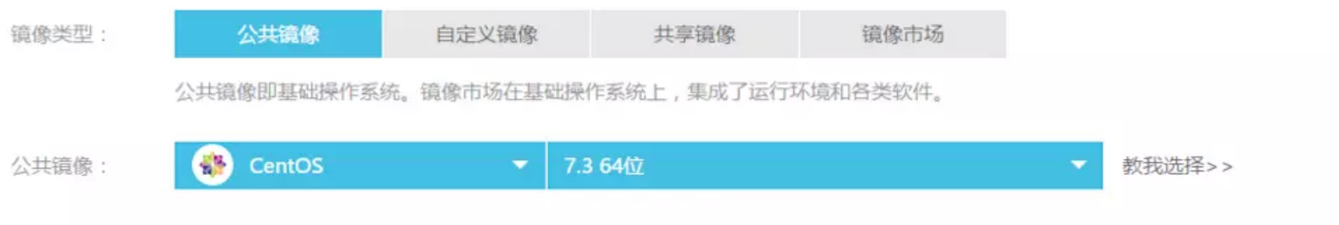
在linux服务器上安装git

git是目前流行的非常好用的版本控制工具，本章教大家如何在linux服务上安装git。这里介绍两种安装方式；1、**yum安装**；2、**从github上下载最新的源码编译后安装**

一、获得一台linux服务器

要在linux下安装git，首先你得先有一台linux服务器，作为小白，手头的机器肯定都是windows的，搞个虚拟机安装对我这种小白简直是折磨人；这里使用最简单的方式获得一台linux服务器，就是从阿里云上租一台。镜像选择CentOS7.3 64位。



