
CentOS 下搭建 Git 版本 控制系统总结

(编号：*****)

二零一二年五月

对本文件资料享受著作权及其它专属权利，未经书面许可，不得将该等文件资料（其全部或任何部分）披露予任何第三方，或进行修改后使用。

文件更改摘要

[illegible]

目 录

- 1. 引言.....4
- 2. Linux 系统的安装.....5
- 3. 安装前准备工作.....7
- 4. Nginx 安装步骤.....9
- 5. MySQL 的安装.....10
- 6. Redmine 的安装.....11
- 7. Java 的安装.....14
- 8. Jenkins 安装：17
- 9. Maven 安装：19
- 10. Nexus 安装：21
- 11. Git 安装：23
- 12. Gerrit 安装：24

1. 引言

1.1 文档目的

本文档是在 CentOS 下进行的，作为搭建 Git 版本控制服务器的指导说明书。

1.2 文档范围

（在此说明本详细设计文档涉及的需求范围以及设计范围（如只包括客户端界面设计还是包括某需求在各层中的实现等））

1.3 定义参考文献

文献名称	作者	版本	日期

1.4 定义、首字母缩写和缩略语

缩写、术语	解 释	访问路径
CentOS 系统	操作系统	
Nginx	代理 http 服务软件	http://localhost
MySql	数据库软件	
Redmine	Bug 跟踪软件	http://localhost/redmine
Git	源代码管理软件	
Java	Java 编程语言	
Maven	Java 项目管理工具	
Nexus	Maven 仓库管理器	http://localhost/nexus
Jenkins	持续集成软件	http://localhost/jenkins
Gerrit	代码审查软件	http://localhost/gerrit

2. Linux系统的安装

2.1 版本服务器环境说明

主机名称		
处理器		
内存		
系统类型(os)	CentOS 6.2	
语言	LANG=c	
Ip 地址		
用户/密码		
文件系统	/opt	500G
	/home	200G
	/swap	30G
	/	所有剩余空间
安装服务		

2.2 安装步骤

1. 安装系统：用光盘或者 U 盘成功引导后，出现第一个界面说明如下：

Install or upgrade an existing system 安装或升级现有的系统

Install system with basic video driver 安装过程中采用 基本的显卡驱动

Rescue installed system 进入系统修复模式

Boot from local drive 退出安装从硬盘启动

这里选择【**第一项**】，安装或升级现有的系统，回车；

2. 出现是否对 CD 媒体进行测试的提问，这里选择【**Skip**】跳过测试；

3. 出现语言选择的时候，这里选择【**简体中文**】或【**English**】

选择语言为：【**中文（简体）**】或【**English**】；

4. 键盘模式，默认前进即可；

5. 硬盘选项，选择第一项：【**是，丢弃所有数据**】，对硬盘进行格式；

6. 设置计算机名，默认前进即可；

7. 时区选择，默认前进即可；

8. 设置 root 密码，注意：如果你使用的密码过于简单，系统会自动提示，这里选择“仍然使用”；
9. 要使用哪种类型的安装：选择【最后一项】，自定义分区；
10. 自定义分区：点【创建】，选择标准分区，点【创建】；
详细的分区方案详见《linux 硬盘分区方案》
Swap 1G(内存小于 2G 时，设置为内存的 2 倍；内存大于或等于 2G 时，设置为 2G)
- 特别说明：用于正式生产的服务器，切记必须把数据盘单独分区，防止系统出问题
时，保证数据的完整性。比如可以再划分一个/data 专门用来存放数据。
11. 设置好分区后选下一步，出现格式化警告，选择【第二项】，将修改写入磁盘；
12. 默认是第三个，这里选择【第一个】，桌面安装；
13. 你现在可以进行软件定制,选择【现在定制】，根据实际选择软件定制；
14. 重新引导，默认前进，选择【第一项】，同意许可协议；
15. 创建普通账号，可以不用创建，前进即可；
16. 设置时间，默认前进即可；
17. 没有足够的内存配置 kdump，确定；
18. 完成，进入登录界面；
19. 点“其他”，用户名输入：root，输入你 root 账号的密码

