**GitLab使用规范**



**Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Change Description** |
| 1.0 | 07/24/2015 | Jonathan Dai | 文档创建 |
| 2.0 | 07/16/2016 | Jonathan Dai | 内容翻新 |
| 2.1 | 07/21/2016 | Jonathan Dai | 更新流程图 |



1 GitLab 2

2 名词解释 2

2.1 代码库 2

2.2 命名空间 / 代码库群组 2

2.3 锁定分支 2

2.4 SSH秘钥 2

3 角色解释 3

3.1 运维管理员 3

3.2 职能经理 3

3.3 普通程序员 3

4 权限系统介绍 3

4.1 权限介绍 3

4.1.1 Site Admin 3

4.1.2 Owner 3

4.1.3 Master 3

4.1.4 Developer 3

4.1.5 Reporter 4

4.2 权限分配 4

4.2.1 命名空间权限 4

4.2.2 代码库权限 4

5 运维日常操作及权限分配 4

5.1 创建账号 4

5.2 创建命名空间 4

5.3 命名空间成员管理 4

5.4 创建代码库 4

5.5 代码库成员管理 5

5.6 运维相关功能 5

6 命名规范 5

6.1 用户 5

6.2 代码库 5

6.3 命名空间 5

7 小结 6

# 



# GitLab

GitLab是开源的类GitHub代码库管理工具。其功能和GitHub非常类似，功能强大，适合大型团队进行代码库的管理。

gitlab的官方文档地址：<http://docs.gitlab.com/ce/>



# 名词解释

## 代码库

最基本的代码管理单位，一个代码库管理一份代码。

## 命名空间 / 代码库群组

多个代码库如果共享相同的权限组、用户组的话，就有需求针对这些代码库创建一个共通的命名空间 / 代码库群组。同一个命名空间共享一份用户列表及其中的用户的权限设定。

## 锁定分支

GitLab里的特定概念，为了保护某些非常重要的分支，某些分支可以被锁定。锁定分支只有固定的几个权限所有者可以向其提交代码。

## SSH秘钥

用户访问代码库的时候可以使用SSH公钥私钥的方式来访问，而不仅仅只是使用用户名和密码的方式，这样增加了访问的便利性和安全性。其所有者是普通用户。



# 角色解释

## 运维管理员

负责gitlab的整体站点搭建和维护，并进行所有资源的创建工作（用户、代码库、命名空间）。同时也负责所有未定义负责人的站点管理工作的执行。

站点的Site Admin权限。

## 职能经理

管理当前职能所属的所有项目和命名空间的。

代码库和命名空间的Owner权限。

## 普通程序员

获取账号，根据被授予的权限进行代码库的访问。

代码库和命名空间的Developer和Master权限。



# 权限系统介绍

详细的权限内容请访问gitlab官方doc查看：<http://docs.gitlab.com/ce/user/permissions.html>

## 权限介绍

### Site Admin

站点管理员账号，由运维控制该账号，妥善保管。拥有所有权限。

### Owner

* 代码库、命名空间的所有者，拥有当前代码库、命名空间最高权限。
* 可以进行人员调整。
* 赋予该角色给代码库或命名空间所属的职能经理。

### Master

* 在代码库中拥有接近于Owner的权限。
* 在命名空间中和非Owner的其他权限相同，无特权。
* 可以对锁定分支进行提交。
* 赋予该角色给代码库对应的项目的技术负责人。

### Developer

* 普通开发程序员。
* 拥有对代码库的读和写权限。
* 无法对锁定分支进行提交。

### Reporter

* 仅具有代码库的读权限。

## 权限分配

### 命名空间权限

多项目、多代码库共享的权限组控制，一般会将相同分类或者相同项目组的多个项目分到同一个命名空间中。

### 代码库权限

代码库可以添加命名空间组员之外的其他成员，该操作可以由Owner或Master两个权限所有者执行。



# 运维日常操作及权限分配

## 创建账号

只能由Site Admin执行该操作。

* 不允许用户自主注册。该行为可以通过全局配置进行关闭。 编辑：gitlab.yml，signup\_enabled改为false
* 创建的账号，默认的私人项目个数请设置为0，方便管理。编辑：gitlab.yml，default\_projects\_limit改为0
* 创建的账号，默认不允许创建命名空间。编辑：gitlab.yml，default\_can\_create\_group改为false
* 创建的账号，默认不允许更改用户名。编辑：gitlab.yml，username\_changing\_enabled改为false

创建的账号请给与默认密码，后续由用户自行修改。

## 创建命名空间

只能由Site Admin执行该操作，创建的命名空间，根据其用途所属，交付其Owner权限给对应的职能经理。

## 命名空间成员管理

由命名空间的Owner（职能经理）进行命名空间其中的人员管理。任何来自项目组及第三方的人员管理请求都应该转到职能经理处，由职能经理判断是否应该被添加进人员列表中。

