围)

1. 新功能开发负责人根据研发计划申请创建feature分支 (角色: 功能开发负责人)
2. 根据命名规则, 从develop分支上拉代码创建分支 (角色: 功能开发人员)
3. 基于新建分支开发新功能 (角色: 开发人员)
4. 开人员完成新功能并自测通过后, 选择合适计划时间合并代码到develop分支 (角色: 功能开发负责人)
5. 在develop 环境进行联调验证 (角色: 开发人员)
7. develop测试题通过后, 对已完成新功能按计划进行代码合并到Sit分支, 到Sit环境测试 (角色: 功能开发负责人, 测试负责人)
8. Sit测试题通过后, 按计划在Release环境进行验证 (角色: 测试人员)
9. 测试题通过后, 对已完成新功能按计划进行发布 (角色: 项目产品负责人)

一: 基于dev切出 feature分支进行开发:

- (0) `git pull` 拉取最新代码
- (1) 切换到分支到dev: `git checkout dev`
- (2) 基于dev切出 feature分支: `git checkout -b feature/20200315_task_1`
- (3) 代码开发
- (4) 提交代码
`git status` 查看本地修改 如发现不需要提交 `git checkout -- test.java`
`git add <some-file>`
`git commit -m "bugfix:修改bug"`
`git push`

二: 开发结束后, 如果需要合并代码到dev

- (1) 切换到分支到dev `git checkout dev`
- (2) 拉取dev最新代码 `git pull (git fetch + git merge,rebase)`
- (3) 合并feature分支代码 `git merge feature/20200315_task_1`
- (4) 把dev代码推送到远端 `git push`

三: 合并dev代码到sit分支

- (1) 进入网址 `https://starsvm.qm.cn/` 选择相应项目
- (2) 选择左侧导航栏中的merge requests选项
- (3) 点击new merge request 按钮
- (4) 选择source branch 为 dev,选择target branch为 sit,点击continue
- (5) 填写相关描述, 选择assignee审批人 点击确定
- (6) 待审批人审批通过后系统自动merge代码

第三套版本管理方案

CoderTanzJ
码龄13年 暂无认证

9 原创

10万+ 阅读

169万+ 粉丝

15万+ 访问

等级

1062 积分

61 粉丝

114 获赞

118 评论

366 收藏

私信

关注

AI图早知道，每日最新动态
了解全球AI新鲜事！
立即参考

大额流量券送不停
多发多得，流量翻倍！
去领取

搜博主文章

HUAWEI 华为云

38元/年
2核2G2M
云服务器

超高性价比，更可靠更安全

立即抢购

热门文章

Java | Cmd运行Jar出现乱码问题 34672

Maven | 使用IDEA将本地的Jar包添加到个人的maven仓库中 28729

项目版本管理的最佳实践：飞流Flow（阿里AoneFlow）篇 25204

基于maven的springboot多环境ymli配置文件切换与隔离 10093

项目版本管理的最佳实践：gitflow基础篇 7402

最新评论

Java | Cmd运行Jar出现乱码问题
persist 执：我也是同样的问题，请问一下你是怎么解决的

Java | Cmd运行Jar出现乱码问题
webkin_44031254：你好，时隔3年，请问还记得如何解决的吗。我在网上试了好多方...

单元测试的最佳实践
Kwan的解忧杂货铺：创新空间代码工作室，支持懂主优质文章，讲解得非常详细，干货...

基于maven的springboot多环境ymli配置...
一个卑微的小白：项目启动报 @ 的错

大家在看

基于matlab的深度学习案例及基础知识专栏
前青 4990

微软蓝屏事件引发的网络安全与系统稳定性深度思考及实践探索
db、dDu、dBM、dBV、dBFs
一键视频转动！DiffSynth Studio的详细本地部署安装方法来了~ 502

《2024年数研》C题第一问：假假级数教学 老外游中国（阿鑫）

最新文章

单元测试的最佳实践

项目版本管理的最佳实践：gitflow基础篇

Spring Cloud集成seata分布式事务—AT模式

2022年 1篇

2020年 4篇

2019年 1篇

2018年 3篇

项目版本管理的最佳实践：飞流Flow（阿里AoneFlow）篇

原创 CoderTanzJ 于 2020-12-29 21:56:05 发布 阅读量2w 收藏 59 点赞数 12

文章标签： 项目管理 AoneFlow 版本管理 最佳实践

开放原子开发者工作坊 文章已被社区收录

加入社区

阿里云云效平台分支管理模型——飞流Flow，也就是阿里的AoneFlow。飞流Flow是基于git的多主干分支模式的版本管理模型（也叫分支模型），区别于传统使用develop分支。release分支和master的gitflow，飞流Flow采用了 feature + n*release + master 的分支形式实现版本管理，而其中，n * release 代表了各环境的发布分支。在本文中的最佳实践章节中，笔者将以阿里云云效平台为载体，介绍飞流Flow(AoneFlow)的最佳实践。