2.3 完成系统安装

现在已经登录系统桌面了，至此，**CentOS 6.2 Linux** 安装完成

3. 安装前准备工作

3.1 配置好 IP、DNS、网关

(切记：在做任何操作之前先备份原文件,我们约定备份文件的名称为：源文件名称+bak，例如原文件名称为：centos.txt，那么备份文件名称为：centos.txtbak)

linux 的网卡 IP 地址是存放在文件中的，这个配置文件在/etc/sysconfig/network-scripts 下，名称分别为 ifcfg-eth0,ifcfg-eth1 等，如果你只有一块网卡，就只有 ifcfg-eth0 一个文件，如果你有两块或者两块以上的网卡，就会有 ifcfg-eth1、ifcfg-eth2 等文件的出现。

先查看电脑有几块网卡，终端命令下面输入：

`cd /etc/sysconfig/network-scripts` #进入/etc/sysconfig/network-scripts 文件夹中

`ls -l` #查看/etc/sysconfig/network-scripts 文件夹中所有的文件，发现只有一个 ifcfg-eth0，说明电脑只有一块网卡

`cd` #返回

`cp`

`/etc/sysconfig/network-scripts/ifcfg-eth0 /etc/sysconfig/network-scripts/ifcfg-eth0bak110816` #备份文件

`vi /etc/sysconfig/network-scripts/ifcfg-eth0` #进入网卡 ifcfg-eth0 配置文件

`cp /etc/resolv.conf /etc/resolv.confbak110816` #备份文件

`vi /etc/resolv.conf` #设置 DNS 服务器

如果不行，就重启一下服务器：`shutdown -r now` 或者 `reboot`

到此，网络配置完成！确保使用远程连接工具能够连接服务器

(本处参考地址：<http://www.osyunwei.com/archives/423.html>)

3.2 配置防火墙，开启 80 端口、3306 端口

`vi /etc/sysconfig/iptables`

特别提示：很多人把规则添加到防火墙配置的最后一行，导致防火墙启动失败，正确的应该是添加到默认的 22 端口这条规则的下面

添加防火墙规则如下所示：

```
#####  
# Firewall configuration written by system-config-firewall  
# Manual customization of this file is not recommended.  
  
*filter  
  
:INPUT ACCEPT [0:0]  
:FORWARD ACCEPT [0:0]  
:OUTPUT ACCEPT [0:0]  
  
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT  
-A INPUT -p icmp -j ACCEPT  
-A INPUT -i lo -j ACCEPT  
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT  
-A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT  
-A INPUT -m state --state NEW -m tcp -p tcp --dport 3306 -j ACCEPT  
-A INPUT -j REJECT --reject-with icmp-host-prohibited  
-A FORWARD -j REJECT --reject-with icmp-host-prohibited  
  
COMMIT  
  
#####  
  
/etc/init.d/iptables restart #最后重启防火墙使配置生效
```

3.3 gcc 安装

安装之前，先安装 gcc

```
#yum install gcc
```


4. Nginx安装步骤

1 . 本文使用 **Nginx-1.2.0** 安装

```
# rpm -ivh
```

```
http://nginx.org/packages/centos/6/x86_64/RPMS/nginx-1.2.0-1.el6ngx.x86_64.rpm
```