镜像类型：公共镜像

公共镜像即基础操作系统。镜像市场在基础操作系统上，集成了运行环境和各类软件。

公共镜像：CentOS 7.3 64位

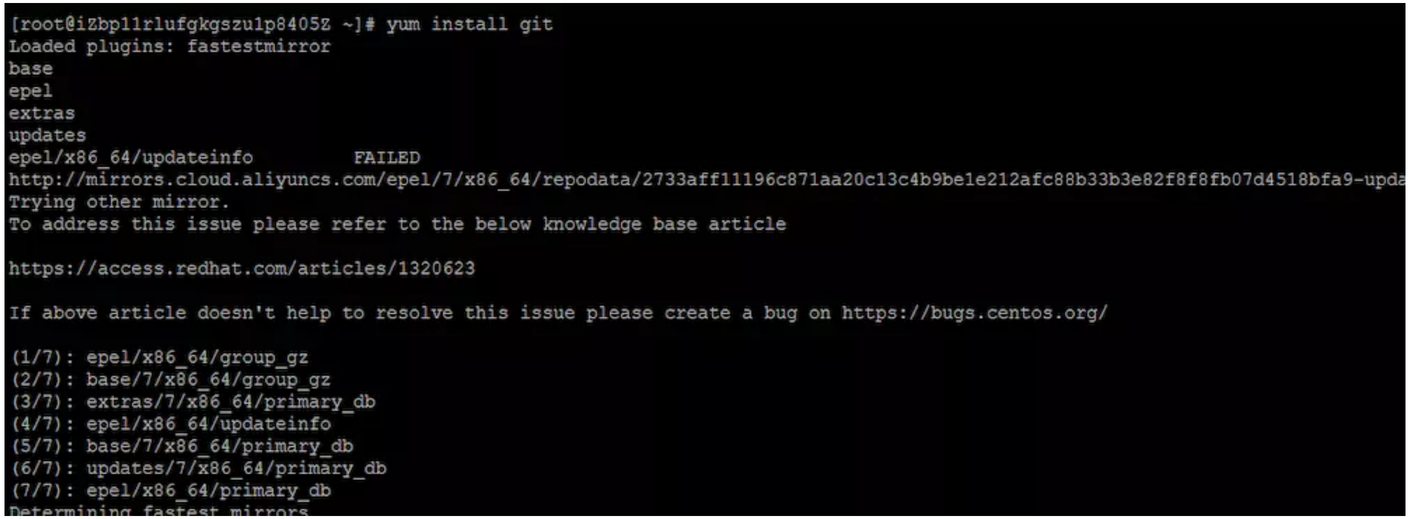
阿里云上租服务器

二、yum安装git

在linux上使用yum安装git非常简单，只需要一行命令

```
yum install git
```

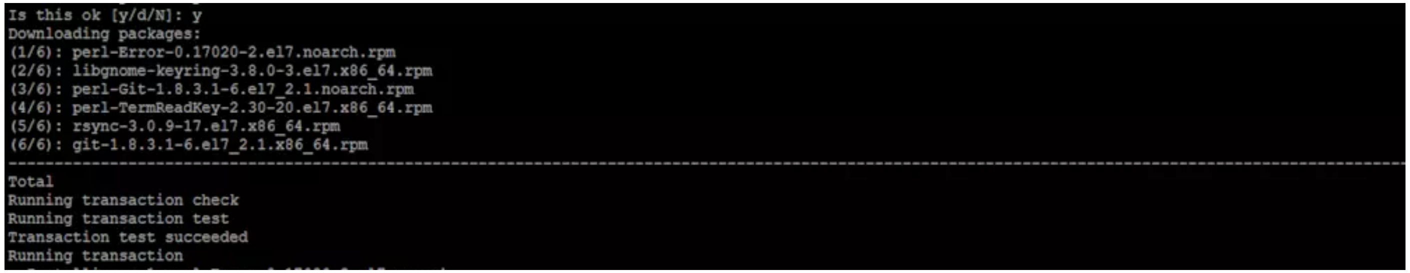
随后就可以看到系统开始自动下载安装



```
[root@iZbp11rlufgkgszu1p8405Z ~]# yum install git
Loaded plugins: fastestmirror
base
epel
extras
updates
epel/x86_64/updateinfo         FAILED
http://mirrors.cloud.aliyuncs.com/epel/7/x86_64/repodata/2733aff11196c871aa20c13c4b9be1e212afc88b33b3e82f8f8fb07d4518bfa9-upda
Trying other mirror.
To address this issue please refer to the below knowledge base article

https://access.redhat.com/articles/1320623

If above article doesn't help to resolve this issue please create a bug on https://bugs.centos.org/

(1/7): epel/x86_64/group_gz
(2/7): base/7/x86_64/group_gz
(3/7): extras/7/x86_64/primary_db
(4/7): epel/x86_64/updateinfo
(5/7): base/7/x86_64/primary_db
(6/7): updates/7/x86_64/primary_db
(7/7): epel/x86_64/primary_db
Determining fastest mirrors
```

yum安装git开始下载

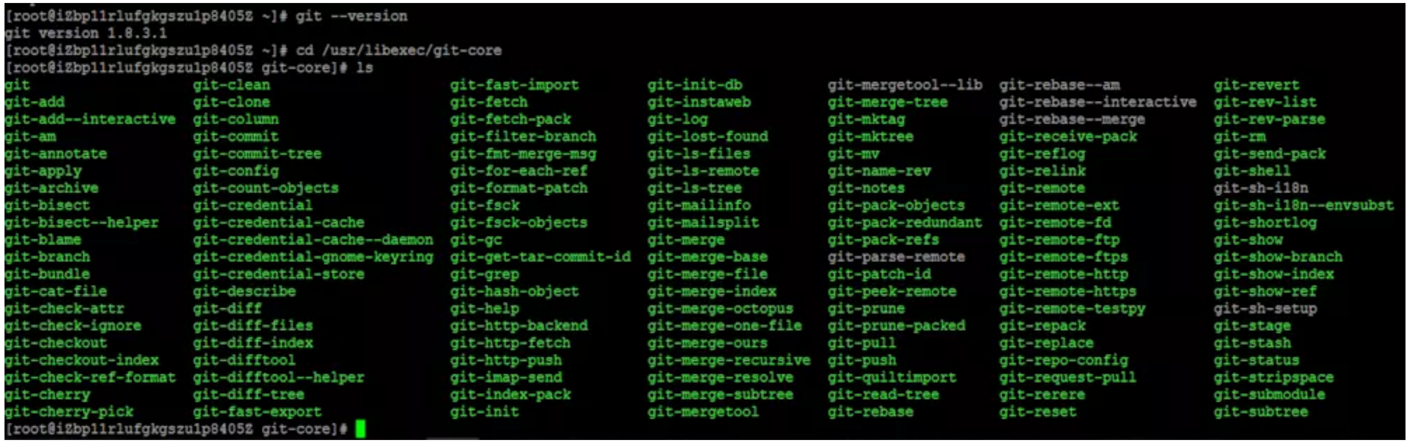
出现提示是否下载的时候输入 y 并按回车。



```
Is this ok [y/d/N]: y
Downloading packages:
(1/6): perl-Error-0.17020-2.el7.noarch.rpm
(2/6): libgnome-keyring-3.8.0-3.el7.x86_64.rpm
(3/6): perl-Git-1.8.3.1-6.el7_2.1.noarch.rpm
(4/6): perl-TermReadKey-2.30-20.el7.x86_64.rpm
(5/6): rsync-3.0.9-17.el7.x86_64.rpm
(6/6): git-1.8.3.1-6.el7_2.1.x86_64.rpm
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
Total
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
```

