

2.1.22 软件安装介绍

学软件开发，各种台的软件熟练安装是必须要熟练掌握。大家都知道，Windows下安装软件时，只需用鼠标双击软件的安装程序，或者用Zip等解压缩软件解压缩即可安装；在Android或者apple中安装软件时，只需要在手机应用商店点击安装即可。而在Linux下安装软件难度高于Windows、Android、ios和windows phone下软件安装。下面我就详细讲解Linux下如何安装软件。

Linux下软件的安装大概可以分为如下几种：

二进制程序的安装(最简单安装)

Rpm程序安装(后缀名为*.rpm)

Yum在线安装(本质也是rpm安装，它只是rpm的一种安装方式而已)

源码安装

2.1.23 二进制安装

Linux下二进制格式的软件是指事先已经在各种平台编译安装好相关软件，然后压缩打包，在安装时只需解压或者执行安装可执行文件即可。这种软件发行格式类似与windows系统，这样做的原因是保护源代码程序不对外泄露，保护知识产权。二进制软件包的

优点是安装简单、容易

缺点是缺乏灵活性，相应的软件包执行在对应平台下安装，离开这个环境软件就无法运行。

二进制软件包提供了很多类型的打包方式，最常见的就是我们上节讲到的RPM格式的包，还有以".tar.gz、.tgz、*.bz2"等形式的二进制软件包，最后还有一个就是提供安装程序进行安装的二进制软件包。下面分别介绍：

1.1 安装***.tar.gz、*.bz2"二进制软件包**

这种格式的软件包，安装其实就是简单的解压过程，根据不同的软件打包格式，我们用相应的解压命令解压即可。

对于.tar.gz软件格式解压：tar -zxvf xxxxxx.tar.gz对于.bz2软件格式解压：tar -jxvf xxxxxx.tar.gz
这类软件的卸载，对于解压后只有单一目录的软件，可以直接删除对应软件目录即可，如果解压后文件分散在几个目录中，需要一一手动删除目录。

例如：我们常用的应用服务器容器tomcat软件就是基于这种打包压缩格式发行的。我们只需下载后解压即可完成安装，如果需要卸载，直接删除对应的tomcat目录即可。

1.2 提供安装程序的软件包

这种软件包都提供了安装脚本或者安装向导程序，只需在下载此类软件包后解压，然后进入安装目录，找到类似的setup、install、install.sh之类的可执行文件运行即可，然后根据提示（比如安装路径，参数设置等等）进行相应的设置，接着安装就自动完成。

这类软件的卸载也提供了相应的卸载脚本或者卸载向导，根据提示即可完成软件卸载。

例如：经常使用的sun jdk就是这种安装方式，从网络下载下来的jdk是个二进制可执行文件，首先设置文件的可执行权限（例如chmod 755 jdk1.6.0_07.bin，权限操作本书后面会有详细讲述），然后运行软件（也即是./jdk1.6.0_07.bin）即可完成安装。安装完成会在当前目录产生一个对应的jdk程序目录。

2.1.24 rpm安装

RPM**软件包管理**

RPM软件包也称为二进制软件包

RPM是RedHat Package Manager (RedHat软件包管理工具) 的缩写，这一文件格式名称虽然打上了RedHat的标志，但是其原始设计理念是开放式的，现在包括RedHat、CentOS、SUSE等Linux的发行版本都有采用，可以算是公认的行业标准了。RPM文件在Linux系统中的安装最为简便

RPM**命令使用**

rpm的常用参数

i : 安装应用程序 (install)

vh : 显示安装进度 ; (verbose hash)

U : 升级软件包 ; (update)

qa: 显示所有已安装软件包 (query all)

e : 卸载应用程序 (erase)

