# 1、Gitlab 创建备份

### 1.1 创建备份文件

首先我们得把老服务器上的Gitlab整体备份，使用Gitlab一键安装包安装Gitlab非常简单, 同样的备份恢复与迁移也非常简单. 使用一条命令即可创建完整的Gitlab备份。

# gitlab-rake gitlab:backup:create

使用以上命令会在/var/opt/gitlab/backups目录下创建一个名称类似为**1503392426\_gitlab\_backup.tar**的压缩包, 这个压缩包就是Gitlab整个的完整部分, 其中开头的**1503392426**是备份创建的日期

* /etc/gitlab/gitlab.rb 配置文件须备份
* /var/opt/gitlab/nginx/conf nginx配置文件
* /etc/postfix/main.cfpostfix 邮件配置备份

生成完后，/var/opt/gitlab/backups目录创建一个名称类似为**1502357536\_2017\_08\_10\_9.4.3\_gitlab\_backup.tar**的压缩包

### 1.2 更改Gitlab备份目录

当然你也可以通过/etc/gitlab/gitlab.rb配置文件来修改默认存放备份文件的目录

gitlab\_rails['backup\_path'] = "/var/opt/gitlab/backups"

/var/opt/gitlab/backups修改为你想存放备份的目录即可

修改完成之后使用下面命令重载配置文件即可.

gitlab-ctl reconfigure

### 1.3 Gitlab自动备份

1.3.1 定时自动备份

#输入命令crontab -e

sudo crontab -e

#输入相应的任务

0 2 \* \* \* /opt/gitlab/bin/gitlab-rake gitlab:backup:create CRON=1

1.3.2 设置备份过期时间

设置只保存最近7天的备份，编辑 **/etc/gitlab/gitlab.rb** 配置文件,找到gitlab\_rails[‘backup\_keep\_time’]，设置为你想要设置的值，然后保存。

gitlab\_rails['backup\_keep\_time'] = 604800

# 2、 Gitlab迁移

2.1 copy老服务器上面的备份文件到新服务器

### 2.1.1 确保新Gitlab服务器和老Gitlab服务器版本相同

第一步，新服务器上的Gitlab的版本必须与创建备份时的Gitlab版本号相同..

#### 2.1.2 copy老服务器上面的备份文件到新服务器

第二步，如果你没修改过默认备份目录的话，需要将老服务器上的备份文件目录（/var/opt/gitlab/backups目录）下的备份文件拷贝到新服务器上的/var/opt/gitlab/backups目录。

使用scp命令从远程服务器copy文件或者目录到本地

### 2.1.3 从备份文件中恢复gitlab

#### 1、将备份文件权限修改为777

第一步，将备份文件权限修改为777，不然可能恢复的时候会出现权限不够，不能解压的问题

#### 2、执行命令停止相关数据连接服务

第二步，执行命令停止相关数据连接服务

# 停止相关数据连接服务

gitlab-ctl stop unicorn

gitlab-ctl stop sidekiq

# 3、执行命令从备份文件中恢复Gitlab

第三步，执行命令从备份文件中恢复Gitlab

gitlab-rake gitlab:backup:restore BACKUP=备份文件编号

例如我们的备份文件的编号是**1503392426\_gitlab\_backup.tar**，因此执行下面的命令即可恢复gitlab

# gitlab-rake gitlab:backup:restore BACKUP=1503392426

敲完命令后，出现第一个交互页面，输入“yes”继续。

恢复过程中。。。。。

出现第二个交互页面，输入“yes”继续。

第四步，启动Gitlab

# sudo gitlab-ctl start

打开迁移后的Gitlab，进行对比