



夕阳下的奔跑

不积跬步 无以至千里



VirtualBox下虚拟机和主机内网互通+虚拟机静态IP的网络配置

By K.X

🕒 发表于 2014-09-05

本文给出了在VirtualBox下搭建一个

- 虚拟机之间，以及虚拟机和主机之间内网互通
- 主机和虚拟机都有静态IP
- 主机和虚拟机都可以访问外网（internet）

的网络配置方案。

前提条件

先定义一下名词：

- **主机**：也叫做host，也就是安装了VirtualBox的环境，物理上存在的实体机
- **虚拟机**：也叫做guest，也就是安装在VirtualBox里面的各个操作系统，不需要硬件，虚拟的系统

环境要求

文章目录

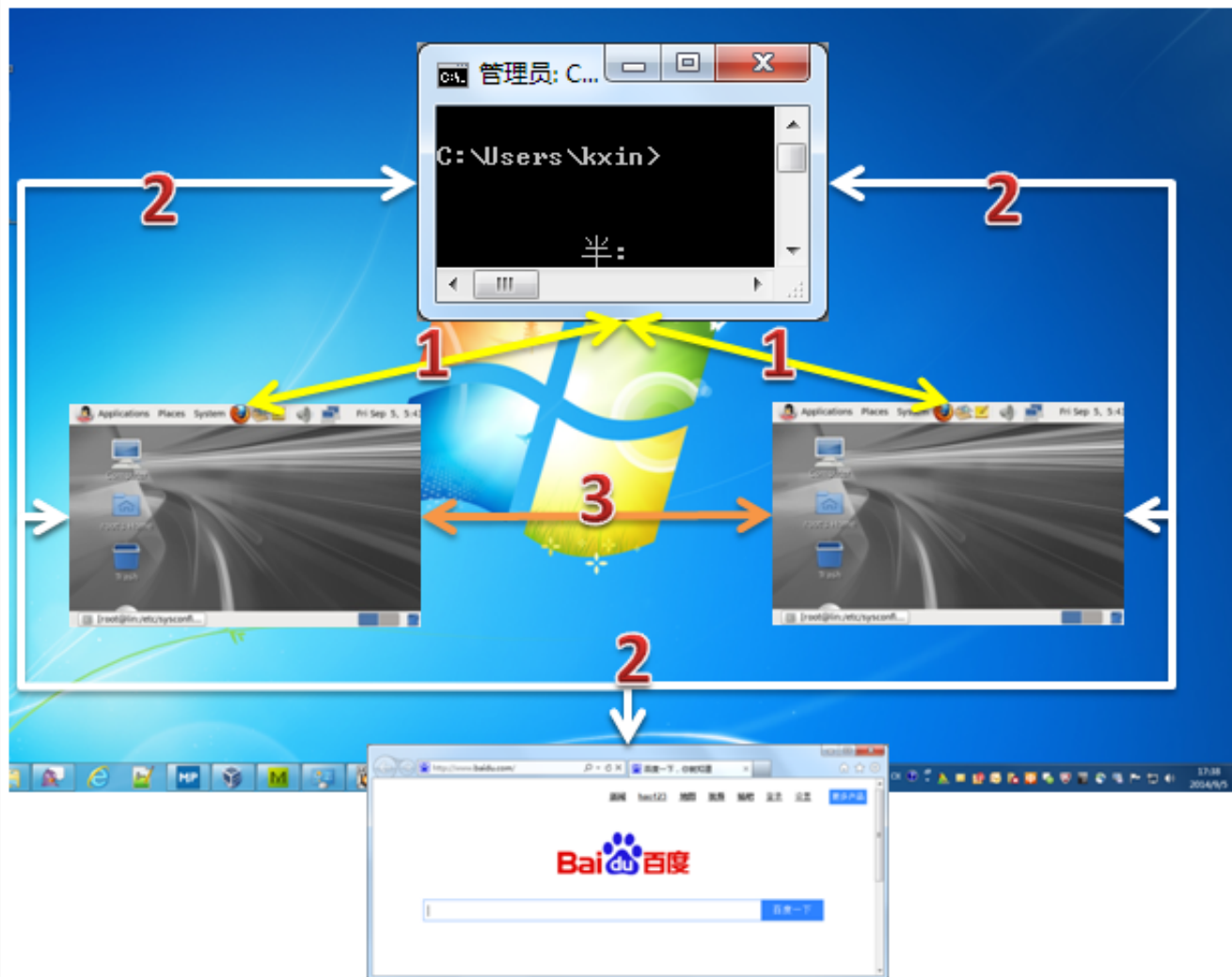
1. 前提条件
2. 环境要求
3. 基本思路
4. 配置主机的内网静态IP
 - 4.1. 配置虚拟机的网卡
 - 4.2. 在虚拟机内部配置网卡
5. 验证
 - 5.1. 对需求1的验证
 - 5.2. 需求2的验证
 - 5.3. 需求3的验证
6. 总结

最近在使用VirtualBox搭建一个实验环境，由于公司规定了所有的机器都不能使用固定IP，都必须由DHCP自动获取。

为了不影响公司整理的网络环境，只能把实验用的网络环境限制在使用内部IP上。因此，这个实验环境必须满足如下的需求（注意这三个需求是AND的关系，必须同时满足）：

- **需求1**：从主机可以通过静态IP访问到每一个虚拟机，从虚拟机中可以访问主机（主机也有一个固定的静态IP）
- **需求2**：虚拟机之间组成一个由静态IP构成的网络，而且虚拟机之间可以互相访问
- **需求3**：从主机，从虚拟机都可以访问internet