注意：如果其它软件包有依赖关系，卸载时会产生提示信息，可使用--nodeps强行卸载。

查询所有安装的rpm包: # rpm -qa

查询mysql相关的包： # rpm -qa | grep mysql

安装：rpm -ivh jdk.rpm

卸载：rpm -e mysql*

强行卸载：rpm -e mysql* --nodeps

图形化界面，手动挂载

2.1.25 yum日常操作

yellowdog updater modified 软件包管理工具

应用**yum的好处:**

- 1,自动解决软件包依赖关系
- 2,方便的软件包升级

查询

yum list 查询所有可用软件包列表

yum search 关键字 搜索服务器上所有和关键字相关的包

可以通过yum info 关键字 来查找包名

安装

yum -y install 包名 -y 自动回答yes

升级

yum -y update 包名

注意**：**如果不加包名，就升级所有的，包括内核。必须加包名升级单个软件包，慎用升级所有的

检测升级 yum check-update

卸载

yum -y remove 包名

帮助

yum --help、man yum

yum clean all 清除缓存和旧的包

yum repolist 查看当前可用的yum源

yum deplist httpd 列出一个包所有依赖的包

2.1.26 本地yum源安装

```
#cat /etc/yum.repos.d/CentOS-Base.repo
```

```
[base]
```

```
name=CentOS-$releasever - Base
```

```
mirrorlist=http://mirrorlist.centos.org/?release=\$releasever&arch=\$basearch&repo=os
```

```
#baseurl=http://mirror.centos.org/centos/\$releasever/os/\$basearch/
```

```
gpgcheck=1
```

```
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6
```

解释：

[base] 容器名称，一定要放在[]中

name 容器说明，这个可以自己随便写

mirrorlist 镜像站点，这个可以注释掉

baseurl yum源服务器的地址。默认是CentOS官方的yum源服务器，

enabled 此容器是否生效，如果不写或写enable=1都是生效，写成enable=0就是不生效

gpgcheck 如果是1是指RPM的数字证书生效，如果是0则不生效

gpgkey 数字证书的公匙文件保存位置，不用修改

1. 挂载光盘

```
# mkdir /mnt/cdrom
```

挂/卸载

命令: mount umount

格式: mount [参数] 存储设备 挂载点

说明: 存储设备对应分区的设备文件 (如 “/dev/sdb1”) 或网络资源路径, 挂载点为用户指定用于挂载的目录。挂载点必须是一个已经存在的目录

参数	说明			
-t 文件系统类型	指定文件系统的类型, 通常不必指定。 mount 会自动选择正确的类型	常用类型	光盘或光盘镜像	iso9660
			fat16文件系统	msdos
			fat32文件系统	vfat
			ntfs文件系统	ntfs
-o options	主要用来描述设备或档案的挂载方式	常用参数	采用只读方式挂接设备	ro
			用来把一个文件当成硬盘分区挂接上系统	loop
			采用读写方式挂接设备	rw
			指定访问文件系统所用字符集	iocharset
-f	用与除错。它会使 mount 并不执行实际挂上的动作, 而是模拟整个挂上的过程。			
-v	显示较详细信息, 通常会和 -f一起使用。			
-F	使每一个 mount 的动作产生一个行程负责执行。挂上大量 NFS 档案系统时可以加快挂上的动作。			
-a	将 /etc/fstab 中定义的所有档案系统挂上。通常和 -F一起使用			

```
##命令格式: mount [-t vfstype] [-o options] device dir
其中:
#1.-t vfstype 指定文件系统的类型, 通常不必指定。mount 会自动选择正确的类型。常用类型有:
光盘或光盘镜像: iso9660
DOS fat16文件系统: msdos
Windows 9x fat32文件系统: vfat
Windows NT ntfs文件系统: ntfs
Mount windows文件网络共享: smbfs
UNIX(LINUX) 文件网络共享: nfs
#2.-o options 主要用来描述设备或档案的挂载方式。常用的参数有:
loop: 用来把一个文件当成硬盘分区挂载上系统
ro: 采用只读方式挂载设备
rw: 采用读写方式挂载设备
iocharset: 指定访问文件系统所用字符集
#3.device 要挂载(mount)的设备。
#4.dir设备在系统上的挂载点(mount point)。
##将镜像文件挂载在/mnt/cdrom下
mkdir /mnt/dvd #在根目录下的mnt目录下创建目录cdrom
mount -t iso9660 -o loop /dev/cdrom /mnt/cdrom ##挂载
umount /mnt/dvd ##卸载
```

2**, 让网络yum源文件失效**

```
cd /etc/yum.repos.d/

rename .repo .repo.bak *      #重命名所有的.repo文件

cp CentOS-Media.repo.bak CentOS-Media.repo  #配置一个.repo文件
```

