**Day1 linux\虚拟网络**

**三种网络：**

**VMware虚拟机三种联网方法及原理**

**一、Brigde——桥接：默认使用VMnet0**

**1、原理：**

**Bridge 桥"就是一个主机，这个机器拥有两块网卡，分别处于两个局域网中，同时在"桥"上，运行着程序，让局域网A中的所有数据包原封不动的流入B，反之亦然。这样，局域网A和B就无缝的在链路层连接起来了，在桥接时，VMWare网卡和物理网卡应该处于同一IP网段 当然要保证两个局域网没有冲突的IP.**

**VMWare 的桥也是同样的道理，只不过，本来作为硬件的一块网卡，现在由VMWare软件虚拟了！当采用桥接时，VMWare会虚拟一块网卡和真正的物理网卡就行桥接，这样，发到物理网卡的所有数据包就到了VMWare虚拟机，而由VMWare发出的数据包也会通过桥从物理网卡的那端发出。**

**所以，如果物理网卡可以上网，那么桥接的软网卡也没有问题了，这就是桥接上网的原理了。**

**2、联网方式：**

**这一种联网方式最简单，在局域网内，你的主机是怎么联网的，你在虚拟机里就怎么连网。把虚拟机看成局域网内的另一台电脑就行了！**

**提示：主机网卡处在一个可以访问Internet的局域网中，虚拟机才能通过Bridge访问Internet。**

**`````````````````````````````````````````````````````````````````````````````````````````````````````````````````````**

**二、NAT——网络地址转换 ：默认使用VMnet8**

**1、原理：**

**NAT 是 Network address translate的简称。NAT技术应用在internet网关和路由器上，比如192.168.0.123这个地址要访问internet，它的数据包就要通过一个网关或者路由器，而网关或者路由器拥有一个能访问internet的ip地址，这样的网关和路由器就要在收发数据包时，对数据包的IP协议层数据进行更改（即 NAT），以使私有网段的主机能够顺利访问internet。此技术解决了IP地址稀缺的问题。同样的私有IP可以网关NAT 上网。**

**VMWare的NAT上网也是同样的道理，它在主机和虚拟机之间用软件伪造出一块网卡，这块网卡和虚拟机的ip处于一个地址段。同时，在这块网卡和主机的网络接口之间进行NAT。虚拟机发出的每一块数据包都会经过虚拟网卡，然后NAT，然后由主机的接口发出。**

**虚拟网卡和虚拟机处于一个地址段，虚拟机和主机不同一个地址段，主机相当于虚拟机的网关，所以虚拟机能ping到主机的IP，但是主机ping不到虚拟机的IP。**

**2、联网方式：**

**方法1、动态IP地址。**

**主机是静态IP或动态IP，都无所谓，将虚拟机设置成使用DHCP方式上网,Windows下选择“自动获取IP“，linux下开启DHCP服务即可。（这种方法最简单，不用过多的设置，但要在VMware中进行“编辑→虚拟网络设置”，将NAT和DHCP都开启了。一般NAT默认开启，DHCP默认关闭）**

**方法2、静态IP地址。**

**如果不想使用DHCP，也可以手动设置：**

**IP设置与vmnet1同网段,网关设置成vmnet8的网关（在“虚拟网络设置”里的Net选项卡里能找到Gateway）通常是xxx.xxx.xxx.2。**

**子网掩码设置与VMnet8相同（设置好IP地址后，子网掩码自动生成）**

**DNS设置与主机相同。**

**例如：主机IP是10.70.54.31,设置虚拟机IP为10.70.54.22。Netmask,Gateway,DNS都与主机相同即可实现 虚拟机 ---主机 虚拟机<---->互联网 通信。**

**提示：使用NAT技术，主机能上网，虚拟机就可以访问Internet，但是主机不能访问虚拟机。**

**`````````````````````````````````````````````````````````````````````````````````````````````````````````````````````**