导航

项目版本管理的最佳实践：gitflow基础篇

项目版本管理的最佳实践：飞流Flow（阿里AoneFlow）篇

目录

- 一、分支规约
- 二、版本号规约
 - 2.1 主版本号（首位版本号）
 - 2.2 次版本号（迭代号）
 - 2.3 小版本号
- 三、飞流Flow的最佳实践（使用阿里云云效）
 - 3.1 总体流程图
 - 3.2 号行同学与阿吉同学的最佳实践
 - 3.2.1 功能分支(feature分支)的创建
 - 3.2.5 危险分支下线
 - 3.2.6 生产环境发布
 - 3.2.7 生产环境发布-写基线
- 四、FAQ

一、分支规约

分支	分支名	说明
特性分支 (开发分支)	origin/feature_\$(specific_feature)	该分支为变更分支或者特性分支，用于承载具体的功能开发与缺陷修复，基于最新的master 拉出，\$(specific_feature) 为具体开发的功能目标，每发起一个变更（特性开发或缺陷修复），都需要创建一个feature与变更进行对应，一般的开发工作都会在此分支上进行。
集成分支	origin/release_\$(build_no)	该分支主要用于集成和发布，即在环境中（如日常、预发和生产）对各feature分支进行集成和部署，不同的环境下可以使用不同的release分支（如 release_daily、release_pre和release_prod）对feature进行集成（即 n * release）。
主干分支	origin/master (只读保护)	master代表当前项目的基线，不允许直接修改。各分支功能在验证通过并发布到生产验证后，将分支集成后release合并回master，保证master代码最新；若某个feature功能验证不通过，可以单独摘下该feature的集成再合并回master，从而实现风险的全局最小影响。

二、版本号规约

在最佳实践中，我们常用的版本号三位数版本号，其构成如下：

V主版本号.次版本号.小版本号

eg: V1.0.0、V1.5.0、V1.13.1等

2.1 主版本号（首位版本号）

主版本号，也叫首位版本号、顶级版本号，即V后第一个版本号。主版本号一般代表项目的期数与产品方向。除非项目合同改变、大规模api不兼容、产品方向改变、底层架构升级等情况外不轻易更新。

另外，项目未正式发布、未正式孵化、未正式上线，则首位版本号为0，一期发布，则为V1，二期发布则为V2。

2.2 次版本号（迭代号）

次版本号，也叫迭代号，一般代表某个迭代发布的功能集合（一个迭代发布会包含若干个功能更新）。

如V1.1.0：第一期项目第一迭代发布版本、V1.2.0：第一期第二迭代发布版本、第一期第十八个迭代发布版本 V1.18.0。

2.3 小版本号

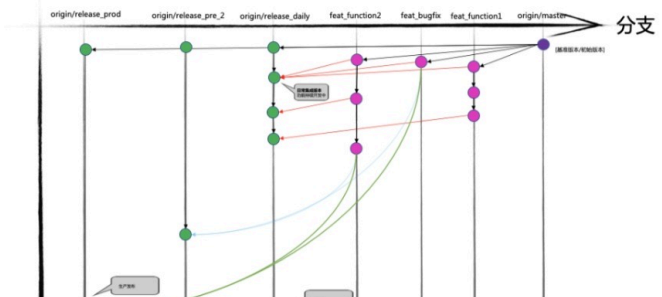
小版本号，是为了某些小功能的临时上线，热修复的临时上线设置的小迭代，通常不包含大的功能性更新，常常是围绕某个功能点进行升级或者某个bug的修复进行上线。

三、飞流Flow的最佳实践（使用阿里云云效）

为了更好地使用飞流Flow，接下来将结合阿里云云效来讲解飞流Flow的最佳实践

3.1 总体流程图

下图为最乐观形式下的飞流Flow模型图，可以见到，release分支是多个feature的集成版本。同时，release又可以通过流水线进行组织，使用在不同的项目环境构建下。



ECS 2核
2G3M固定带
宽不限流量

立即购买安全稳定的云服务器

爆款云服务器2核2G 3M
82元/年，助力低成本上云

阿里云

了解详情

目录

导航

目录

一、分支规约

二、版本号规约

2.1 主版本号（首位版本号）

2.2 次版本号（迭代号）

2.3 小版本号

三、飞流Flow的最佳实践（使用阿里云...）

3.1 总体流程图

3.2 号行同学与阿吉同学的最佳实践

3.2.1 功能分支(feature分支)的...