类似于这样：



✎ 网络配置图

为了满足以上需求，我在网上找了很多教程，终于有一个叫做 JOB OPEREL 的老哥，给出了一个比较靠谱的解决方法，你可以在 [这里](#) 找到他的文章。

我借鉴了他的方法，但是在连接外网的时候，和他的做法（他用的是桥接连接外网，最终还是需要影响外部网络的；而我用的是NAT地址转换来连接外网，不需要外部网络的，如果你不知道我在这里碎碎念什么，那么，无视吧.....）不一样。废话少说，开始干活！

基本思路

为了满足上面的三个需求，我准备在虚拟机里面加三个网卡来完成，思路如下：

- **eth0**: 网卡1，使用 **仅主机(Host-Only)适配器** 的连接方式，完成 **需求1**，即：从主机可以通过一个静态IP访问到每一个虚拟机，从虚拟机中可以访问主机（主机也有一个固定的静态IP）
- **eth1**: 网卡2，使用 **内部网络** 的连接方式，完成 **需求2**，即：虚拟机之间组成一个由静态IP构成的网络，而且虚拟机之间可以互相访问
- **eth2**: 网卡3，使用 **网络地址转换(NAT)** 的连接方式，完成 **需求3**，即：从主机，从虚拟机都可以访问internet

说了这么多，其实还差一块，那么到底主机与虚拟机之间是怎么连接的呢？

细心的人可能会发现，在我们安装好VirtualBox之后，主机的系统里面会多出一个连接，**VirtualBox Host-Only Network**，这个是VirtualBox自动生成的一个连接；而另外一个 **本地连接** 就是实际的网卡的连接：



✎ VirtualBox Host-Only Network

上图的两个连接，以及虚拟机里面的三块网卡互相配合，组成了下面的网络结构图：



网络结构图

配置主机的内网静态IP

打开 Oracle VM VirtualBox 管理器，从菜单中选择 管理 – 全局设定；在弹出的窗口中选择 网络，然后再选择 仅主机 (Host-Only) 网络 (H) 标签页，双击 VirtualBox Host-Only Ethernet Adapter，修改里面的 IP 地址，为了尊重传统，我选

