

使用 VirtualBox 虚拟机应用

搭建 redis 存储系统

集群测试环境

真实机器环境

系统：Windows7 旗舰版

CPU: Celeron 赛扬双核处理器及以上

内存：4G及以上

系统类型：64 位操作系统

使用软件说明

VirtualBox 虚拟机

用途：构建虚拟机的运行平台

版本：4.3.24 用于 64 位 win7 系统



VirtualBox 虚拟机
安装.docx

安装步骤：请参见：

Vagrant 管理软件

用途：管理 VirtualBox 虚拟机的命令库

版本：1.7.4



Windows上安装 Vag
rant.docx

安装步骤：请参见：

整体搭建步骤

1. 使用 Vagrant 命令构建 VirtualBox 虚拟机

离线添加 VagrantBox

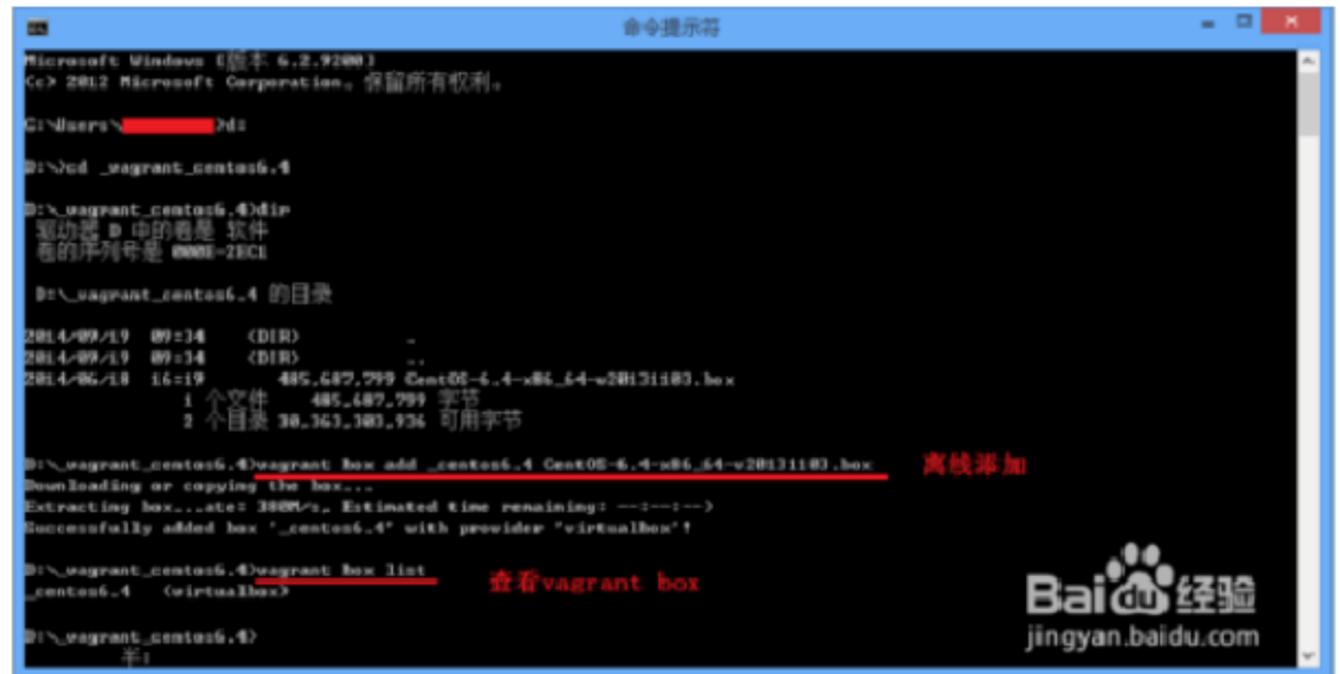
Vagrant 添加 VagrantBox 有两种方式：离线添加与在线添加。演示离线添加方式：在 F 盘创建 xuni_vdev1 目录。

- 1) 将目录中文件 CentOS-6.4-x86_64-v20131103.box 复制到 F:\xuni_vdev1 目录
- 2) 离线添加 VagrantBox ，命令语法：
vagrant box add {title} {box} ，如下图。

title : 为添加的 VagrantBox 进行重命名, 命名方式不限

box : 为使用的 VagrantBox 文件

3) 查看 Vagrant box, 输入命令: `vagrant box list`。



```
Microsoft Windows [版本 6.2.9200]
C:\> cd D:\Users\<redacted>\Documents

D:\> cd _vagrant_centos6.4

D:\_vagrant_centos6.4> dir
驱动器 B 中的卷是 软件
卷的序列号是 880E-2ECC

D:\_vagrant_centos6.4 的目录

2014/09/19 09:34 <DIR>          .
2014/09/19 09:34 <DIR>          ..
2014/06/18 16:19                485,687,799 CentOS-6.4-x86_64-w20131103.box
                1 个文件      485,687,799 字节
                2 个目录  30,363,383,936 可用字节

D:\_vagrant_centos6.4> vagrant box add _centos6.4 CentOS-6.4-x86_64-w20131103.box 离线添加
Downloading or copying the box...
Extracting box...ate: 3000%. Estimated time remaining: --:--:--
Successfully added box '_centos6.4' with provider 'virtualbox'

D:\_vagrant_centos6.4> vagrant box list 查看vagrant box
_centos6.4 (virtualbox)

D:\_vagrant_centos6.4>
```

