

# 环境安装

## JDK

在linux上使用yum安装是非常粗暴无脑的，但仍然有需要注意的点，不然会掉坑里。这里说一下步骤。

1.执行命令yum -y list java\*查看可安装java版本。执行成功后可以看见如下的结果



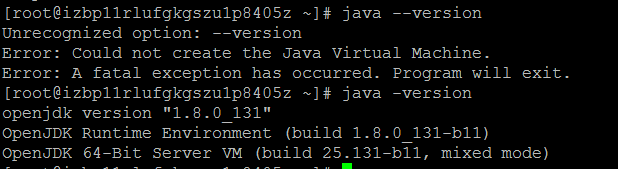
2.选择一个java版本进行安装，这里我们希望安装java1.8，因为我们的机器是64位的，所以选择安装java-1.8.0-openjdk-devel.x86\_64。

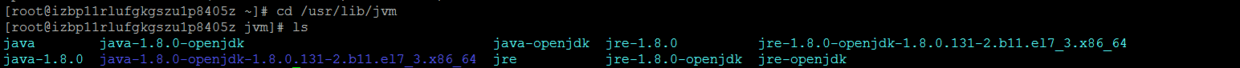
**这里有个地方要注意，上图中我用红框圈起来的两个java版本，要选择-devel的安装，因为这个安装的是jdk，而那个不带-devel的安装完了其实是jre。**

3.执行命令yum install -y java-1.8.0-openjdk-devel.x86\_64。执行完后会看见控制台刷出很多输出。

耐心等待至自动安装完成

输入java --version查看已安装的jdk版本，当出现如下输出表示安装成功。

 5.你可能好奇，yum安装的jdk，被安装到哪里去了？你可以在/usr/lib/jvm目录下找到他们。



Paste\_Image.png

至此，yum安装jdk完成。

### 三、手动安装jdk

如果你不喜欢yum安装的方式，这个很多目录不好控制，想要使用官方提供的安装包进行传统方式的安装，可以使用如下步骤。

#### .tar.gz安装

tar zxvf jdk-8u151-linux-x64.tar.gz

[root@yuanchuangke jdk8]# mv jdk1.8.0\_151 jdk8

[root@yuanchuangke lib]# mkdir kjvm

[root@yuanchuangke kjvm]# mv /home/keyguan/jdk8 .

[root@yuanchuangke etc]# vi /etc/profile

•在profile文件末尾加入：

export JAVA\_HOME=/usr/share/jdk1.6.0\_14

export PATH=$JAVA\_HOME/bin:$PATH

export CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

保存后执行立即生效

. /etc/profile  或者　source /etc/profile

无法更新java的版本

[root@yuanchuangke usr]# java -version

java version "1.7.0\_21"

Java(TM) SE Runtime Environment (build 1.7.0\_21-b11)

Java HotSpot(TM) 64-Bit Server VM (build 23.21-b01, mixed mode)

[root@yuanchuangke usr]# echo $JAVA\_HOME

/usr/lib/kjvm/jdk8

#### ·用文本编辑器打开/etc/profile   rpm包安装

去官网下载对应的rpm格式jdk-8u151-linux-x64.rpm为例

1.执行命令useradd java，新建用户java

Paste\_Image.png

2.执行命令passwd java，设置java用户密码

Paste\_Image.png

3.进入oracle官网，java8下载页面http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html。

Paste\_Image.png

4.选择Accept License Agreement，点击jdk-8u131-linux-x64.tar.gz，获取到下载链接。**注意，获取到的下载链接有时效（具体有效多久不清楚，反正隔天肯定不能用），请尽快复制到linux上进行下载。**