择192.168.1.1，如下图所示：



✎ 修改里面的IP地址

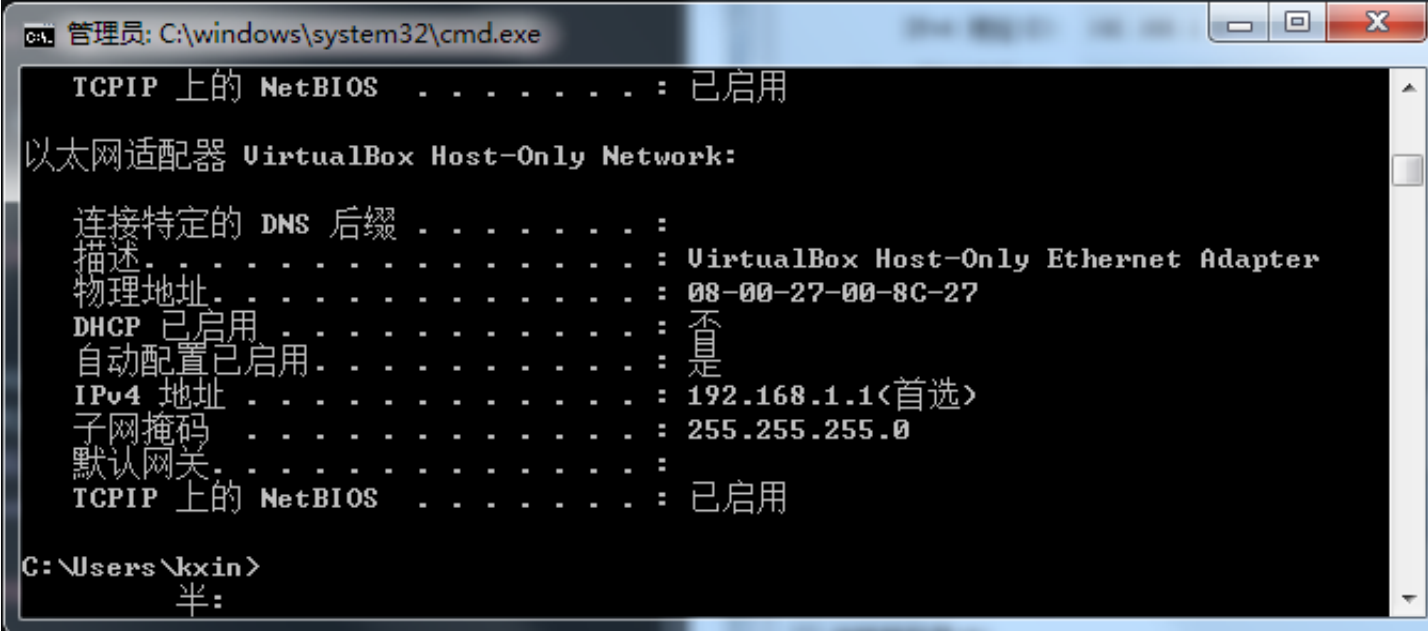
接着，关闭DHCP服务器，如下图所示：



✎ 关闭DHCP服务器

也就是说，现在主机由一个静态IP：192.168.1.1。之后，虚拟机就可以通过这个IP地址来访问主机了。

上述设定完成后，查看主机的地址，可以看到如下的信息：



```
管理员: C:\windows\system32\cmd.exe

TCPIP 上的 NetBIOS . . . . . : 已启用

以太网适配器 VirtualBox Host-Only Network:

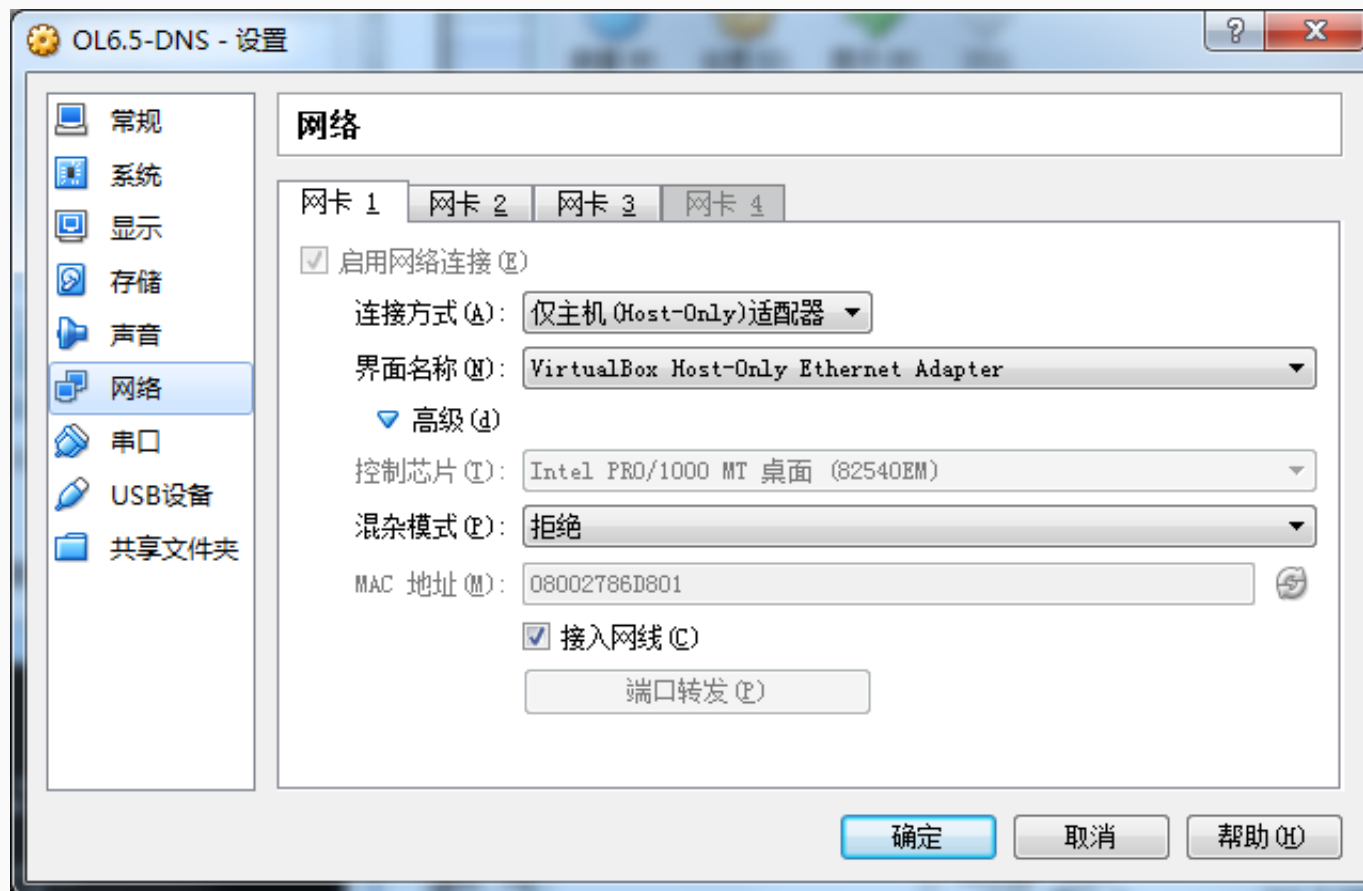
    连接特定的 DNS 后缀 . . . . . :
    描述. . . . . : VirtualBox Host-Only Ethernet Adapter
    物理地址. . . . . : 08-00-27-00-8C-27
    DHCP 已启用 . . . . . : 否
    自动配置已启用. . . . . : 是
    IPv4 地址 . . . . . : 192.168.1.1<首选>
    子网掩码 . . . . . : 255.255.255.0
    默认网关. . . . . :
    TCPIP 上的 NetBIOS . . . . . : 已启用

C:\Users\kxin>
```