yum安装git完成

输入 `git --version` 检查git是否安全完成，以及查看其版本号。
顺便说一下，yum安装git被安装在 `/usr/libexec/git-core` 目录下。



```
[root@iZbp11rlufgkgszu1p8405Z ~]# git --version
git version 1.8.3.1
[root@iZbp11rlufgkgszu1p8405Z ~]# cd /usr/libexec/git-core
[root@iZbp11rlufgkgszu1p8405Z git-core]# ls
git                   git-clean                     git-fast-import        git-init-db          git-mergetool--lib  git-rebase--am           git-revert
git-add               git-clone                     git-fetch              git-instaweb         git-merge-tree      git-rebase--interactive  git-rev-list
git-add--interactive  git-column                    git-fetch-pack         git-log              git-mktag           git-rebase--merge        git-rev-parse
git-am                git-commit                    git-filter-branch      git-lost-found       git-mktree          git-receive-pack         git-rm
git-annotate          git-commit-tree               git-fmt-merge-msg      git-ls-files         git-mv              git-reflog               git-send-pack
git-apply             git-config                    git-for-each-ref       git-ls-remote        git-name-rev        git-relink               git-shell
git-archive           git-count-objects             git-format-patch       git-ls-tree          git-notes           git-remote               git-sh-i18n
git-bisect            git-credential                git-fsck               git-mailinfo         git-pack-objects    git-remote-ext           git-sh-i18n--envsubst
git-bisect--helper    git-credential-cache          git-fsck-objects       git-mailsplit        git-pack-redundant  git-remote-fd            git-shortlog
git-blame             git-credential-cache--daemon  git-gc                 git-merge            git-pack-refs       git-remote-ftp           git-show
git-branch            git-credential-gnome-keyring  git-get-tar-commit-id  git-merge-base       git-parse-remote    git-remote-ftps          git-show-branch
git-bundle            git-credential-store          git-grep               git-merge-file       git-patch-id        git-remote-http          git-show-index
git-cat-file          git-describe                  git-hash-object        git-merge-index      git-peek-remote     git-remote-https         git-show-ref
git-check-attr        git-diff                      git-help               git-merge-octopus    git-prune           git-remote-testpy        git-sh-setup
git-check-ignore      git-diff-files                git-http-backend       git-merge-one-file   git-prune-packed    git-repack               git-stage
git-checkout          git-diff-index                git-http-fetch         git-merge-ours       git-pull            git-replace              git-stash
git-checkout-index    git-difftool                  git-http-push          git-merge-recursive  git-push            git-repo-config          git-status
git-check-ref-format  git-difftool--helper          git-imap-send          git-merge-resolve    git-quiltimport     git-request-pull         git-stripspace
git-cherry            git-diff-tree                 git-index-pack         git-merge-subtree    git-read-tree       git-rerere               git-submodule
git-cherry-pick       git-fast-export               git-init               git-mergetool        git-rebase          git-reset                git-subtree
[root@iZbp11rlufgkgszu1p8405Z git-core]#
```

校验yum安装git

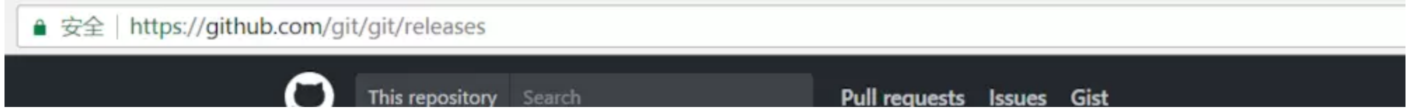
至此，yum安装git完成。

二、从github上下载最新的源码编译后安装git

yum安装这么简单，为什么还要学从github上下载最新的源码编译后安装呢?
刚才输入 `git --version` 命令的时候相信大家也看到了，是 `1.8.3.1` 版本，这个版本还是蛮旧的。yum安装就是这个缺点，版本你不好控制。如果想要使用最新版的git，那还是得自己下载源码安装。具体怎么做呢?

我们还是从一个什么都没安装的linux服务器开始示范。

1.进入git在github上的发布版本页面 `https://github.com/git/git/releases` 。在这个页面我们可以找到所有git已发布的版本。这里我们选择最新版的 `tar.gz` 包。



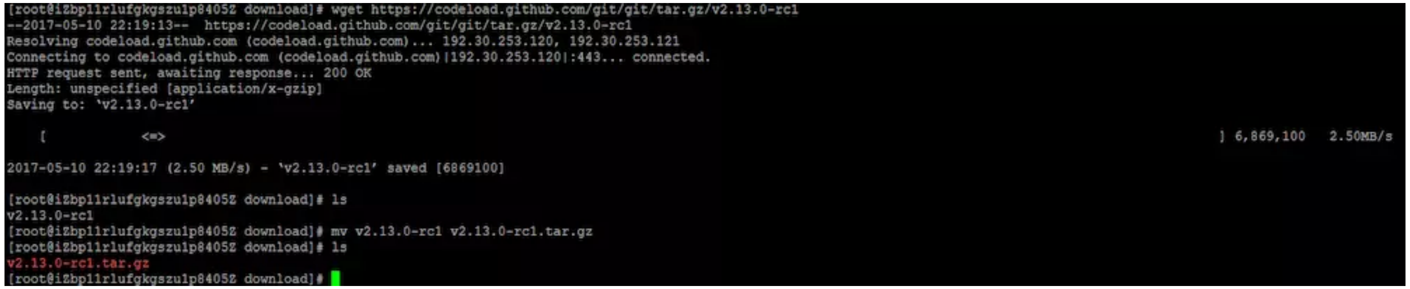
git在github上的仓库

2.获取到最新包的下载链接后，我们进入linux服务器，开始下载。

```
wget https://codeload.github.com/git/git/tar.gz/v2.13.0-rc1
```

