operation day 07

================================================

**git服务的使用方式：**

**1，使用ssh方式**

[root@web1 ~]# git init --bare /var/lib/git/test //如果web1没

有仓库就创建

[root@web2 ~]# ssh-keygen -f /root/.ssh/id\_rsa -N '' //在客户

端web2创建秘钥，且无密码

[root@web2 ~]# ssh-copy-id 192.168.2.100 //传入git服务端

[root@web2 ~]# git clone 192.168.2.100:/var/lib/git/test //克

隆仓库，已经无需密码

[root@web2 test]#echo abc > abc

[root@web2 test]#git add .

[root@web2 test]#git commit -m "abc"

[root@web2 test]#git push

**2, 使用git协议**

[root@web1 ~]# yum -y install git-daemon //装包

[root@web1 ~]# vim /usr/lib/systemd/system/git@.service

--enable=receive-pack //在第7行的末尾增加本句配置，可

以实现对仓库的写入

[root@web1 ~]# systemctl start git.socket //起服务

关闭防火墙和selinux

[root@web1 ~]# chmod -R 777 /var/lib/git

[root@web2 ~]#rm -rf test //到客户端，删除仓库

[root@web2 ~]# git clone git://192.168.2.100/test //用git协议的

方式重新克隆已有仓库

[root@web2 ~]#cd test //进入仓库

[root@web2 test]# echo xyz > abc //修改或者创建文件

[root@web2 test]# git add . //提交到暂存区

[root@web2 test]# git commit -m "abc++" //提交到仓库

[root@web2 test]# git push //提交到远程仓库

**3，使用http访问git，只读**

[root@web1 ~]# yum -y install httpd gitweb //装包

[root@web1 ~]# vim +10 /etc/gitweb.conf //进入配置文件并转到第

10行

our $projectroot = "/var/lib/git"; //去掉第10行的注释

[root@web1 ~]# killall nginx //如果有其他占用80端口的服务要关闭

[root@web1 ~]# systemctl restart httpd //开启httpd

使用火狐访问http://192.168.2.100/git/

-----------------------------------------------------------------

**rpm包 安装便利 更新速度慢**

**源码包 安装繁琐 更新速度快**

**如何将源码包制作为rpm包？**

[root@web1 ~]# yum -y install rpm-build //安装rpm包制作软件

[root@web1 ~]# rpmbuild -ba nginx.spec //制作rpm包，但是没有配

置文件会报错，报错也需要敲，会产生所需的目录

[root@web1 ~]# ls rpmbuild

[root@web1 ~]# cp lnmp\_soft/nginx-1.17.6.tar.gz rpmbuild/SOURCES/ //拷

贝源码包到指定目录

[root@web1 ~]# vim rpmbuild/SPECS/nginx.spec //编写转换nginx为rpm

的配置文件

Name:nginx //软件名称

Version:1.17.6 //版本号

Release:1 //制作的rpm包版本号

Summary:test //简单描述

#Group:

License:GPL //授权协议

URL:www.abc.com //网址

Source0:nginx-1.17.6.tar.gz //源码包文件名

#BuildRequires:

#Requires:

%description //详细描述

test test test

%post //可选执行指令

useradd abcabc

echo 123345 | passwd –stdin abcabc

%prep

%setup -q

%build

./configure //将%修改为./ 如果还需要选项模块等内容可以添加

make %{?\_smp\_mflags}

%install

make install DESTDIR=%{buildroot}

%files

%doc

/usr/local/nginx/\* //定义打rpm包的文件

%changelog

[root@web1 ~]# yum -y install gcc pcre-devel openssl-devel //安装nginx

所需的依赖包

[root@web1 ~]# rpmbuild -ba rpmbuild/SPECS/nginx.spec //制作nginx

的rpm包

[root@web1 ~]# ls rpmbuild/RPMS/x86\_64/nginx-1.17.6-1.x86\_64.rpm

// 查看诞生的nginx的rpm包

[root@web1 ~]#cd rpmbuild/RPMS/x86\_64/

rpm -ivh nginx-1.17.6-1.x86\_64.rpm //装包

rpm -qpi nginx-1.17.6-1.x86\_64.rpm //查包

---------------------------------------------------------

**vpn 虚拟专用网**

**1，使用gre技术实现vpn (两台linux)**

[root@web1 ~]# modprobe ip\_gre //在内核中开启gre功能模块

[root@web1 ~]# lsmod | grep ip\_gre //查询

ip tunnel add tun0 mode gre remote 192.168.2.200 local 192.168.2.100 //创建

vpn隧道，名字叫tun0，使用gre技术，与2.200连接，自己ip是2.100

[root@web1 ~]# ip tunnel del tun0 //如果配置错误，就删除

[root@web1 ~]# ip addr show //查看

[root@web1 ~]# ip addr add 10.10.10.10/8 peer 10.10.10.5/8 dev tun0 //在

tun0隧道中使用私有ip地址，本机是10.10 ，对面是10.5

[root@web1 ~]# ip link set tun0 up //激活之前的配置

**接下来web2主机也按照上述配置进行创建vpn，ip要反着写**

selinux、防火墙都要关闭

**2，使用pptp搭建vpn (支持Windows)**

[root@web1 ~]# cd ~/lnmp\_soft/vpn/

[root@web1 vpn]# yum -y install pptpd-1.4.0-2.el7.x86\_64.rpm

[root@web1 vpn]# vim +102 /etc/pptpd.conf //修改102、103行

localip 192.168.2.100 //本机ip,如果是实际环境要写公网ip

remoteip 10.10.10.10-18 //客户机分配的ip范围,自定义

[root@web1 vpn]# vim +66 /etc/ppp/options.pptpd //修改66行，添

加dns地址，此处的dns地址可以是公共网络中任意一个可用dns即可

ms-dns 8.8.8.8

[root@web1 vpn]# vim /etc/ppp/chap-secrets

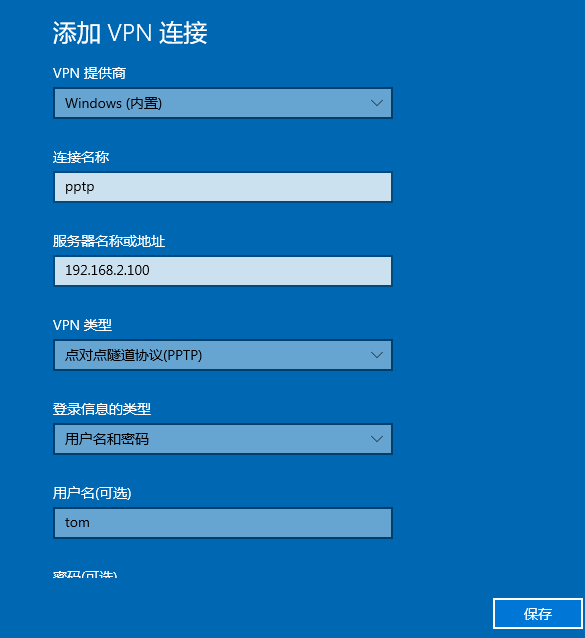
tom \* 123456 \* //添加账户tom，密码123456

[root@web1 vpn]# systemctl restart pptpd //开启服务

[root@web1 vpn]# ss -ntulp | grep pptpd //查询

之后使用windows系统连接服务器

用户名tom下面的密码是123456



**3，使用l2tp+IPsec配置vpn，相比pptp安全性更好**

[root@web1 vpn]# systemctl stop pptpd

[root@web1 vpn]# yum -y install libreswan

[root@web1 vpn]# cp myipsec.conf /etc/ipsec.d/ //拷贝配置

文件

[root@web1 vpn]# vim /etc/ipsec.d/myipsec.conf //修改第16

行为本机ip

left=192.168.2.100

[root@web1 vpn]# vim /etc/ipsec.secrets //修改安全相关配置

192.168.2.100 %any: PSK "randpass" //添加该行，2.100

是本机ip，%any是任何客户可以连接本机，PSK是预共享密钥

密码是randpass

yum -y install xl2tpd-1.3.8-2.el7.x86\_64.rpm //安装l2tp软件包

[root@web1 vpn]# vim /etc/xl2tpd/xl2tpd.conf //修改32、33行

ip range = 10.10.10.10-10.10.10.18 //分配给客户的ip

local ip = 192.168.2.100 //vpn服务器本身的ip

[root@web1 vpn]# vim /etc/ppp/options.xl2tpd //修改配置，

把第10、16行添加注释，21行删除注释与首部的空格

systemctl restart ipsec //启动服务

systemctl restart xl2tpd //启动服务

ss -ntulp | grep xl2tpd //查xl2tp服务

ss -ntulp | grep :500 //查ipsec服务



-------------------------------------------------------

**配置Unit文件实现自定义systemctl控制的服务**

**在linux系统中，有些程序无法利用systemctl控制，就需要添加Unit文件**

**unit文件可以通过其他类似的服务拷贝作为模板使用**

[root@web1 vpn]# cd /usr/lib/systemd/system //该目录存储了很多服务的Unit文件

[root@web1 system]# cp httpd.service nginx.service //拷贝其他服

务的配置文件当做nginx服务的模板

[root@web1 system]# vim nginx.service //修改

[Unit]

Description=nginx //描述

After=network.target remote-fs.target nss-lookup.target //在网络服务、远程文件系统服务、域名相关服务启动之后再开启nginx

[Service]

Type=forking //由于nginx是多进程服务，要设置forking

ExecStart=/usr/local/nginx/sbin/nginx //当执行systemctl start nginx时，所执行的命令

ExecReload=/usr/local/nginx/sbin/nginx -s reload //当执行systemctl reload nginx时，所执行的命令

ExecStop=/bin/kill -s QUIT ${MAINPID} //当执行systemctl stop nginx时，所执行的命令,-s QUIT是发送退出信号，${MAINPID}是nginx的主进程号

[Install]

WantedBy=multi-user.target //支持开机自启

[root@web1 system]#systemctl stop httpd //关闭目前占用80端口的服务

[root@web1 system]#systemctl restart nginx //尝试开启nginx

服务，本机需要提前安装nginx

[root@web1 system]#systemctl daemon-reload //如果开启失败，可以刷新

[root@web1 system]# systemctl status nginx //查询nginx状态