

安装 git 与本地建立连接

1.0: 安装 git

yum install git

1.1: 创建 git 用户:

adduser git

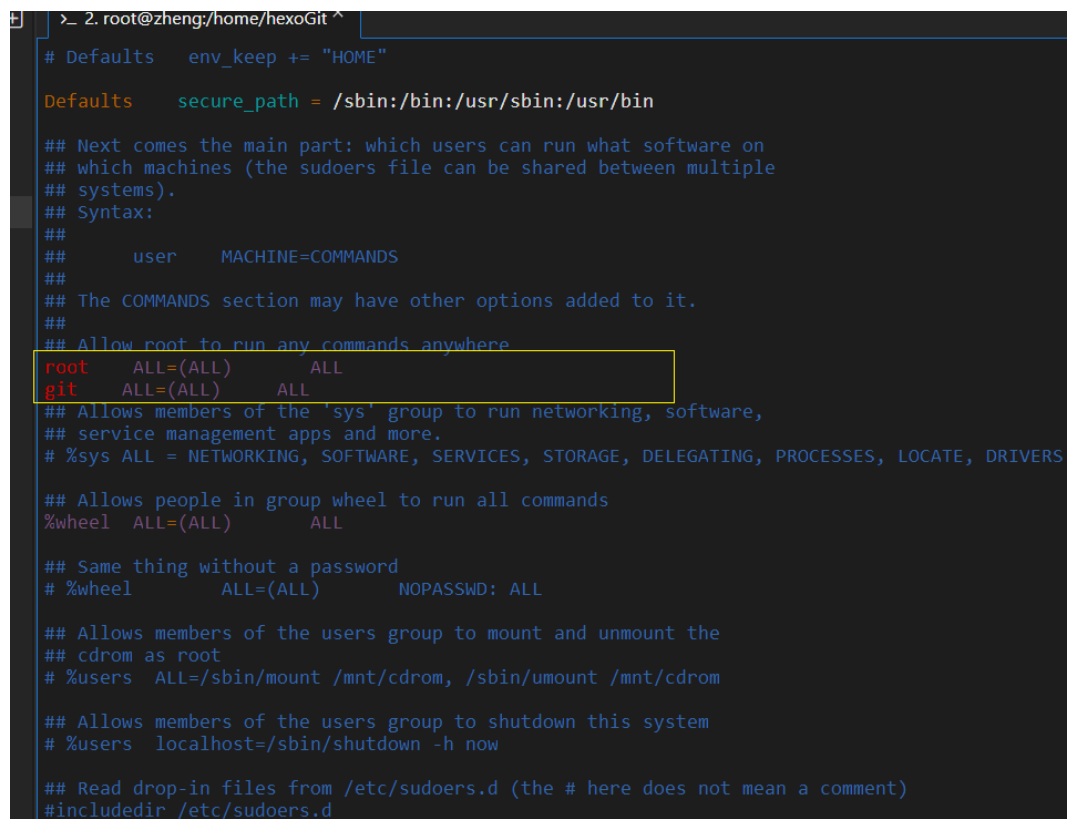
1.2: 添加 git 账户权限:

chmod 740 /etc/sudoers

1.3: 然后使用 vim 编辑器打开文件, 如果你不会使用 vim, 移步 [Vim 基础使用](#):

vim /etc/sudoers

1.4: 找到这个位置, 添加:



```
# Defaults    env_keep += "HOME"

Defaults     secure_path = /sbin:/bin:/usr/sbin:/usr/bin

## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##      user    MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)    ALL
git     ALL=(ALL)    ALL
## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS

## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)    ALL

## Same thing without a password
# %wheel    ALL=(ALL)    NOPASSWD: ALL

## Allows members of the users group to mount and unmount the
## cdrom as root
# %users    ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

## Allows members of the users group to shutdown this system
# %users    localhost=/sbin/shutdown -h now

## Read drop-in files from /etc/sudoers.d (the # here does not mean a comment)
#includedir /etc/sudoers.d
```

git ALL=(ALL) ALL

1.5: 修改权限:

chmod 400 /etc/sudoers

1.6: 设置 git 账户密码:

sudo passwd git

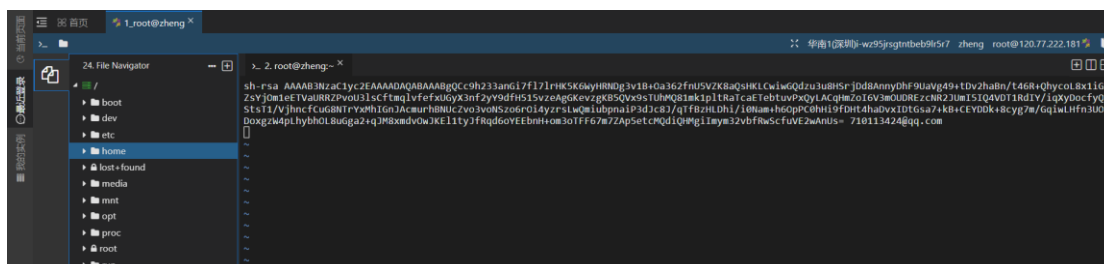
1.7: 切换至 git 用户, 创建 `~/.ssh` 文件夹和 `~/.ssh/authorized_keys` 文件, 并赋予相应的权限

su git

mkdir ~/.ssh

vim ~/.ssh/authorized_keys

1.8: 将我们在 win10 中生成的 id_rsa.pub 文件中的公钥复制到 authorized_keys 中。



1.9: 接着，输入一下命令，赋予权限：

chmod 600 /home/git/.ssh/authorized_keys

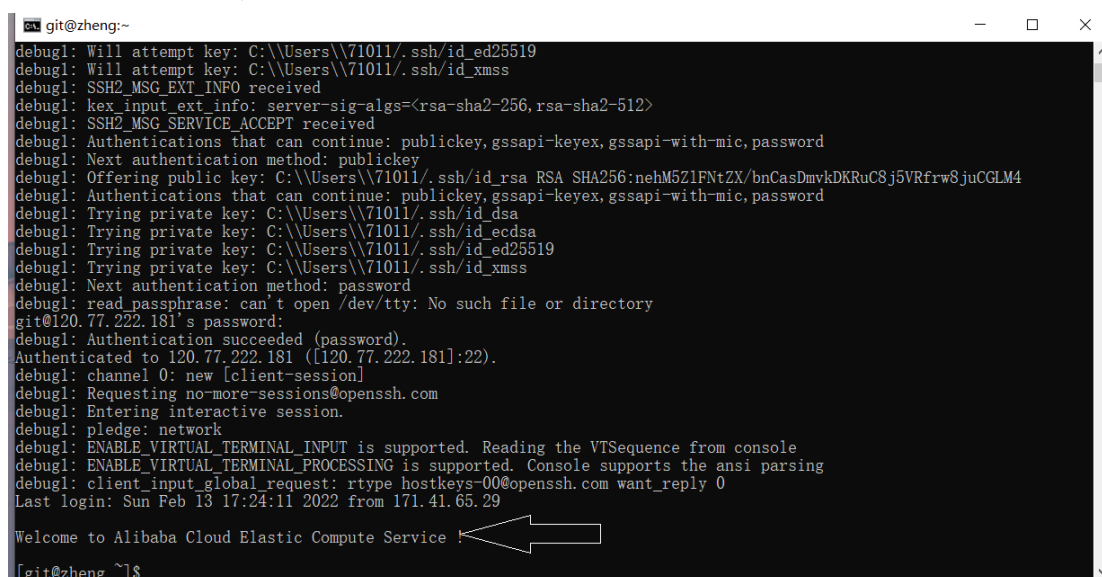
chmod 700 /home/git/.ssh

2.0: 打开 cmd，输入：

ssh -v git@SERVER

其中 SERVER 填写自己的云主机 ip

看到这句话表明成功：



到此为止就完成了服务器与主机的连接了。

配置 git 钩子

1.0: 返回服务端命令行切换到 root 账户：

sudo su

1.1: 创建目录在 home 目录下创建 hexoGit 作为 Git 仓库目录，然后输入：

mkdir /home/hexoGit

1.2: 这个目录就作为我们的仓库地址，接下来赋予权限：

chown -R git:git /home/hexoGit

chmod -R 755 /home/hexoGit

1.3: 接下来创建 hexo 目录作为网站根目录，并赋予权限：

```
mkdir /home/hexo
chown -R git:git /home/hexo
chmod -R 755 /home/hexo
```

1.4: 接下来创建一个空白的 git 仓库:

```
cd /home/hexoGit
git init --bare hexo.git
```

创建一个新的 Git 钩子用于自动部署

1.0: 在 /var/repo/hexo.git 下, 有一个自动生成的 hooks 文件夹。我们需要在里边新建一个新的钩子文件 post-receive。

```
vim /home/hexoGit/hexo.git/hooks/post-receive
```