✎ 主机的地址

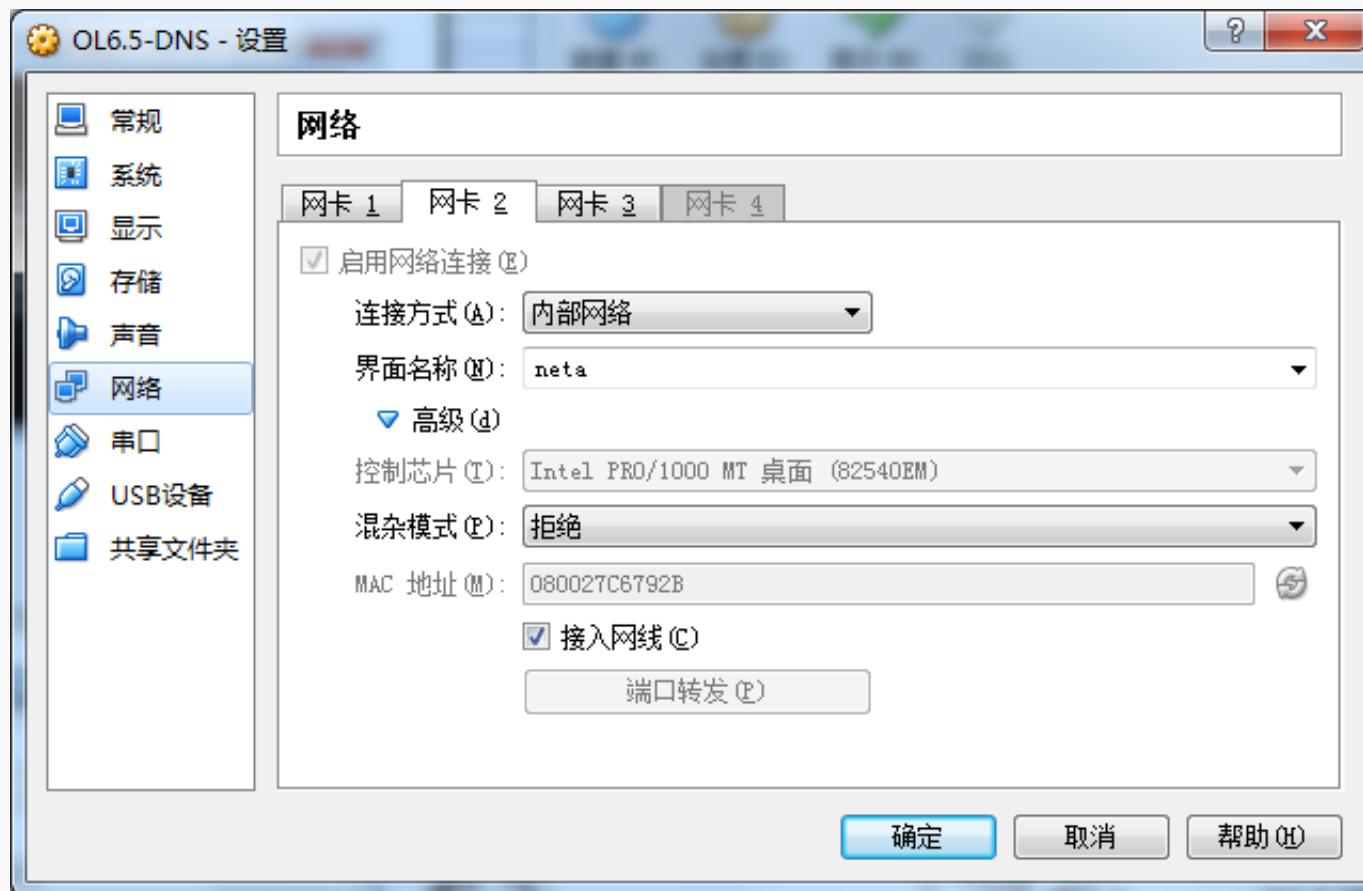
配置虚拟机的网卡

加入第一块网卡 eth0，将 eth0 配置成如下图的方式：



✎ eth0配置

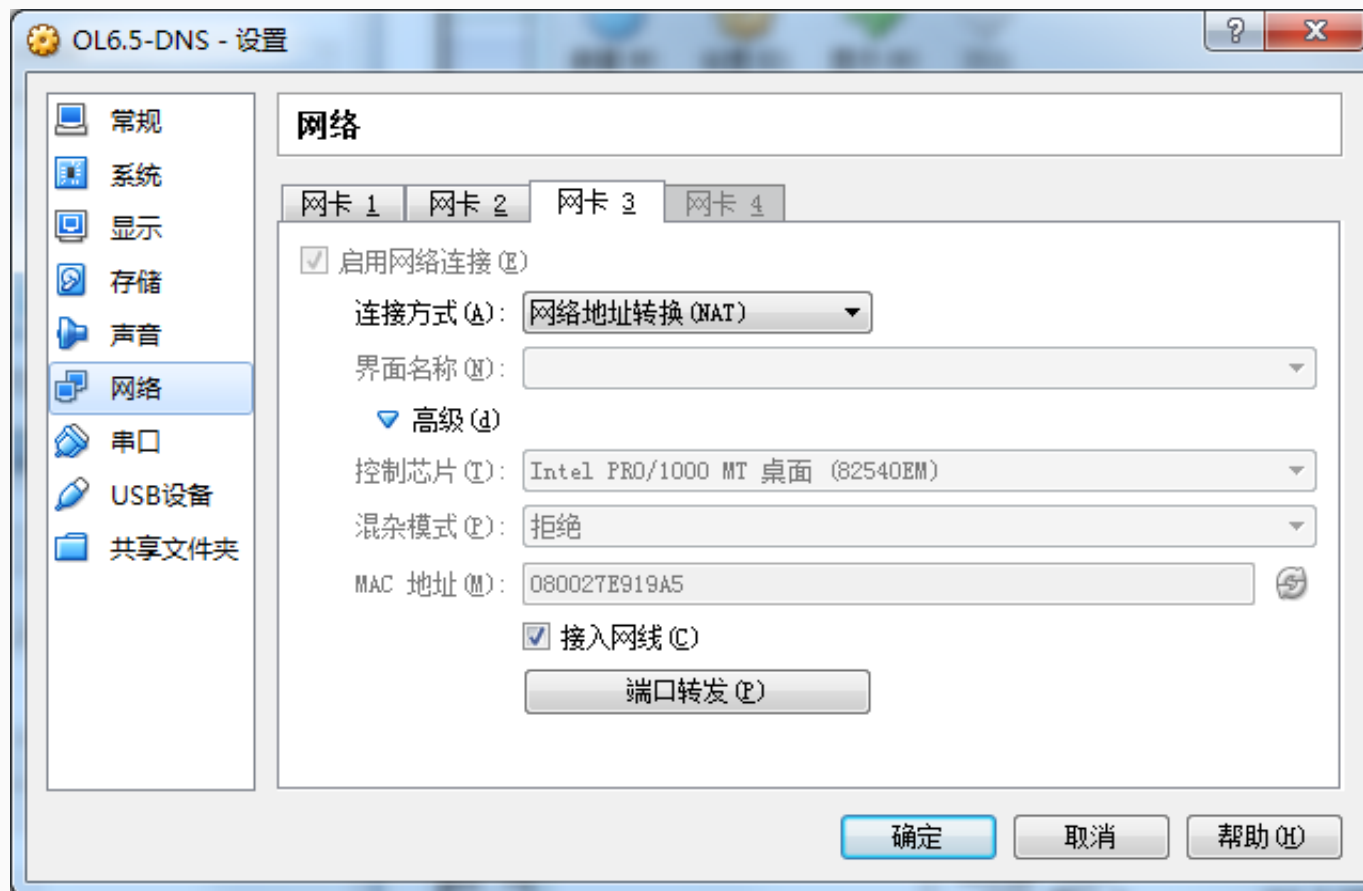
接着加入第二块网卡 eth1，将 eth1 配置成如下图的方式：



✎ eth1配置

注意，虚拟机之间的内网名字可以随便起，这里叫做 neta。如果在配置在第二个虚拟机的时候，也要记得选择 netca 这个内网名字，这样虚拟机之间才可以互相通信。

最后，加入第三块网卡 eth2，配置如下：



✎ eth2配置

保存，启动虚拟机，安装系统。

在虚拟机内部配置网卡

在这里我使用了Oracle Linux（其实和CentOS，RHEL是一毛一样的）。在配置IP的时候，用命令行来配置各个网卡的IP信息，如果你喜欢，也可以使用GUI（NetworkManager）。

首先 eth0，这里要与主机保持的一个网段（192.168.1.），把 eth0 的IP地址配置为 192.168.1.2：

```
1 $ vim /etc/sysconfig/network-scripts/ifcfg-eth0
```

```
2 DEVICE=eth0
3 TYPE=Ethernet
4 ONBOOT=yes
5 NM_CONTROLLED=yes
6 BOOTPROTO=none
7 IPADDR=192.168.1.2 #这里要与主机保持的一个网段：192.168.1.
8 PREFIX=24
9 DEFROUTE=no
10 IPV4_FAILURE_FATAL=yes
11 IPV6INIT=no
12 NAME="System eth0"
13 NETWORK=192.168.1.0
14 BROADCAST=192.168.1.255
15 HWADDR=08:00:27:86:D8:01
16 MTU=1500
17 UUID=5fb06bd0-0bb0-7ffb-45f1-d6edd65f3e03