3**, 修改光盘yum源文件**

vi CentOS-Media.repo

[c6-media]

name=CentOS-\$releasever - Media

baseurl=file:///mnt/cdrom

#这里的地址为自己光盘挂载地址，并把不存在的地址注释掉，在行首注释

file:///media/cdrom/

file:///media/cdrecorder/

gpgcheck=1

enabled=1 #把原来的0改为1，让这个yum源配资文件生效

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6

可使用 yum repolist 查看 当前可用yum源

yum clean all

yum repolist

2.1.27 源码编译安装

由于linux操作系统开放源代码，因而在其上安装的软件大部分也都是开源软件，例如apache、tomcat、php等软件。开源软件基本都提供源码下载，源码安装的方式；

源码安装的优点是：

- 1、用户可以自己定制软件功能，安装需要的模块，不需要的功能可以不用安装。
- 2、用户还可以自己选择安装路径，方便管理。
- 3、卸载软件也很方便，只需删除对应的安装目录即可。
- 4、能最大程度和服务平台融合，效率稍微比其他方式高。
- 5、没有windows所谓的注册表之说。

源码安装的缺点是：

- 1、安装较为繁琐，需要自己配置
- 2、安装较为耗时，需要自己编译源码
- 3、安装较为容易出错，出错也难以解决

源码安装软件一般由以下几个步骤组成：下载解压源码、分析安装平台环境（ifconfigure）、编译安装软件（make,make install）。下面我们分别介绍。

1 下载解压源码

linux下软件的源码一般都是C或者C++语言编写的，并且都会在软件的官方网站上提供源码包下载，例如apache开源项目的官方网站为<http://www.apache.org>，此外在网上的很多开源社区一般也提供一些常用软件的源码下载，源码都打包成压缩文件，常见的源码打包压缩格式有“tar.gz”、“tar.baz2”等。我们可以从网站上首先下载源码文件，然后在传到所在的linux系统下，如果你的linux系统处于联网状态下，也可以直接在系统内通过wget之类的下载命令将源码包直接下载到linux系统下。

下载完成，把相应的软件解包即可，针对下载软件包的不同，使用不同的方法进行解压，解压完毕后进入解压生成的目录中，在这个目录下，一般都存在一个README文件，这个文本文件非常重要，它详细的介绍了这个软件所能完成的功能、授权许可、安装需求、安装注意事项、安装方式等等，由于linux各个版本的差异，以及安装环境的不同，软件的安装方式也不尽相同，所以在安装软件前，一定要事先阅读这个README文件，确保安装的正确性。

2 分析安装平台环境

在软件包解压完毕之后，我们进入源码目录，一般都会存在configure和README这么两个文件，README就是上面我们讲到的，是对软件的介绍和安装说明；linux下软件的安装受到操作系统安装环境的影响，比如某些软件在安装或者运行过程中需要调用操作系统本身的库文件，或者需要运行系统的某个工具等等，但是系统中不存在这个库文件或者这个工具没有安装，那么安装或者运行软件就会失败。

为了避免这个问题的出现，在安装软件的时候就需要分析操作系统环境，检测当前系统是否拥有安装软件所需的所有文件和工具，如果系统缺少某个文件，就给出提示，直到满足软件的所有需求为止。这个就是configure文件的功能，configure文件一般是个可执行文件，可以在当前目录下直接输入“./configure”进行软件安装的环境测试，如果提示缺少某些安装包，就需要进行安装，直到测试通过。通常的，源码安装都需要GCC或者CC编译器，这些编译器一般在安装系统时定制安装包中的开发工具选项下，其实这也是我们在第二章安装linux系统中要求选择开发工具包的原因所在。

此外，在执行configure分析软件需求时，还可以在“./configure”后面加上软件的安装路径以及安装所需的模块等等一些选项，来定制用户需要的软件功能。

3 编译安装软件

在验证软件安装环境后，正式进入软件的编译步骤，在进行编译前，首先了解下关于编译的一些基础知识。

在linux系统下，make是我们经常用到的编译命令，无论是安装软件还是项目开发，都会经常用到编译安装命令，也就是make和make install，对于一个包含很多源文件的应用程序，使用make和makefile工具可以简单快速的解决各个源文件之间复杂的依赖关系，同时，make工具还可以自动完成所有源码文件的编译工作，并且可以只对上次编译后修改过的文件进行增量编译，因此，熟练掌握了make和makefile工具之后，源码安装软件就变的像windows下安装软件一样简单。