1.1: 在该文件里面输入以下内容然后保存退出:

```
#!/bin/sh
git --work-tree=/var/hexo --git-dir=/var/repo/hexo.git checkout -f
```

这其实是一个脚本文件, 每当我们推送的时候就会触发这个脚本文件的执行。这样一旦本地仓库变更提交就会触发 Git 钩子, 驱动自动部署。

1.2: 下面修改权限:

```
chown -R git:git /home/hexoGit/hexo.git/hooks/post-receive
chmod +x /home/hexoGit/hexo.git/hooks/post-receive
```

到这里 Git 仓库和钩子 已经搭建完毕了。

安装并配置 nginx

1.1: 安装 nginx

```
yum install -y nginx
```

1.2: 启动服务器:

```
systemctl start nginx
systemctl enable nginx
```

我们要记住的是/etc/nginx/是 nginx 默认的配置路径, 一会要用到。

1.3: 打开 /etc/nginx/nginx.conf 目录

```
vim /etc/nginx/nginx.conf
```

1.4: 修改 nginx 里 server 配置项:

```
server {
    listen      80;          #请先检查服务器是否开启80端口
    listen      [::]:80;
    server_name 120.77.222.181; #此处填写自己域名“www.czcgx.com”, 没有就填写服务器公网ip
    root        /home/hexo;   #此处填写博客存放目录

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;

    error_page 404 /404.html;
    location = /404.html {
    }

    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
    }
}
```

```
104  deploy:
105      type: git
106      repo: git@120.77.222.181:/home/hexoGit/hexo.git
107      branch: master
108      aplayer:
109      meting: true
110
```

修改本地博客文件夹里的 config 文件

在本地编译发布上传一次博客

1.1: 清除缓存文件:

hexo clean

1.2 : 编 译 :

hexo g

1.3: 上传:

hexo d

****: 如果此处出现权限问题请在服务器运行以下命令

chown -R git:git /home/hexoGit/

chown -R git:git /home/hexo

打开服务器重启 nginx 服务

nginx -s reload

此时 home/hexo 文件夹下应该有被推送上来的博客文件

Nginx 安装 ssl 证书

首先开启服务器 443 端口：

入方向

出方向

手动添加

快速添加

Q 输入端口或者授权对象进行搜索

授权策略	优先级	协议类型	端口范围	授权对象	描述	创建时间	操作
<input checked="" type="checkbox"/> 允许	1	自定义 TCP	目的: 443/443	源: 0.0.0.0/0		2022年2月11日 16:12:04	编辑 复制 删除
<input checked="" type="checkbox"/> 允许	1	自定义 TCP	目的: 80/80	源: 0.0.0.0/0		2022年2月11日 12:12:00	编辑 复制 删除
<input checked="" type="checkbox"/> 允许	100	自定义 TCP	目的: 22/22	源: 0.0.0.0/0	System created rule.	2022年1月29日 16:09:08	编辑 复制 删除
<input checked="" type="checkbox"/> 允许	100	全部 (ICMPv4)	目的: -1/-1	源: 0.0.0.0/0	System created rule.	2022年1月29日 16:09:08	编辑 复制 删除
<input checked="" type="checkbox"/> 允许	100	自定义 TCP	目的: 3389/3389	源: 0.0.0.0/0	System created rule.	2022年1月29日 16:09:08	编辑 复制 删除

删除

首先查看有没有安装 ssl 模块

1.0: 基于 nginx 版本不同所以 nginx 文件目录也会有所不同

找到 nginx 的 sbin 目录

首先查看此目录有没有 sbin:

`/usr/local/nginx/sbin/`

如果没有则在此目录下查看:

`/usr/sbin/`

1.1: 查看有没有安装 ssl 模块 (基于实际 sbin 目录输入命令)