耐心等待下载完成。我们可以看到下载后的文件名是 `v2.13.0-rc1` ，并不是压缩包的格式，不用担心，这只是链接的问题，手动修改文件名为 `v2.13.0-rc1.tar.gz` 。

```
mv v2.13.0-rc1 v2.13.0-rc1.tar.gz
```



```
[root@iZbp11rlufgkgszu1p8405Z download]# wget https://codeload.github.com/git/git/tar.gz/v2.13.0-rc1
--2017-05-10 22:19:13--  https://codeload.github.com/git/git/tar.gz/v2.13.0-rc1
Resolving codeload.github.com (codeload.github.com)... 192.30.253.120, 192.30.253.121
Connecting to codeload.github.com (codeload.github.com)|192.30.253.120|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/x-gzip]
Saving to: ‘v2.13.0-rc1’

    [            <=>                                                                                           ] 6,869,100   2.50MB/s

2017-05-10 22:19:17 (2.50 MB/s) - ‘v2.13.0-rc1’ saved [6869100]

[root@iZbp11rlufgkgszu1p8405Z download]# ls
v2.13.0-rc1
[root@iZbp11rlufgkgszu1p8405Z download]# mv v2.13.0-rc1 v2.13.0-rc1.tar.gz
[root@iZbp11rlufgkgszu1p8405Z download]# ls
v2.13.0-rc1.tar.gz
[root@iZbp11rlufgkgszu1p8405Z download]# 
```

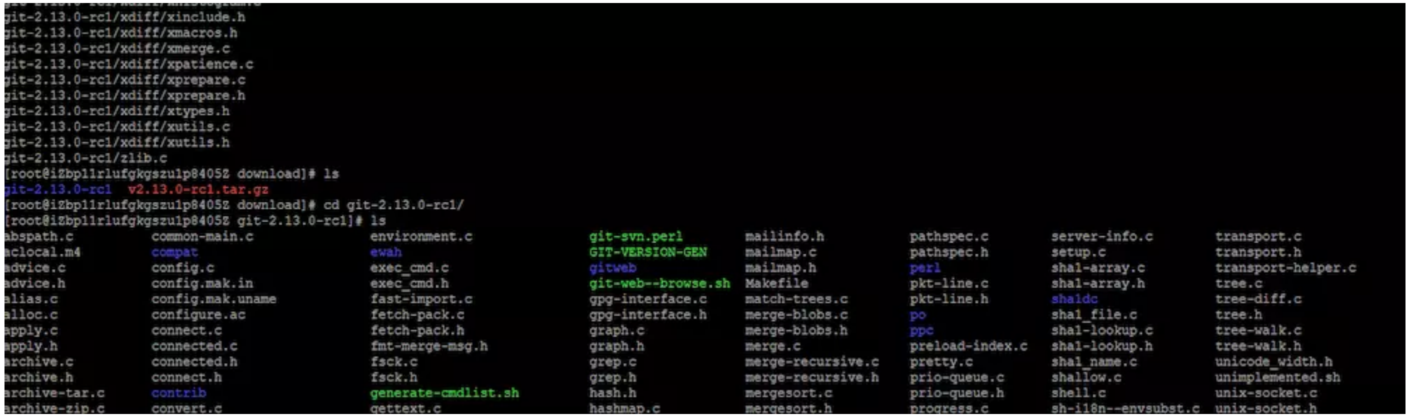
下载git源码

解压压缩包

```
tar -zxvf v2.13.0-rc1.tar.gz
```

进入解压后的文件夹

```
cd git-2.13.0-rc1
```



```
git-2.13.0-rc1/xdiff/xinclude.h
git-2.13.0-rc1/xdiff/xmacros.h
git-2.13.0-rc1/xdiff/xmerge.c
git-2.13.0-rc1/xdiff/xpatience.c
git-2.13.0-rc1/xdiff/xprepare.c
git-2.13.0-rc1/xdiff/xprepare.h
git-2.13.0-rc1/xdiff/xtypes.h
git-2.13.0-rc1/xdiff/xutils.c
git-2.13.0-rc1/xdiff/xutils.h
git-2.13.0-rc1/zlib.c
[root@iZbp11rlufgkgszu1p8405Z download]# ls
git-2.13.0-rc1  v2.13.0-rc1.tar.gz
[root@iZbp11rlufgkgszu1p8405Z download]# cd git-2.13.0-rc1/
[root@iZbp11rlufgkgszu1p8405Z git-2.13.0-rc1]# ls
abspath.c          common-main.c         environment.c            git-svn.perl          mailinfo.h          pathspec.c          server-info.c          transport.c
aclocal.m4         compat                ewah                     GIT-VERSION-GEN       mailmap.c           pathspec.h          setup.c                transport.h
advice.c           config.c              exec_cmd.c               gitweb                mailmap.h           perl                sha1-array.c           transport-helper.c
advice.h           config.mak.in         exec_cmd.h               git-web--browse.sh    Makefile            pkt-line.c          sha1-array.h           tree.c
alias.c            config.mak.uname      fast-import.c            gpg-interface.c       match-trees.c       pkt-line.h          sha1dc                 tree-diff.c
alloc.c            configure.ac          fetch-pack.c             gpg-interface.h       merge-blobs.c       po                  sha1_file.c            tree.h
apply.c            connect.c             fetch-pack.h             graph.c               merge-blobs.h       ppc                 sha1-lookup.c          tree-walk.c
apply.h            connected.c           fmt-merge-msg.h          graph.h               merge.c             preload-index.c     sha1-lookup.h          tree-walk.h
archive.c          connected.h           fsck.c                   grep.c                merge-recursive.c   pretty.c            sha1_name.c            unicode_width.h
archive.h          connect.h             fsck.h                   grep.h                merge-recursive.h   prio-queue.c        shallow.c              unimplemented.sh
archive-tar.c      contrib               generate-cmdlist.sh      hash.h                mergesort.c         prio-queue.h        shell.c                unix-socket.c
archive-zip.c      convert.c             gettext.c                hashmap.c             mergesort.h         progress.c          sh-i18n--envsubst.c    unix-socket.h
```