5.进入到/home/java目录下，输入wget +地址，即可开始下载

Paste\_Image.png

6.等待下载成功

Paste\_Image.png

7.查看文件，发现下载后的文件名有奇怪的后缀，重命名下载文件

Paste\_Image.png

8.输入命令tar zxvf jdk-8u131-linux-x64.tar.gz解压安装包

Paste\_Image.png

9.输入命令vim /etc/profile，打开环境变量配置文件

在文件底部输入以下信息，并保存

JAVA\_HOME=/home/java/jdk1.8.0\_131

JRE\_HOME=$JAVA\_HOME/jre

PATH=$PATH:$JAVA\_HOME/bin

CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

export JAVA\_HOME

export JRE\_HOME

export PATH

export CLASSPATH

Paste\_Image.png

10.输入命令source /etc/profile，刷新环境变量配置文件使其立刻生效；输入java --version查看已安装的jdk版本

Paste\_Image.png

**你要以为这就完成了，那就掉坑里了。虽然大部分时候这就够了，但还有一步操作最好做一下。建一个/usr/bin/java的java的超链接。**

ln -s /home/java/jdk1.8.0\_131/bin/java /usr/bin/java

**为什么要建这个超链接，因为一些自己注册的linux服务（如springboot的jar注册的服务），默认情况下从/usr/bin/java路径使用java，yum安装的时候，这个超链接会自动创建，如果你自己下载包安装的话，这个超链接就需要你手动创建了。**

至此，从官网下载包安装jdk完成。

作者：低调的微胖  
链接：http://www.jianshu.com/p/10949f44ce9c  
來源：简书  
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

# GitLab

## 一、系统环境

* 服务器：阿里云主机
* 操作系统：Centos7.0 64位
* 已装软件：Nginx（80端口）、Apache（8080端口）、PHP-FPM（9000端口）

## 二、安装版本

* GitLab分为社区版（GitLab Community Edition）和企业版（GitLab Enterprise Edition）。社区版免费，企业版收费，但是功能比社区版多。根据目前的需求，选择安装社区版（GitLab-CE）。
* 版本号：8.5.4

## 三、安装方式

以前试过源码安装，过程痛苦无比。此次选择官方提供的GitLab-CE Omnibus安装包。GitLab官网上有详细的安装说明，根据自己的操作系统选择相应的版本，按步骤操作即可。  
<https://about.gitlab.com/downloads>

## 四、安装过程

由于国内阿里云主机无法连接国外的GitLab Yum源，所以只能从GitLab中文社区直接下载rpm包进行安装。

curl -LJO https://mirror.tuna.tsinghua.edu.cn/gitlab-ce/yum/el7/gitlab-ce-8.5.4-ce.0.el7.x86\_64.rpm

rpm -i gitlab-ce-8.5.4-ce.0.el7.x86\_64.rpm

GitLab中文社区：<http://www.gitlab.cc>

## 五、GitLab服务构成

GitLab由以下服务构成：

* nginx：静态Web服务器
* gitlab-shell：用于处理Git命令和修改authorized keys列表
* gitlab-workhorse:轻量级的反向代理服务器
* logrotate：日志文件管理工具
* postgresql：数据库
* redis：缓存数据库
* sidekiq：用于在后台执行队列任务（异步执行）
* unicorn：An HTTP server for Rack applications，GitLab Rails应用是托管在这个服务器上面的。

重点讲一下gitlab-shell和gitlab-workhorse。

#### Gitlab Shell

GitLab Shell有两个作用：为GitLab处理Git命令、修改authorized keys列表。

当通过SSH访问GitLab Server时，GitLab Shell会：

1. 限制执行预定义好的Git命令（git push, git pull, git annex）
2. 调用GitLab Rails API 检查权限
3. 执行pre-receive钩子（在GitLab企业版中叫做Git钩子）
4. 执行你请求的动作
5. 处理GitLab的post-receive动作
6. 处理自定义的post-receive动作

当通过http(s)访问GitLab Server时，工作流程取决于你是从Git仓库拉取(pull)代码还是向git仓库推送(push)代码。如果你是从Git仓库拉取(pull)代码，GitLab Rails应用会全权负责处理用户鉴权和执行Git命令的工作；如果你是向Git仓库推送(push)代码，GitLab Rails应用既不会进行用户鉴权也不会执行Git命令，它会把以下工作交由GitLab Shell进行处理：