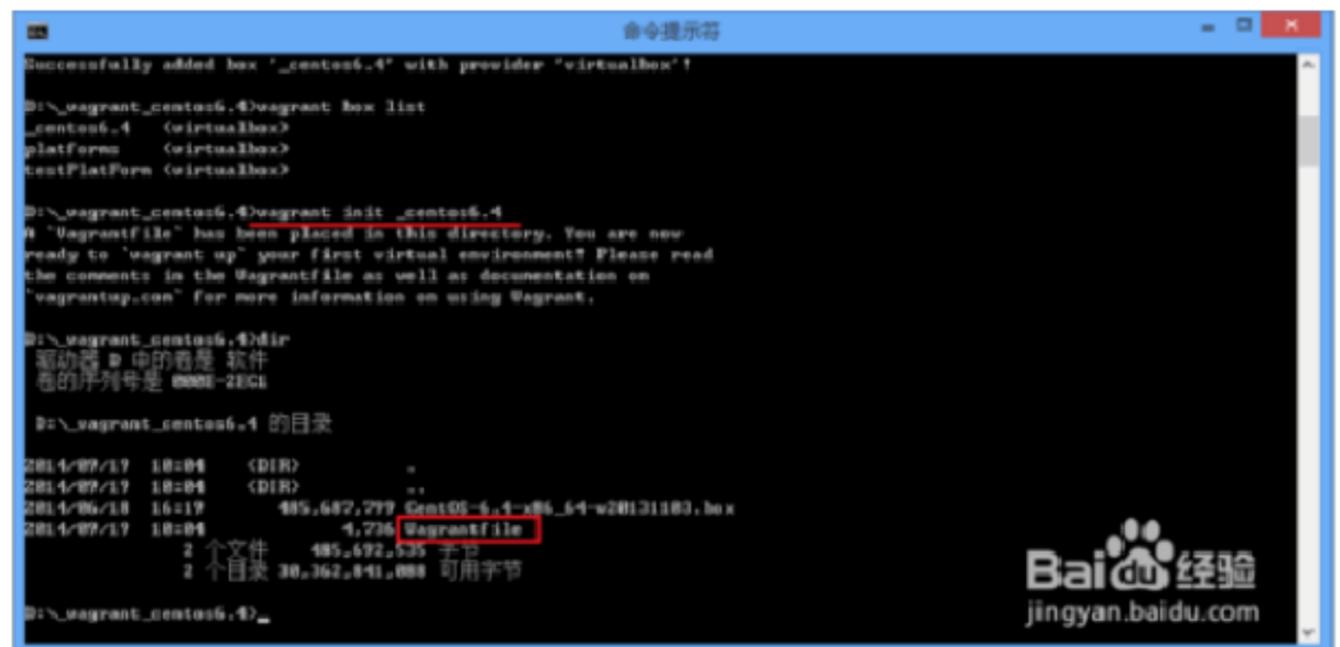
初始化 Vagrant 工程, 以 “_centos6.4” 为例

初始化完成后会生成 Vagrantfile 文件, Vagrantfile 是配置 vagrant 的重要文件。如: 配置网络、端口、ssh 等。

1) 初始化 Vagrant 工程, 命令语法: `vagrant init {title}`, 如下图。

title: 为 Vagrantfile 中 config.vm.box 重命名, title 可略则默认的 config.vm.box= “base”。

2) 查看当前目录是否存在 Vagrantfile 文件。



```
Successfully added box '_centos6.4' with provider 'virtualbox'

D:\_vagrant_centos6.4> vagrant box list
_centos6.4 (virtualbox)
platform (virtualbox)
testPlatform (virtualbox)

D:\_vagrant_centos6.4> vagrant init _centos6.4
# "Vagrantfile" has been placed in this directory. You are now
ready to 'vagrant up' your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
"vagrantup.com" for more information on using Vagrant.

D:\_vagrant_centos6.4> dir
驱动器 B 中的卷是 软件
卷的序列号是 880E-2ECC

D:\_vagrant_centos6.4 的目录

2014/09/17 18:04 <DIR>          .
2014/09/17 18:04 <DIR>          ..
2014/06/18 16:19                485,687,799 CentOS-6.4-x86_64-w20131103.box
2014/09/17 18:04                1,736 Vagrantfile
                2 个文件      485,692,535 字节
                2 个目录  30,362,841,088 可用字节

D:\_vagrant_centos6.4>
```

启动虚拟机 “_centos6.4”