Makefile**文件**

make工具最主要的功能就是通过makefile文件来实现的，makefile文件是按照某种语法来进行编写的，文件中定义了各个源文件之间的依赖关系，并说明了如何编译源文件并生成可执行文件，它通过描述各个源程序之间的关系让make工具自动完成编译工作。

在linux系统下，习惯用Makefile替代makefile文件，我们在执行configure通过后，就会在当前目录下生成这个文件，一般情况下在命令行输入make时，make命令会默认去寻找当前目录下的Makefile文件，如果我们用其它文件作为makefile，就要在make命令选项后指定makefile文件，例如我们设定编译规则文件为make_file，就需要使用以下命令指定：

```
[root@WEBServer ~]#make -f make_file
```

下面我们通过举例了解下Makefile的编写原则和内容。

例如有一个test程序，有三个源文件file1.c、file2.c、file3.c以及库文件LS编译生成，这三个源文件还分别包含了自己的头文件1.h、2.h、3.h。通常，C编译器将会输出三个目标文件file1.o、file2.o、file3.o。假定file1.c和file3.c都要声明用到一个apcs的文件，但是file2.c不用，也就是在file1.c和file3.c文件中都有类似“#include “apcs””的声明。

下面这段代码就是对test程序的描述：

```
1:#It is a example for describing makefile2:test: file1.o file2.o file3.o3:cc file1.o file2.o file3.o -
LS -o test
```

```
4:file1.o: file1.c 1.h apcs
```

```
5:gcc -c file1.c
```

```
6:file3.o: file3.c 3.h apcs
```

```
7:gcc -c file3.c
```

```
8:file2.o: file2.c c.h
```

```
9: gcc -c file2.c
```

这个描述文档其实就是一个makefile文件，第一行为注释行，第二行指定test程序有file1.o、file2.o、file3.o链接生成，第三行描述了如何从test所依赖的文件建立可执行文件，接下来的第四、六、八行定义了三个目标文件和它们依赖的“.c”、“.h”和“apcs”文件。剩下的第五、七、九行定义了如何从目标依赖的文件生成可执行文件。

在了解Makefile文件结构后，我们就可以控制编译的选项，定制自己所需的软件功能。

接下来，我们只需在命令行输入make命令，即可进入编译阶段，根据软件源程序的大小和系统的硬件配置，编译时间不定，编译完毕，会在当前目录以及子目录下生成相应的可执行文件，接下来就进入最后的安装软件阶段，在命令行输入make install命令，开始安装软件，安装进程会首先创建安装目录，如果我们没有指定安装的目录，安装程序默认会安装在系统的/usr/local目录下创建安装目录，然后将相应的文件和可执行程序从源码目录拷贝到安装目录下，这样安装就完毕了。

2.1.28 克隆

安装minimal版本虚拟机(自行安装)



克隆后所需要进行的操作：

1.eth0删掉 ,eth1改成eth0

```
vim /etc/udev/rules.d/70-persistent-net.rules
```

```
# program, run by the persistent-net-generator.rules rules file.
#
# You can modify it, as long as you keep each rule on a single
# line, and change only the value of the NAME= key.

# PCI device 0x8086:0x100f (e1000)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="00:0c:29:cd:18:1c",
ATTR{type}=="1", KERNEL=="eth*", NAME="eth0" 删除这部分

# PCI device 0x8086:0x100f (e1000)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}=="00:0c:29:3b:7a:af",
ATTR{type}=="1", KERNEL=="eth*", NAME="eth1"

~
~
~
```

改外eth0

复制这个硬件地址

2.虚拟机改ip地址 , mac地址,删除uuid 或者 将机器重启也可以

```
vim /etc/sysconfig/network-scripts/ifcfg-eth0
```

用复制的硬件地址替换掉ifcfg-eth0中的硬件地址

删除UUID那一行或者将uuid中的数字或字母改变一个

##（如果删除了UUID那一行，所有配置完成后要重新启动机器）

3.改主机名和映射文件

```
##改主机名
```

```
hostname 主机名
```

```
vim /etc/sysconfig/network
```

```
##改ip映射
```

