|  |  |
| --- | --- |
| 文档编号 | EayunCloud-T01 |
| 文档版本 | 1.0.0 |
| 拟制人 | 徐光柳 |
| 日期 | 2016.8.2 |

EayunCloud配置库管理规范

目录

[变更说明 3](#_Toc457900442)

[1 背景概述 4](#_Toc457900443)

[2 配置内容 4](#_Toc457900444)

[3 安全保障 4](#_Toc457900445)

[4 代码库管理 5](#_Toc457900446)

[4.1 分支管理模型 5](#_Toc457900447)

[4.2 新版本发布流程 6](#_Toc457900448)

[4.3 更新包发布流程 6](#_Toc457900449)

[4.4 分支详细说明 6](#_Toc457900450)

[4.5 角色和责任 9](#_Toc457900451)

[5 研发文档库管理 9](#_Toc457900452)

[5.1 文档库目录结构 9](#_Toc457900453)

[5.2 角色和责任 10](#_Toc457900454)

[6 管理文档库管理 10](#_Toc457900455)

[6.1 部门管理目录结构 10](#_Toc457900456)

[6.2 角色和责任 11](#_Toc457900457)

变更说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **日期** | **版本** | **变更位置** | **变更说明** | **作者** |
| 2016-5-12 | 0.1.0 |  | **新建** | 陈浩 |
| 2016-5-27 | 0.2.0 |  | **评审修订** | 陈浩 |
| 2016-6-15 | 1.0.0 |  | **评审修订** | 陈浩 |
| 2016-8-2 | 1.0.0 |  | **修订** | 徐光柳 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 背景概述

鉴于EayunCloud产品存在多版本同时迭代开发的情况，为了代码管理的便捷和可控，EayunCloud配置库管理工具从SVN全部迁移至Git。EayunCloud研发测试团队在公司线上公有云平台下建了一台云主机，安装GitLab作为Git管理工具，EayunCloud全体研发测试团队利用该GitLab作为配置库进行日常代码和文档跟踪管理。

# 配置内容

|  |  |  |
| --- | --- | --- |
| **配置库项目** | **地址** | **用户** |
| ECMC前台代码 | ssh://git@175.25.23.162:10694/eayuncloud/web-ecmc.git | 所有开发人员 |
| Cloud开发文档 | ssh://git@175.25.23.162:10694/eayuncloud/document.git | 所有开发、测试人员 |
| ECSC前后台代码  ECMC后台代码 | ssh://git@175.25.23.162:10694/eayuncloud/business.git | 所有开发人员 |
| 部门管理文档 | ssh://git@175.25.23.162:10694/eayuncloud\_manage/document.git | 部门经理、技术经理 |
| Cloud平台代码 | ssh://git@175.25.23.162:10694/eayuncloud\_platform/platform.git | 部门经理、技术经理和部分骨干 |

# 安全保障

考虑到GitLab服务器在公网上，我们采取了以下措施保证安全性：

1. 在防火墙和安全组上做了限制，只允许北京办公区IP和成都办公区IP可以访问到GitLab服务器；
2. 管理上要求所有开发测试人员只能用SSH的方式操作Git，不允许用HTTP方式操作Git。技术上对成都办公区封了80端口，只对北京办公区放开80端口；
3. SSH端口从默认的22改为10694，增加入侵难度；
4. GitLab服务器禁止用户名密码访问，只允许SSH KEY的方式访问；
5. GitLab配置库数据每天备份一次，滚动保留14个备份，备份数据自动提交至线上其他备份服务器；
6. 所有Git项目只有注册用户才允许访问，根据不同项目，给不同人员分配不同角色的权限（参见上述配置内容）；
7. 对重要的分支（比如master、develop、release和tag等分支），做了保护处理，不允许开发人员直接提交变更，只能用merge的方式修改而merge只有少数几个技术经理和部门经理才有权限，确保了同时迭代开发的代码可控性。