源码解压完成

3.拿到解压后的源码以后我们需要编译源码了，不过在此之前需要安装编译所需要的依赖。输入如下命令。

```
yum install curl-devel expat-devel gettext-devel openssl-devel zlib-devel gcc perl-ExtUtils-MakeMaker
```

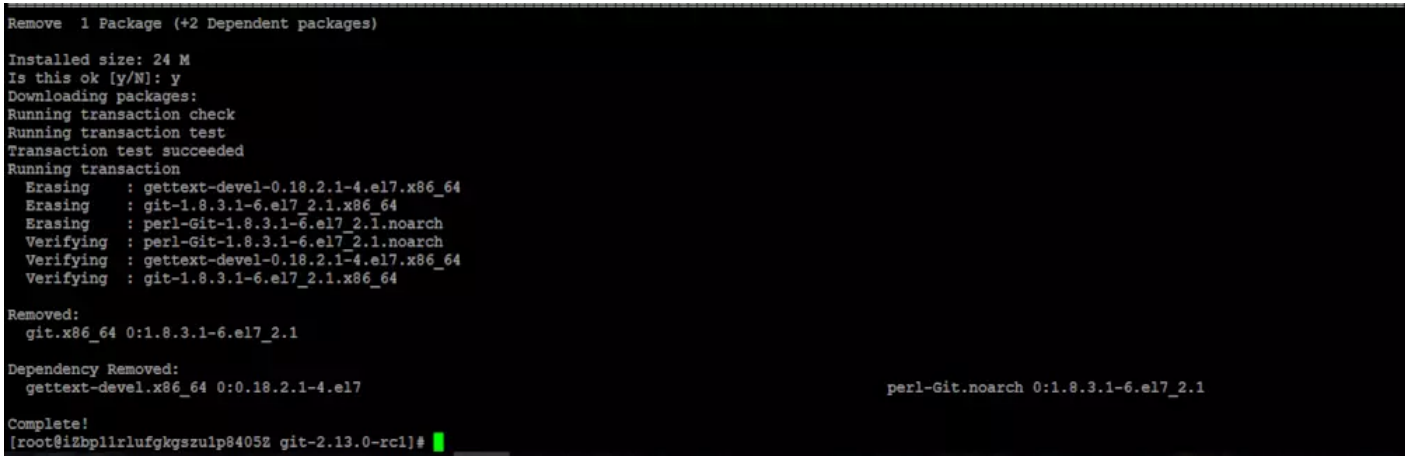
耐心等待安装完成，中途出现提示的时候输入 y 并按回车。

编译源码依赖安装完成

4.提示，安装编译源码所需依赖的时候，yum自动帮你安装了git，这时候你需要先卸载这个旧版的git。

```
yum remove git
```

耐心等待删除完成，中途出现提示的时候输入 y 并按回车。



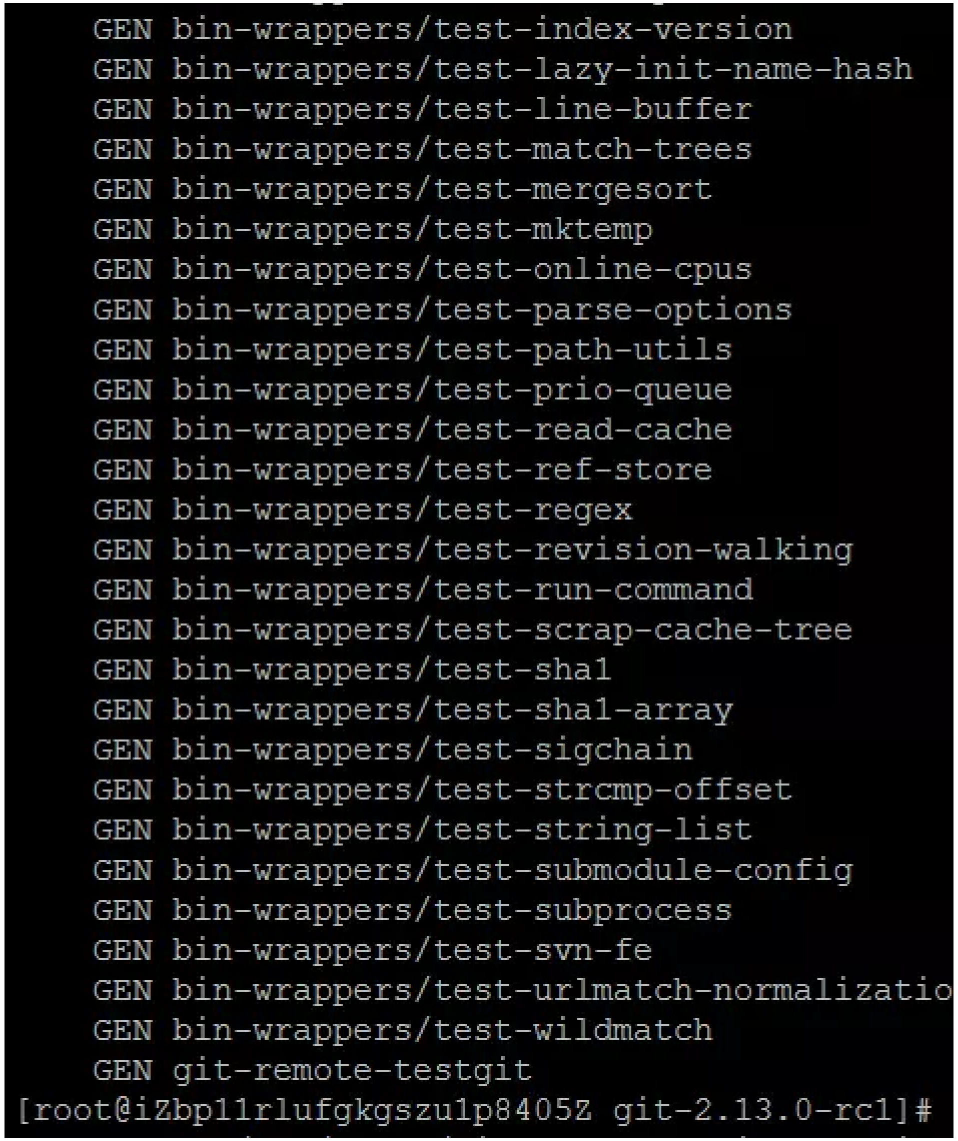
```
Remove  1 Package (+2 Dependent packages)

Installed size: 24 M
Is this ok [y/N]: y
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Erasing    : gettext-devel-0.18.2.1-4.el7.x86_64
  Erasing    : git-1.8.3.1-6.el7_2.1.x86_64
  Erasing    : perl-Git-1.8.3.1-6.el7_2.1.noarch
  Verifying  : perl-Git-1.8.3.1-6.el7_2.1.noarch
  Verifying  : gettext-devel-0.18.2.1-4.el7.x86_64
  Verifying  : git-1.8.3.1-6.el7_2.1.x86_64

Removed:
  git.x86_64 0:1.8.3.1-6.el7_2.1

Dependency Removed:
  gettext-devel.x86_64 0:0.18.2.1-4.el7                                                     perl-Git.noarch 0:1.8.3.1-6.el7_2.1

Complete!
[root@iZbp11rlufgkgszu1p8405Z git-2.13.0-rc1]#
```