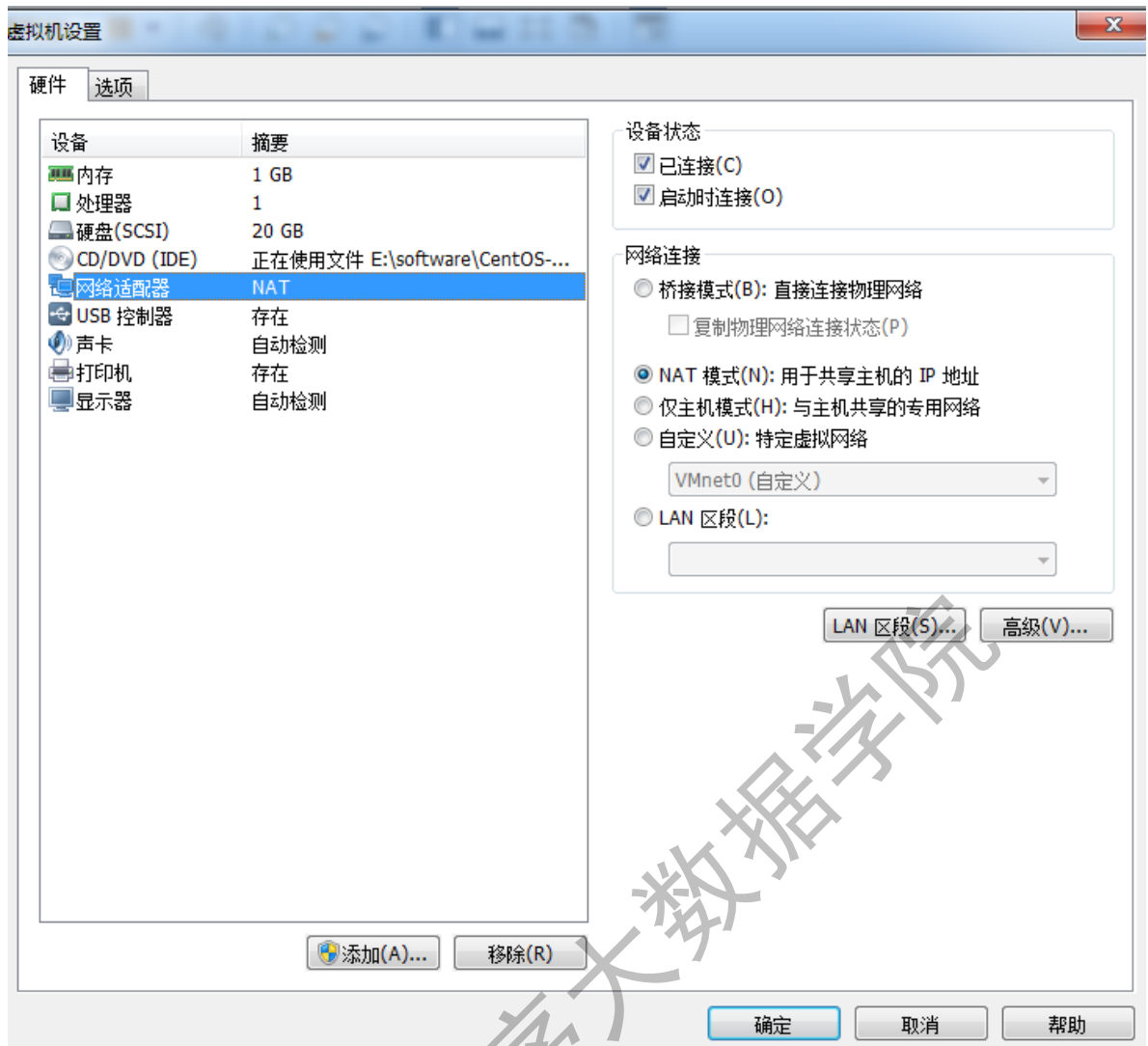
```
vim /etc/hosts
```

4.克隆后,修改了eth0的ip地址,但是ipconfig后,ip没有变化

方法:到network-scripts下,将ifcfg-eth0删掉,重启网络

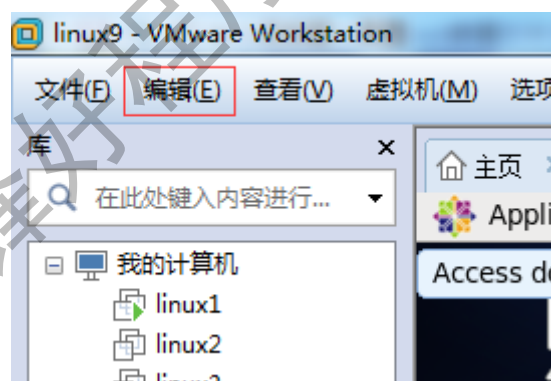
5. 使用nat模式实现两台虚拟机局域网通讯

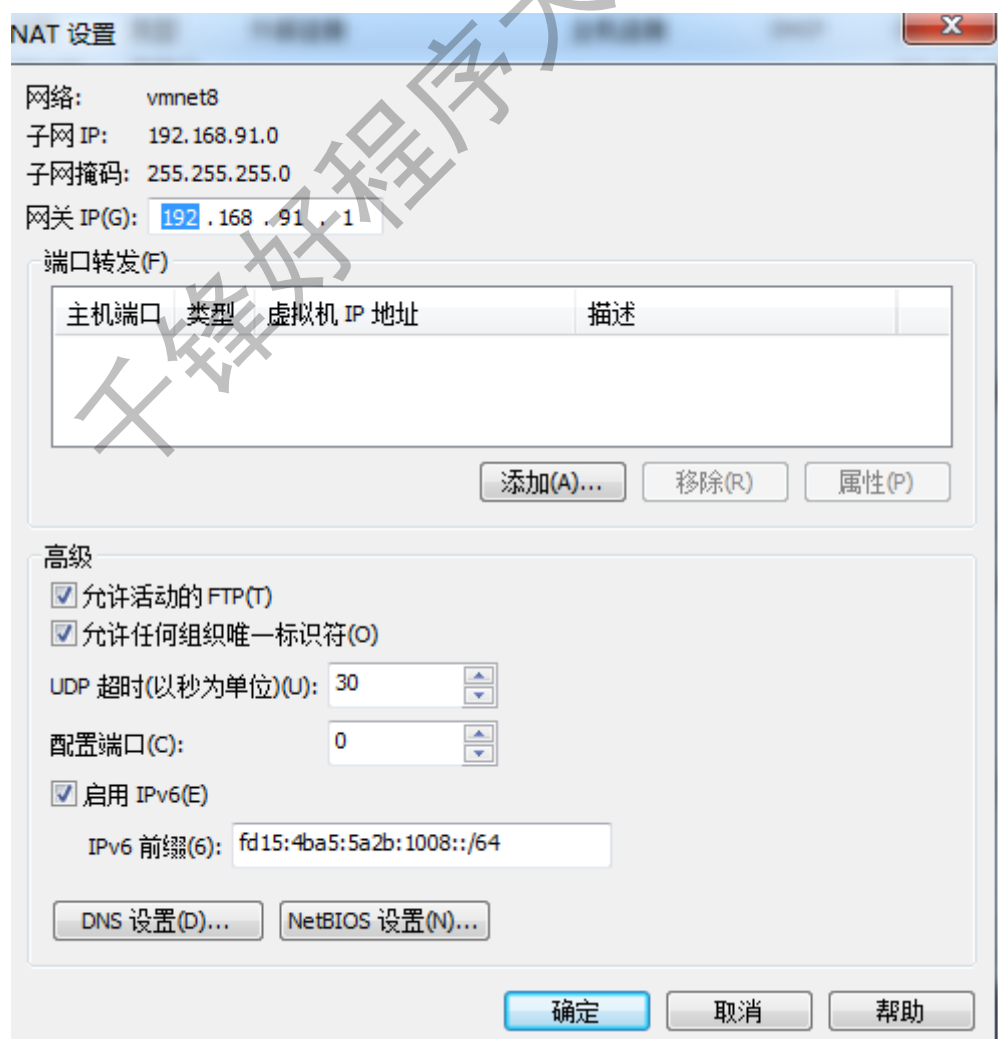
- 1.更改主机名临时+永久
- 2.检查网卡是否启动
- 3.关闭NetworkManager网络管理服务
- 4.关闭防火墙
- 5.选择nat模式





6.查看vmnet8网关





- 7.更改虚拟机ip地址，网关，DNS
- 8.更改主机vmnet8的ip地址，网关，DNS（为了使用远程连接工具连接虚拟机）
- 9.虚拟机互相ping，进行测试

7. 操作新linux系统流程

- 1.更改主机名临时+永久
- 2.检查网卡是否启动
- 3.关闭NetworkManager网络管理服务
- 4.关闭防火墙
- 5.更改ip地址
- 6.更改主机名称和主机映射
- 7.使用连接工具（xshell）进行连接
- 8.如果出现问题按照以上流程检查