1. 调用GitLab Rails API 检查权限
2. 执行pre-receive钩子（在GitLab企业版中叫做Git钩子
3. 执行你请求的动作
4. 处理GitLab的post-receive动作
5. 处理自定义的post-receive动作

也许你会奇怪在通过http(s)推送(push)代码的情况下，GitLab Rails应用为什么不在GitLab Shell之前进行鉴权。这是因为GitLab Rails应用没有解析git push命令的逻辑。好的方法是将这些解析代码放在一个地方，这个地方就是GitLab Shell，这样我们就可以在通过SSH进行访问时重用这段代码。实际上，GitLabShell在执行git push命令时根本不会进行权限检查，它是依赖于pre-receive钩子进行权限检查的。而当你执行git pull命令时，权限检查是在命令执行之前的。对git pull命令的权限检查要简单得多，因为你只需要检查一个用户是否可以访问这个仓库就可以了（不需要检查分支权限）。

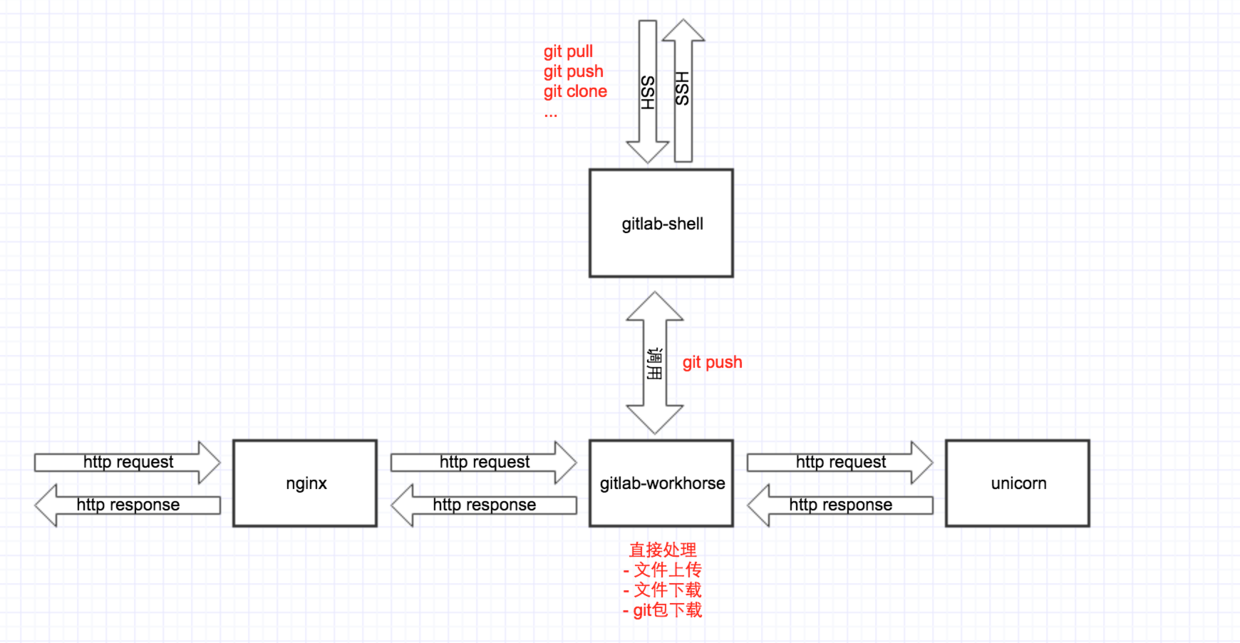
好吧，GitLab Shell这段话都是翻译官网的。链接在这里  
<https://gitlab.com/gitlab-org/gitlab-shell/blob/master/README.md>

