1.配置yum仓库

```
#centos7配置yum仓库
mkdir /dvd
mkdir -p /opt/centos
mount /dev/cdrom /dvd
cp -arf /dvd/* /opt/centos
cat >/etc/yum.repos.d/<< loca.repo EOF
[centos]
name=centos
baseurl=file:///opt/centos/
enabled=1
gpgcheck=0
EOF
yum clean all; yum makecache
```

2.配置ftp服务

FTP服务的介绍：

FTP文件传输协议 基于客户端/服务器模式（c/s），默认使用20、21号端口，其中端口20（数据端口）用于进行数据传输，端口21（命令端口）用于接受客户端发出的相关FTP命令与参数。 用于文件的上传和下载

模式有哪些：

主动模式：简称“psvr” FTP服务器主动向客户端发起连接请求

工作原理：

被动模式：FTP服务器等待客户端发起连接请求（FTP的默认工作模式）。

工作原理：

配置文件的介绍：

chroot_local_user #是否将所有用户限制在主目录、

chroot_list_enable #是否启动限制用户的名单 YES为启用 NO禁用(包括注释掉也为禁用)；

chroot_list_file=/etc/vsftpd/chroot_list #是否限制在主目录下的用户名单，至于是限制名单还是排除名单，这取决于chroot_local_user的值，

我们可以这样记忆： chroot_local_user总是一个全局性的设定，其为YES时，全部用户被锁定于主目录，

其为NO时，全部用户不被锁定于主目录。

那么我们势必需要在全局设定下能做出一些“微调”，即，我们总是需要一种“例外机制"，所以当chroot_list_enable=YES时，表示我们“需要例外”。

而”例外“的含义总是有一个上下文的，即，当”全部用户被锁定于主目录“时（即chroot_local_user=YES），

"例外"就是：不被锁定的用户是哪些；

当"全部用户不被锁定于主目录"时（即chroot_local_user=NO），

"例外""就是：要被锁定的用户是哪些。这样解释和记忆两者之间的关系就很清晰了！

三种认证模式：

本地用户模式

安装vsftpd服务

```
 yum install -y vsftpd ftp
```

创建本地用户并设置密码：

```
useradd tom
echo 123 | passwd --stdin tom
```

配置/etc/vsftpd/vsftpd.conf

```
anonymous_enable=no
local_enable=yes
write_enable=yes
lcoal_umask=yes
userlist_enable=yes
```

重新启动服务 systemctl restart vsftpd ; systemctl enable vsftpd

注意：在采用本地用户模式登录FTP服务器后，默认访问的是该用户的家目录，也就是说，访问的是/home/linuxprobe目录。而且该目录的默认所有者、所属组都是该用户自己，因此不存在写入权限不足的情况。

设置selinux的上下文

```
getsebool -a | grep ftp
ftpd_full_access --> off
setsebool -P ftpd_full_access=on
```

测试服务器：

ftp://192.168.100.10

匿名用户模式

```
yum install -y vsftpd ftp
```

配置/etc/vsftpd/vsftpd.conf 文件

```
anonymous_enable=YES

anon_umask=022

anon_upload_enable=yes

anon_mkdir_write_enable=yes

anon_other_write_enable=yes
```

重新启动服务 systemctl restart vsftpd ;systemctl enable vsftpd

修改/etc/var/ftp/pub的所有者：

```
chown -Rf ftp:ftp /etc/ftp/pub
```

设置selinux的上下文

```
#getsebool -a | grep ftp
ftpd_full_access --> off //找到这一项
setsebool -P ftpd_full_access=on
```

FTP客户端测试：

```
ftp://192.168.100.10
```

虚拟用户模式：

	Linux环境	**Windows环境**
FTP服务器	vsftpd	IIS
proftpd	Serv-U	
wu-ftpd		
FTP客户端	ftp/ncftp/lftp命令行工具	ftp命令行工具
gftp	CuteFTPpro	
浏览器firefox	浏览器IE	

ftp的账户类型：

本地用户

用户在FTP服务器上拥有账号，且该账号为为本地用户的账号

可以通过输入自己的账号和口令进行授权登录

登录目录为自己的home目录($HOME)

虚拟用户

用户在FTP服务器上拥有账号，但该账号只能用于文件传输服务

登录目录为某一指定的目录

通常可以上传和下载

匿名用户

用户在FTP服务器上没有账号

登录目录为/var/ftp

配置简单的ftp服务器

```
yum install -y vsftpd
echo "anon_root=/opt" >> /etc/vsftpd.vsftpd.conf
systemctl start vsftpd &&systemctl enable vsftpd
netstat -ntpl    | grep 21
```

3.逻辑卷进行创建

```
lsblk
fdisk /dev/sdb
添加3个5G空间 划分出来
最后w保存
pvcreate /dev/sdb[1-2]
vgcreate myvg /dev/sdb[1-2]
lvcreate -L +5G -n mylv myvg
mkfs.ex4 /dev/mapper/myvg-mylv
mkdir /lvm
mount /dev/mapper/myvg-mylv /lvm
df -TH
扩展逻辑卷:
lvextend -L +1G /dev/mapper/myvg-mylv
resize2fs /dev/mapper/myvg-mylv     不刷新文件系统永久是5G     刷新文件系统才会先大小
```

4.配置nfs服务

```
1.配置yum文件    yum clean all; yum makecache
```

```
2.修改主机名  hostnamectl set-hostname nfs-server        查看名称是不是生效
hostnamectl
    设置客户端名称   hostnamectl set-hostname nfs-client        查看   hostnamectl
3. 安装服务
   nfs-server节点yum install -y nfs-utils rpcbind
   nfs-client节点yum install -y nfs-utils rpcbind
4.创建目录编写配置文件  mkdir /mnt/test
    vi /etc/exports
    /mnt/test   192.168.200.0/24
(rw,no_root_squash,no_all_squash,sync,anonuid=501)
    exportfs -r
5.重新启动服务
    systemctl restart nfs rpcbind
6.查看挂载情况
    echo " 192.168.200.128 nfs-server" >>/etc/hosts
    showmount -e
7.关闭防火墙
    setenforce 0 && systemctl stop firewalld
8.客户端进行挂载
    mount -t nfs 192.168.200.10:/mnt/test /mnt
    df -h
9.验证
    先在nfs-client 的mnt目录下创建文件abc.txt    touch abc.txt
    然后在nfs-server 下的/mnt/test/是不是有abc.txt     cd /mnt/test/
```

解释：
/mnt/vcdrom/ 192.168.94.5(rw,async,no_root_squash,no_subtree_check)　　192.168.94.5可以为*
　　注：配置文件说明：/mnt/vcdrom/为共享的目录，使用绝路径。对192.168.94.5(rw,no_root_squash,no_all_squash,sync) 为客户端的地址及权限，地址可以是一个网段，一个IP地址或者是一个域名，域名支持通配符，如：*youxia.com。
　　权限说明：rw：read-write，可读写；ro：read-only，只读；sync：文件同时写入硬盘和内存；async：文件暂存于内存，而不是直接写入内存；no_root_squash：NFS客户端连接服务端时如果使用的是root的话，那么对服务端分享的目录来说，也拥有root权限。显然开启这项是不安全的。root_squash：NFS客户端连接服务端时如果使用的是root的话，那么对服务端分享的目录来说，拥有匿名用户权限，通常他将使用nobody或nfsnobody身份；all_squash：不论NFS客户端连接服务端时使用什么用户，对服务端分享的目录来说都是拥有匿名用户权限；anonuid：匿名用户的UID值，通常是nobody或nfsnobody，可以在此处自行设定；anongid：匿名用户的GID值。第六，参考yum源配置方法二，将/opt/tools/下的ios文件挂载到/mnt/vcdrom，需要设置开机自动挂载。