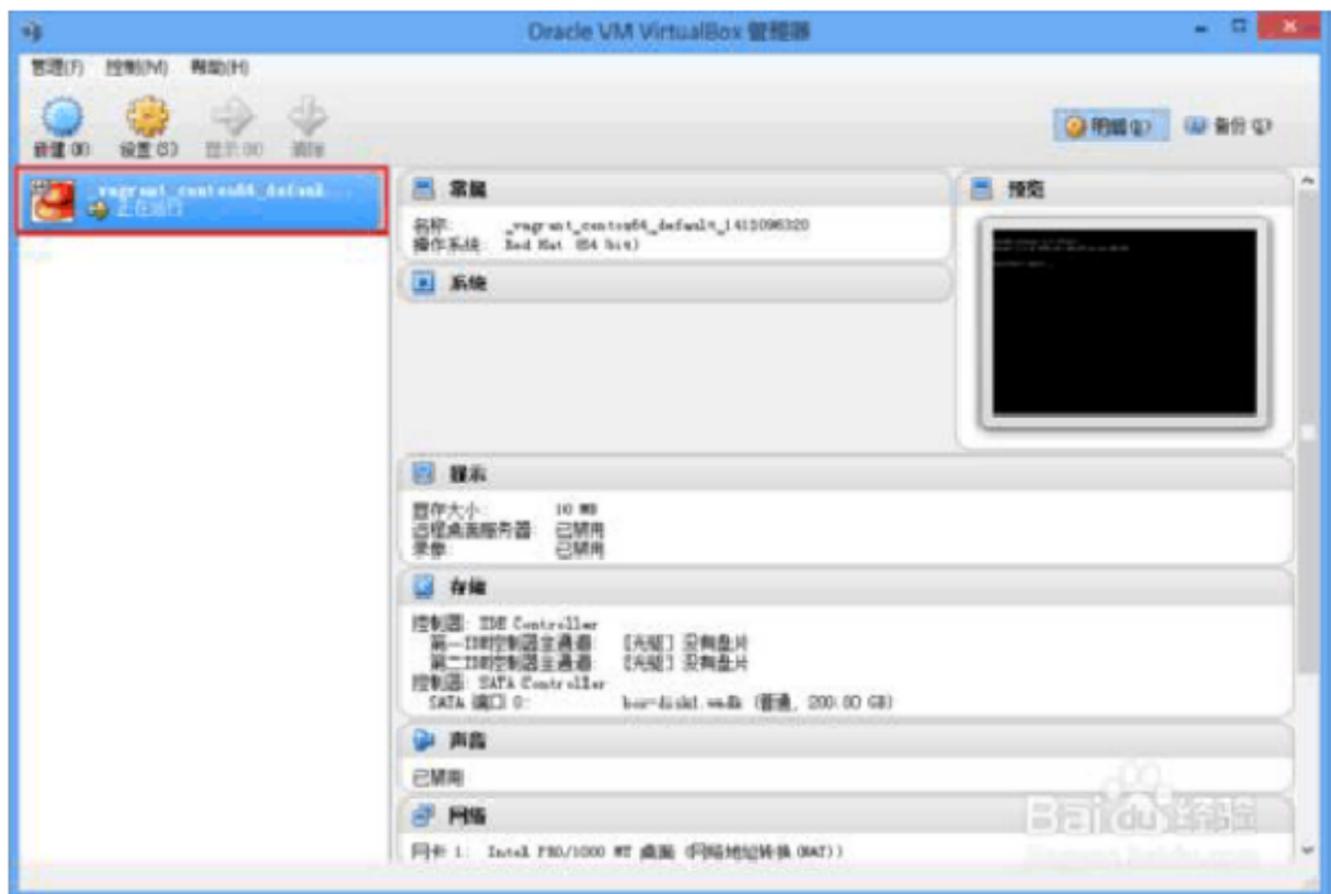
启动: `vagrant up` 关闭: `vagrant halt` 重启: `vagrant reload`

注: vagrant 命令需在对应虚拟机目录下运行才有效。

```
命令提示符
D:\_vagrant_centos6.4>dir
2014/07/17  18:04    <DIR>        .
2014/07/17  18:04    <DIR>        ..
2014/06/18  16:17          485,687,777 CentOS-6.4-x86_64-v20121102.box
2014/07/17  18:04           4,726 Vagrantfile
                2 个文件  485,692,503 字节
                2 个目录  38,362,841,088 可用字节

D:\_vagrant_centos6.4>vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
[default] Importing base box '_centos6.4'...
[default] Matching NIC address for NAT networking...
[default] Setting the name of the VM...
[default] Clearing any previously set forwarded ports...
[default] Creating shared folders metadata...
[default] Clearing any previously set network interfaces...
[default] Preparing network interfaces based on configuration...
[default] Forwarding ports...
[default] -- 22 => 2222 (adapter 1)
[default] Booting VM...
[default] Waiting for machine to boot. This may take a few minutes...
[default] Machine booted and ready!
[default] Mounting shared folders...
[default] -- /vagrant
D:\_vagrant_centos6.4>
```