最后一段话有点拗口，我对此还是有一点问题的：既然你把git push的逻辑都放在GitLab Shell里面了，为什么不把git pull的逻辑也都放在里面提供重用呢？  
猜想：git pull这段逻辑无法重用，因为通过http(s)方式访问时，要读取仓库的数据并且把这些数据封装成http包返回给客户端；而通过ssh方式访问时，仓库代码数据是通过ssh数据包返回的。两种访问方式返回数据的封装方式不一样，所以也没有必要提供重用。但是我觉得读取仓库数据这段逻辑应该还是重用了的。

#### GitLab Workhorse

GitLab Workhorse是一个敏捷的反向代理。它会处理一些大的HTTP请求，比如文件上传、文件下载、Git push/pull和Git包下载。其它请求会反向代理到GitLab Rails应用，即反向代理给后端的unicorn。官网对GitLab Workhorse的介绍在这里：<https://gitlab.com/gitlab-org/gitlab-workhorse/>

## 六、GitLab工作流程



GitLab工作流程图

## 七、配置

#### 配置考量

* 要求能通过子域名git.zn2studio.com访问GitLab站点并且站点内的仓库地址也要用子域名显示。
* 要求使用腾讯企业邮箱的SMTP服务器发送邮件。
* 要求使用HTTP请求方式。
* 要求能使用SSH连接方式。
* 要求避免与已装软件的端口冲突
* 要求使用系统已安装的Nginx服务器

#### 配置过程

* 修改GitLab配置文件，停用GitLab内置Nginx  
  nginx['enable'] = false
* 使用系统已经安装的Nginx给gitlab-workhorse作反向代理
* 因为unicorn的默认端口是8080,与系统已存在的Apache端口冲突，修改Apache端口为8000（也可以修改unicorn的端口）
* 修改GitLab配置文件中的external\_url  
  external\_url 'http://git.zn2studio.com'  
  修改这个配置会影响GitLab里面显示的仓库链接
* 修改GitLab邮件服务配置，使用腾讯企业邮箱的SMTP服务器
* gitlab\_rails['smtp\_enable'] = true
* gitlab\_rails['smtp\_address'] = "smtp.exmail.qq.com"
* gitlab\_rails['smtp\_port'] = 25
* gitlab\_rails['smtp\_user\_name'] = "xxx"
* gitlab\_rails['smtp\_password'] = "xxx"
* gitlab\_rails['smtp\_domain'] = "smtp.qq.com"
* gitlab\_rails['smtp\_authentication'] = 'plain'

gitlab\_rails['smtp\_enable\_starttls\_auto'] = true

## 参考

[GitLab搭建以及配置](http://www.jianshu.com/p/a22eaa1fcfe7)

# Jenkins

Jenkins是用java编写的开源持续集成工具，目前被国内外各公司广泛使用。本章教大家如何在linux服务器上安装Jenkins。

## 安装环境

CentOS7.3 64

CPU:1.2K\*40

内存：64g，目前还剩3%。。。

lscpu

free –m

软件：JDK1.8及以上

## 安装过程

Tomcat：修改端口号（<http://blog.csdn.net/jay_1989/article/details/52870760>）

从Jenkins官方网站<https://jenkins.io/>下载最新的war包

sudo wget http://mirrors.jenkins.io/war/latest/jenkins.war

## 参考

[依赖JDK](http://www.jianshu.com/p/10949f44ce9c)

# Tomcat

[如何设置tomcat显示文件列表](https://jingyan.baidu.com/article/ff42efa9262df5c19e220219.html): **conf/web.xml**

**修改listings属性**为true即可。

# 文件下载S端设计

请求文件列表：

请求缩列图列表：

文件名字，文件长度，文件格式，创建时间，修改时间

下载单个文件：重定向到其他sdcard位置

断点下载：

## 参考

[Java Web端程序实现文件下载的方法分享](https://www.teakki.com/p/57df76511201d4c1629b8ec1)

[java实现文件断点续传下载功能](https://www.teakki.com/p/57df76561201d4c1629b8ef9)