出现的错误
nfs配置没有问题就服务启动不起来writing fd to kernel failed: errno 111
重启服务　systemctl restart rpcbind ;systemctl restart nfs

```
#server
hostnamectl set-hostname nfs-server
echo "192.168.200.5 nfs-server" >> /etc/hosts
yum install -y nfs-utils rpcbind
mkdir /mnt/test -p
cat >/etc/exports<<EOF
/mnt/test
192.168.200.0/24(rw,no_root_squash,no_all_squash,sync,anonuid=501,anongid=501)
EOF
systemctl restart nfs
systemctl restart  rpcbind
```

```
showmount -e nfs-server
systemctl stop firewalld
systemctl disable firewalld
sed -i 's/SELINUX=.*/SELINUX=disbaled/g' /etc/selinux/config
setenforce 0
mkdir  /nfs-test
mount -t nfs nfs-server:/mnt/test  /nfs-test
df -Th
#client
systemctl stop firewalld
systemctl disable firewalld
sed -i 's/SELINUX=.*/SELINUX=disbaled/g' /etc/selinux/config
setenforce 0
echo "192.168.200.5 nfs-server" >> /etc/hosts
mount -t nfs nfs-server:/mnt/test /nfs-test
```

```
yum install nfs-utils -y

mkdir /data

vim /etc/exports

/data 10.0.0.0/24(rw,async,no_root_squash,no_all_squash)

systemctl restart rpcbind

systemctl restart nfs
```

5.samba文件系统

Samba介绍：
　Samba的作用：
它能够使windows用户通过“网上邻居”，等熟悉的方式直接访问Linux上的资源，也能使linux利用SMB客户端程序访问Windows的共享资源。
SMB(Server Message Block,服务信息块)，看作是局域网上的共享文件夹/打印机的一种协议.

```
Samba主要功能：
```

1、提供windows风格的文件和打印机共享。
2、在Windows网络中解析NetBios的名字
3、提供SMB客户端，linux用户可以利用smbclient利用类似于ftp的形式访问windows资源.
4,、提供命令行工具，利用该工具可以有限制地支持windows的某些管理功能.
Samba服务器配置基础

Samba服务器搭建
测试环境centos7.3或者centos7.2

服务端：

```
samba配置文件：
    /etc/samba/smb.conf
介绍：
    [global]    定义全局配置
    [homes]    定义对用户家目录的共享配置
    [printers]    定义打印共享属性
```

配置工作组：
workgroup = test
server string =This is testgroup!
配置安全信息：
user 默认
share 创建匿名共享时
domain
ads
server

```
samba服务可以作为独立服务器使用。也可以加入域    独立服务器需要使用用户名和密码      在域中使用域的用户名密码
```

两中安全模式：
user-level 默认安全模式 使用用户名和密码验证
share-level 只需要密码访问不推荐使用

客户端：
Windows访问：
\192.168.100.128\test
Linux图形界面访问
smb://192.168.100.128/test -U test
Linux使用smbclient连接：
smbclient://192.168.100.128/test -U test
使用挂载命令：
mount -t cifs //192.168.100.128/test /mnt -o username=test,password=test

案例：用户家目录共享
samba 服务配置默认共享家目录 每个用户家里目录均可以使用samba共享
[homes]
comment = Home Directories
browseable = no
writable = yes
为要通过smb协议访问家目录的用户创建samba密码 创建的用户必须是系统已经存在的用户

smpasswd -a test
启动服务:
systemctl start smb
通过smbclient登录验证用户共享:
smbclient //192.168.100.10/test -U test

创建独立共享
/etc/samba/smb.conf
[test-sub]
comment = this is tes!
path=/test
valid users = test
public = no
browsable = yes
writable = yes
printable = no
create mask = 0765

创建匿名共享:
[global]
security = share
[test]
comment =this is test!
path =/test
read only =yes
guest only =yes
注意share的一些迷人级别可能与windows客户端不兼容

检查配置是否正确
testparm
使用smbclient查看一个服务器都有哪些共享
smbclient -L 192.168.100.10
smbclient -L 192.168.100.10 -U test

安装服务
yum install -y samba
配置 Samba 的配置文件/etc/samba/smb.conf。
① 修改[global]中的内容如下（找到配置文件中的字段并修改，disable spoolss = yes 是新增的）：
disable spoolss = yes

在配置文件的最后，添加如下内容:
[share]
path = /opt/share
browseable = yes
public = yes
writable = yes

启动服务:
systemctl start smb&& systemctl enable smb

创建共享的目录并且设置权限
mkdir -p /opt/share
chmod 777 /opt/share

创建smb的用户
这里的用户名得是系统存在的用户
smbpasswd -a root 然后设置密码 可以和root密码不相同

重启服务
systemctl restart smb

最后windows访问共享

\ip地址

如果win10不能访问就直接
https://jingyan.baidu.com/article/48a420579dbcf0e825250411.html
别忘记最后要重启电脑

```
hostnamectl set-hostname samba-server
echo "192.168.200.6 samba-server" >> /etc/hosts
yum install -y samba
cat >/etc/samba/smb.conf<<EOF
[share]
     path = /opt/share
     browseable = yes
     public = yes
     writable = yes
EOF
mkdir /opt/share -p
chmod 777 /opt/share
systemctl start smb
systemctl enable smb
systemctl status smb
smbpasswd -a root
输入密码可以不和root相同
最后win+r进行修改
\\192.168.200.5
输入用户名和密码
```

6.搭建分布式lnmp架构

网络规划:

mysql1: 192.168.200.5

mysql2: 192.168.200.6

php: 192.168.200.7

nginx: 192.168.200.8

主从复制服务器

mysql1

```
systemctl stop firewalld
systemctl disable firewalld
sed -i 's/SELINUX=.*/SELINUX=disbaled/g' /etc/selinux/config
setenforce 0
hostnamectl set-hostname mysql1
echo "192.168.200.5 mysql1" >> /etc/hosts
echo "192.168.200.6 mysql2" >> /etc/hosts
echo "192.168.200.7 nginx" >> /etc/hosts
echo "192.168.200.8 php" >> /etc/hosts
cat >/etc/yum.repos.d/local.repo<<EOF
[centos]
name=centos
baseurl=http://192.168.200.3/centos/
enabled=1
gpgcheck=0
EOF
yum clean all; yum makecache
```

mysql2

```
systemctl stop firewalld
systemctl disable firewalld
sed -i 's/SELINUX=.*/SELINUX=disbaled/g' /etc/selinux/config
setenforce 0
hostnamectl set-hostname mysql2
echo "192.168.200.5 mysql1" >> /etc/hosts
echo "192.168.200.6 mysql2" >> /etc/hosts
echo "192.168.200.7 nginx" >> /etc/hosts
echo "192.168.200.8 php" >> /etc/hosts
cat >/etc/yum.repos.d/local.repo<<EOF
[centos]
name=centos
baseurl=http://192.168.200.3/centos/
enabled=1
gpgcheck=0
EOF
yum clean all; yum makecache
```

nginx

```
systemctl stop firewalld
systemctl disable firewalld
sed -i 's/SELINUX=.*/SELINUX=disbaled/g' /etc/selinux/config
setenforce 0
hostnamectl set-hostname nginx
```

```
echo "192.168.200.5 mysql1" >> /etc/hosts
echo "192.168.200.6 mysql2" >> /etc/hosts
echo "192.168.200.7 nginx" >> /etc/hosts
echo "192.168.200.8 php" >> /etc/hosts
cat >/etc/yum.repos.d/local.repo<<EOF
[centos]
name=centos
baseurl=http://192.168.200.3/centos/
enabled=1
gpgcheck=0
EOF
yum clean all; yum makecache
```

php

```
systemctl stop firewalld
systemctl disable firewalld
sed -i 's/SELINUX=.*/SELINUX=disbaled/g' /etc/selinux/config
setenforce 0
hostnamectl set-hostname mysql2
echo "192.168.200.5 mysql1" >> /etc/hosts
echo "192.168.200.6 mysql2" >> /etc/hosts
echo "192.168.200.7 nginx" >> /etc/hosts
echo "192.168.200.8 php" >> /etc/hosts
cat >/etc/yum.repos.d/local.repo<<EOF
[centos]
name=centos
baseurl=http://192.168.200.3/centos/
enabled=1
gpgcheck=0
EOF
yum clean all; yum makecache
```