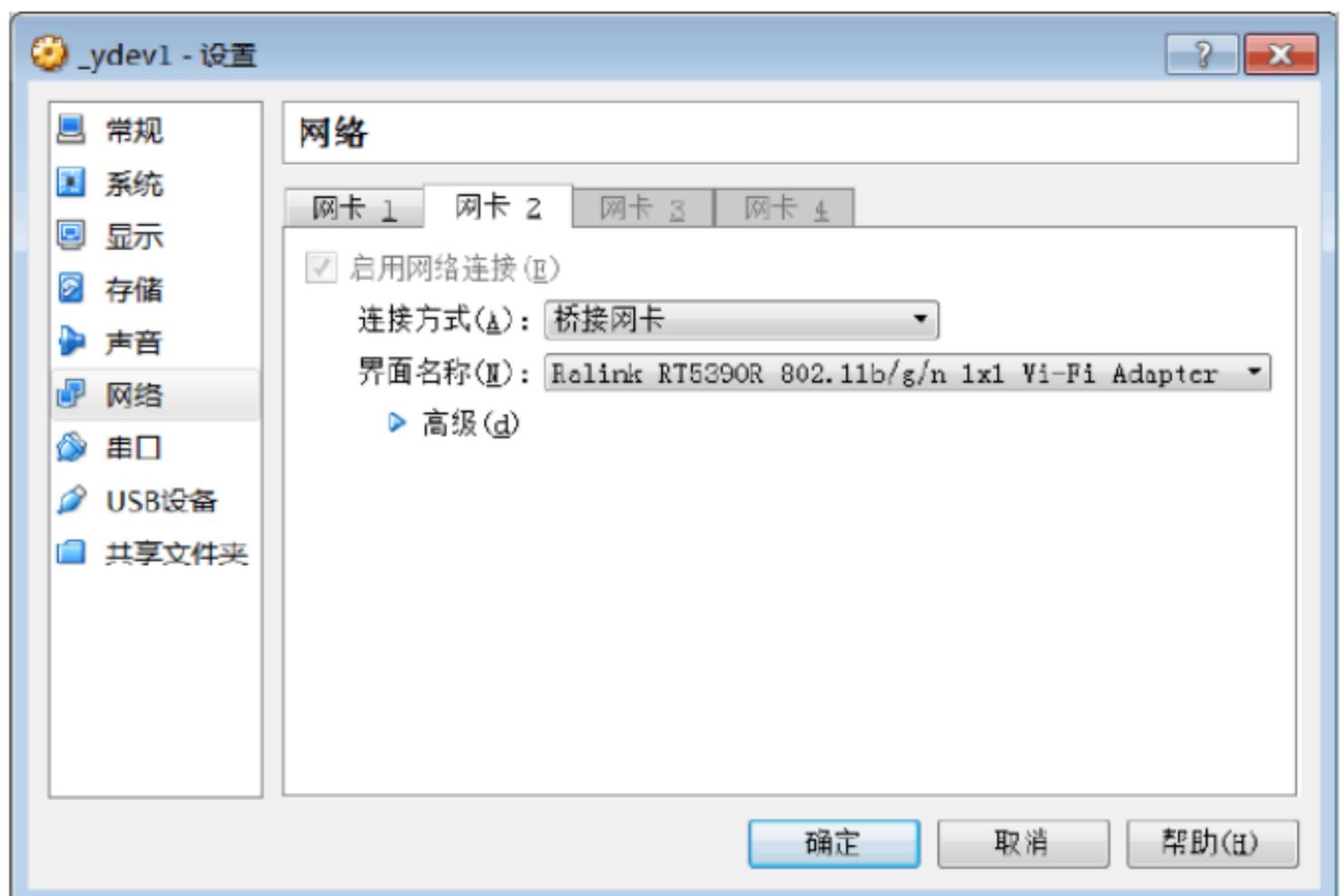
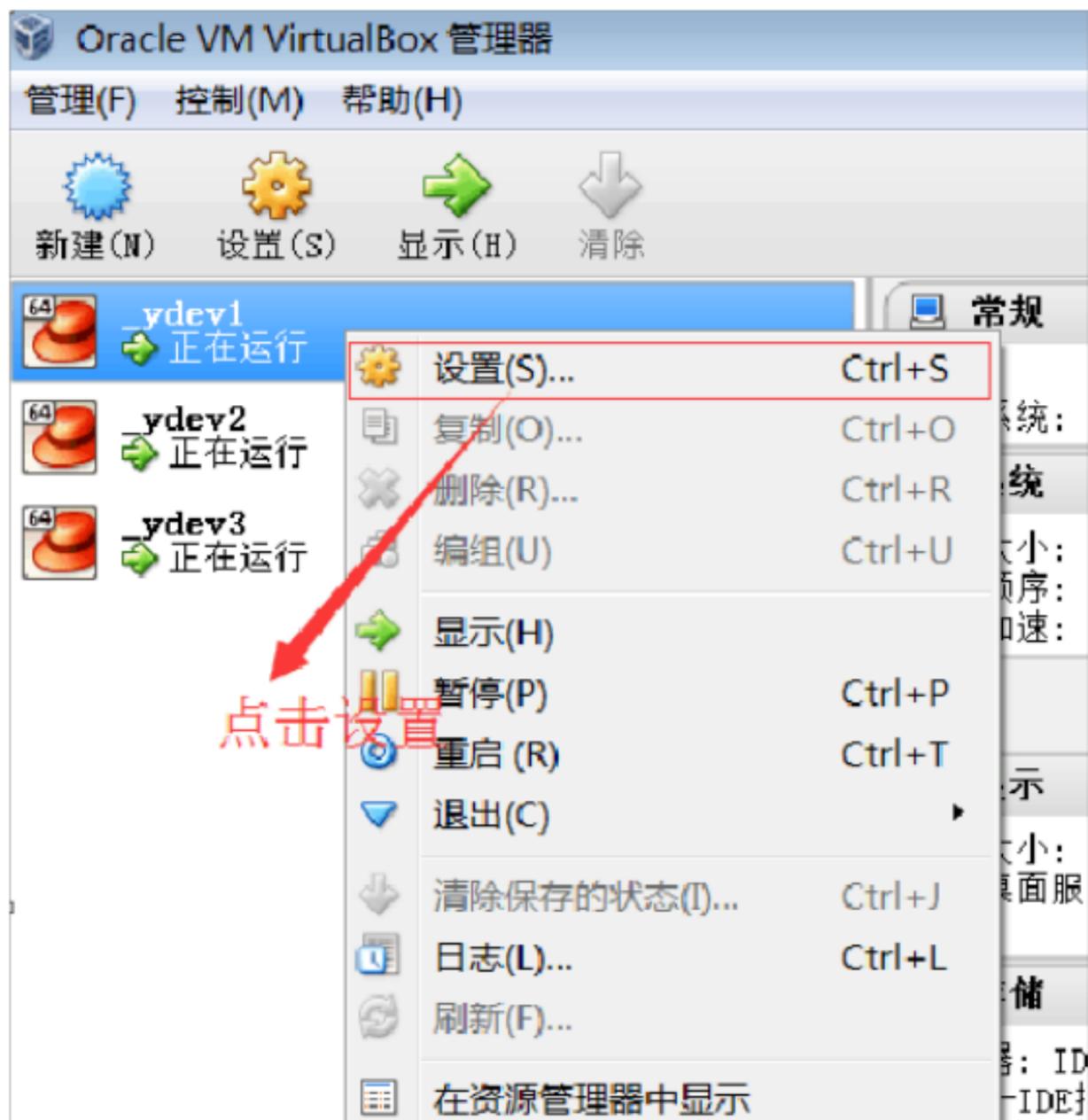
运行 VirtualBox, 查看虚拟机现状