18 LAST_CONNECT=1409908253
```

其次，配置 eth1，要注意内网我们选择 192.168.0. 这个网段，eth1 的IP地址配置为 192.168.0.2:

```
1 $ vim /etc/sysconfig/network-scripts/ifcfg-eth1
2 DEVICE=eth1
3 TYPE=Ethernet
4 ONBOOT=yes
5 NM_CONTROLLED=yes
6 BOOTPROTO=none
7 IPADDR=192.168.0.2 #内网我们选择 192.168.0\ 这个网段
8 NETMASK=255.255.255.0
9 PREFIX=24
10 DEFROUTE=no
11 IPV4_FAILURE_FATAL=yes
12 IPV6INIT=no
13 NAME="System eth1"
14 HWADDR=08:00:27:C6:79:2B
15 MTU=1500
16 UUID=9c92fad9-6ecb-3e6c-eb4d-8a47c6f50c04
17 LAST_CONNECT=1409908255
```

再次，配置 eth2, 因为是NAT地址转换的，这块网卡使用DHCP来自动获取IP，具体如下：

```
1 $ vim /etc/sysconfig/network-scripts/ifcfg-eth2
2 DEVICE=eth2
3 TYPE=Ethernet
4 ONBOOT=yes
5 NM_CONTROLLED=yes
6 BOOTPROTO=dhcp #注意这里是DHCP
7 NAME="System eth2"
8 HWADDR=08:00:27:E9:19:A5
9 DEFROUTE=yes
10 PEERDNS=yes
11 PEERROUTES=yes
12 IPV4_FAILURE_FATAL=yes
13 IPV6INIT=no
14 UUID=3a73717e-65ab-93e8-b518-24f5af32dc0d
15 LAST_CONNECT=1409908256
```