7.Yum安装Tomcat

前期准备：
关闭Selinux

```
setenforce 0
```

关闭防火墙

关闭防火墙

```
systemctl stop firewalld.service
systemctl stop iptables
```

操作步骤：

```
yum -y install wget
wget https://mirrors.tuna.tsinghua.edu.cn/apache/tomcat/tomcat-
9/v9.0.30/bin/apache-tomcat-9.0.30.tar.gz
```

这个地址如果不能用，可以从这里再找最新的下载地址：https://tomcat.apache.org/

这个地址如果不能用，可以从这里再找最新的下载地址：https://tomcat.apache.org/

```
tar -zxvf apache-tomcat-9.0.30.tar.gz
复制tomcat文件到/opt目录并重命名
  mv apache-tomcat-9.0.30 /opt
  cd /opt
  mv apache-tomcat-9.0.30  tomcat9
为启动的脚本文件添加环境变量
cd tomcat9/bin
vi startup.sh
  JAVA_HOME=/usr/java/jdk1.8.0_211
  JRE_HOME=/usr/java/jdk1.8.0_211/jre
  PATH=$JAVA_HOME/bin:$JRE_HOME:$PATH
  CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar:$CLASSPATH
  TOMCAT_HOME=/opt/tomcat9
启动tomcat
./startup.sh
验证tomcat是否能够启动起来
```

8.安装java环境

```
yum list java*
检索1.8的列表
yum list java-1.8*
安装1.8.0的所有文件
yum install java-1.8.0-openjdk* -y
使用命令检查是否安装成功
java -version
我们可以在终端中输入java来检测是否配置成功
java
如果配置成功便会显示提示信息
```

9.Python3.7.5 安装

1.安装依赖环境

```
yum -y install zlib-devel bzip2-devel openssl-devel ncurses-devel sqlite-devel
readline-devel tk-devel gdbm-devel db4-devel libpcap-devel xz-devel  gcc  wget
gcc-c++ libffi-devel
```

2.下载python3

```
wget https://www.python.org/ftp/python/3.7.5/Python-3.7.5.tgz
```

3.安装python3

3.1创建目录

```
mkdir -p /usr/local/Python3
```

3.2解压下载好的Python-3.7.5.tgz包

```
tar -xf Python-3.7.5.tgz
```

3.3编译安装
进入解压后的目录，编译安装
如果编译安装过程有报错提示，可参考文档：http://www.cnblogs.com/shwee/p/9013851.html

```
cd ~
cd Python-3.7.5
./configure --prefix=/usr/local/Python3
```

4.然后：make

```
make && make install
```

5.建立软链接

```
ln -s /usr/local/Python3/bin/python3 /usr/bin/python3
```

6.设置环境变量

```
echo "export PATH=$PATH:$HOME/bin:/usr/local/Python3/bin" >> /etc/profile
source /etc/profile
```

7.验证Python3

```
[root@localhost Python-3.7.5]# python3 -V
Python 3.7.5
[root@localhost Python-3.7.5]# pip3 -V
pip 19.2.3 from /usr/local/Python3/lib/python3.7/site-packages/pip (python 3.7)
```

10.iftop的使用网络监听

```
1.配置yum文件    epel源
2.yum install -y iftop
3.使用命令
 iftop -i eno16777728 -n  -P
```

11.安装gcc和gcc-c++

```
安装命令：yum -y install gcc gcc-c++
安装成功会提示Complete!
```

12.安装 python 2.7.15 和 pip

```
1、先安装 GCC 包，如果没安装 GCC包 就输入以下命令行安装；
yum install gcc openssl-devel bzip2-devel
2、用 wget 下载 python 2.7 并解压
yum -y install wget
cd /usr/src
wget https://www.python.org/ftp/python/2.7.15/Python-2.7.15.tgz
再解压 python2.7
tar -zxvf Python-2.7.15.tgz
3. 安装 python 2.7
   进入上面解压的 Python-2.7.15 解压文件中使用下面命令行安装
   cd Python-2.7.15
   ./configure --enable-optimizations
   make altinstall
4. 查看安装版本
   python -V
       可以看到输出 Python 2.7.15 就安装完成。

5、安装 PIP
curl "https://bootstrap.pypa.io/get-pip.py" -o "get-pip.py"
python2.7 get-pip.py
```

13.安装vncserver

https://blog.csdn.net/niaooer/article/details/87907132

安装vnc-server

```
yum install tigervnc-server
```

修改配置文件

```
cp /lib/systemd/system/vncserver@.service
/etc/systemd/system/vncserver@:1.service
```

```
修改内容，如下将这两行修改成：
ExecStart=/usr/sbin/runuser -l root -c "/usr/bin/vncserver %i"
PIDFile=/root/.vnc/%H%i.pid

备注：这里最好不要用root启动，所以可以将<USER>换成你要开通vnc账户的用户名
```

```
如果需要修改默认启动分辨率，在此行添加：

ExecStart=/usr/sbin/runuser -l root -c "/usr/bin/vncserver %i -geometry
1920x1080"

文件名： vncserver@_8.service
==========================
[Unit]
Description=Remote desktop service (VNC)
After=syslog.target network.target

[Service]
Type=simple

# Clean any existing files in /tmp/.X11-unix environment
ExecStartPre=/bin/sh -c '/usr/bin/vncserver -kill %i > /dev/null 2>&1 || :'
ExecStart=/usr/bin/vncserver_wrapper root %i
ExecStop=/bin/sh -c '/usr/bin/vncserver -kill %i > /dev/null 2>&1 || :'
PIDFile=/root/.vnc/%H%i.pid
[Install]
WantedBy=multi-user.target
====================================
```

2,保存重新加载配置文件

```
systemctl daemon-reload
```

四，关闭防火墙，或者配置防火墙规则

1.关闭防火墙

```
systemctl stop firewalld.service
```

五，切换需要配置vnc 的账户 su user

1.启动vnc

```
systemctl start vncserver@:1.service
这里需要输入，root账户的密码
启动后会提示配置vnc登录密码
```

2.查看状态:

```
systemctl status vncserver@:1.service
```

14.时间服务器

一 .安装chrony软件包

```
        1.yum -y install chrony
```

二.修改配置文件

```
1.vim /etc/chrony.conf

  在配置文件增加允许访问该ntp服务器的网段,比如允许192.168.4.0/24,格式如下：

    添加：

      allow 192.168.4.0/24

   假如需要禁止192.168.2.0/24这个网段访问该NTP服务器，则添加：

      deny 192.168.2.0/24

   设置NTP服务器的层数量

    local stratum 10       //将这个注释去掉，后面的数字由自己定义
```