宝塔面板

简单高效管理服务器

1500万台服务器资源，2分钟安装好，一键管理服务器

广告

分类专栏

Java 3篇

Maven 2篇

Git 1篇

yapi

docker

3.2.3 日常环境发布

完成了流水线设置后，可以点击【运行】对流水线进行测试。在运行时，由于开启了分支模式，此时需要将本次加入【DEV日常流水线】的分支加入到构建列表中。

添加运行分支 ×

代码源	分支	状态	操作
...33520129ba4e9/tanzj/teapot	feature_bugfix	待集成	🗑
	feature_function1	待集成	🗑
	feature_function2	待集成	🗑

👤 添加运行分支

代码源	分支/标签
//github.com/tanzj/teapot-ui	develop

取消 确定

运行后，分支管理器会对feature_bugfix、feature_function1、feature_function2等三个分支进行集成，并生成一个新的【origin/release】分支（如下图），而这个release分支就是专门服务于日常环境的发布分支了。

分支信息 ×

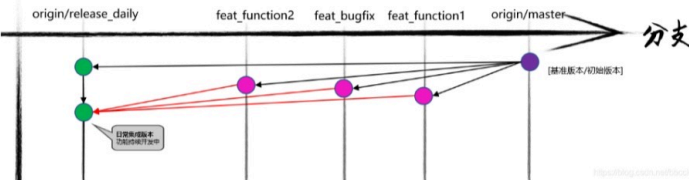
发布分支: release/20201229-011724356606291_release_1206719_2

运行分支

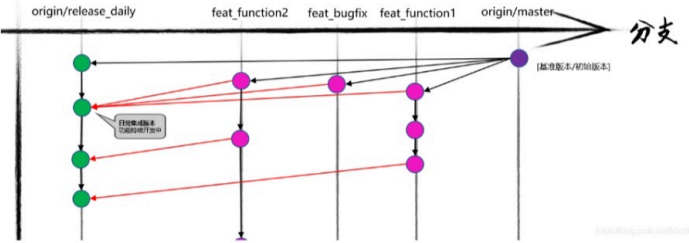
分支	commitId	状态
feature_function2	2d05cc096ddb016dd475b6850f5d28472c...	已集成
feature_function1	2d05cc096ddb016dd475b6850f5d28472c...	已集成
feature_bugfix	2d05cc096ddb016dd475b6850f5d28472c...	已集成

https://blog.csdn.net/606666666 关闭

此时，我们的版本线是这样的（红线代表由云效分支管理器的自动集成）。需要注意的是，release分支的我们不应该直接修改（除了解决冲突外）



而随日常开发的持续进行，每当分支上有同学提交了代码并触发了流水线的重新运行，分支管理器会对分支进行集成处理，形成包含最新分支代码的commit



3.2.4 预发环境发布

经过每天辛辛苦苦的搬砖，由阿吉同学负责的function1功能和弓行同学负责的bugfix通过了自测和日常冒烟，可以上预发进行验证了。此时则需要到预发的流水线中，对这两条分支进行集成操作。

运行配置 ×

代码源	分支	状态	操作
...33520129ba4e9/tanzj/teapot	feature_bugfix	已集成	🗑
	feature_function1	已集成	🗑