2.1.30 scp命令

集群模式下，各节点之间如何方便的拷贝文件？

命令格式：

scp file 远程用户名@远程服务器IP:~/（注意：冒号和目录之间不能有空格）

如果拷贝目录，需要加-r 选项。

使用root用户

eg: scp /etc/profile root@node2:/etc

eg: scp -r /usr/jdk1.8 node2:/usr/java

eg: scp hello.log node2:\$PWD

注：使用\$PWD 默认到当前目录。注意：目标主机要有与当前相同的目录

~:到当前用户的宿主目录

scp /etc/services node2:/root/service.hard

可以通过这种方式修改拷贝的文件名。

2.1.31 ssh免密登录

如何方便的切换到其他节点？？

ssh node2

免密登录

1. 在第一台机器上生成一对钥匙，公钥和私钥

```

[root@node1 ~]# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
bd:8c:39:36:1c:e2:f8:c1:a6:53:98:09:e4:0b:63:e5 root@node1
The key's randomart image is:
+--[ RSA 2048 ]-----+
|
|..
|oo
|o.oE
|.o o +. S .
|. ++.o = .
|..= B o
|. + o o
|...
+-----+
[root@node1 ~]#

```

输入三次回车

当前用户的宿主目录下的.ssh目录多了两个文件

```

[root@node1 ~]# cd .ssh/
[root@node1 .ssh]# ll
total 8
-rw-----. 1 root root 1675 Jul 23 18:12 id_rsa
-rw-r--r--. 1 root root 392 Jul 23 18:12 id_rsa.pub
[root@node1 .ssh]#

```

将公钥拷贝给要免密码登录的机器

注意：主机名和ip都可以（确保配置了主机名ip的映射）

还需要输入密码

```

[root@node1 .ssh]# ssh-copy-id root@node2
root@node2's password:
Now try logging into the machine, with "ssh 'root@node2'", and check in:

  .ssh/authorized_keys

to make sure we haven't added extra keys that you weren't expecting.
[root@node1 .ssh]#

```

主机名或ip地址都可以

拷贝完成之后，会在要免密登录的机器上生成授权密码文件

```

[root@node2 .ssh]#
[root@node2 .ssh]# pwd
/root/.ssh
[root@node2 .ssh]# ll
total 4
-rw-----. 1 root root 392 Jul 23 18:18 authorized_keys
[root@node2 .ssh]#

```

注意：免密码登录是单向的

扩展：

ssh免密登录原理：

1, node1 使用 ssh-keygen -t rsa 生成公钥和私钥

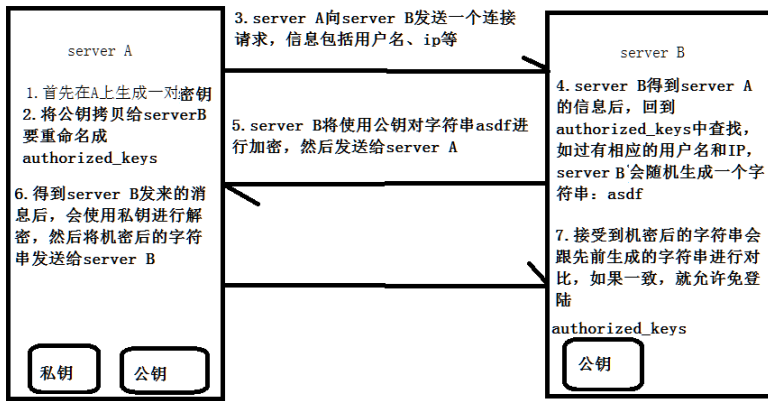
1. ssh-copy-id root@node2 将公钥拷贝到node2上，实际上是把公钥内容追加到 authorized_keys 文件中。

请求时：

1, node1向node2发送连接请求时，附带主机，ip地址等信息

1. node2收到请求后，去授权文件里查找node1的公钥，找到之后，随机生成一个字符串，并用公钥加密，并发送给node1
2. node1接收到密文之后，用私钥解密，并把解密结果返回给node2
3. node2拿到解密结果之后，与之前的字符串进行比较，如果相同，则可以登录成功。

原理图



千锋好程序大数据学院