**三、Host-Only——私有网络共享主机：默认使用VMnet1**

**1、原理：**

**提供的是主机和虚拟机之间的网络互访。只想让虚拟机和主机之间有数据交换，而不想让虚拟机访问Internet，就要采用这个设置了。**

**Host-only的条件下，VMWare在真正的Windows系统中，建立一块软网卡。这块网卡可以在网络连接中看到，一般是VMNET1，这块网卡的作用就是使Windows看到虚拟机的IP。**

**2、联网方法：**

**方法1、动态IP地址。**

**像上面那样开启DHCP后，虚拟机直接自动获取IP地址和DNS。就可以和主机相连了。当然，还要进行一些局域网共享的操作，这里不再赘述。**

**方法2、静态IP地址。**

**也可以手动设置，将虚拟机IP设置与VMnet1同网段,网关设置成VMnet1的网关相同,其余设置与VMnet1相同,DNS设置与主机相同。**

**例如：VMnet1 IP:172.16.249.1 Gateway :172.16.249.2**

**那么虚拟机 IP:172.16.249.100 Gateway: 172.16.249.2**

**这样、 虚拟机<--->主机 可以通信**

**但是、 虚拟机<--->互联网 无法通信**

**提示：Host-only技术只用于主机和虚拟机互访，于访问internet无关。**

# Host-Only



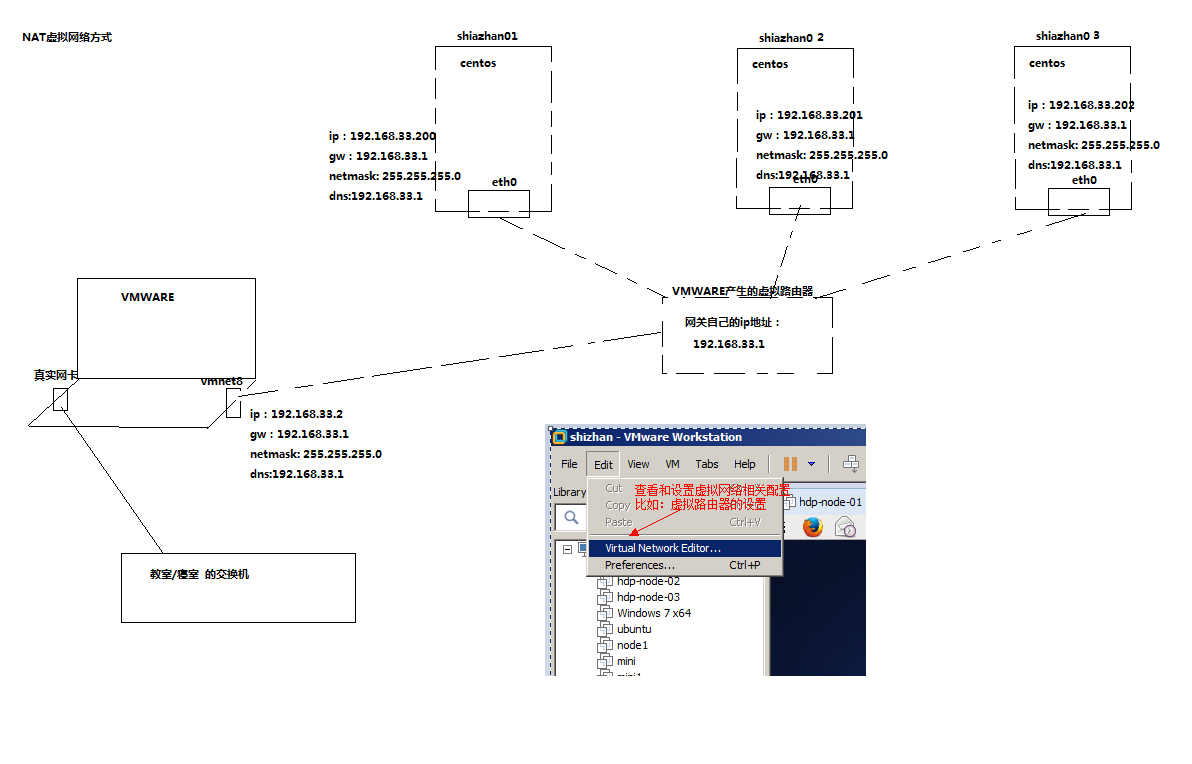
# 桥接



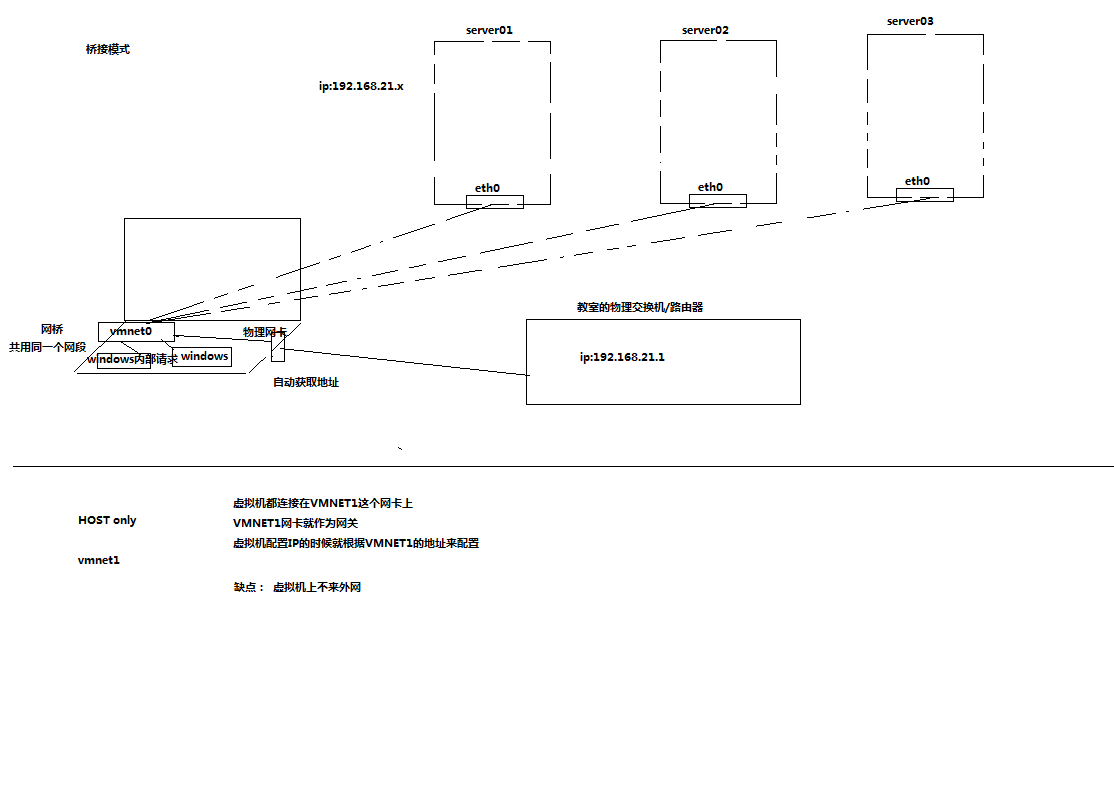
# NAT



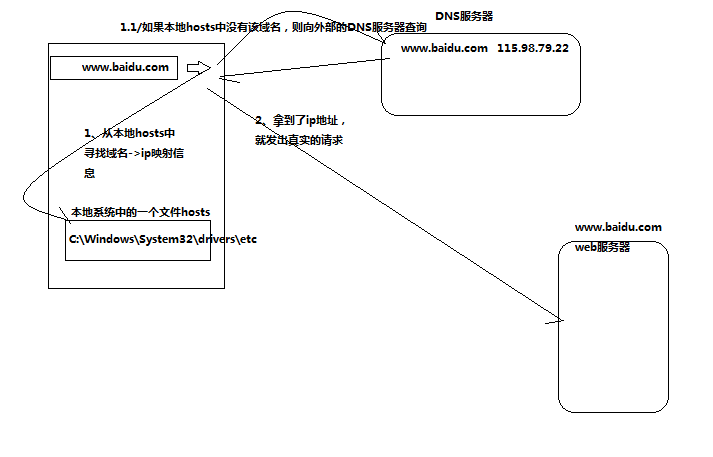
NAT虚拟网络配置：