## 创建代码库

一个代码库必须归属于一个命名空间，即便当前的代码库没有兄弟库，不存在群组关系。为了后面的扩展性考虑，必须先创建一个命名空间，将代码库安置到命名空间中。

只能由Site Admin执行该操作，创建的代码库，根据其所属，交付Owner权限给相关的职能经理。

## 代码库成员管理

同样由代码库的Owner（职能经理）进行人员管理，一般来说只要管理好项目所属的命名空间的成员就可以了。当然偶尔会有例外。任何来自项目组及第三方的人员管理请求都应该转到职能经理处，由职能经理判断是否应该被添加进人员列表中。

## 运维相关功能

gitlab支持LDAP等授权认证方式，具体操作请查询手册。

<http://docs.gitlab.com/ce/administration/auth/README.html>

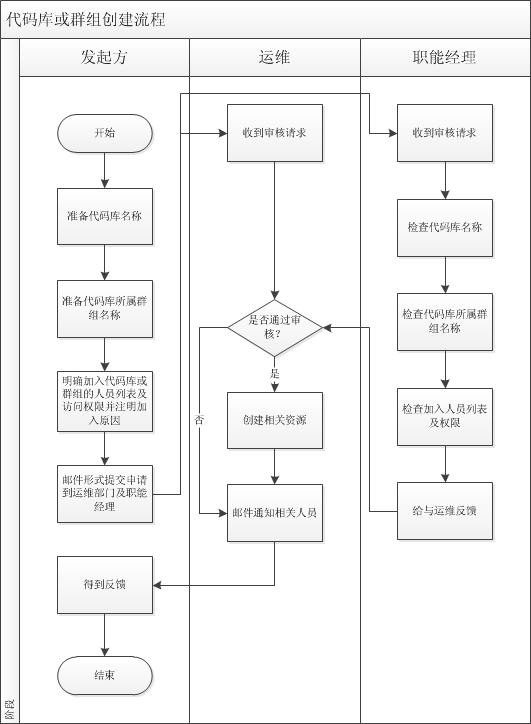
其余运维关心的功能，可以到gitlab的官方文档中查找：

<http://docs.gitlab.com/ce/>

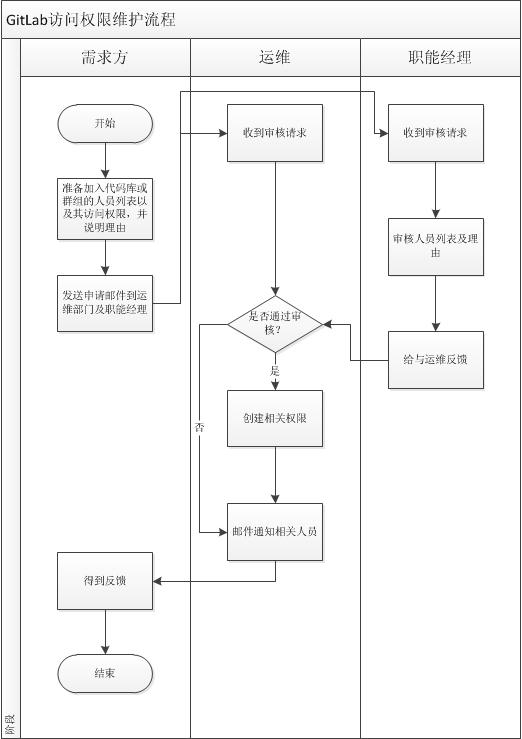


# 维护流程

## 代码库或群组创建流程



## GitLab访问权限维护流程





# 命名规范

## 用户

* 显示名使用员工的中文名字。
* 用户名使用员工的邮箱。

## 代码库

* 命名规范：代码所属项目-仓库功能
* 说明：名称都使用英文单词，驼峰样式，不允许拼音的出现
* 代码库创建的时候，请运维将命名规范告知给创建的需求方，要求对方提供名称，并验证该名称符合规范。如果不符合规范，则打回重新命名，如果多次不符合规范，则可以提交职能经理取名

e.g

FlowerVille-Flash

FlowerVille-PHP

GloryLand-Design

DefenseOfLegend-Art

## 命名空间

* 命名规范：命名空间所属项目-功能
* 说明：名称都使用英文单词，驼峰样式，不允许拼音的出现
* 代码库创建的时候，请运维将命名规范告知给创建的需求方，要求对方提供名称，并验证该名称符合规范。如果不符合规范，则打回重新命名，如果多次不符合规范，则可以提交职能经理取名

e.g

FlowerVille-Flash

FlowerVille-PHP

GloryLand-Design

DefenseOfLegend-Art



# 小结

* 运维控制站点
* 按命名规范创建用户，用户权限最小化，不要给不应该给的权限
* 按命名规范创建命名空间，Owner权限交给职能经理
* 按命名规范创建代码库，必须事先创建代码库归属的命名空间，Owner权限交给职能经理
* 职能经理根据情况维护命名空间和代码库的人员