最后，重新启动一下网络服务：

```
# service network restart
```

查看一下我们配置后的结果：

```
1 $ ip addr show
2 1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
3     link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
4     inet 127.0.0.1/8 scope host lo
5     inet6 ::1/128 scope host
6         valid_lft forever preferred_lft forever
7 2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
8     link/ether 08:00:27:86:d8:01 brd ff:ff:ff:ff:ff:ff
9     inet 192.168.1.2/24 brd 192.168.1.255 scope global eth0
10    inet6 fe80::a00:27ff:fe86:d801/64 scope link
11        valid_lft forever preferred_lft forever
12 3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
13    link/ether 08:00:27:c6:79:2b brd ff:ff:ff:ff:ff:ff
```

```
14      inet 192.168.0.2/24 brd 192.168.0.255 scope global eth1
15      inet6 fe80::a00:27ff:fec6:792b/64 scope link
16          valid_lft forever preferred_lft forever
17      4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
18          link/ether 08:00:27:e9:19:a5 brd ff:ff:ff:ff:ff:ff
19      inet 10.0.4.15/24 brd 10.0.4.255 scope global eth2
20      inet6 fe80::a00:27ff:fee9:19a5/64 scope link
21          valid_lft forever preferred_lft forever
22  ``language
```

也就是说，现在这台虚拟机有三个IP：

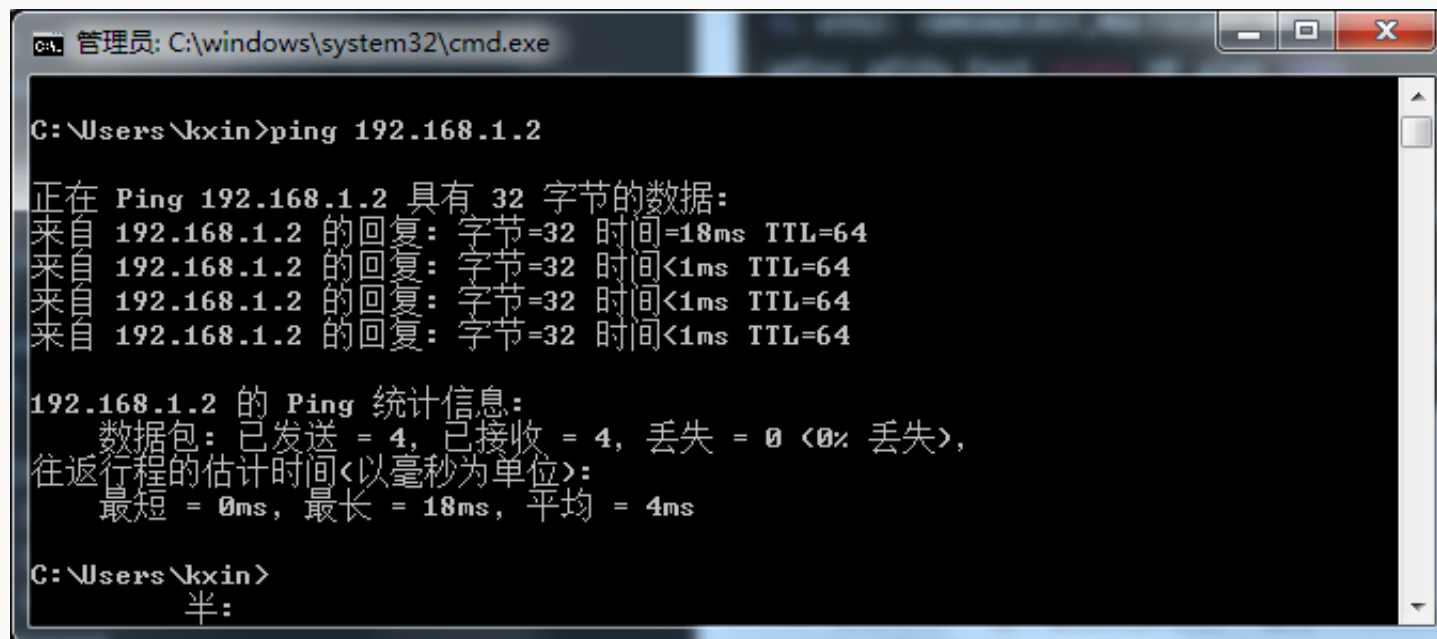
- 192.168.1.2：和主机通信使用
- 192.168.0.2：和其他的虚拟机通信使用
- 10.0.4.15：是自动获取的，通过VirtualBox的地址转换功能，用来访问外网internet