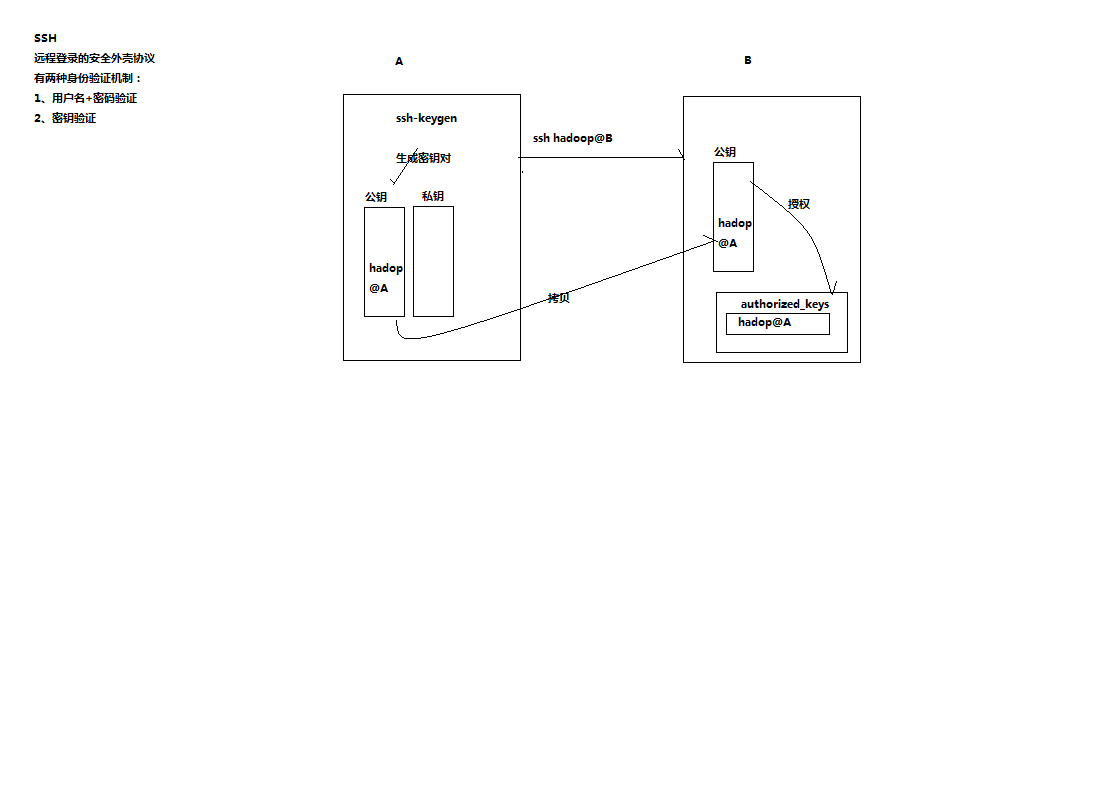
桥接和hostonly虚拟网络配置：



域名服务工作流程：



Ssh免密登陆：



# Linux常用软件安装

# 课程目标

## 熟练使用Linux常用命令

## 熟练修改Linux的基本配置

## 熟练在Linux上安装JDK、Tomcat、MySQL软件

# 修改Linux的基本配置

\*\*IP地址配置，最简单的一种

在命令运行 setup，带提示，按照提示修改即可

1. 修改主机名

vi /etc/sysconfig/network

|  |
| --- |
| NETWORKING=yes  HOSTNAME=server1.itcast.cn |

1. 修改ip地址

vi /etc/sysconfig/network-scripts/ifcfg-eth0

|  |
| --- |
| DEVICE=eth0  TYPE=Ethernet  ONBOOT=yes #是否开机启用  BOOTPROTO=static #ip地址设置为静态  IPADDR=192.168.0.101  NETMASK=255.255.255.0 |

service network restart

1. 修改ip地址和主机名的映射关系

vi /etc/hosts

|  |
| --- |
| 127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4  ::1 localhost localhost.localdomain localhost6 localhost6.localdomain6  192.168.0.101 server1.itcast.cn |