do passwd git

下载地址：<http://nginx.org/packages/centos/6/>

2 . 查看安装的 **Nginx** 的属性

```
# nginx -V
```

3 . 设置为服务

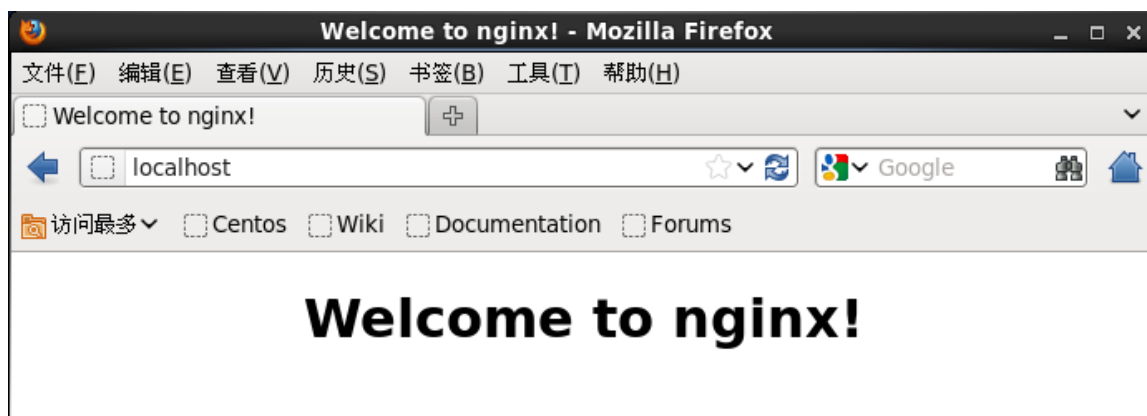
```
# chkconfig nginx on
```

4 . 测试

```
# service nginx start
```

Starting nginx;

5 . 访问：<http://localhost> 成功如下：



5. MySQL的安装

1、本文使用 **yum** 安装

```
# yum install mysql-server
```

```
# chkconfig mysqld on
```

```
# service mysqld start
```

2、修改 **root** 的密码为'**root**' (演示用，请自行设置密码)

```
# /usr/bin/mysqladmin -u root password 'root'
```

3、测试

```
# mysql -u root -p
```

```
Enter password:
```

```
mysql> exit;
```

```
Bye
```

6. Redmine的安装

1. Redmine 需要安装一些头文件

```
# yum install mysql-devel ImageMagick ImageMagick-devel
```

2. 安装 ruby

```
# yum install ruby ruby-devel rubygems
```

3. 版本查看

```
# ruby --version
```

```
ruby 1.8.7 (2011-06-30 patchlevel 352) [x86_64-linux]
```

```
# gem --version
```

```
1.3.7
```

4. Redmine 1.4.1 使用 bundler 安装

```
#gem install bundler
```

5. 查看 gem 包

```
# gem list --local
```

```
*** LOCAL GEMS ***
```

```
bundler (1.1.3)
```

6. 安装 Redmine

```
# wget http://rubyforge.org/frs/download.php/76033/redmine-1.4.1.tar.gz
```

```
# cp redmine-1.4.1.tar.gz /usr/local
```

```
# cd /usr/local
```

```
# tar zxvf redmine-1.4.1.tar.gz
```

7. 建立链接

```
# ln -s /usr/local/redmine-1.4.1 /usr/local/redmine
```

```
# cd /usr/local/redmine
```

8. 使用 bundle 安装 (本文不是用 development, test 模式 , postgresql, sqlite 数据库)

```
# bundle install --without development test postgresql sqlite
```

9. 建立数据库

```
# mysql -u root -p
```

```
mysql> create database redmine character set utf8;

mysql> create user 'redmine'@'localhost' identified by 'my_password';

mysql> grant all privileges on redmine.* to 'redmine'@'localhost';

mysql> exit;
```

10. 修改数据库配置文件

```
# cd /usr/local/redmine/config

# cp database.yml.example database.yml

# vi database.yml
```

11. 主要修改 **production** 部分(数据库的用户名和密码)

```
production:

  adapter: mysql

  database: redmine

  host: localhost

  username: redmine

  password: my_password

  encoding: utf8
```

12. 生成会话存储

```
# rake generate_session_store

Please install RDoc 2.4.2+ to generate documentation.

可以忽略这句:Please install RDoc 2.4.2+ to generate documentation.
```

13. 创建数据库结构

```
# RAILS_ENV=production rake db:migrate
```

14. 初始化数据库

```
# RAILS_ENV=production rake redmine:load_default_data
```

15. 测试

```
ruby script/server webrick -e production

访问：http://localhost:3000
```

16. Nginx 代理，需修改 **redmine** 的 **environment.rb**

```
# vi /usr/local/redmine/config/environment.rb

在最下面添加如下代码

Redmine::Utils::relative_url_root = "/redmine"
```