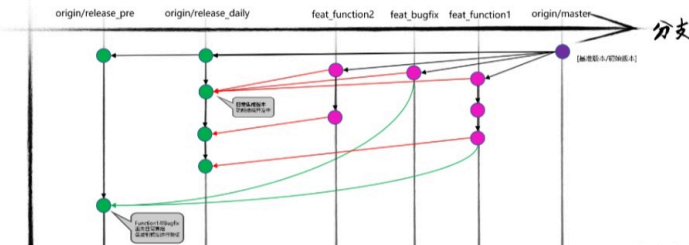
👤 添加运行分支

https://blog.csdn.net/606666666

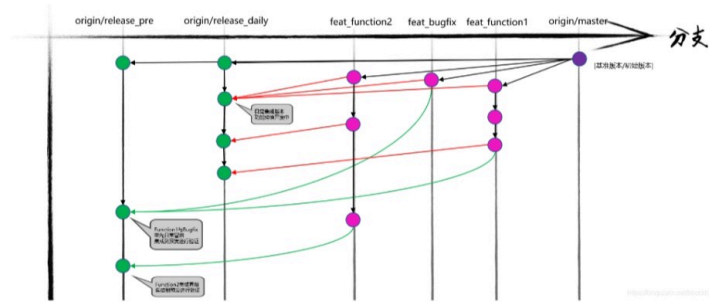
选择完需要集成的分支之后，点击运行，便可以在预发环境发布这两条分支。

此时的版本线是这样的（绿线代表由预发流水线分支管理器的集成），如此一来，预发环境便得到了只包含bugfix和function1而不含没有冒烟通过的function2的最新代码的纯净提交。

测试同学和开发同学便可以在预发环境对功能进行预发验证。



同理，当弓行同学的function2功能也开发自测完、在日常冒烟验证后，在预发流水线添加他的分支，便可以完成对function2的集成了，至此，整个版本线如下所示：

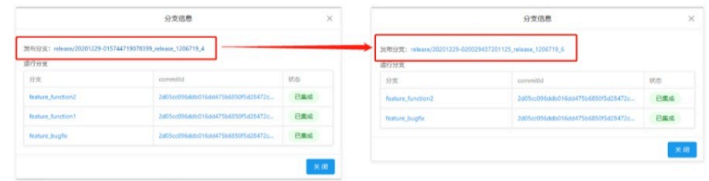


3.2.5 危险分支下线

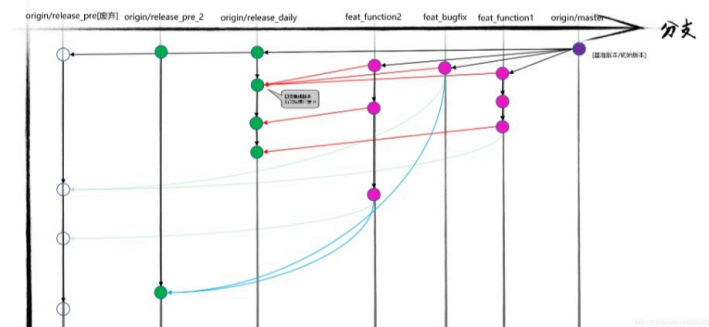
在预发环境进行预发验证和测试时，测试同学发现由【阿吉】同学开发的function1功能虽然完成了开发，但是他的改动会影响某个功能正常运行，而发布日迫在眉睫，现在改动一定是来不及的，此时阿吉同学的feature_function1分支便是一个危险分支，不能够上线。此时，需要在预发流水线对阿吉同学的代码进行下线操作。



下线后，因为涉及到的改动会比较多，此时云效的分支管理器会自动将feature_function2和feature_bugfix两条分支重新集成到我们创建的另一条预发环境使用的发布分支【release_pre_2】中从而达到去除危险分支的目的（下线分支需要重新集成的原因是无法将危险分支的commit从集成分支中剥离）。



此时，版本线如下图所示（蓝线为云效分支管理器集成，而原origin/release_pre分支已经废除，取而代之的是origin/release_pre2）：



3.2.6 生产环境发布

将通过测试的分支在生产流水线中添加（如3.2.4步）并实现构建便可完成生产环境的发布，生产环境运行的分支也是一条release分支。

在实践中，推荐将生产环境的发布流程增加人工卡点（审批），即流水线的设置可以如下：

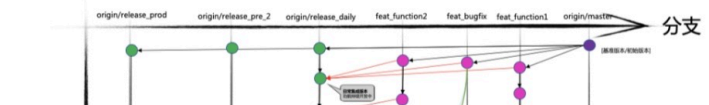
【构建】-【部署审批（人工卡点）】-【灰度部署(分批)】-【生产部署(分批)】-【生产验证(人工卡点)】-【写基线】

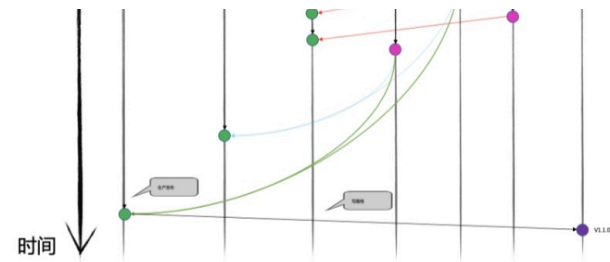
3.2.7 生产环境发布:写基线

写基线是指将发布分支的代码合并到当前master分支中，一般在完成生产验证之后执行。



完成发布后，整体个版本流程图是





四、FAQ

Q1: AoneFlow下如何进行code review和拉取请求?