移除旧版git

5.编译git源码

```
make prefix=/usr/local/git all
```

耐心等待编译完成，中途可能会花费几分钟的时间。



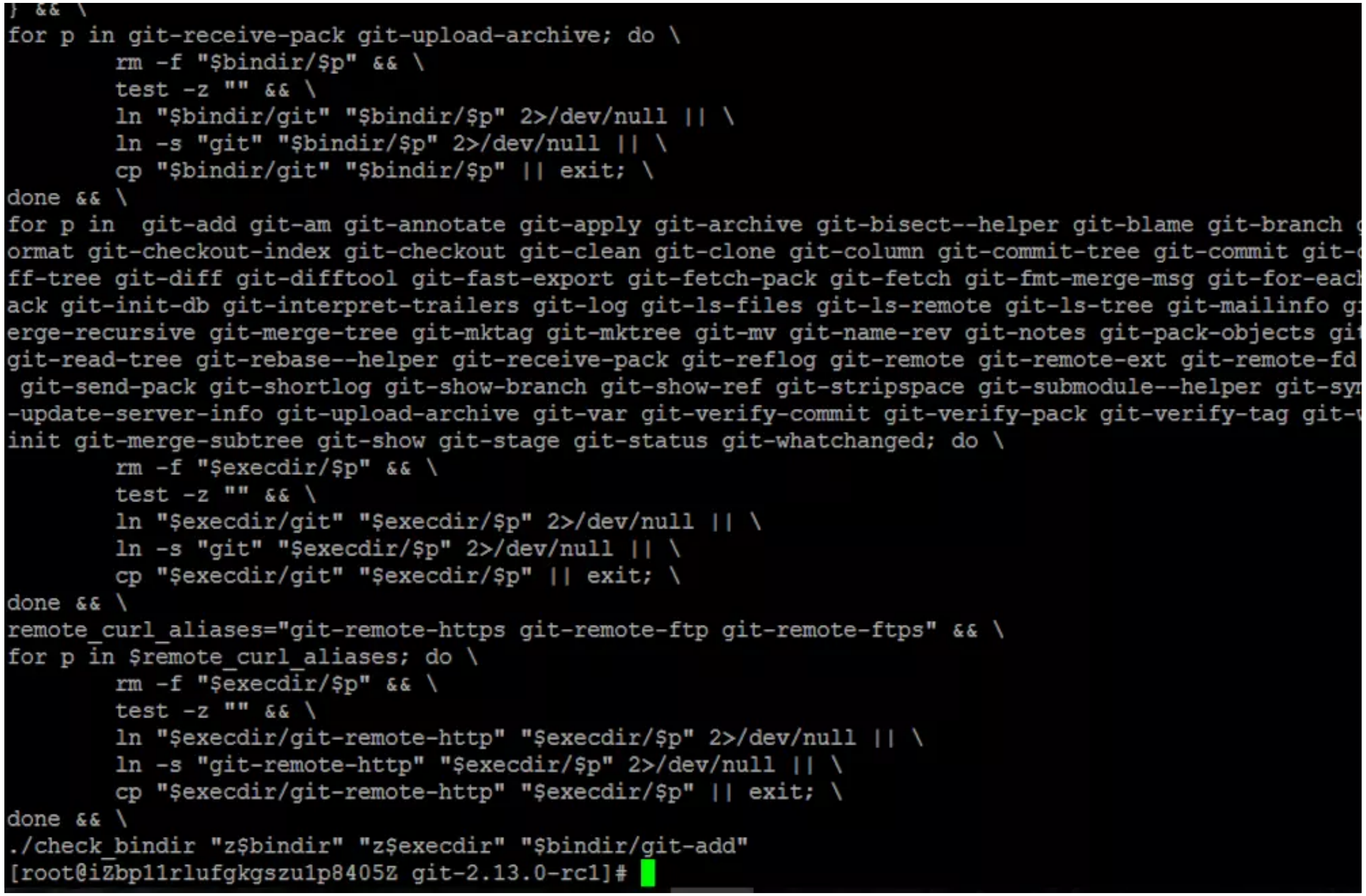
```
    GEN bin-wrappers/test-index-version
    GEN bin-wrappers/test-lazy-init-name-hash
    GEN bin-wrappers/test-line-buffer
    GEN bin-wrappers/test-match-trees
    GEN bin-wrappers/test-mergesort
    GEN bin-wrappers/test-mktemp
    GEN bin-wrappers/test-online-cpus
    GEN bin-wrappers/test-parse-options
    GEN bin-wrappers/test-path-utils
    GEN bin-wrappers/test-prio-queue
    GEN bin-wrappers/test-read-cache
    GEN bin-wrappers/test-ref-store
    GEN bin-wrappers/test-regex
    GEN bin-wrappers/test-revision-walking
    GEN bin-wrappers/test-run-command
    GEN bin-wrappers/test-scrap-cache-tree
    GEN bin-wrappers/test-sha1
    GEN bin-wrappers/test-sha1-array
    GEN bin-wrappers/test-sigchain
    GEN bin-wrappers/test-strcmp-offset
    GEN bin-wrappers/test-string-list
    GEN bin-wrappers/test-submodule-config
    GEN bin-wrappers/test-subprocess
    GEN bin-wrappers/test-svn-fe
    GEN bin-wrappers/test-urlmatch-normalizatio
    GEN bin-wrappers/test-wildmatch
    GEN git-remote-testgit
[root@iZbp11rlufgkgszu1p8405Z git-2.13.0-rc1]#
```

git源码编译完成

6.安装git至 `/usr/local/git` 路径

```
make prefix=/usr/local/git install
```

等待安装完成



```
} && \
for p in git-receive-pack git-upload-archive; do \
        rm -f "$bindir/$p" && \
        test -z "" && \
        ln "$bindir/git" "$bindir/$p" 2>/dev/null || \
        ln -s "git" "$bindir/$p" 2>/dev/null || \
        cp "$bindir/git" "$bindir/$p" || exit; \
done && \