# 3.文件下载C端设计

缩列图，原始图三级缓存策略

# Busybox

[android安装busybox](https://www.bbsmax.com/A/RnJW2qnrzq/)

|  |  |  |
| --- | --- | --- |
| Kw-GOOGLE |  |  |
| busybox |  |  |
| linux busybox web server-中文 | [设备产品WEB管理界面实现](https://segmentfault.com/a/1190000004963423) | 轻量级WEB Server 对比， |
|  | [编译内核+BusyBox定制一个Linux提供ssh和web服务](http://chenpipi.blog.51cto.com/8563610/1390874) | 定制一个linux系统；能让其远程登录和提供web服务 |
| android busybox web server | [android busybox httpd搭建简单web服务器](http://www.cnblogs.com/flycn/p/3677835.html) | busybox httpd -p 8080 -h /sdcard/www  关闭:*busybox ps | grep httpd*  6573 httpd  *kill 6573* |
| Android busybox web 服务器 |  |  |

## 命令学习

[Busybos文件文件命令](http://blog.csdn.net/jeffreyst_zb/article/details/7160984)

### find . –mtime

[Find–atime –ctime –mtime的用法与区别总结](http://blog.csdn.net/abcdef0966/article/details/7607545)

ind . –mtime n:  File waslast modified n\*24 hours ago.

最后一次修改发生在距离当前时间n\*24小时至(n+1)\*24 小时

find . –mtime +n:

最后一次修改发生在n+1天以前，距离当前时间为(n+1)\*24小时或者更早

find . –mtime –n:

最后一次修改发生在n天以内，距离当前时间为n\*24小时以内

### 从其他文件目录拷贝出来？

总部可能一个文件

## 有用的命令

不处理缩略图吧。

查找png文件：find /sdcard/ -name '\*.png'

busybox find /sdcard/ -name '\*.png' -and -mtime +1

## 3.3 busybox httpd

# 其他方案

[它以NGINX作为Web Server](https://wenku.baidu.com/view/a278e4c8aa00b52acfc7cac2.html?from=search)：类似于其他集成开发环境的名称，我们简单的称之为ANMPP，即Android Nginx Mysql Php-fpm Postgresql的缩写，安卓ANMPP集成环境提供了完善的PHP和NGINX模块支持。

ANMPP是需要ROOT的

[i-jetty](https://wenku.baidu.com/view/a83565e32cc58bd63186bd87.html?re=view)

[教你把安卓手机改造成web服务器,支持外网访问](https://wenku.baidu.com/view/7dd8420602768e9950e7384f.html)：[这里也有案例](https://jingyan.baidu.com/article/90bc8fc865c795f652640c59.html)

Baidu

关键字：android web服务器

1. [AndServer，一个Android端的web服务器](http://blog.csdn.net/yanzhenjie1003/article/details/64090436)：580kb，是一个android端的Web服务器，类似Apache或者Tomcat，但又有不同，它是一个普通的Android Library，[开源的](https://github.com/yanzhenjie/AndServer)
2. [NanoHttpd](http://blog.csdn.net/u012455330/article/details/48676835)
3. [如果只是一般的web service应用不依赖servlet，其实真心不必tomcat或者jetty。因为android已经带了org.apache的包，只需要简单的使用httpentity的类就可以实现简单的web server](https://zhidao.baidu.com/question/555463599.html)

Guge:

关键字：

# [android安装busybox](https://www.bbsmax.com/A/RnJW2qnrzq/)

0. 把手机给Root。

1. [下载BusyBox的binary](http://www.busybox.net/downloads/binaries) ，选择最新版本，然后下载对应你的设备架构的版本 （ /proc/cpuinfo），这里我下载了busybox-armv6l，重命名为busybox，下面将以这个文件名为示例。

# NanoHTTPD

KW: NanoHTTPD

[NanoHttpd嵌入式服务器](http://www.jianshu.com/p/1d53ceb6b4ac)

[**基于NanoHttpd的Android视频服务器开发**](http://ticktick.blog.51cto.com/823160/1705767)