建立链接，否则 javascript,css 文件会找不到

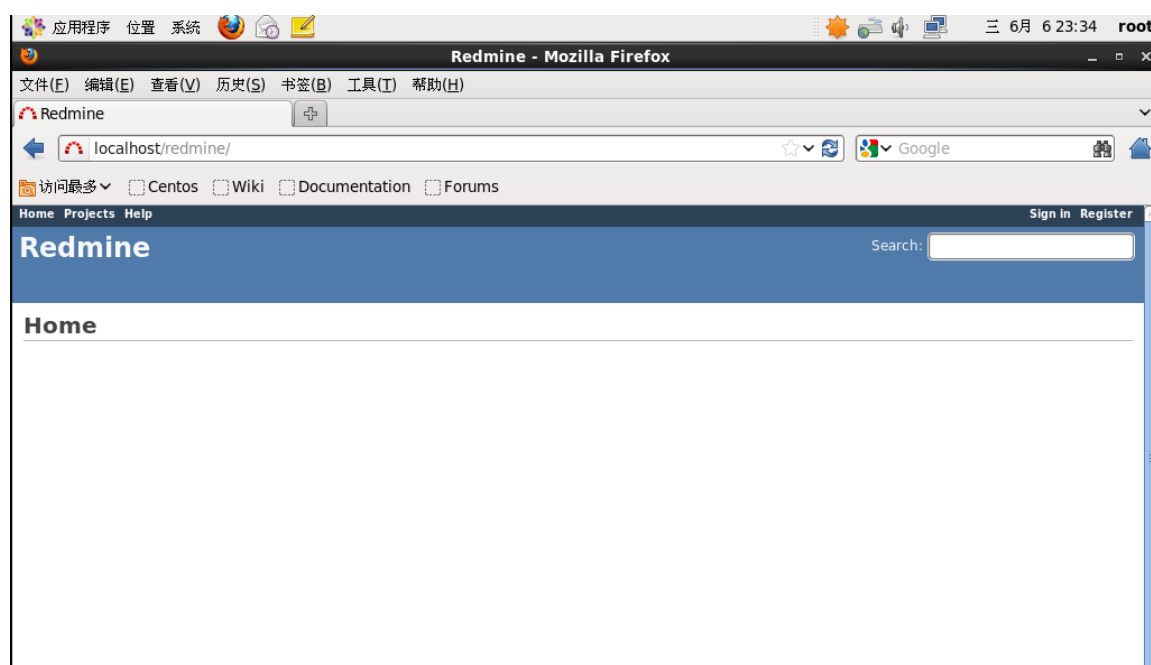
```
ln -s /usr/local/redmine/public /usr/local/redmine/public/redmine
```

17. 更改 nginx 配置

```
# vi /etc/nginx/conf.d/default.conf
```

```
location /redmine/ {  
    proxy_pass http://127.0.0.1:3000;  
}
```

测试访问：<http://localhost/redmine>



7. Java的安装

7.1 安装 JDK

```
# rpm -ivh jdk-7u4-linux-x64.rpm
Preparing... ##### [100%]
1:jdk ##### [100%]
Unpacking JAR files...
    rt.jar...
    jsse.jar...
    charsets.jar...
    tools.jar...
    localedata.jar...
```

默认安装位置为/usr/java

```
# cd /usr/java/
# ll
total 4
lrwxrwxrwx. 1 root root 16 May 6 10:06 default -> /usr/java/latest
drwxr-xr-x. 8 root root 4096 May 6 10:06 jdk1.7.0_04
lrwxrwxrwx. 1 root root 21 May 6 10:06 latest -> /usr/java/jdk1.7.0_04
```

7.2 设置环境变量

```
# update-alternatives --install /usr/bin/java java /usr/java/default/bin/java 2
# update-alternatives --config java
```

There are 2 programs which provide 'java'.