for p in  git-add git-am git-annotate git-apply git-archive git-bisect--helper git-blame git-branch g
ormat git-checkout-index git-checkout git-clean git-clone git-column git-commit-tree git-commit git-c
ff-tree git-diff git-difftool git-fast-export git-fetch-pack git-fetch git-fmt-merge-msg git-for-each
ack git-init-db git-interpret-trailers git-log git-ls-files git-ls-remote git-ls-tree git-mailinfo gi
erge-recursive git-merge-tree git-mktag git-mktree git-mv git-name-rev git-notes git-pack-objects git
git-read-tree git-rebase--helper git-receive-pack git-reflog git-remote git-remote-ext git-remote-fd
 git-send-pack git-shortlog git-show-branch git-show-ref git-stripspace git-submodule--helper git-sym
-update-server-info git-upload-archive git-var git-verify-commit git-verify-pack git-verify-tag git-w
init git-merge-subtree git-show git-stage git-status git-whatchanged; do \
        rm -f "$execdir/$p" && \
        test -z "" && \
        ln "$execdir/git" "$execdir/$p" 2>/dev/null || \
        ln -s "git" "$execdir/$p" 2>/dev/null || \
        cp "$execdir/git" "$execdir/$p" || exit; \
done && \
remote_curl_aliases="git-remote-https git-remote-ftp git-remote-ftps" && \
for p in $remote_curl_aliases; do \
        rm -f "$execdir/$p" && \
        test -z "" && \
        ln "$execdir/git-remote-http" "$execdir/$p" 2>/dev/null || \
        ln -s "git-remote-http" "$execdir/$p" 2>/dev/null || \
        cp "$execdir/git-remote-http" "$execdir/$p" || exit; \
done && \
./check_bindir "z$bindir" "z$execdir" "$bindir/git-add"
[root@iZbp11rlufgkgszu1p8405Z git-2.13.0-rc1]#
```