[**源码**](https://github.com/NanoHttpd/nanohttpd)**,**

[**源码分析cn**](http://blog.csdn.net/new_abc/article/details/52876681)**1，**

**[源码分析cn2](http://www.it165.net/pro/html/201402/9405.html)**

[Simple webserver](http://www.voidcn.com/blog/new_abc/article/p-6241847.html)

[该是浏览器多请求了一个 /favicon.ico](http://www.jianshu.com/p/1d53ceb6b4ac)

# 跨进程通信

## 传输大数据传输

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| RPC/AIDL文件 | Binder进程间通信，需要内核中拷贝一次内存 | 声明权限保护好 |  |
| LocalSocket | 如果想要Java和C之间更灵活的通信方式，可以使用，只需要Internet权限，实质是Unix Domain Socket（不是通常的network socket，虽然二者接口基本一致），需要内核中拷贝两次内存 | 需要在两端做好校验 |  |
| 共享内存 | 最高效的方法   1. Android里边有个Ashmem, 不需要内存拷贝，需要同步，为何有人说不适合 2. android有匿名共享内存，用MemoryFile | 系统级的权限。 |  |

搜索：Android 共享内存，

## 参考

[在Android中两个进程之间传输大数据](https://www.zhihu.com/question/51478014)

# Linux工具

## RPM

在 Linux 操作系统下，几乎所有的软件均通过RPM 进行安装、卸载及管理等操作。RPM 的全称为Redhat Package Manager ，是由Redhat 公司提出的，用于管理Linux 下软件包的软件。Linux 安装时，除了几个核心模块以外，其余几乎所有的模块均通过RPM 完成安装。RPM 有五种操作模式，分别为：安装、卸载、升级、查询和验证。

PM 安装操作

命令：

rpm -i 需要安装的包文件名

举例如下：

rpm -i example.rpm 安装 example.rpm 包；

rpm -iv example.rpm 安装 example.rpm 包并在安装过程中显示正在安装的文件信息；

rpm -ivh example.rpm 安装 example.rpm 包并在安装过程中显示正在安装的文件信息及安装进度；

RPM 查询操作

命令：

rpm -q …

附加查询命令：

a 查询所有已经安装的包以下两个附加命令用于查询安装包的信息；

i 显示安装包的信息；

l 显示安装包中的所有文件被安装到哪些目录下；

s 显示安装版中的所有文件状态及被安装到哪些目录下；以下两个附加命令用于指定需要查询的是安装包还是已安装后的文件；

p 查询的是安装包的信息；

f 查询的是已安装的某文件信息；

举例如下：

rpm -qa | grep tomcat4 查看 tomcat4 是否被安装；

rpm -qip example.rpm 查看 example.rpm 安装包的信息；

rpm -qif /bin/df 查看/bin/df 文件所在安装包的信息；

rpm -qlf /bin/df 查看/bin/df 文件所在安装包中的各个文件分别被安装到哪个目录下；

RPM 卸载操作

命令：

rpm -e 需要卸载的安装包

在卸载之前，通常需要使用rpm -q …命令查出需要卸载的安装包名称。

举例如下：

rpm -e tomcat4 卸载 tomcat4 软件包

RPM 升级操作

命令：

rpm -U 需要升级的包

举例如下：

rpm -Uvh example.rpm 升级 example.rpm 软件包

RPM 验证操作

命令：

rpm -V 需要验证的包

举例如下：

rpm -Vf /etc/tomcat4/tomcat4.conf

输出信息类似如下：

S.5....T c /etc/tomcat4/tomcat4.conf

其中，S 表示文件大小修改过，T 表示文件日期修改过。限于篇幅，更多的验证信息请您参考rpm 帮助文件：man rpm

RPM 的其他附加命令

--force 强制操作 如强制安装删除等；

--requires 显示该包的依赖关系；

--nodeps 忽略依赖关系并继续操作；

.

## .tar.gz

.tar.gz 和 .tgz

解压：tar zxvf FileName.tar.gz

压缩：tar zcvf FileName.tar.gz DirName