配置虚拟机 “_centos6.4 ” 网络环境实现内外互联

由于要实现内外互联，采用桥接模式，在这种模式下，VirtualBox 虚拟出来的操作系统就像是局域网中的一台独立的主机，它可以访问网内任何一台机器。在桥接模式下，你需要手工为虚拟系统配置 IP 地址、子网掩码，而且还要和宿主机处于同一网段，这样虚拟系统才能和宿主机进行通信。同时，由于这个虚拟系统是局域网中的一个独立的主机系统，那么就可以手工配置它的 TCP/IP 配置信息，以实现通过局域网的网关或路由器访问互联网。

步骤：在 VirtualBox 中设置网络连接，开启桥接网卡连接模式



启动 _centos6.4 , 发现 /etc/sysconfig/network-scripts 多了一个未配置的网卡 ifcfg-eth1 文件, vi 编辑该文件, 重启网络 service network restart , 如下图所示

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

[root@localhost network-scripts]# vi ifcfg-eth1

DEVICE="eth0"
BOOTPROTO="static" 静态网址
IPV6INIT="yes"
NM_CONTROLLED="no" yes 激活网卡
ONBOOT="yes"
TYPE="Ethernet" ip地址要和真实机在一个网段上
IPADDR=172.16.0.193 子网掩码
NETMASK=255.255.255.0 网关
GATEWAY=172.16.0.254
#UUID="7249a920-7c1d-4fb5-a910-db264545d7c6"

[root@localhost network-scripts]# vi ifcfg-eth1
[root@localhost network-scripts]# service network restart
```

2. 在虚拟机中安装 redis 系统

下载 redis 版本是 2.6.16

```
cd /home/
wget http://download.redis.io/releases/redis-2.6.16.tar.gz
```

解压到当前目录

```
tar -zxvf redis-2.6.16.tar.gz
```

安装

```
make PREFIX=/home/redis-2.6.16 install

cd redis-2.6.16/ utils

ln -s /home/redis-2.6.16/src/redis-sentinel /usr/local/bin/ #添加到环境变量中

./install_server.sh #执行这个文件他会默认生成一些 redis 的配置文件

##### 如果执行这脚本的时候发生下面的错误 #####

[root@localhost utils]# ./install_server.sh
```

```

Welcome to the redis service installer

This script will help you easily set up a running redis server

Please select the redis port for this instance : [ 6379] #redis 配置的端口

Selecting default : 6379

Please select the redis config file name [ /etc/ redis / 6379.conf ]

#redis 默认文件存放路径

Selected default - /etc/ redis / 6379.conf

Please select the redis log file name [ /var/ log / redis_6379 .log ]

#redis 日志文件路径

Selected default - /var/ log / redis_6379 .log

Please select the data directory for this instance [ /var/ lib / redis / 6379 ]

Selected default - /var/ lib / redis / 6379

Please select the redis executable path [ /usr/ local / bin / redis - server ]

s#^port [ 0- 9]{ 4} $#port 6379 #; s#^logfile .+ $#logfile / var / log / redis_6379 .
log #; s#^dir .+ $#dir / var / lib / redis / 6379 #; s#^pidfile .+ $#pidfile / var / ru
n/ redis_6379 .pid #; s#^ daemonize no$ #daemonize yes #;

Copied / tmp / 6379.conf => / etc / init . d/ redis_6379

Installing service ...

./ install_server .sh: line 178: update - rc .d: command not found

exists , process is already running or crashed

```

注意，你安装的时候，有可能出现的 bug

```

##### 因为 install_server.sh 这个安装脚本有 bug#####
#####

vim install_server .sh

## 第一个点

### 修改前

- if [ ! `which chkconfig` ] ; then

### 修改后

```

```

+if [ ! `which chkconfig` ] ; then

    #combine the header and the template (which is actually a static footer)

    echo $REDIS_INIT_HEADER > $TMP_FILE && cat $INIT_TPL_FILE >> $TMP_FILE || die "Could not write init script to $TMP_FILE"

else

## 第二个点

### 修改前

- if [ ! `which chkconfig` ] ; then

### 修改后

+if [ ! `which chkconfig` ] ; then

    #if we're not a chkconfig box assume we're able to use update-rc.d

    update-rc.d redis_$REDIS_PORT defaults && echo "Success!"

else

### 修改完成后，再次执行，应该没问题的了

```

启动 redis

```

cd /home/redis-2.6.16

redis-server redis.conf

redis-cli

[root@localhost utils]# redis-cli

redis 127.0.0.1:6379> set name yyc

OK

redis 127.0.0.1:6379> get name

"yyc"