三.重启服务器

```
systemctl restart chronyd

 systemctl enable chronyd
```

客户端配置：

一. 安装chrony软件包

```
yum -y install chrony
```

二.修改配置文件

```
vim /etc/chrony.conf

 server 192.168.4.5 iburst   //设置与192.168.4.5这台服务器同步时间，其中iburst 表示重启服务器以后尽快同步时间。
```

三.重启服务

```
systemctl restart chronyd
```

四 查看时间

```
时间查看  date
查看系统时钟与时区
使用timedatectl list-timezones列出可用的时区
设定系统时钟与时区
timedatectl set-timezone timezone
timedatectl set-time hour:min:sec
设定是否启用网络时间同步
timedatectl set-ntp true|false
```

15.ssh服务器

SSH介绍：

ssh是linux下远程管理工具，全称就是安全的shell 采用客户端和服务器的架构

ssh的版本有v1版本和v2版本

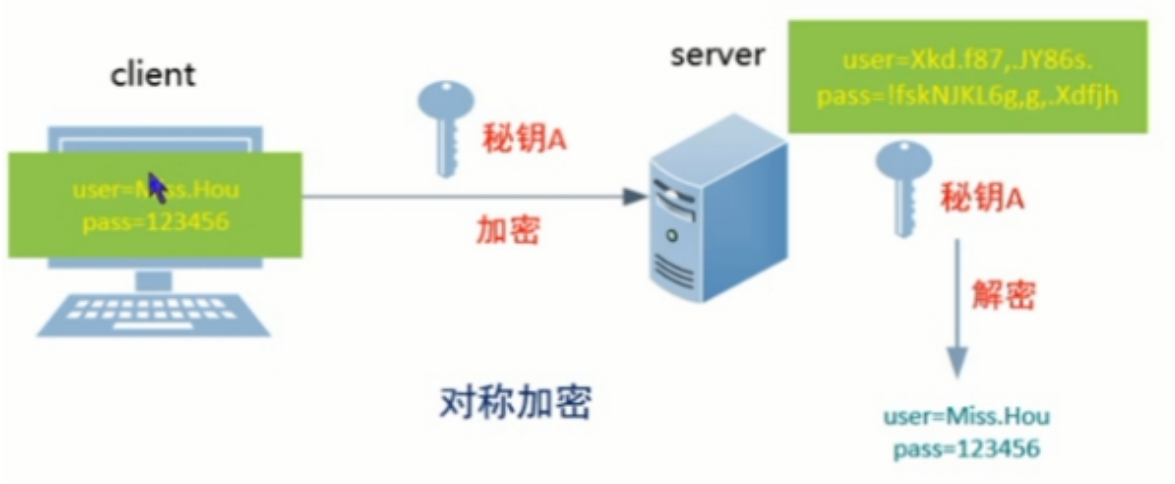
ssh免密登录的原理：

主控端生成一对密钥，将公钥传递到远程主机上,当主控端想要连接远程主机时，远程主机会随机发送一串字符给主控端，主控端将这串字符用私钥加密，返回给远程主机，远程主机使用公钥将加密的字符解密，如果和自己生成的字符一致，则验证通过，可以进行登录。