1. 关闭iptables并设置其开机启动/不启动

service iptables stop

chkconfig iptables on

chkconfig iptables off

# 安装JDK

1. 上传jdk-7u45-linux-x64.tar.gz到Linux上
2. 解压jdk到/usr/local目录

tar -zxvf jdk-7u45-linux-x64.tar.gz -C /usr/local/

1. 设置环境变量，在/etc/profile文件最后追加相关内容

vi /etc/profile

|  |
| --- |
| export JAVA\_HOME=/usr/local/jdk1.7.0\_45  export PATH=$PATH:$JAVA\_HOME/bin |

1. 刷新环境变量

source /etc/profile

1. 测试java命令是否可用

java -version

# 安装Tomcat

1. 上传apache-tomcat-7.0.68.tar.gz到Linux上
2. 解压tomcat

tar -zxvf apache-tomcat-7.0.68.tar.gz -C /usr/local/

1. 启动tomcat

/usr/local/apache-tomcat-7.0.68/bin/startup.sh

1. 查看tomcat进程是否启动

jps

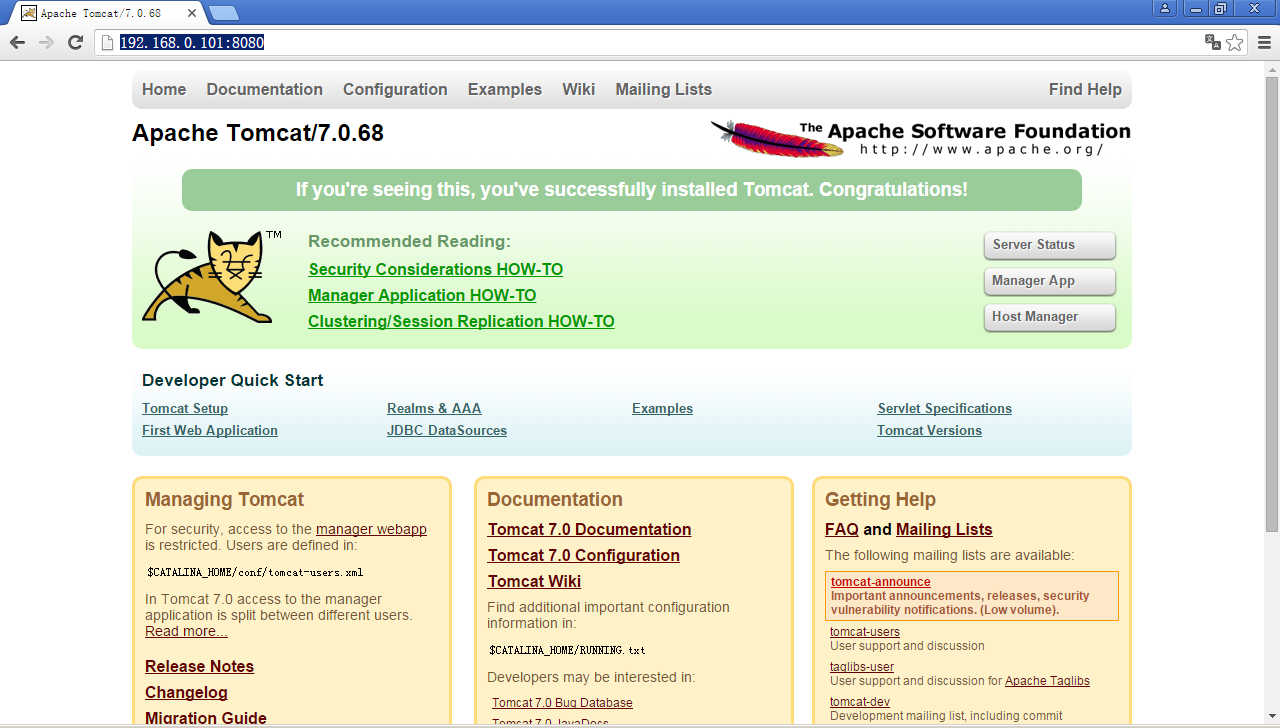


1. 查看tomcat进程端口

netstat -anpt | grep 2465

1. 通过浏览器访问tomcat

<http://192.168.0.101:8080/>



# 安装MySQL

1. 上传MySQL-server-5.5.48-1.linux2.6.x86\_64.rpm、MySQL-client-5.5.48-1.linux2.6.x86\_64.rpm到Linux上
2. 使用rpm命令安装MySQL-server-5.5.48-1.linux2.6.x86\_64.rpm，缺少perl依赖