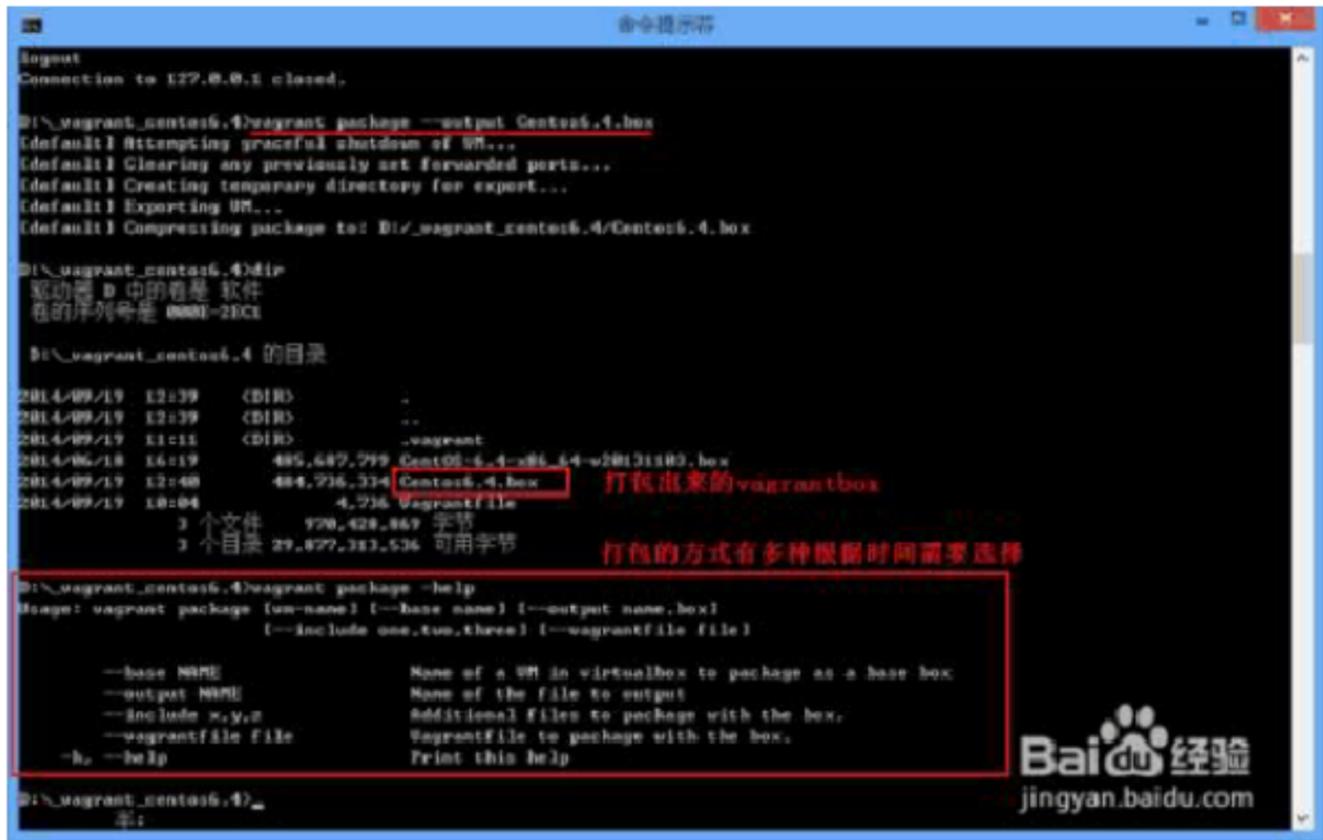
redis 127.0.0.1:6379>

安装完成

```

3. 打包虚拟机

打包虚拟机输入命令：`vagrant package --output Centos6.4.box`

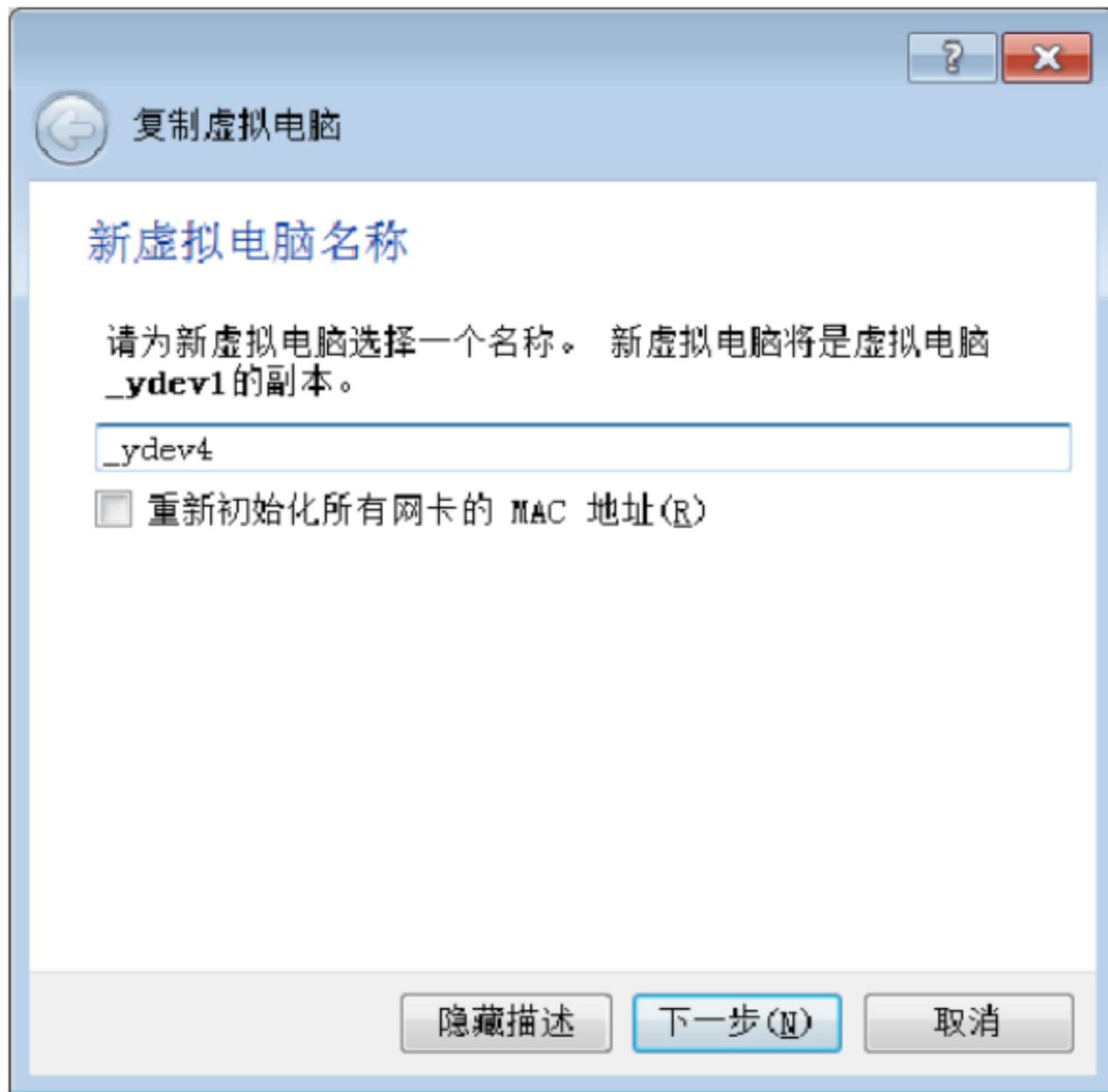


4. 克隆虚拟机，实现多机部署

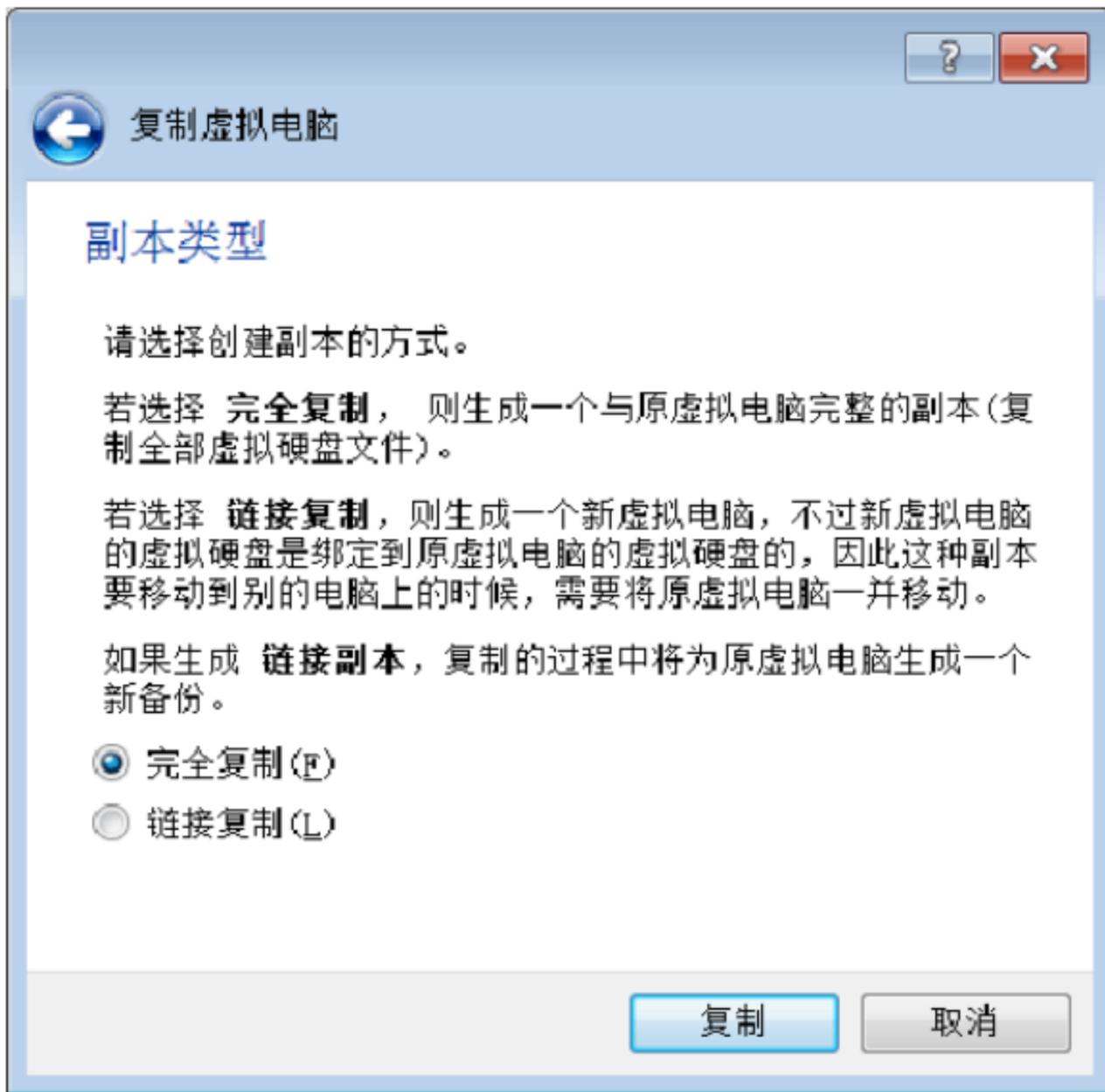
右键虚拟机“_ydev1”点击复制



编辑虚拟机名称，点击下一步



选择完全复制，点击复制，实现克隆。



5. 配置主从和集群监听服务

集群配置最少需要三台机器，那么我就三台虚拟机，三台虚拟机分别安装同样的 redis 的环境

ip 分别：

172.16.0.192(redis sentinel 集群监控)

172.16.0.193 (redis 主)

172.16.0.194 (redis 从)

redis 配置：

主的 redis 配置文件，使用默认的配置就可以，如果你需要设计其他参数

从的 redis 配置文件，添加