Selection	Command
-----------	---------

```
-----
*+ 1      /usr/lib/jvm/jre-1.6.0-openjdk.x86_64/bin/java
      2      /usr/java/default/bin/java

Enter to keep the current selection[+], or type selection number: 2

# java -version

java version "1.7.0_04"

Java(TM) SE Runtime Environment (build 1.7.0_04-b20)

Java HotSpot(TM) 64-Bit Server VM (build 23.0-b21, mixed mode)
```

有时需要设置 **JAVA_HOME**, 可以放到系统环境变量中, 创建 **shell** 脚本(两种)
Create the Bourne script in /etc/profile.d/java.sh

```
# vi /etc/profile.d/jdk.sh

# Oracle jdk

if [ -d /usr/java/default ]; then

    JAVA_HOME=/usr/java/default

    PATH=$JAVA_HOME/bin:$PATH

    export JAVA_HOME PATH

fi
```

Create the C-shell script in /etc/profile.d/java.csh

```
# vi /etc/profile.d/jdk.csh

# Oracle jdk

if ( -d /usr/java/default ) then

    setenv JAVA_HOME "/usr/java/default"

    setenv PATH "$JAVA_HOME/bin:$PATH"

endif
```

使之立即生效

```
# source /etc/profile
```

7.3 测试

```
# echo $JAVA_HOME
```

```
/usr/java/default
```


8. Jenkins安装：

8.1 Jenkins 安装

```
# wget -O /etc/yum.repos.d/jenkins.repo http://pkg.jenkins-ci.org/redhat/jenkins.repo
# rpm --import http://pkg.jenkins-ci.org/redhat/jenkins-ci.org.key
# yum install jenkins
```

安装的文件信息在 `/etc/init.d/jenkins` 中可以找到,需要修改配置,默认是 `JENKINS_JAVA_CMD=""`, `jenkins` 会查找 `/usr/bin/java`, 正常是可能找到的但是我在安装时,始终出错,改为绝对路径就 OK 了。

```
# vi /etc/sysconfig/jenkins
```

修改 `JENKINS_JAVA_CMD` 这个变量

```
JENKINS_JAVA_CMD="/usr/java/default/bin/java"
```

8.2 测试

```
service jenkins start
```

访问：<http://localhost:8080>

8.3 Nginx 代理

1.使用 nginx 代理，修改配置文件

```
# vi /etc/sysconfig/jenkins
```

```
JENKINS_ARGS="--prefix=/jenkins"
```

2.此时，jenkins 的入口地址为：<http://localhost:8080/jenkins>

```
# vi /etc/nginx/conf.d/default.conf
```

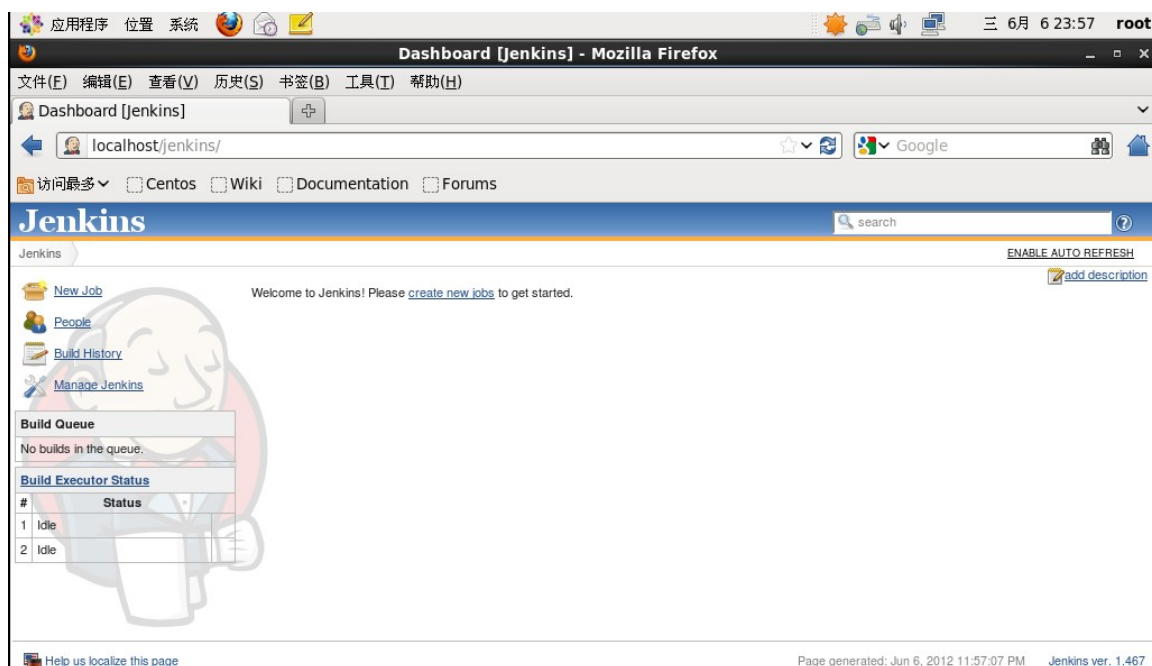
```
location /jenkins/ {
    proxy_pass http://127.0.0.1:8080;
}
```