rpm -ivh MySQL-server-5.5.48-1.linux2.6.x86\_64.rpm



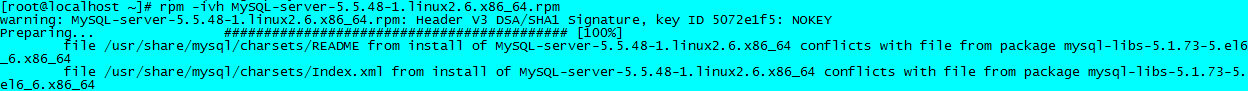
1. 安装perl依赖，上传6个perl相关的rpm包



rpm -e perl-\*

1. 再安装MySQL-server，rpm包冲突

rpm -ivh MySQL-server-5.5.48-1.linux2.6.x86\_64.rpm



1. 卸载冲突的rpm包

rpm -e mysql-libs-5.1.73-5.el6\_6.x86\_64 --nodeps

1. 再安装MySQL-client和MySQL-server

rpm -ivh MySQL-client-5.5.48-1.linux2.6.x86\_64.rpm

rpm -ivh MySQL-server-5.5.48-1.linux2.6.x86\_64.rpm

1. 启动MySQL服务，然后初始化MySQL

service mysql start

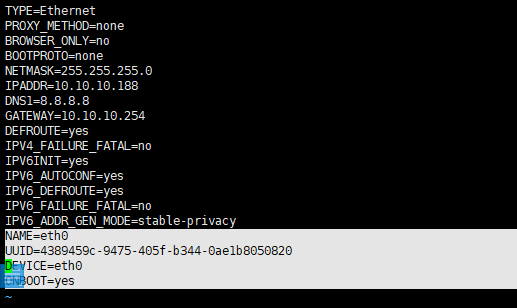
/usr/bin/mysql\_secure\_installation

1. 测试MySQL

mysql -u root –p

# [修改CENTOS7的网卡ens33修改为eth0](https://www.cnblogs.com/w787815/p/9277583.html)

1、先编辑网卡的配置文件将里面的NAME DEVICE项修改为eth0   
vim /etc/sysconfig/network-scripts/ifcfg-ens33



2、[root@linux-node1~ network-scripts]# mv ifcfg-ens33 ifcfg-eth0

3、禁用该可预测命名规则。对于这一点，你可以在启动时传递“net.ifnames=0 biosdevname=0 ”的内核参数。这是通过编辑/etc/default/grub并加入“net.ifnames=0 biosdevname=0 ”到GRUBCMDLINELINUX变量来实现的。

[root@linux-node1 ~]# vim /etc/default/grub

[复制代码](javascript:void(0);)

GRUB\_TIMEOUT=5

GRUB\_DISTRIBUTOR="$(sed 's, release .\*$,,g' /etc/system-release)"

GRUB\_DEFAULT=saved

GRUB\_DISABLE\_SUBMENU=true

GRUB\_TERMINAL\_OUTPUT="console"

GRUB\_CMDLINE\_LINUX="crashkernel=auto rd.lvm.lv=centos/root rd.lvm.lv=centos/swap rhgb quiet "net.ifnames=0 biosdevname=0""

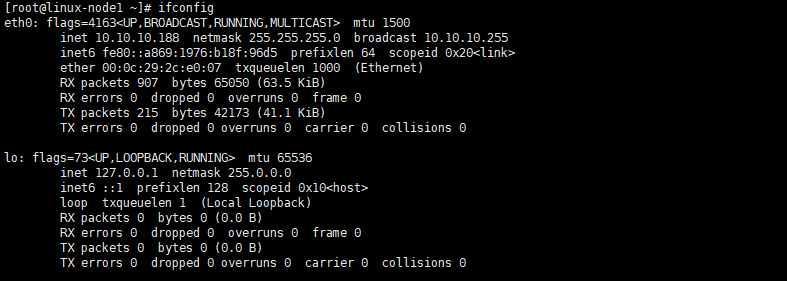
GRUB\_DISABLE\_RECOVERY="true"

[复制代码](javascript:void(0);)

