PaaS内源代码社区同步操作简介

# 简介

Cloudfoudry官方库上会定期更新版本，由于内源要与官方版本保持同步，就需要定期从官方的github上进行更新，然后将这些更新到内网的PaaS内源库的develop分支，开发人员要基于新版本修改Cloudfoudry某个功能模块，只要从PaaS内源库的develop分支的新版本中拉一个功能分支进行开发即可。

# Paas组github库有哪些？

## 内外源远程库关系图

## 内源远程库各分支含义介绍

社区Repo（官方库）：<https://github.com/cloudfoundry>；

镜像Repo（镜像库）：<http://rnd-github.huawei.com/cloudfoundry>；

内源Repo（PaaS内源库）：<http://rnd-github.huawei.com/PaaS>；

Master：仓库主分支，当预发布分支代码验收ok后合入master分支，在master分支上打tag发布版本；

Develop：仓库开发分支，外源同步、白区同步、内源贡献及特性开发完成后都合入develop分支；

Feature：仓库特性分支，特性开发启动时基于develop分支新建一个feature分支，在feature分支上进行代码开发，当特性开发完成后结束feature分支合入develop分支；

Release：仓库预发布分支，发版本前从develop分支拉出预发布分支，待发布前集成验证，验收通过后合入master分支及develop分支；

# 更新内网镜像库

定期从cloudfoudry官方库(<https://github.com/cloudfoundry>)中更新同步代码到内网镜像库（http://rnd-github.huawei.com/cloudfoundry），以cf-release为示例说明怎么将外源代码更新到内网镜像库具体的操作：

*假设:外部官方库为remote1，内网镜像库为remote2*

1. 克隆remote2的cf-release模块代码到本地

git clone http://rnd-github.huawei.com/cloudfoundry/cf-release.git

1. 进入cf-release目录，给该模块增加外网的remote定义

git remote add cloudfoudry <https://github.com/cloudfoundry/cf-release>.git

1. 查看当前模块有哪些远程库

$ git remote -v

cloudfoudry https://github.com/cloudfoundry/cf-release.git (fetch)

cloudfoudry https://github.com/cloudfoundry/cf-release.git (push)

origin http://rnd-github.huawei.com/cloudfoundry/cf-release.git (fetch)

origin http://rnd-github.huawei.com/cloudfoundry/cf-release.git (push)

1. 在cf-release目录下获取cloudfoudry官方库的主干分支（master）代码，并与前面从内网镜像库更新的代码进行合并（merge）

git pull cloudfoudry master

1. 在合并过程.gitmodules文件会有冲突，需要手动解决冲突，然后在将修改后的代码上传到cloudfoudry内部镜像库中

git add .

git commmit -m "提交日志"

注意：

* 如果是某个文件夹有冲突，可以直接用mergetool解决，不用手动操作
* cf-release的子模块还没有更新，所以现在不能将上述的修改，推入到内网镜像库中，需待其子模块更新完毕才能推进，即push操作

1. 查询cf-release有哪些子模块

$ git submodule

-97461df55878d8bdef45cfc870ba13c22cca69dd packages/ruby

-a33125b7b5d8abba1c0e3cbdb34132f7d4e7fdf4 shared

-400e9be0ea2eef735d5a81d094d8a98862c5d623 src/acceptance-tests

-6f67aa2d472e73ab117731ee5c928c69697e3ba3 src/cloud\_controller\_ng

-cddedff8282fbd93d68abc4618bbc3beaa3cdf59 src/collector

-dca6f83aab94d5b780b68e742f4eb5e98681a000 src/dea\_next

-7da1e8774815661263282bde4278b40f080bd7df src/gnatsd

-a977dfd5eb6ee4eaf535b5e1b715d40253df18de src/gorouter

-998cb7aceea55b3e4a546bf7e7077154a9b5bd3d src/hm9000

-66b57a3df2f8b5b139102b4ff5dd55eb25e8bf4f src/loggregator

-9119c09cb5ed96dfbda0a6d801be291f81484f2e src/login

-bdbc99d415b157895db91827fc678a77b074b323 src/uaa

-d31b3b4884592f261c5e6b889aa47fad53219080 src/warden

1. 这些子模块要克隆到本地，和cf-release相同的处理

# 内网镜像库打标签

以给内网镜像库打CFv164版本标签为示例介绍内网镜像库打标签过程：

1. 首先克隆内源cloudfoudry镜像库中的cf-release代码到本地，进入cf-release目录
2. 查询v164（外网打好的标签）对应的commit ID

git ls-remote --tags cloudfoudry | grep v164

1. 将cf-release的上下文环境切换到这个commit点

git checkout -b v164 8b361c3dfb2607f1341dfd652ae1ca72f25ad00d

1. 进入cf-release的src目录，逐个修改该目录下每个子模块的.gitmodules文件为内网镜像库地址，但是这个子模块的子模块不需要再打标签，以dea\_ng子模块为例：

* 进入cf-release\src\dea\_next目录
* 修改.gitmodules文件
* add 并 commit 这个修改

git commit -a -m "update tag"

* 然后将这个修改推进内网镜像库

git push

1. 将cf-release的所有子模块对应侧commit点都打完标签后，最后修改cf-release本身的.gitmodules文件，然后进行类似4点的操作，即打标签也是从子模块开始再上层的父模块的

# 更新PaaS内源库的develop分支到最新版本

从内网cloudfoudry镜像库更新版本到PaaS内源库的develop分支，以更新dea\_ng模块到CFv164版本为例：

1. 从PaaS内源库克隆dea\_ng（http://rnd-github.huawei.com/paas/dea\_ng）代码到本地
2. 进入dea\_ng目录
3. 从内源的cloudfoudry镜像库，抓取获取dea\_ng分支的CFv164版本的代码，并进行合并

git pull cloudfoudry master --tags CFv164

1. 如果有冲突，先将冲突解决
2. add 并 commit 这个修改
3. 将这个修改推入PaaS的develop分支

这样就更新dea\_ng模块到CFv164版本了，其他模块处理类似

注意：如果需要的模块没有tag标记，需要按最新版本cf-release对应到该子模块的commit点进行更新