3.重新加载 nginx 配置

```
service nginx reload
```

4.测试

访问：<http://localhost/jenkins>



9. Maven安装：

9.1 安装

```
# wget http://labs.renren.com/apache-mirror/maven/binaries/apache-maven-3.0.4-bin.tar.gz
# cp apache-maven-3.0.4-bin.tar.gz /usr/local
# cd /usr/local
# tar -xzf apache-maven-3.0.4-bin.tar.gz
# ln -s /usr/local/apache-maven-3.0.4 /usr/local/maven
# rm -f apache-maven-3.0.4-bin.tar.gz
```

1.添加到系统环境变量

```
# vi /etc/profile.d/maven.sh
```

2.添加如下内容

```
# Maven Path
```

```
if [ -d /usr/local/maven ]; then
    M2_HOME=/usr/local/maven
    PATH=$PATH:$M2_HOME/bin
    export M2_HOME PATH
fi
```

3.使之生效

```
# source /etc/profile
```

9.2 测试

```
# mvn -version

Apache Maven 3.0.4 (r1232337; 2012-01-17 16:44:56+0800)

Maven home: /usr/local/maven

Java version: 1.7.0_04, vendor: Oracle Corporation
```

Java home: /usr/java/jdk1.7.0_04/jre

Default locale: en_US, platform encoding: UTF-8

OS name: "linux", version: "2.6.32-220.13.1.el6.x86_64", arch: "amd64", family: "unix"

10. Nexus安装：

10.1 安装

```
# wget http://www.sonatype.org/downloads/nexus-2.0.4-1-bundle.tar.gz
# cp nexus-2.0.4-1-bundle.tar.gz /usr/local
# cd /usr/local
# tar zxvf nexus-2.0.4-1-bundle.tar.gz
# ln -s /usr/local/nexus-2.0.4-1 /usr/local/nexus
# rm -rf nexus-2.0.4-1-bundle.tar.gz
```

10.2 设置为服务

```
# cp /usr/local/nexus/bin/nexus /etc/init.d
# vi /etc/init.d/nexus
```

1.主要修改如下几项：

```
NEXUS_HOME="/usr/local/nexus"
RUN_AS_USER=root
PIDDIR="/var/run"
```