4.运行命令grub2-mkconfig -o /boot/grub2/grub.cfg 来重新生成GRUB配置并更新内核参数。

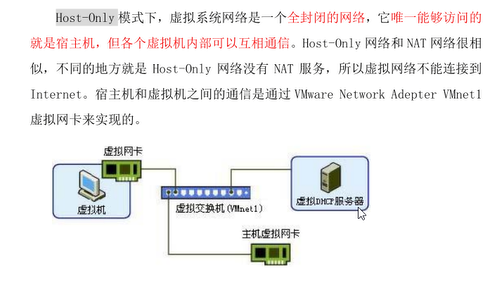
grub2-mkconfig -o /boot/grub2/grub.cfg

5、重启认证

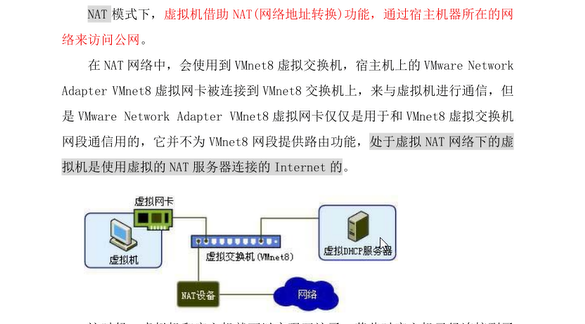


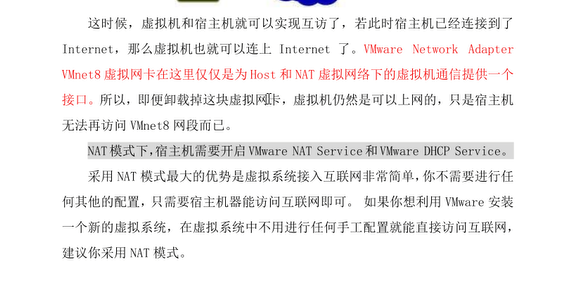
 桥接模式：通过VMnet0，虚拟机和宿主机处于同一个网段；

Host-only:通过Vmnet1,不能上网；



NAT模式：：VMNET8,可以上网，

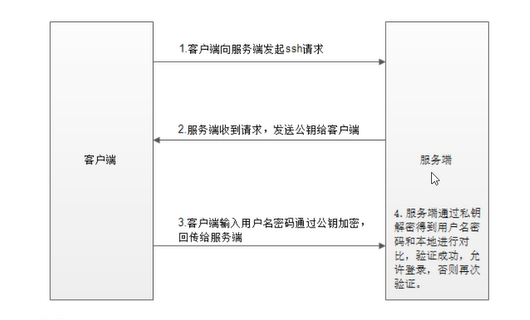


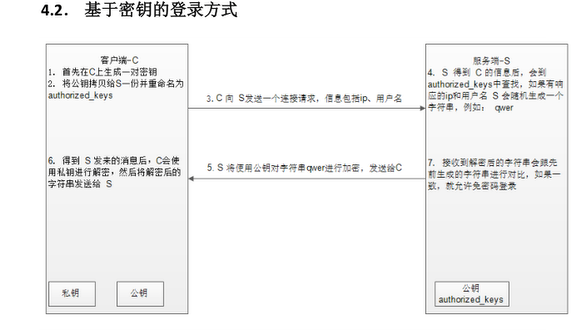


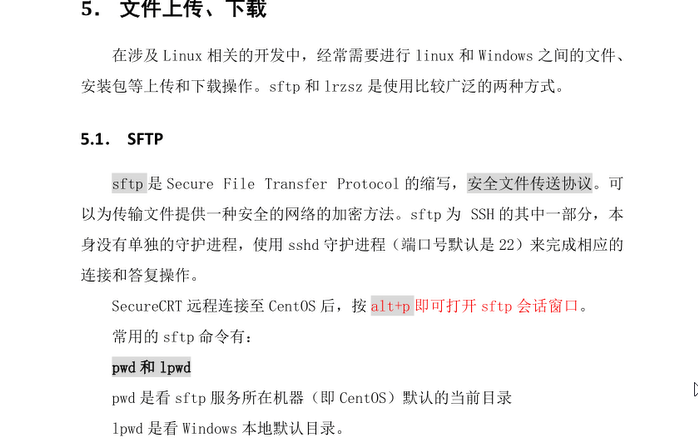
SSH:

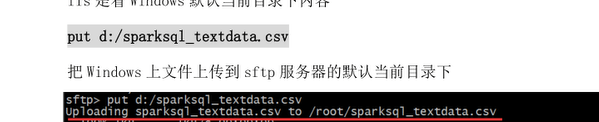
非对称加密；有一对秘钥：公钥、私钥。公加密，私解密；

1、加密

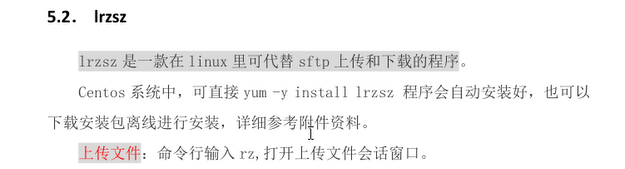






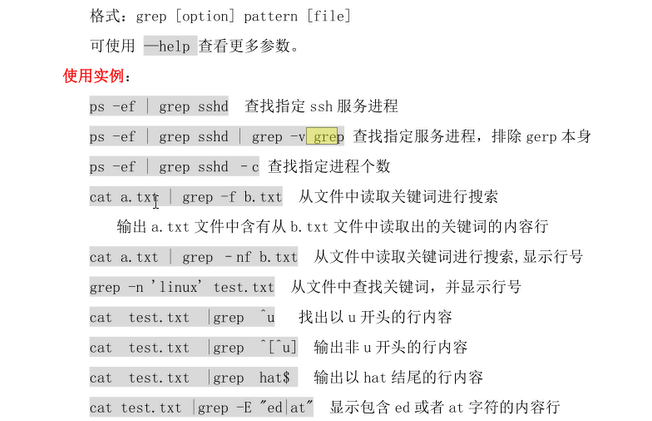


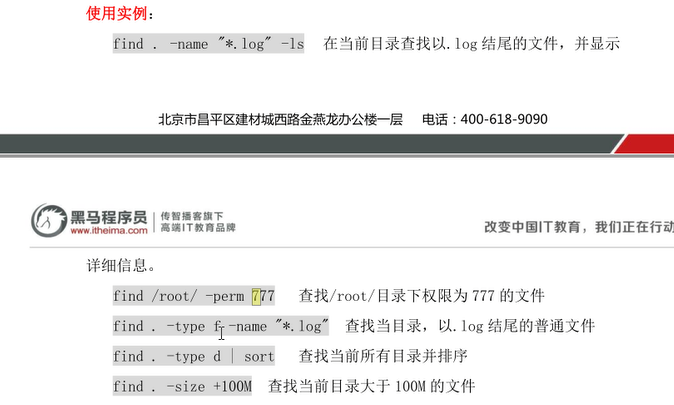
下载：get xxxx把服务器上的文件下载到本地

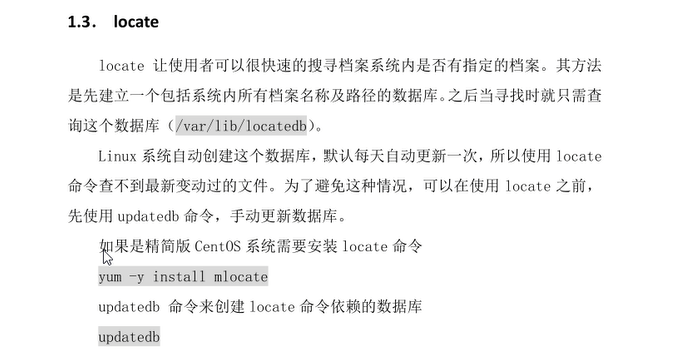


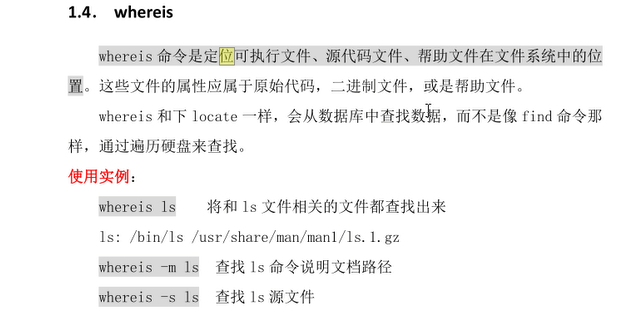
或者直接拖拽就可以。

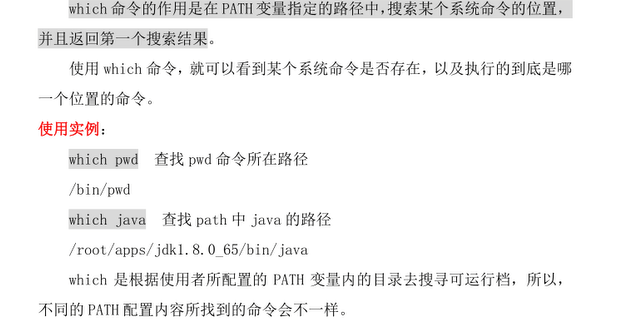
命令：

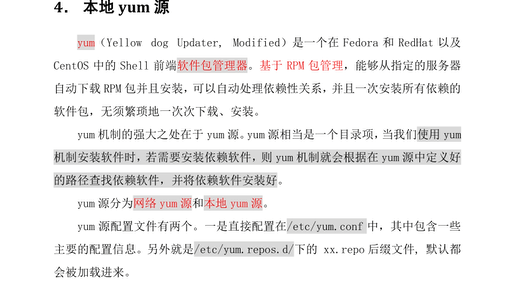


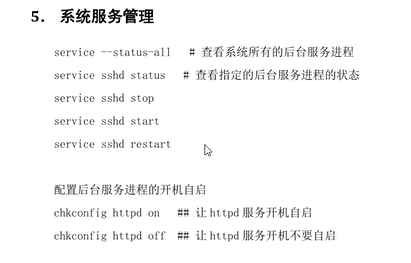


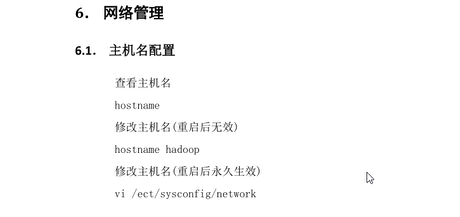


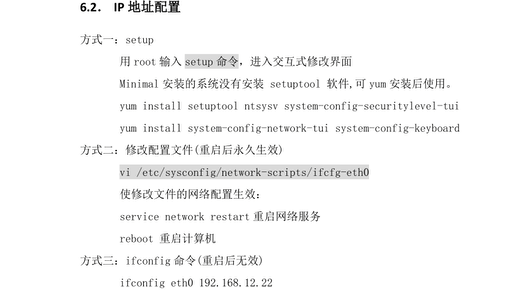