# 代码库管理

代码库存放EayunCloud部门相关产品的源代码，包括ECMC前台代码（html、js等）、ECMC和ECSC后台代码（java、xml等）、平台性组件代码以及构建配置等源代码。

## 分支管理模型

图4-1代码库管理分支管理模型图

## 新版本发布流程

所谓新版本上线，是指完成产品指定版本开发测试且有计划的版本发布。

1. 首先从develop分支拉出feature分支，开发完毕且自测通过。提交测试时，合并到develop分支，再以此拉出release分支，交付测试；
2. 测试在release分支上进行，发现的问题在feature分支上修改，在下一轮测试开始前，合并到develop分支，再从develop分支合并到release分支；
3. 下一轮测试时重复上述过程；
4. 当测试通过后，从release分支拉出tag分支，待上线时合并到master分支，feature分支和release分支删除；
5. 同一个上线计划中，原则上只有一个feature分支计划上线。如果产品部有多个模块的版本计划在一个时间上线，则在同一个feature分支中开发。

## 更新包发布流程

所谓更新包发布是指生产环境中发现的缺陷在紧急修复后上线。

1. 从master分支中拉出hotfix分支，自测通过后，合并到master分支，在master中打出增量包，交付测试。
2. 若测试不通过，继续在hotfix中修改，合并到master分支，循环此过程。
3. 若测试通过，增量包上线，hotfix合并到develop分支，其他未结束的hotfix分支，开发中的feature分支，最近已发版的release分支。所有相关分支合并后，删除此hotfix分支。
4. 短时间内发现的几个缺陷在一个hotfix分支中体现，**原则上同时只能存在一个hotfix分支**。
5. **若有缺陷修复时间较长，单独拉分支，合并到master时注意冲突解决。**

## 分支详细说明

1. master分支

|  |  |
| --- | --- |
| **分支名称** | master |
| **说明** | 稳定分支，和线上环境一致 |
| **允许删除** | 否 |
| **允许直接编码** | 否 |

1. develop分支

|  |  |
| --- | --- |
| **分支名称** | develop |
| **说明** | 不稳定分支，最新开发分支 |
| **分支来源** | master |
| **合并目标** | release |
| **合并时机** | 提交测试 |
| **允许删除** | 否 |
| **允许直接编码** | 否 |

1. feature分支

|  |  |
| --- | --- |
| **分支名称** | feature |
| **说明** | 由产品需求发起，一个分支包含一个或多个特定的需求，以上线计划拉feature分支 |
| **命名规范** | feature-8位创建日期-需求简称 |
| **分支来源** | develop |
| **创建时机** | 确定了需求 |
| **合并目标** | develop |
| **合并时机** | 提交测试 |
| **删除时机** | release发版 |
| **允许直接编码** | 是 |
| **提交代码标准** | 自测通过，提交后不产生编译问题 |

1. hotfix分支

|  |  |
| --- | --- |
| **分支名称** | hotfix |
| **说明** | 生产环境缺陷发起，一个分支包含一个或多个特定的缺陷修复 |
| **命名规范** | hotfix-8位创建日期-问题简称 |
| **分支来源** | master |
| **创建时机** | 确认线上发现了缺陷 |
| **合并目标1** | develop，未删除的feature分支，未删除的hotfix分支 |
| **合并时机1** | hotfix测试通过 |
| **合并目标2** | master，最近的tag分支 |
| **合并时机2** | hotfix上线 |
| **删除时机** | 所有相关分支已合并 |
| **允许直接编码** | 是 |
| **提交代码标准** | 自测通过，提交后不产生编译问题 |

1. release分支

|  |  |
| --- | --- |
| **分支名称** | release |
| **说明** | 由发版计划发起，交付测试用  一个分支可以包含一批特定的需求 |
| **命名规范** | hotfix-8位创建日期-版本简称 |
| **分支来源** | develop |
| **创建时机** | 提交测试 |
| **合并目标** | tag，其他未删除的feature分支 |
| **合并时机** | 发版 |
| **删除时机** | 拉出tag分支 |
| **允许直接编码** | 否 |