2.修改权限

```
chmod 755 /etc/init.d/nexus
```

3.启动服务

```
service nexus start
```

访问：<http://localhost:8081/nexus>

10.3 Nginx 代理

```
# vi /etc/nginx/conf.d/default.conf
```

```
location /nexus/ {
```

```
proxy_pass http://127.0.0.1:8081;

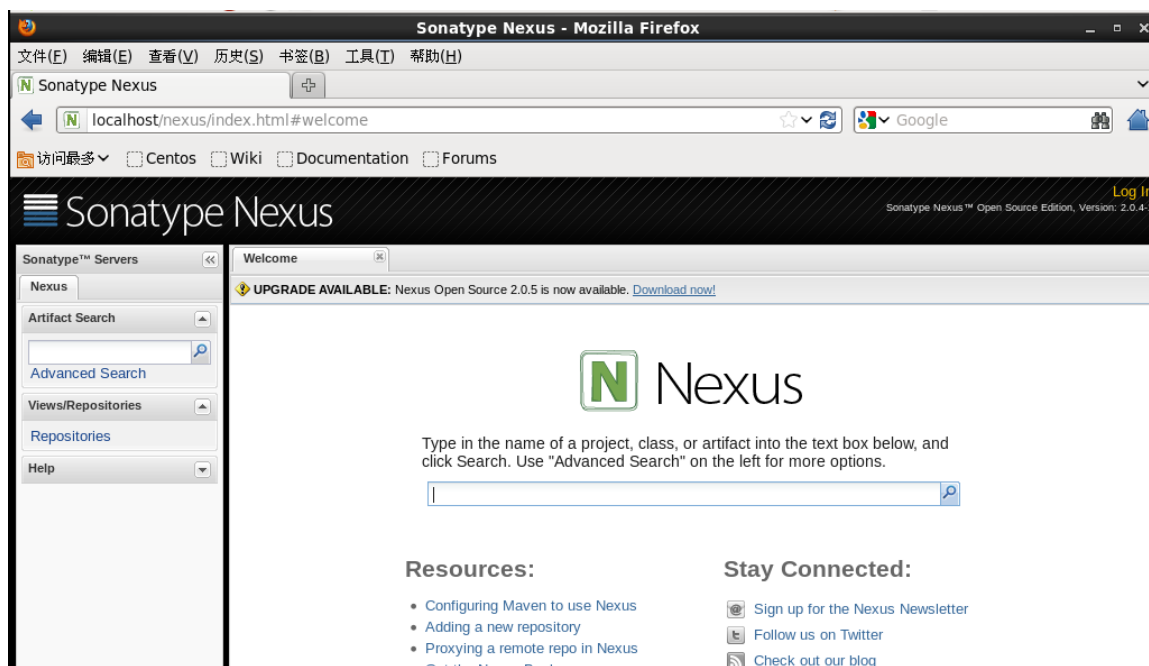
}
```

注: nexus 默认使用 8081 端口, 相关配置信息在\$NEXUS_HOME/bin/nexus.properties 中;

重新加载 nginx 配置

service nginx reload

测试访问: <http://localhost/nexus>



11. Git安装：

11.1 安装

```
#yum install git
```

11.2 使用图形化用户界面

```
# yum install git-gui
```

12. Gerrit安装：

12.1 安装

下载安装包，访问 <http://code.google.com/p/gerrit>，本文下载的是 gerrit-2.4.war 创建数据库 (本文以 MySQL 为例)

```
# mysql -u root -p

mysql> CREATE USER 'gerrit2'@'localhost' IDENTIFIED BY 'gerrit2';

mysql> CREATE DATABASE reviewdb;

mysql> ALTER DATABASE reviewdb charset=latin1;

mysql> GRANT ALL ON reviewdb.* TO 'gerrit2'@'localhost';

mysql> FLUSH PRIVILEGES;
```

1.安装 gerrit

```
# cp gerrit-2.3.war /usr/local

# cd /usr/local

# java -jar gerrit-2.3.war init -d review_site
```

2.安装过程一般直接回车即可，主要注意一下地方：数据库(本文是用 MySQL,默认为 H2)

```
*** SQL Database
***

Database server type      [H2/?]: mysql
```

3.认证类型(本文使用 http,默认为 OPENID)

```
Authentication method    [OPENID/?]: http
```

4.端口(本文使用 8082，因为前面安装 nexus 已经使用了 8081 端口)

```
*** HTTP Daemon
***
```



```
Behind reverse proxy      [y/N]? y
Proxy uses SSL (https://)  [y/N]?
Subdirectory on proxy server [/]: /gerrit
Listen on address         [*]:
Listen on port            [8081]:
Canonical URL             [http://localhost/gerrit]:
```

```
Initialized /usr/local/gerrit
Executing /usr/local/gerrit/bin/gerrit.sh start
Starting Gerrit Code Review: OK
Waiting for server to start ... OK
Opening browser ...
No protocol specified
```