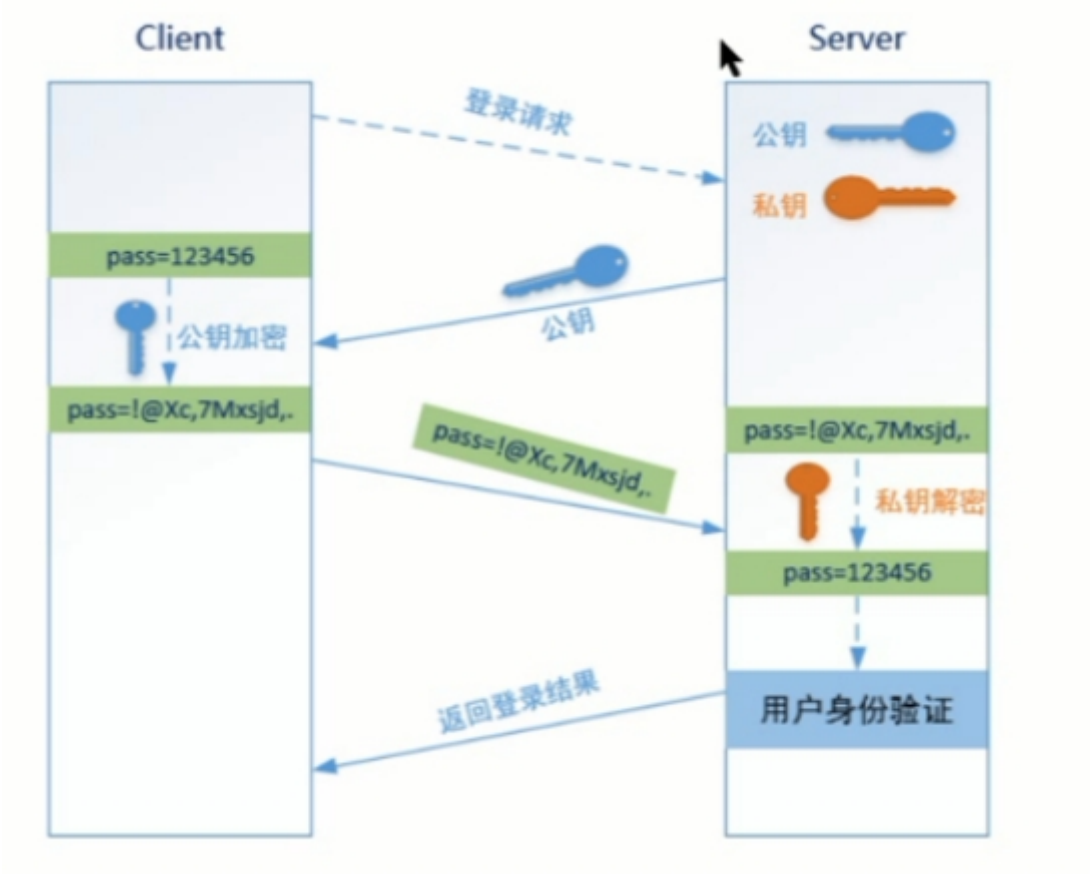
ssh的加密算法：

dsa对称加密安全低 数据传输速度快使用同一个密钥加密和解密

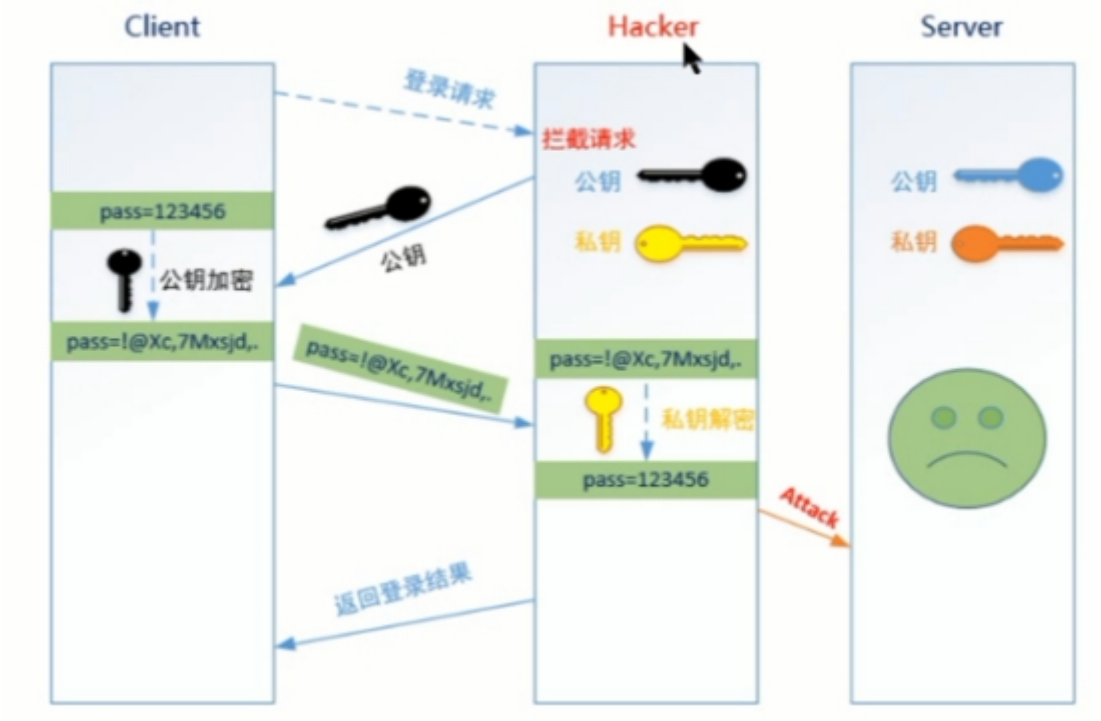
rsa非对称加密算法 安全 数据传输慢 ssh默认的加密算法



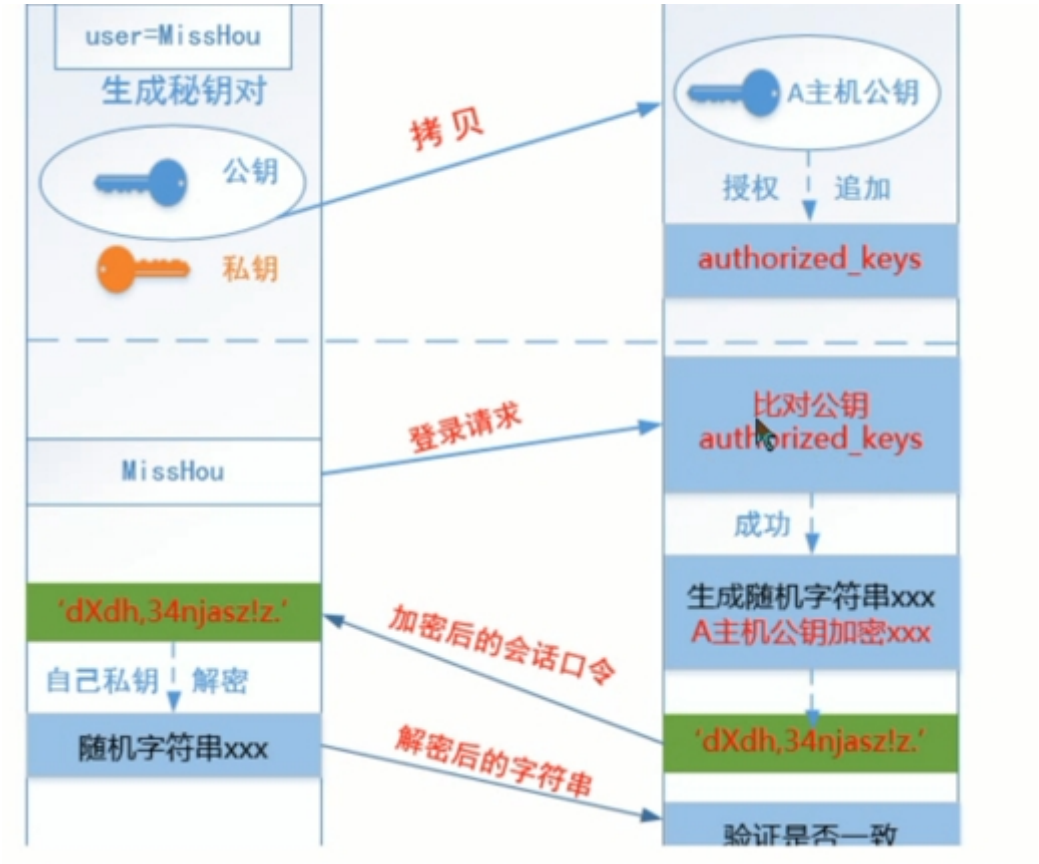
工作过程：



中间人劫持的过程：



非对称加密的过程：



认证原理:

有2种认证方式:

基于账号和口令的验证方式 和 基于公钥和私钥的验证方式

ssh的登录过程分为5个阶段

1、版本号协商阶段

2、密钥和算法协商阶段

3、认证阶段

4、会话请求阶段

5、会话交互阶段

1、版本号协商阶段

服务端打开端口22，等待客户连接。

客户端向服务端发起TCP连接，连接建立后，服务端向客户端发送第一个报文，包括版本标志字符串，格式为“协议版本号 次协议版本号 软件版本号”。

客户端收到报文后，解析协议版本号，如果服务端的协议版本号比自己的低，且客户端能支持服务端的低版本，就使用服务端的协议号，否则使用自己的协议版本号。

客户端回复服务端一个报文，包含了客户端决定使用的协议版本号。

服务端比较客户端发过来的版本号，决定是否能同客户端交互。

如果协商成功，就进入密钥和算法协商阶段。否则服务端断开TCP连接。

2、密钥和算法协商阶段

服务端和客户端分别发送算法协商报文给对方，报文中包含自己支持的公钥算法列表、加密算法列表、消息验证码算法列表、压缩算法列表等。

服务端和客户端根据对方和自己支持的算法得出最终使用的算法。

服务端和客户端利用DH交换算法、主机密钥对等参数，生成会话密钥和会话ID。

c公 客户端公钥

c密 客户端密钥

s公 服务端公钥

s密 服务端密钥

在版本号协商阶段完成后：

服务端将 s公 发送给客户端。

服务端生成会话ID，设为 id，发送给客户端。

客户端生成会话密钥，设为 key，并计算 res = id 异或 key。

客户端将 res 用 s公 进行加密，将结果发送给服务端。

服务端用 s密 进行解密，得到 res。

服务器计算 res 异或 id，得到 key。

至此服务端和客户端都知道了会话密钥和会话ID，以后的数据传输都使用会话密钥进行加密和解密。

3、认证阶段

基于账号和口令的验证方式：

客户端使用密钥和算法协商阶段生成的会话密钥加密账号、认证方法、口令，将结果发送给服务器。

服务端使用获得的会话密钥解密报文，得到账号和口令。

服务端对这个账号和口令进行判断，如果失败，向客户端发送认证失败报文，其中包含了可以再次认证的方法列表。

客户端从认证方法列表中选择一种方法进行再次认证。

这个过程反复进行，直到认证成功或者认证次数达到上限，服务端关闭本次TCP连接。

基于公钥和私钥的验证方式：

使用ssh-keygen程序生成公钥 id_dsa.pub 和私钥 id_dsa，一般是在客户端上生成，然后把 id_dsa.pub 通过某种方式发送给服务端。

服务端放在将要远程登录过来的那个账号的目录的.ssh目录下面。

客户端使用密钥和算法协商阶段生成的会话密钥加密账号、认证方法、id_dsa.pub，将结果发送给服务端。

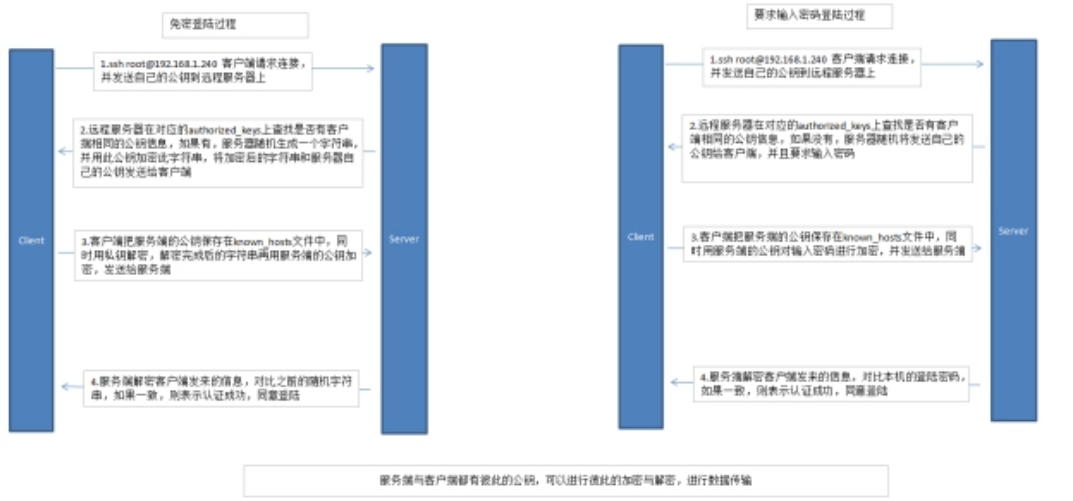
服务端使用会话密钥解密报文，得到账号、id_dsa.pub。 服务端在这个账号的目录的.ssh目录下找对应的公钥，如果没有找到，发送失败消息给客户端，如果找到，比较客户发送过来的这个公钥和找到的公钥，如果内容相同，服务端生成一个随机的字符串，简称“质询”，然后使用找到的公钥加密这个质询，然后使用会话密钥再次加密。

服务端把这个双重加密的数据发送给客户端。

客户端使用会话密钥解密报文，然后使用id_dsa再次解密数据，得到质询。

客户端使用会话密钥加密质询，发送给服务端。

服务端使用会话密钥解密报文，得到质询，判断是不是自己生成的那个质询，如果不相同，发送失败消息给客户端，如果相同，认证通过。




```
ssh-keygen
cd .ssh
cat id_rsa.pub >>authorized_keys
chmod 600 authorized_keys
cd /root
chmod 700 ~/root/.ssh


[root@localhost ~]# cat /etc/ssh/sshd_config  | grep -v "^#" | grep -v "^$"
```

需要修改的内容

```
PermitRootLogin yes
PubkeyAuthentication yes
AuthorizedKeysFile  .ssh/authorized_keys
PasswordAuthentication no
RSAAuthentication yes



systemclt restart ssd
```

在生产环境中，为了提高安全性，经常需要对server的访问进行限制，比如一些重要的server只允许某些特定的主机或者特定IP网段才能访问。

对Linux 进行IP 访问限制可以通过修改 /etc/hosts.allow 和 /etc/hosts.deny, hosts.allow 优先级高于 hosts.deny. 具体配置如下：

1. 允许某个IP 或 IP 段 能 SSH 到 Linux Server （IP 是 客户端的ip，配置是在Linux Server上修改），在 /etc/hosts.allow 中添加以下信息：

```
sshd:10.74.61.3   #允许特定的一个IP ssh 登录
sshd:10.74.133.0/255.255.255.0  #允许10.74.133.0 网段的地址 ssh 登录
```

2. 拒绝所有的 ssh 登录，在 /etc/hosts.deny中添加一下信息：

```
sshd:ALL   #拒绝所有的ssh 登录
```

因为 /etc/hosts.allow 优先级高于 /etc/hosts.deny， 所以 1 和 2 的配置 结合起来就是该台 server 只能从 10.74.61.3 以及 10.74.133.x 的 地址 ssh 登录。

16.搭建Mariadb

```
[root@localhost ~]# yum install mariadb mariadb-server
Complete!
[root@localhost ~]# systemctl start mariadb
[root@localhost ~]# systemctl enable  mariadb
Created symlink from /etc/systemd/system/multi-user.target.wants/mariadb.service
to /usr/lib/systemd/system/mariadb.service.
[root@localhost ~]# mysql_secure_installation
NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!
In order to log into MariaDB to secure it, we'll need the current
password for the root user.  If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.
Enter current password for root (enter for none):
OK, successfully used password, moving on...
Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.
Set root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
 ... Success!
By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them.  This is intended only for testing, and to make the installation
```

```
go a bit smoother.  You should remove them before moving into a
production environment.
Remove anonymous users? [Y/n] y
 ... Success!
Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.
Disallow root login remotely? [Y/n] n
 ... skipping.
By default, MariaDB comes with a database named 'test' that anyone can
access.  This is also intended only for testing, and should be removed
before moving into a production environment.
Remove test database and access to it? [Y/n] y
 - Dropping test database...
 ... Success!
 - Removing privileges on test database...
 ... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
 ... Success!

Cleaning up...

All done!  If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
[root@localhost ~]#
[root@localhost ~]# mysql -uroot -proot
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 9
Server version: 5.5.68-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
+--------------------+
3 rows in set (0.00 sec)

MariaDB [(none)]> select version();
+----------------+
| version()      |
+----------------+
| 5.5.68-MariaDB |
+----------------+
1 row in set (0.00 sec)

MariaDB [(none)]> Ctrl-C -- exit!
```

```
Aborted
[root@localhost ~]#
```

17.配置DHCP

一、创建DHCP作用域

1、安装DHCP服务并且给本机设置一个在后边dhcp中同一个网段的IP地址

```
[root@localhost Packages]#
[root@localhost Packages]# yum -y install dhcp-4.1.1-38.P1.el6.centos.x86_64.rpm
 _
```

2、配置文件在/etc/dhcp，配置文件是dhcpd.conf

```
[root@localhost dhcp]#
[root@localhost dhcp]#
[root@localhost dhcp]# pwd
/etc/dhcp
[root@localhost dhcp]# ll
total 12
drwxr-xr-x. 2 root root 4096 Nov 22  2013 dhclient.d
-rw-r--r--. 1 root root  193 Nov 22  2013 dhcpd6.conf
-rw-r--r--. 1 root root  112 Nov 22  2013 dhcpd.conf
[root@localhost dhcp]# _
```

3、打开显示配置文件

```
[root@localhost dhcp]# cat dhcpd.conf
#
# DHCP Server Configuration file.
#   see /usr/share/doc/dhcp*/dhcpd.conf.sample
#   see 'man 5 dhcpd.conf'
#
[root@localhost dhcp]# _
```

4、拷贝这个目录的文件到dhcpd.conf

```
[root@localhost dhcp]#
[root@localhost dhcp]# \cp -p /usr/share/doc/dhcp-4.1.1/dhcpd.conf.sample dhcpd.
conf
[root@localhost dhcp]# _
```

5、创建作用域，打开配置文件，修改第46-55行

```
# A slightly different configuration for an internal subnet.
subnet 100.10.10.0 netmask 255.255.255.0 {网络号和子网掩码
  range 100.10.10.2 100.10.10.5;起始地址段
  option domain-name-servers 100.10.10.213; 分配dns的IP地址
  option domain-name "dns.jnds.net";域名
  option routers 100.10.10.254;分配的网关
  option broadcast-address 100.10.10.255;广播地址
  default-lease-time 172800;租约期限
  max-lease-time 259200; 最大租约期限
}
```