```
#从的 redis 配置文件，需要添加
vi /home/redis-2.6.16 / redis.conf
slaveof 172.16 .0.193 6379
```

启动之前请把机器的防火墙清除一下 iptables -F

启动主从 redis

```
#启动主 redis ( 172.16 . 0.193 )
cd /home/redis-2.6.16/
redis-server redis.conf

#启动从 redis ( 172.16 . 0.194 )
cd /home/redis-2.6.16/
redis-server redis.conf
```

查看主 redis 信息

```
#查看主 redis 的信息
redis -cli -h 172.16 . 0.193 info Replication

# Replication
role : master #代表 172.16 . 0.193 : 6379 这台 redis 是主
connected_slaves : 1
slave0 : 172.16 . 0.193 , 6379, online
```

查看从 redis 信息

```
#查看主 redis 的信息
redis -cli -h 172.16 . 0.194 info Replication

# Replication
role : slave #代表 172.16 . 0.193 : 6379 这台 redis 是主
master_host : 172.16 . 0.193
master_port : 6379
master_link_status : up
master_last_io_seconds_ago : 4
master_sync_in_progress : 0
slave_priority : 100
slave_read_only : 1
connected_slaves : 0
```

配置 redis sentinel 集群监控服务

1. 添加一份 redis sentinel 配置文件

```
vi /home/redis-2.6.16/ / sentinel .conf

##redis - 0
##sentinel 实例之间的通讯端口
port 26379

#master1
sentinel monitor master1 172.16 . 0.193 6379 1
sentinel down - after - milliseconds master1 5000
sentinel failover - timeout master1 900000
sentinel can - failover master1 yes
sentinel parallel - syncs master1 2
```

```
#master2 可以添加多组主从的 redis 监听
...
..
..
```

2. 有配置文件了，那么启动 redis sentinel 做 redis 集群监听

```
redis -sentinel sentinel .conf -- sentinel
```

6. 正常演示

把主的 redis 启动

把从的 redis 启动

把 redis sentinel 集群监听启动

观察 redis sentinel 日志信息

这里很清楚地看到，从的 redis 加入了集群

```
[ 4925] 15 Oct 03:42:21.889 * +slave slave 172.16.0.194:6379 172.16.0.194 6379 @
master1 172.16.0.193 6379
```

执行以下命令，查看 redis 主从信息

```
[root@localhost vagrant]# redis -cli -h 172.16.0.192 -p 26379 info Sentinel
# Sentinel
sentinel_masters :1
sentinel_tilt :0
sentinel_running_scripts :0
sentinel_scripts_queue_length :0
master0 : name=master1 , status =ok, address =172.16.0.193 :6379, slaves =1, sentinels =1
```

那么表示一切都正常了。你的 redis sentinel 集群已经配置成功！

7. 故障演示

执行以下命令使用主的 redis 服务停止

```
redis -cli -h 172.16.0.193 -p 6379 shutdown #表示把 172.16.0.193 这台 redis 关闭
```

关闭后，我们再查看 redis sentinel 的日志情况

redis sentinel 监控到主的 redis 服务停止，然后自动把从的 redis 切换到主。

再执行以下命令，查看 redis 主从信息

```
[root@localhost vagrant]# redis -cli -h 172.16.0.192 -p 26379 info Sentinel
# Sentinel
sentinel_masters :1
sentinel_tilt :0
sentinel_running_scripts :0
sentinel_scripts_queue_length :0
master0 : name=master1 , status =ok, address =172.16.0.194 :6379, slaves =1, sentinels =1
```

把从已经升为主了。那么自动切换就已经成功了！

当我们将发现，一台 redis 发生故障了，可能会收到一些故障信息，那么再把服务已关闭的 redis 恢复服务状态，会发生什么样的情况呢？

```
[ root@localhost vagrant ] 15 Oct 03:59:18.615 * +slave slave 172.16.0.193:6379 172.16.0.193:6379 @master1 172.16.0.194:6379
```

redis sentinel 集群服务，会把上次主 redis 重新加入服务中，但是他再以不是主 redis 了，变成从的 reids 。