A1: 基于AoneFlow进行团队协作开发时,可以围绕feature分支进行code review和pr操作,即除了保护release分支外,还保护feature分支,不允许直接提交到feature分支,且另外创建origin/feature_xxx_pr分支进行拉取请求。不仅如此,在最终发布到生产之前,设置一个人工卡点来进行code review操作也是可行的,只是code review的粒度不一样(前者基于每个commit,后者基于发布的整个功能),如果团队的发布节奏比较紧急且人力资源不太充足,可以采取发布前进行人工卡点+团队code review的形式。

Q2: AoneFlow适合什么样的开发场景或者开发团队?

A2: AoneFlow适合团队规模适中,一个迭代中所需要开发的backlogs涉及到不同的业务域,且存在分支发布风险或存在迭代周期交叉情况(如1.2.1与1.3.0同时开发并测试)的敏捷团队。如上述最佳实践中,【阿吉】同学开发的function1在临近上线前发现会影响其他业务功能开发,需要临时下车不发布;如果一个开发团队中只有两三个人,那么一切从简便可。

Q3: 我可以不使用云效来实现AoneFlow吗?

A3: 目前来看,使用云效来实现AoneFlow是最省时间的,若不使用云效,可以采用人工管理release分支的构建+jenkins流水线的方式也是可以实现AoneFlow的(或者采用脚本自动合并分支)

Q4: 远程feature分支可以不删除吗?

A4: 远程feature分支可以不删除,但是由于feature在发布后已经合并到了基线,不删除留存在远程版本库意义不大。

Q5: 多个分支同时开发,遇到代码冲突怎么办?

A5: 云效提供了完成的冲突解决教程。最安全的做法是将集成分支拉到本地,在本地解决冲突后,构建成功后再提交到远程release分支



Q6: 下一次迭代,还需要重新创建流水线吗?

A6: 不需要,只需要在原先的流水线中将原来需要集成的分支删除(实际上发布后也会自动删除),重新添加需要发布的功能分支上去便可

Q7: 预发、日常都集成了同一个feature,重新构建的话新提交会影响两个环境吗?

A7: 一旦预发流水线、日常流水线都集成了同一个feature分支,那么开发者提交代码后触发重新部署,在预发环境和日常环境都会呈现最新的功能特性

Q8: 几条release分支会互相合并吗(如日常的release和预发的release)?

A8: 不会,release分支相互独立,完全没有一点关系,他们的相同也只是名字上的部分相同而已。

Q9: 对比了gitflow、AoneFlow感觉更加灵活和自由,对风险的控制也是比较稳妥的,那么AoneFlow是最好的版本管理模式吗?

A9: 没有最好的版本管理模式,适合自己生产的具体情况才是最好的

以上是项目版本管理的最佳实践:飞流Flow(阿里AoneFlow)篇的所有内容,欢迎在评论区讨论与提出改进意见!

让AI简单且强大:深度学习引擎OneFlow技术实践	cynthia_chen的博文	3543
本文内容节选自msup主办的第七届TOP100summit,北京一流科技有限公司首席科学家袁进程(老师木)分享的《让AI简单且强大:深度学习引擎One...		
开发分支管理模式之阿里AoneFlow	weixin_34224941的博文	860
谈到分支管理模式,令人最为熟悉的莫过于TrunkBased和GitFlow。TrunkBased模型是持续集成思想所崇尚的工作方式,它由单个master分支和许多rel...		
WHAT - Git工作流(含Gitflow和AoneFlow)_aoneflowgitflow-CSDN...		8-1
目录 一、介绍 1.集中式工作流(Centralized Workflow) 2.功能分支工作流(Feature Branch Workflow) 3. Gitflow 工作流 4. Pull Request 工作流 5. GitOps ...		
阿里AoneFlow分支管理		8-4
在AoneFlow上你能看到许多其他分支模式的影子。它基本上借鉴了TrunkBased的“易于持续集成”和GitFlow的“易于管理需求”特点,同时规避掉GitFlow...		
AoneFlow和云效快速使用DevOps	white_bird_shd的博文	1935
今天学习一下git分支管理模式之阿里的AoneFlow模型,以及阿里的云效代码管理平台分支管理模式简介,使用过git一段时间之后,开始深入了解git分支管理...		
Git-flow分支管理与Aone-flow分支管理对比 最新发布	Andrew_Chenwg的博文	585
比如上线后的Hotfix,正常的处理方法应该是,创建一条新的发布分支,对应线上环境(相当于Hotfix分支),同时为这个分支创建临时流水线,以确保...		
在阿里,我们如何管理代码分支?		7-31
在阿里内部,使用AoneFlow流程的团队基本上不用自己运行Git,来处理分支的事情,而是由阿里巴巴集团内部名叫Aone的协同研发平台(以下简称平台)来...		
开发分支管理模式之阿里AoneFlow		7-30
AoneFlow只使用三种分支类型:master分支、feature分支、release分支,以及三条基本规则。规则一:开始工作前,从master创建feature分支。从代表最新...		
阿里分支模式AoneFlow	陶凤桃月的专栏	1348