租约期限以秒为单位

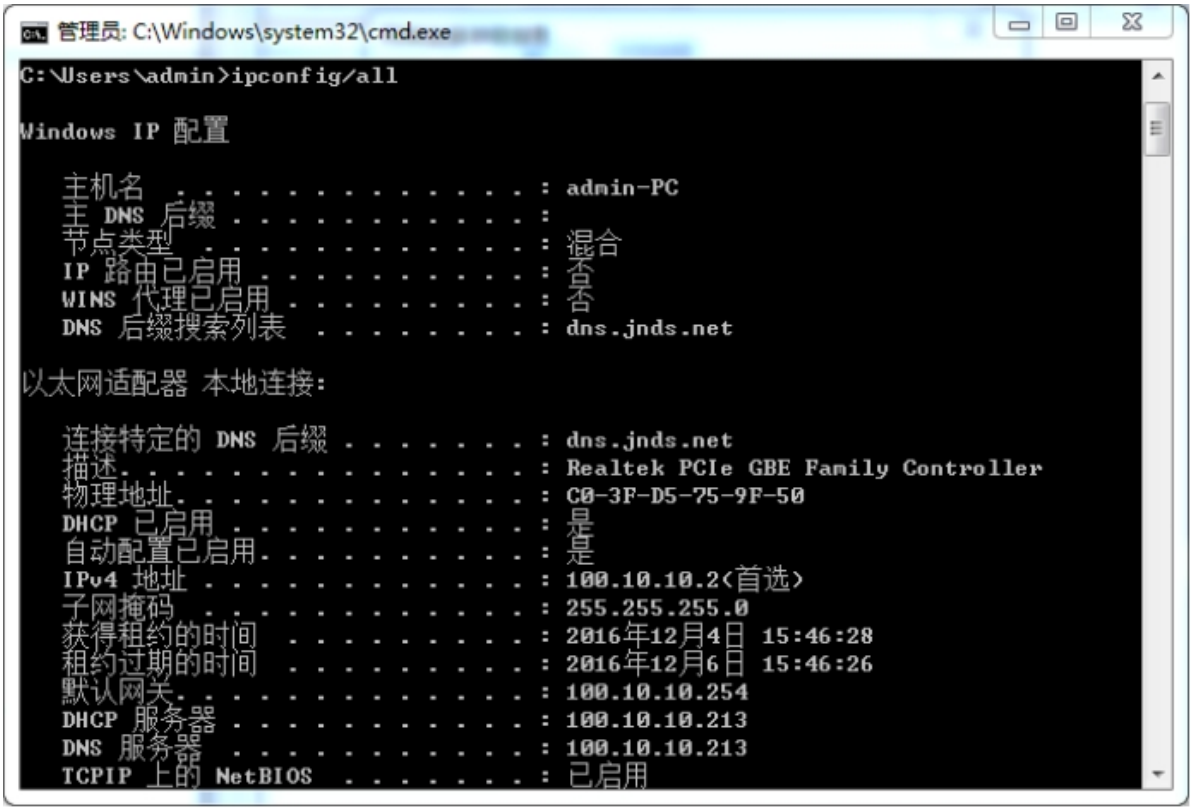
6、保存退出启动DHCP服务器，如果配置文件有错误服务器不会正常启动

```
[root@localhost dhcp]# service dhcpd restart
Starting dhcpd:                                            [  OK  ]
[root@localhost dhcp]# _
```

7、查看服务器运行的端口，DHCP使用的是udp协议的67号端口

```
[root@localhost dhcp]# netstat -nulp | grep dhcp
udp        0      0 0.0.0.0:67                  0.0.0.0:*
        1628/dhcpd
[root@localhost dhcp]# _
```

8、查看客户端获取地址

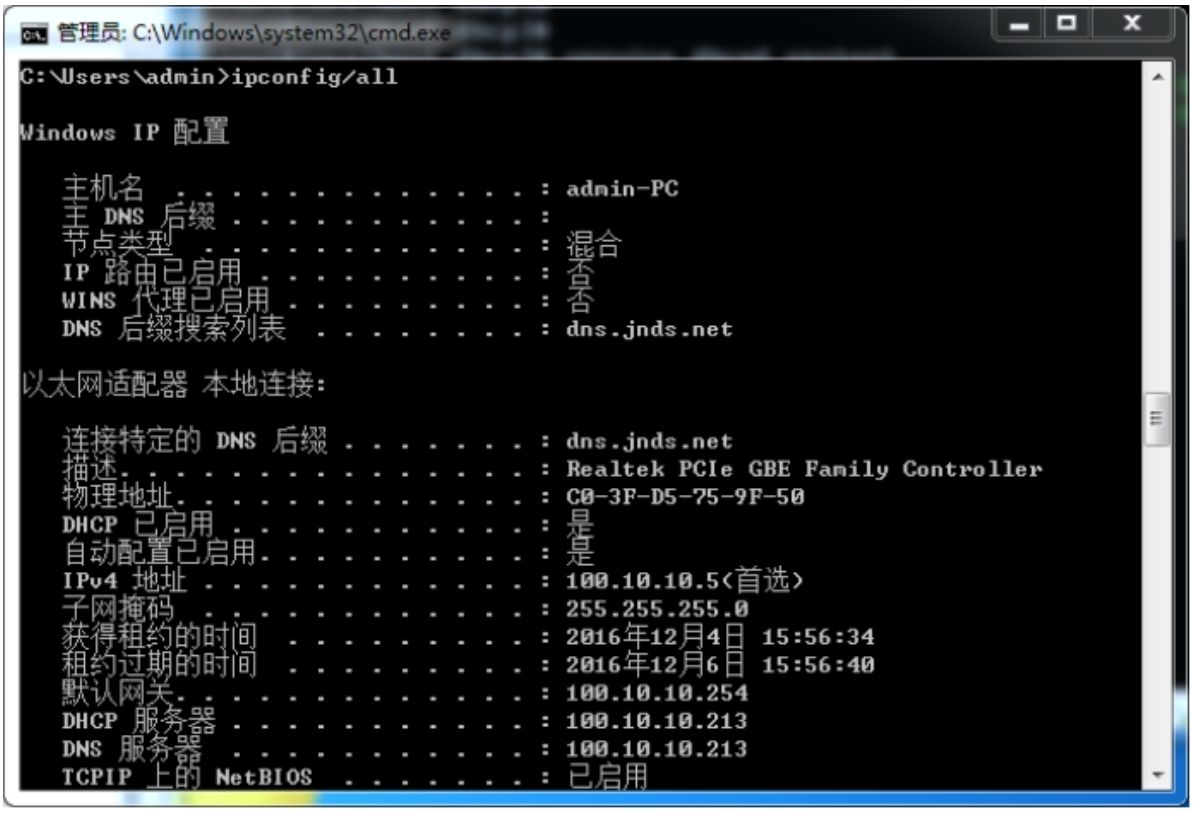


二、为DHCP客户端保留IP地址

1、打开dhcp的配置文件，修改的位置在75-78行

```
host windowsclient { 保留IP地址的客户端名称
  hardware ethernet c0:3f:d5:75:9f:50; 绑定客户端的MAC地址
  fixed-address 100.10.10.5; 为客户端保留的IP地址
}
```

2、打开客户端查看获取的IP地址



三、创建DHCP的超级作用域

1、打开dhcpd.conf配置文件，超级作用域以大括号开始和大括号结束。

超级作用域格式：

```
shared-network mydhcpdomain {

subnet 100.10.10.0 netmask 255.255.255.0 {

 range 100.10.10.2 100.10.10.5;

 option domain-name-servers 100.10.10.213;

 option domain-name "dns.jnds.net";

 option routers 100.10.10.254;

 option broadcast-address 100.10.10.255;

 default-lease-time 172800;

 max-lease-time 259200;

}

subnet 100.10.11.0 netmask 255.255.255.0 {

 range 100.10.11.2 100.10.11.10;

 option domain-name-servers 100.10.10.213;

 option domain-name "dns.jnds.net";

 option routers 100.10.11.254;

 option broadcast-address 100.10.11.255;

 default-lease-time 172800;

 max-lease-time 259200;

}
```