`/usr/sbin/nginx -V`

出现下图则是已经安装过 ssl 模块

```
[root@zheng nginx-1.20.2]# /usr/sbin/nginx -V
nginx version: nginx/1.20.1
built by gcc 4.8.5 20150623 (Red Hat 4.8.5-44) (GCC)
built with OpenSSL 1.1.1g FIPS 21 Apr 2020 (running with OpenSSL 1.1.1k FIPS 25 Mar 2021)
TLS SNI support enabled
configure arguments: --prefix=/usr/share/nginx --sbin-path=/usr/sbin/nginx --modules-path=/usr/lib64/nginx/modules --conf-path=/etc/nginx/nginx.conf --error-log-path=/var/log/nginx/error.log --http-log-path=/var/log/nginx/access.log --http-client-body-temp-path=/var/lib/nginx/tmp/client_body --http-proxy-temp-path=/var/lib/nginx/tmp/proxy --http-fastcgi-temp-path=/var/lib/nginx/tmp/fastcgi --http-uwsgi-temp-path=/var/lib/nginx/tmp/uwsgi --http-scgi-temp-path=/var/lib/nginx/tmp/scgi --pid-path=/run/nginx.pid --lock-path=/run/lock/subsys/nginx --user=nginx --group=nginx --with-compat --with-debug --with-file-aio --with-google_perftools_module --with-http_addition_module --with-http_auth_request_module --with-http_dav_module --with-http_degradation_module --with-http_flv_module --with-http_gunzip_module --with-http_gzip_static_module --with-http_image_filter_module=dynamic --with-http_mp4_module --with-http_perl_module=dynamic --with-http_random_index_module --with-http_realip_module --with-http_secure_link_module --with-http_slice_module --with-http_ssl_module --with-http_stub_status_module --with-http_sub_module --with-http_v2_module --with-http_xslt_module=dynamic --with-mail=dynamic --with-mail_ssl_module --with-pcre --with-pcre-jit --with-stream=dynamic --with-stream_ssl_module --with-stream_ssl_preread_module --with-threads --with-cc-opt='-O2 -g -pipe -Wall -Wp,-D_FORTIFY_SOURCE=2 -fexceptions -fstack-protector-strong --param=ssp-buffer-size=4 -grecord-gcc-switches -specs=/usr/lib/rpm/redhat/redhat-hardened-cc1 -m64 -mtune=generic' --with-ld-opt='-Wl,-z,relro -specs=/usr/lib/rpm/redhat/redhat-hardened-ld -Wl,-E'
```

2.0: 如果没有安装则执行以下命令开始安装:

2.1: 去官网找到下载连接, 进入/usr/local 路径 使用指令

`wget` 下载链接

2.2: 下载并且解压:

`tar -xvf` 压缩包全名

2.3: 进入解压后的文件夹路径执行指令:

`./configure --prefix=/usr/local/nginx --with-http_stub_status_module --with-http_ssl_module --with-file-aio --with-http_realip_module`

之后执行：

```
make
```

报错的话执行下面的，不报错就算了：

```
yum -y install openssl openssl-devel
```

2.4：继续执行下面的指令。将 nginx 启动文件覆盖。第一条指令是备份
备份：

```
cp /usr/local/nginx/sbin/nginx /usr/local/nginx/sbin/nginx.bak
```

替换、覆盖：

```
cp objs/nginx /usr/local/nginx/sbin/nginx
```

2.5 重启 nginx:

```
nginx -s reload
```

修改 nginx.conf 使 ssl 生效

1.0 添加 ssl 证书作用：

使用私密链接更加安全并且不会弹出不安全提示



1.1 新建目录 etc/ssl

```
mkdir etc/ssl
```

1.2 编辑 nginx 的配置文件

```
vim etc/nginx/nginx.conf
```

```
{
```

```
1 http{
2     #http 节点中可以添加多个 server 节点
3     server{
4         #监听 443 端口
5         listen 443;
6
7         server_name 你的域名;
8         ssl on;
9         ssl_certificate /etc/ssl/1.crt.pem;
10        ssl_certificate_key /etc/ssl/1.key;
11    }
```

```

12     ssl_session_timeout 5m;
13     ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
14     ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:HIGH:!aNULL:!MD5:!RC4:!DHE;
15     ssl_prefer_server_ciphers on;
16     #这是我的主页访问地址，因为使用的是静态的 html 网页，所以直接使用 location 就
17可以完成了。
18     location / {
19         #文件夹
20         root /usr/local/service/ROOT;
21         #主页文件
22         index index.html;
23     }
24 }
25 server{
26     listen 80;
27     server_name 你的域名;
28     rewrite ^/(.*)$ https://你的域名:443/$1 permanent;
29 }

```

}

1.3 每次更改配置文件都需要重启 nginx 服务
 nginx -s reload

/usr/sbin/nginx -V