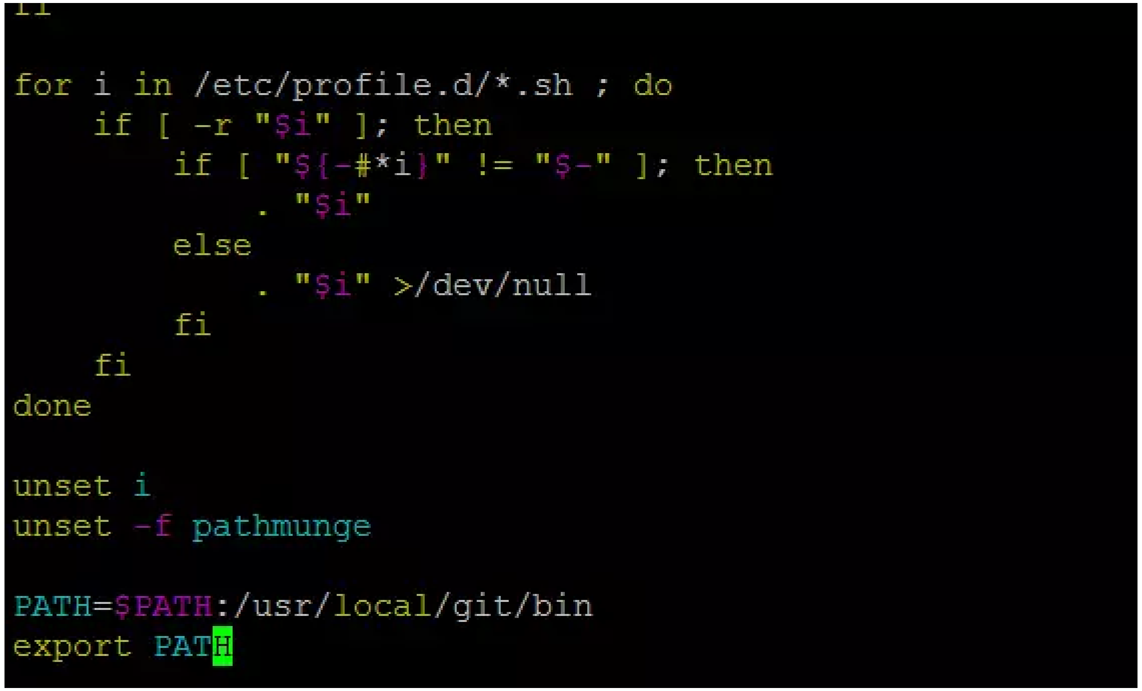
git安装完成

7.打开环境变量配置文件

```
vim /etc/profile
```

在底部加上git相关配置

```
PATH=$PATH:/usr/local/git/bin
export PATH
```



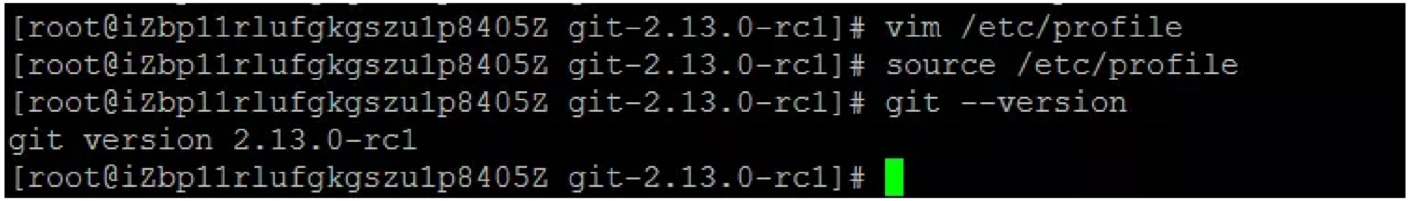
```
fi

for i in /etc/profile.d/*.sh ; do
    if [ -r "$i" ]; then
        if [ "${-#*i}" != "$-" ]; then
            . "$i"
        else
            . "$i" >/dev/null
        fi
    fi
done

unset i
unset -f pathmunge

PATH=$PATH:/usr/local/git/bin
export PATH
```

修改环境变量

刷新环境变量

```
source /etc/profile
```

8.输入 `git --version` 检查git是否安全完成，以及查看其版本号。



```
[root@iZbp11rlufgkgszu1p8405Z git-2.13.0-rc1]# vim /etc/profile
[root@iZbp11rlufgkgszu1p8405Z git-2.13.0-rc1]# source /etc/profile
[root@iZbp11rlufgkgszu1p8405Z git-2.13.0-rc1]# git --version
git version 2.13.0-rc1
[root@iZbp11rlufgkgszu1p8405Z git-2.13.0-rc1]# 
```

git安装完成

至此，从github上下载最新的源码编译后安装git完成。