}

```
shared-network mydhcpdomain { 超级作用域的名称
subnet 100.10.10.0 netmask 255.255.255.0 {
  range 100.10.10.2 100.10.10.5;
  option domain-name-servers 100.10.10.213;
  option domain-name "dns.jnds.net";
  option routers 100.10.10.254;
  option broadcast-address 100.10.10.255;
  default-lease-time 172800;
  max-lease-time 259200;
}
subnet 100.10.11.0 netmask 255.255.255.0 {
  range 100.10.11.2 100.10.11.10;
  option domain-name-servers 100.10.10.213;
  option domain-name "dns.jnds.net";
  option routers 100.10.11.254;
  option broadcast-address 100.10.11.255;
  default-lease-time 172800;
  max-lease-time 259200;
}
}
```

2、启动服务器

```
[root@localhost dhcp]# service dhcpd restart
Starting dhcpd:                                            [  OK  ]
[root@localhost dhcp]# _
```

四、DHCP中继，配置DHCP中继需要两块网卡

1、添加两块网卡，到/etc/sysconfig/network-scripts目录下，拷贝ifcfg-eth0为ifcfg-eth1

```
[root@localhost network-scripts]# ls
ifcfg-eth0    ifdown-ipv6    ifup          ifup-plip    ifup-wireless
ifcfg-eth1    ifdown-isdn    ifup-aliases  ifup-plusb   init.ipv6-global
ifcfg-lo      ifdown-post    ifup-bnep     ifup-post    net.hotplug
ifdown        ifdown-ppp     ifup-eth      ifup-ppp     network-functions
ifdown-bnep   ifdown-routes  ifup-ippp     ifup-routes  network-functions-ipv6
ifdown-eth    ifdown-sit     ifup-ipv6     ifup-sit
ifdown-ippp   ifdown-tunnel  ifup-isdn     ifup-tunnel
[root@localhost network-scripts]# _
```

2、打开ifcfg-eth1手动更改网卡的MAC地址

```
DEVICE=eth1
HWADDR=00:0c:29:10:69:74
TYPE=Ethernet
UUID=168db9c4-96cb-4f00-ba96-91e13a2e43f6
ONBOOT=yes
NM_CONTROLLED=yes
BOOTPROTO=none
IPADDR=100.10.11.214
NETMASK=255.255.255.0
GATEWAY=100.10.11.254
IPV6INIT=no
USERCTL=no
~
~
~
~
~
~
~
~
~
~
~
"ifcfg-eth1" 12L, 225C
```

3、启动网卡，MAC地址配置或者其它有错误重启不会成功

```
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# service network restart
Shutting down interface eth0:                              [  OK  ]
Shutting down interface eth1:                              [  OK  ]
Shutting down loopback interface:                          [  OK  ]
Bringing up loopback interface:                            [  OK  ]
Bringing up interface eth0:  Determining if ip address 100.10.10.213 is already
in use for device eth0...
                                                           [  OK  ]
Bringing up interface eth1:  Determining if ip address 100.10.11.214 is already
in use for device eth1...
                                                           [  OK  ]
[root@localhost ~]# _
```

4、配置超级作用域，这里使用上面配置的超级作用域

```
shared-network mydhcpdomain {
subnet 100.10.10.0 netmask 255.255.255.0 {
  range 100.10.10.2 100.10.10.5;
  option domain-name-servers 100.10.10.213;
  option domain-name "dns.jnds.net";
  option routers 100.10.10.254;
  option broadcast-address 100.10.10.255;
  default-lease-time 172800;
  max-lease-time 259200;
}
subnet 100.10.11.0 netmask 255.255.255.0 {
  range 100.10.11.2 100.10.11.10;
  option domain-name-servers 100.10.10.213;
  option domain-name "dns.jnds.net";
  option routers 100.10.11.254;
  option broadcast-address 100.10.11.255;
  default-lease-time 172800;
  max-lease-time 259200;
}
}
```

超级作用域的名称

5、重启DHCP服务器

```
[root@localhost dhcp]# service dhcpd restart
Starting dhcpd:                                            [  OK  ]
[root@localhost dhcp]# _
```

6、配置DHCP中继，打开/etc/sysconfig/dhcrelay文件

```
# Command line options here
DHCRELAYARGS=""
# DHCPv4 only
INTERFACES="eth0 eth1"
# DHCPv4 only
DHCPSERVERS=""
DHCPSERVERS="100.10.10.213"
~
"/etc/sysconfig/dhcrelay" 8L, 139C
```

使用的网卡

DHCP服务器的地址

配置文件的路径

7、启动中继服务

```
[root@localhost ~]# service dhcrelay restart
Shutting down dhcrelay:                                    [  OK  ]
Starting dhcrelay:                                         [  OK  ]
[root@localhost ~]# _
```

8、开启路由转发功能，打开/etc/sysctl.conf(vi)

```
# Kernel sysctl configuration file for Red Hat Linux
#
# For binary values, 0 is disabled, 1 is enabled.  See sysctl(8) and
# sysctl.conf(5) for more details.

# Controls IP packet forwarding
net.ipv4.ip_forward = 1   默认是0，改为1

# Controls source route verification
net.ipv4.conf.default.rp_filter = 1

# Do not accept source routing
net.ipv4.conf.default.accept_source_route = 0
```

使路由转发功能生效，sysctl -p

```
[root@localhost ~]# sysctl -p
net.ipv4.ip_forward = 1
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.default.accept_source_route = 0
kernel.sysrq = 0
kernel.core_uses_pid = 1
net.ipv4.tcp_syncookies = 1
error: "net.bridge.bridge-nf-call-ip6tables" is an unknown key
error: "net.bridge.bridge-nf-call-iptables" is an unknown key
error: "net.bridge.bridge-nf-call-arptables" is an unknown key
kernel.msgmnb = 65536
kernel.msgmax = 65536
kernel.shmmax = 68719476736
kernel.shmall = 4294967296
[root@localhost ~]#
[root@localhost ~]# _
```

9、客户端获取IP地址

现在客户端获取到了100.10.10.11.0网段分配的地址

```
管理员: C:\Windows\system32\cmd.exe
C:\Users\admin>ipconfig/all

Windows IP 配置

   主机名  . . . . . . . . . . . . . : admin-PC
   主 DNS 后缀 . . . . . . . . . . . :
   节点类型 . . . . . . . . . . . . : 混合
   IP 路由已启用 . . . . . . . . . . : 否
   WINS 代理已启用 . . . . . . . . . : 否
   DNS 后缀搜索列表  . . . . . . . . : dns.jnds.net

以太网适配器 本地连接:

   连接特定的 DNS 后缀 . . . . . . . : dns.jnds.net
   描述. . . . . . . . . . . . . . . : Realtek PCIe GBE Family Controller
   物理地址. . . . . . . . . . . . . : C0-3F-D5-75-A8-09
   DHCP 已启用 . . . . . . . . . . . : 是
   自动配置已启用. . . . . . . . . . : 是
   本地链接 IPv6 地址. . . . . . . . : fe80::d4ff:4024:84ad:4f61%11(首选)
   IPv4 地址 . . . . . . . . . . . . : 100.10.11.2(首选)
   子网掩码 . . . . . . . . . . . . : 255.255.255.0
   获得租约的时间  . . . . . . . . . : 2016年12月4日 18:27:59
   租约过期的时间  . . . . . . . . . : 2016年12月6日 18:27:58
   默认网关. . . . . . . . . . . . . : 100.10.11.254
   DHCP 服务器 . . . . . . . . . . . : 100.10.11.214
```