5.新建 passwd 文件

```
# mkdir /etc/nginx/passwd
# htpasswd -c /etc/nginx/passwd/gerrit2.passwd gerrit2
New password:
Re-type new password:
Adding password for user gerrit2
```

6.查看 gerrit 的配置文件，整个配置文件是这样的

```
# vi /usr/local/gerrit/etc/gerrit.config

[gerrit]
    basePath = git
    canonicalWebUrl = http://localhost/gerrit/

[database]
    type = MYSQL
    hostname = localhost
```

```
database = reviewdb

username = gerrit2

[auth]

type = HTTP

[sendemail]

smtpServer = localhost

[container]

user = root

javaHome = /usr/java/jdk1.7.0_04/jre

[sshd]

listenAddress = *:29418

[httpd]

listenUrl = proxy-http://*:8082/gerrit/

[cache]

directory = cache
```

12.2 Nginx 代理

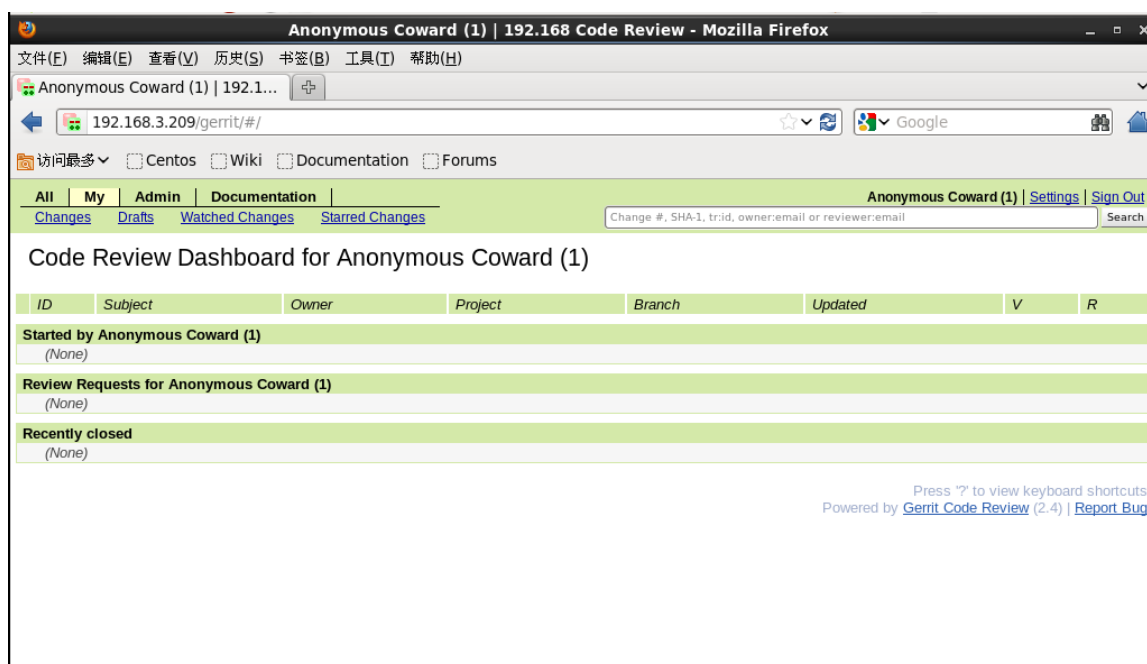
```
# vi /etc/nginx/conf.d/default.conf

location /gerrit/ {
    proxy_pass      http://127.0.0.1:8082;
    proxy_set_header X-Forwarded-For $remote_addr;
    proxy_set_header Host $host;
    auth_basic      "Gerrit Code Review";
    auth_basic_user_file /etc/nginx/passwd/gerrit2.passwd;
}
```

12.3 测试访问

<http://localhost/gerrit>

输入用户名 gerrit2，密码为刚才设置的密码(本文设为 gerrit2);



以上资料参考地址：<http://qizhanming.com/2012/05/05/centos-dev-env-setup/>

<http://fatalove.iteye.com/blog/1332881>

<http://www.infoq.com/cn/articles/Gerrit-jenkins-hudson>