

# Gitlab进行备份恢复与迁移

## 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1502357536\_2017\_08\_10\_9.4.3\_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1502357536\_2017\_08\_10\_9.4.3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，例如下面代码将备份路径修改为/mnt/backups  
  
gitlab_rails['backup_path'] = '/mnt/backups'
```

修改完成之后使用下面命令重载配置文件即可。

```
gitlabctl reconfigure
```

### 1.3 Gitlab自动备份

#### 1.3.1 定时自动备份

#输入命令

```
crontab -e
```

#输入相应的任务 --- 添加定时任务，每天凌晨两点，执行gitlab备份

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

#重新加载cron配置文件

```
/usr/sbin/service cron reload
```

#重启cron服务

```
/usr/sbin/service cron restart
```

### 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 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

```
chmod 777 1502357536_2017_08_10_9.4.3_gitlab_backup.tar
```

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

```
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1502357536\_2017\_08\_10\_9.4.3，因此执行下面的命令即可恢复gitlab

```
gitlab-rake gitlab:backup:restore BACKUP=1502357536_2017_08_10_9.4.3
```

敲完命令后，出现第一个交互页面，

```
root@ubuntu4146:/var/opt/gitlab/backups# gitlab-rake gitlab:backup:restore  
BACKUP=1502357536_2017_08_10_9.4.3  
Unpacking backup ... done  
Before restoring the database we recommend removing all existing  
tables to avoid future upgrade problems. Be aware that if you have  
custom tables in the GitLab database these tables and all data will be  
removed.  
  
Do you want to continue (yes/no)?  
输入“yes”继续。
```

恢复过程中。。。。。

出现第二个交互页面，

```
Put GitLab hooks in repositories dirs [DONE]
done
Restoring uploads ...
done
Restoring builds ...
done
Restoring artifacts ...
done
Restoring pages ...
done
Restoring lfs objects ...
done
This will rebuild an authorized_keys file.
You will lose any data stored in authorized_keys file.
Do you want to continue (yes/no)?
```

输入“yes”继续。

#### 4、执行命令从备份文件中恢复Gitlab

第四步，启动Gitlab

```
gitlabctl start
```

#### 5、打开迁移后的Gitlab，进行对比