如果再新建一台虚拟机，还是需要参考上面的步骤，为每个虚拟机都设置三块网卡。

验证

对需求1的验证

通过主机来 ping 虚拟机，结果如图，成功！



```
管理员: C:\windows\system32\cmd.exe

C:\Users\kxin>ping 192.168.1.2

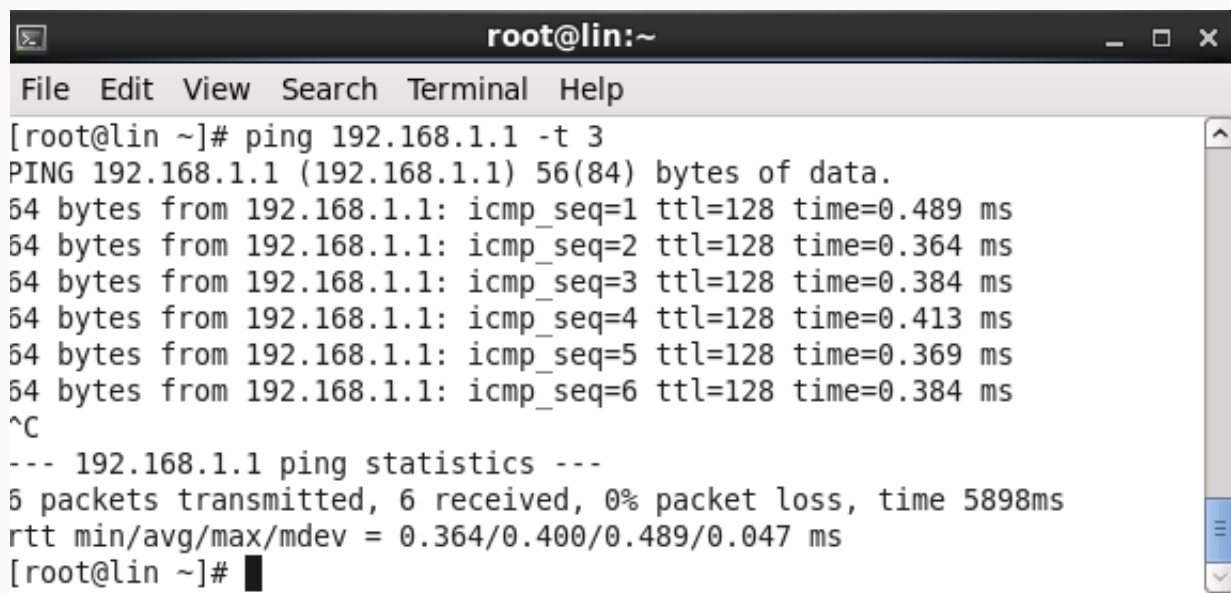
正在 Ping 192.168.1.2 具有 32 字节的数据:
来自 192.168.1.2 的回复: 字节=32 时间=18ms TTL=64
来自 192.168.1.2 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.1.2 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.1.2 的回复: 字节=32 时间<1ms TTL=64

192.168.1.2 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
    往返行程的估计时间<以毫秒为单位>:
        最短 = 0ms, 最长 = 18ms, 平均 = 4ms

C:\Users\kxin>
```

✎ 主机来ping虚拟机

反过来，从虚拟机中ping主机，结果如图，成功！



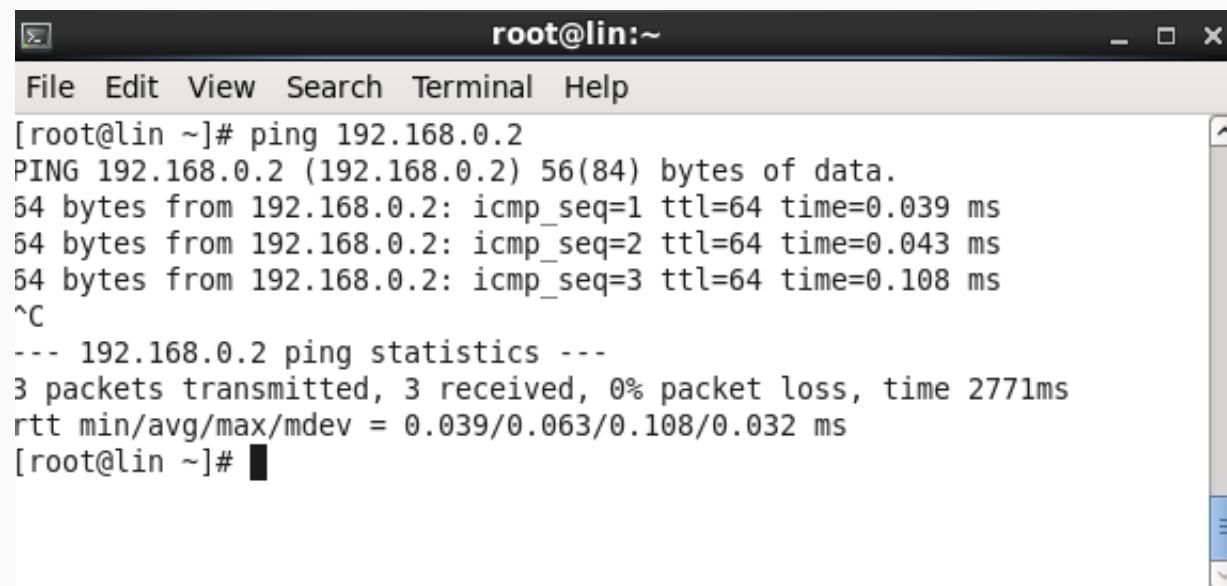
```
root@lin:~
File Edit View Search Terminal Help

[root@lin ~]# ping 192.168.1.1 -t 3
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
 64 bytes from 192.168.1.1: icmp_seq=1 ttl=128 time=0.489 ms
 64 bytes from 192.168.1.1: icmp_seq=2 ttl=128 time=0.364 ms
 64 bytes from 192.168.1.1: icmp_seq=3 ttl=128 time=0.384 ms
 64 bytes from 192.168.1.1: icmp_seq=4 ttl=128 time=0.413 ms
 64 bytes from 192.168.1.1: icmp_seq=5 ttl=128 time=0.369 ms
 64 bytes from 192.168.1.1: icmp_seq=6 ttl=128 time=0.384 ms
^C
--- 192.168.1.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5898ms
rtt min/avg/max/mdev = 0.364/0.400/0.489/0.047 ms
[root@lin ~]#
```