1. tag分支

|  |  |
| --- | --- |
| **分支名称** | tag |
| **说明** | 由发版计划发起，以此为程序基线  一个分支可以包含一批特定的需求 |
| **命名规范** | tag-8位创建日期-版本简称 |
| **分支来源** | release |
| **创建时机** | 提交测试 |
| **合并目标** | master |
| **合并时机** | 上线 |
| **允许删除** | 否 |
| **允许直接编码** | 否 |

## 角色和责任

|  |  |
| --- | --- |
| **角色** | **责任** |
| 配置管理员  （部门经理和技术经理兼任） | 管理Gitlab和维护项目、人员、角色、权限  适时拉出feature，hotfix，release，tag分支  适时合并master，develop，release，tag分支  适时删除feature，hotfix，release分支 |
| 研发人员 | 根据需要在feature，hotfix分支上开发，提交代码 |

# 研发文档库管理

研发文档库存放EayunCloud在开发和测试相关的过程文档。考虑到文档修订频次和版本要求，为了管理便捷只建立master分支进行统一管理。

## 文档库目录结构

|  |  |  |  |
| --- | --- | --- | --- |
| **一级目录** | **二级目录** | **三级目录** | **备注** |
| 01需求定义 | 01需求调研 |  | 空 |
| 02需求草稿 |  | 《需求说明书》等 |
| 03需求确认 |  | 《需求确认单》 |
| 02设计开发 | 01设计草稿 | 模块1 | 《概要设计说明书》  《数据库设计说明书》等 |
| 模块2 |
| 数据库变更脚本 |
| 初始数据库脚本 |
| 02设计确认 |  | 评审通过后的文档 |
| 03技术预研 |  | 《技术路线说明书》 |
| 03系统稳定 | 01测试计划 |  | 《测试计划》 |
| 02测试用例 | 版本1测试用例 | 《测试用例》 |
| 版本2测试用例 |
| 03测试报告 | 版本1测试报告 | 《测试报告》 |
| 版本2测试报告 |
| 04用户手册 |  | 《用户手册》 |
| 05线上问题记录跟踪 |  | 《线上问题跟踪表》 |
| 04版本发布 |  |  | 《版本发布记录》 |
| 05会议评审 |  |  | 《会议纪要》  《评审报告》 |

## 角色和责任

|  |  |
| --- | --- |
| **角色** | **责任** |
| 技术经理 | 维护人员和权限 |
| 开发和测试人员 | 读写文件 |

# 管理文档库管理

管理文档库存放EayunCloud部门管理相关的过程文档。考虑到文档修订频次和版本要求，为了管理便捷只建立master分支进行统一管理。

## 部门管理目录结构

|  |  |  |
| --- | --- | --- |
| **一级目录** | **二级目录** | **备注** |
| 01制度规范 | 01管理制度 | 部门管理流程和制度 |
| 02技术规范 | 技术管理规范 |
| 03笔试考题 | 入职考题 |
| 04岗位职责 | 岗位分级和职责描述 |
| 02文档模板 | 01需求开发 |  |
| 02计划管理 |  |
| 03过程管理 |  |
| 04设计开发 |  |
| 05测试执行 |  |
| 06版本发布 |  |
| 07日常维护 |  |
| 99最佳实践 |  |
| 03技术架构 |  |  |
| 04日常维护 | 01本地维护 |  |
| 02线上维护 |  |
| 05过程管理 |  |  |
| 06总结汇报 |  |  |

## 角色和责任

|  |  |
| --- | --- |
| **角色** | **责任** |
| 部门经理 | 维护人员和权限 |
| 部门经理、技术经理、测试经理 | 读写文件 |