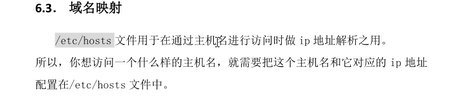


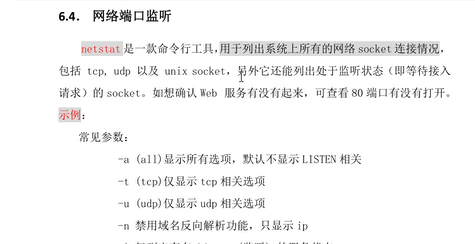




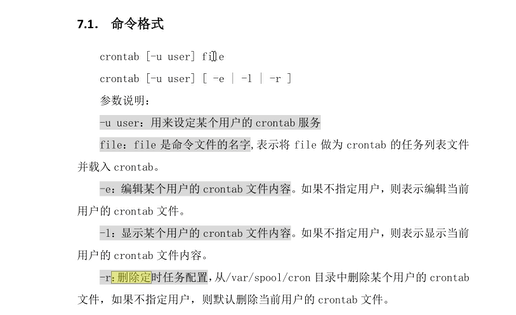
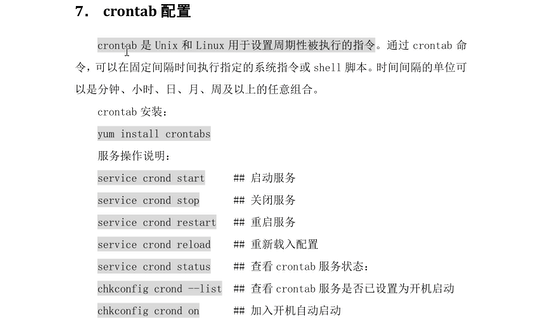




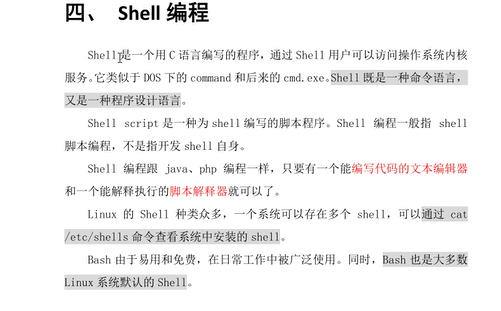


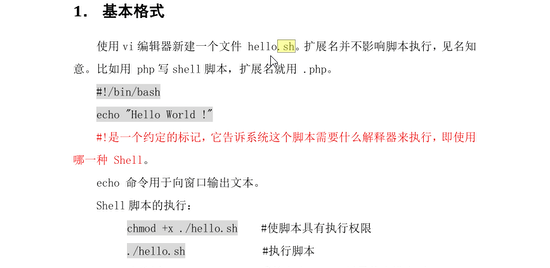


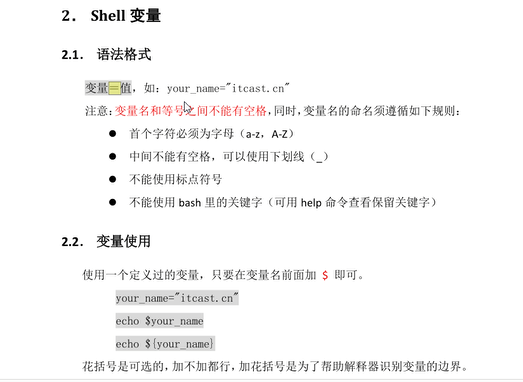


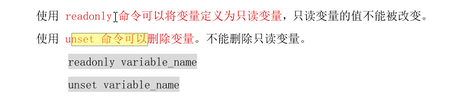
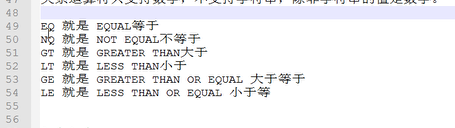
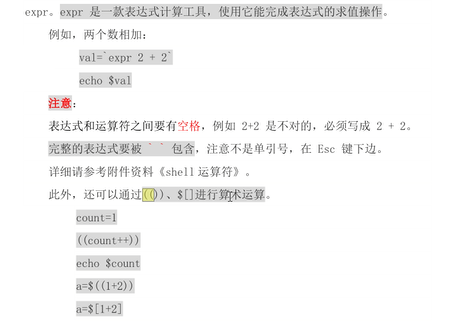
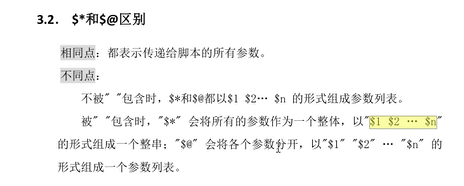
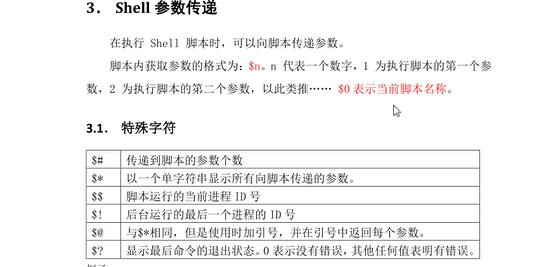




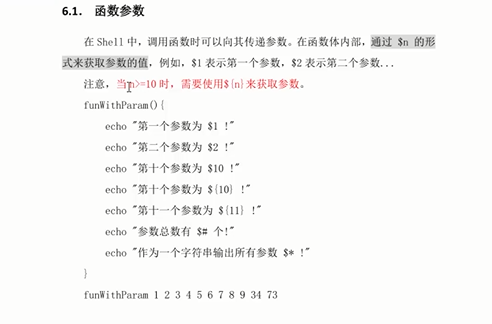












局域网工作机制：