✎ 从虚拟机中ping主机

需求2的验证

虚拟机之间互 ping，结果如图，成功！



```
root@lin:~  
File Edit View Search Terminal Help  
[root@lin ~]# ping 192.168.0.2  
PING 192.168.0.2 (192.168.0.2) 56(84) bytes of data.  
64 bytes from 192.168.0.2: icmp_seq=1 ttl=64 time=0.039 ms  
64 bytes from 192.168.0.2: icmp_seq=2 ttl=64 time=0.043 ms  
64 bytes from 192.168.0.2: icmp_seq=3 ttl=64 time=0.108 ms  
^C  
--- 192.168.0.2 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2771ms  
rtt min/avg/max/mdev = 0.039/0.063/0.108/0.032 ms  
[root@lin ~]#
```

✎ 虚拟机之间互ping

需求3的验证

从虚拟机直接上外网，如图，成功！（注意：请确认你的主机和虚拟机都使用了相同的代理服务器设置）

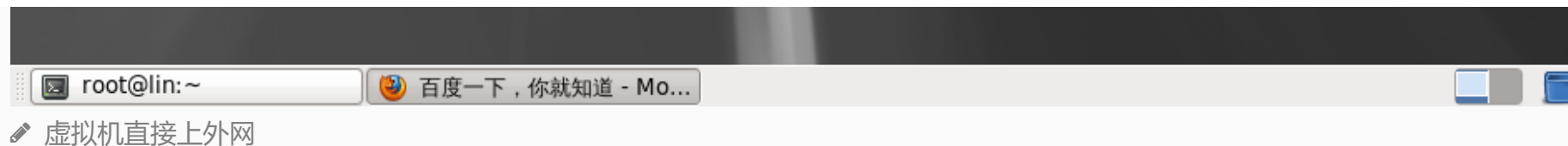


[新闻](#) [hao123](#) [地图](#) [视频](#) [贴吧](#) [登录](#) [设置](#) [更多产品](#)



[把百度设为主页](#) [关于百度](#) [About Baidu](#)

©2014 Baidu [使用百度前必读](#) 京ICP证030173号



总结

通过把 VirtualBox 的几种联网方式的混合运用，就能满足我们大多场合的需求。其实回过头想想，如果一种解决方法解决不了问题，可以尝试一下混合几种解决方案，取长补短，这样带来的加乘效果，往往会让你吃惊。

(END)

虚拟化  dhcp ip virtualbox 网络配置 虚拟机

上一篇：

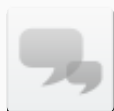
◀ Oracle Linux下内网DNS服务器的配置

下一篇：

▶ Virtualbox中找不到eth0的解决方法

1条评论

最新 最早 最热



anonymous

呵呵😊

3月28日 ◀ 回复 ❤ 顶 ➡ 转发

社交帐号登录:  微信  微博  QQ  人人 [更多»](#)



发布

夕阳下的奔跑正在使用多说

分类

Mac ¹

WEB基础 ²

java ²

linux ¹²

内家功 ²

数据库 ³

文本处理 ³

虚拟化 ¹⁵

随笔 ⁷

标签云

12c ORA-00838 ORA-00845 acrobat ad adobe aip application provision application virtulization archlinux array bind cdm centos cjk code createrepo database
debian demo dhcp django dn timer em12c eth0 fc fcitx fdisk firewall gateway github gs gvim ha heapsort hostname ip iptable java jni kpi latex ldap linux lv

lvm mac markdown mysql nameserver netfilter nslookup oam octpress oem ome oms oracle oracle linux oracle12c osx ovdi packagekit pdf pdfreader poc
poem prim proxmox pvecm python rdp reddhat redhat remote repolist reposync rhce rmi
samba sbc selinux sgd sgd5.2 sizing smb ssh ssl sso swig tarantella tds tex ubuntu vbox vdi vi vim virtual desktop virtualbox virtualization vpn web
webgate windows windows server yum 专用网 企业管理器 估算 协议 单点登录 合并 图论 域名 堆
堆排序 安全访问 带宽 应用程序供应 应用程序虚拟化 指标 数据库 替换 本地接口 模板 正则表达式 比较 水质 环境需求 算法 经验 编码 网络性能 网络配置 脆弱性
自签名 自适应协议 虚拟化 虚拟机 虚拟桌面 认证 论文 身份认证 逻辑卷 部署 配置 防火墙 阵列 集群 鱼缸

友情链接

开源中国

Oracle SGD

 RSS 订阅



Hello ,I'm K.X, working for Oracle as Senior Sales Consultant.
This is my blog, believe it or not.

© 2016 K.X