本文介绍了目前阿里的分支模式，兼顾了TrunkBased的易于持续集成和 GitFlow 的易于管理需求特点。看一下具体套路，AoneFlow 是阿里比较通用的...

项目版本与分支管理-阿里AoneFlow模式分析 weixin_30413739的博文 608
前言 在我前期的项目管理的经验中，一个项目需要维护多个产品及多个版本，这给版本与分支的管理增加了难度。前期没有重视，使得分支太多太乱，版...

...vs GitFlow vs AoneFlow vs OneFlow vs ExeFlow 7-19
AoneFlow 阿里的研发效能事业部专家基于TrunkBased和GitFlow提出了一套新思路 AoneFlow。原理 AoneFlow 只使用三种分支类型 主干分支、特性分...

阿里AoneFlow分支管理模式-个人理解 7-28
阿里AoneFlow分支管理模式-个人理解 原文地址 https://blog.csdn.net/u012562943/article/details/79754587 转载于 https://my.oschina.net/u/2526015/blog/...

关于阿里云的一键部署工具【飞流】的使用 (详细) 李某的博文 4771
1.飞流是啥？ 飞流是Teambition的一款应用，功能不限于一键部署服务，但下面讲的就是一键部署的使用。 2.使用详解 2.1在https://flow.teambition.com/...

flyd-dom-delegate 从dom事件创建飞流 04-30
它允许我们将数据流的概念应用于各种场景，包括事件处理、状态管理等。飞流的核心思想是创建可以监听和订阅的数据流，当流中的值发生变化时，订...

震惊~~飞流android版使用体验！！ 03-19
飞流可能是一款为开发者设计的Android工具，帮助他们加速下载、调试或者管理项目。压缩包中的'android_download_tool.apk'文件名表明这是一个And...

机场起飞流可预测性研究 02-11
这对于空中交通流量管理 (ATFM) 具有重要的实践意义。例如，通过优化调度策略和提高预测精度，可以在一定程度上缓解拥堵问题，提高机场的吞吐量...

Rapidminer中文手册 06-12
RapidMiner中文手册，对于初学RapidMiner者有很好的帮助

飞流下载 (Android程序) 08-11
Android操作系统上用的下载程序，有很多分类，很好用！

项目拉去某个版本的_项目版本管理的最佳实践：飞流Flow楠 weixin_42510900的博文 236
飞流Flow是基于git的多主干分支模式的版本管理模式(也叫分支模型)，区别于传统使用develop分支、release分支和master的gitflow，飞流Flow采用了 feat...

Git分支模型(参考阿里Aone Flow) weixin_41563161的博文 649
Git分支模型(参考阿里Aone Flow) https://www.dazhuantian.com/2019/11/30/5de16a1f9dd7/

版本控制工具总结 zbird1988的专栏 636
1.svn 2.git https://www.atlassian.com/git/?_mid=bf77afce13ad4a68dc368e5940a3b036&gclid=COOT_NqggMYCFUGXvQodLxYJog

代码版本方案Trunk Based, Git Flow, Aone Flow abcLinux 1954
git 分支管理

在阿里，我们如何管理代码分支？ weixin_34161063的博文 1094
点击关注 InfoQ，置顶公众号接收程序员的技术早餐作者 | 林帆编辑 | 小智写在前面 在阿里内部，流行着许多有思想的工程实践。有些实践通过工具和流...

用python生成一个庆余年社交网络图 03-26
由于题目没有给定数据，我会根据剧情以及角色关系使用假数据来生成社交网络图。首先，我需要使用Python中的网络库networkx来创建一个空的图形对...