

## Task2 操作指南

### 1. 前提提要

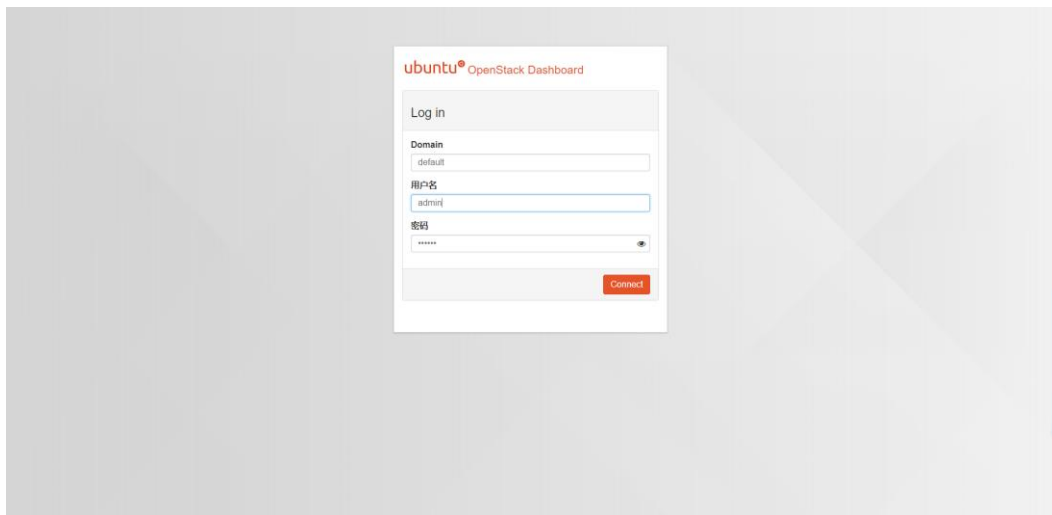
Task1 我们使用脚本部署了 Openstack 在大家的虚拟机上，Task2 我们主要是使用 Openstack，以及网络的设置。

在 Task1 中，部署 Openstack 的虚拟机需要两块网卡，我们把一块设为“**NAT 模式**”，另一块设为了“**仅主机模式**”，并且在安装中设定控制节点以及内部网络为“NAT 模式”的网卡，而“**仅主机模式**”的网卡设为**外部网络**的接口。但实际上，你们将这两块网卡的交换，将 NAT 设为外部网络的网卡，仅主机模式的网卡设为内部网络及控制节点网络也是可以的。在文章的后面，我也会给出这 2 种方式对应的外部网络设置方法。（注意，已经安装好 Openstack 的虚拟机你就不要尝试更换两个网卡的模式，不然可能需要重新装 Openstack 或重新配置）

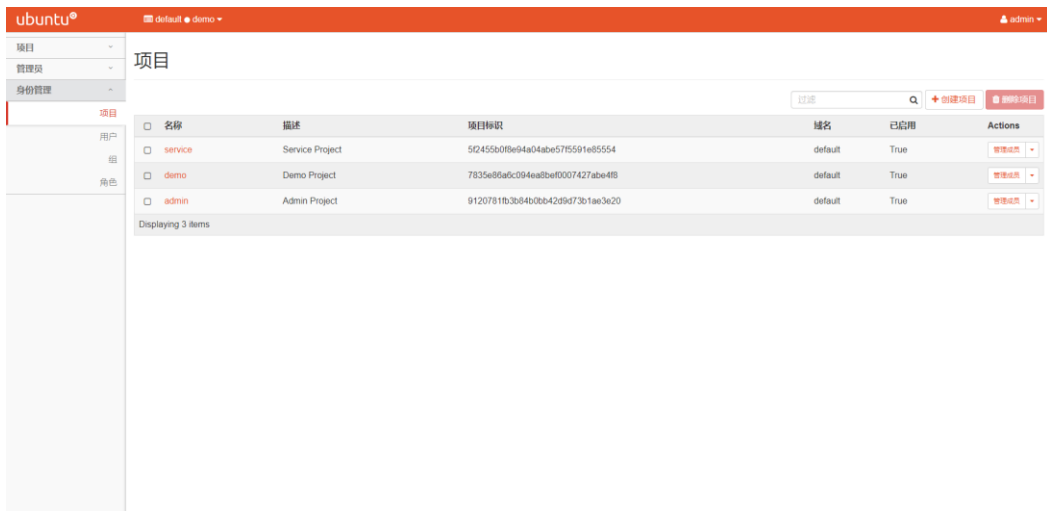
如果大家是“仅主机模式”网卡作为外部网络网卡的话，就按下面的教程操作即可，没必要去尝试以“NAT 模式”为外部网络网卡，当然，有兴趣有时间的同学可以尝试。

### 2. 登陆 Dashboard

首先根据你在部署 Openstack 时设定控制节点的 IP 地址，如 192.168.182.132，在浏览器中输入 dashboard 的访问地址，如 192.168.182.132:8088/horizon。加载出如下网站：



其中 domain 为 default，用户名为 admin，密码为你部署 Openstack 时所设置的密码。点击 connect 按钮，登录进入 Dashboard。



### 3. 查看系统信息

点击左侧导航栏管理员栏目，再点击系统信息，查看 Openstack 启用了哪些服务。

The screenshot shows the OpenStack dashboard interface. On the left sidebar, the 'Admin' (管理员) menu is expanded, and 'System Information' (系统信息) is highlighted. The main content area displays the 'System Information' page. At the top, there are tabs for 'Services' (服务), 'Compute Services' (计算服务), and 'Network Proxy Services' (网络代理服务). Below these tabs is a table listing the services running on the system.

名称	服务	主机	状态
keystone	identity (native 后端)	192.168.182.132	已启用
neutron	network	192.168.182.132	已启用
glance	image	192.168.182.132	已启用
nova	compute	192.168.182.132	已启用

Below the table, it says 'Displaying 4 items'. In the bottom right corner, the version '版本: 9.1.2' is displayed.

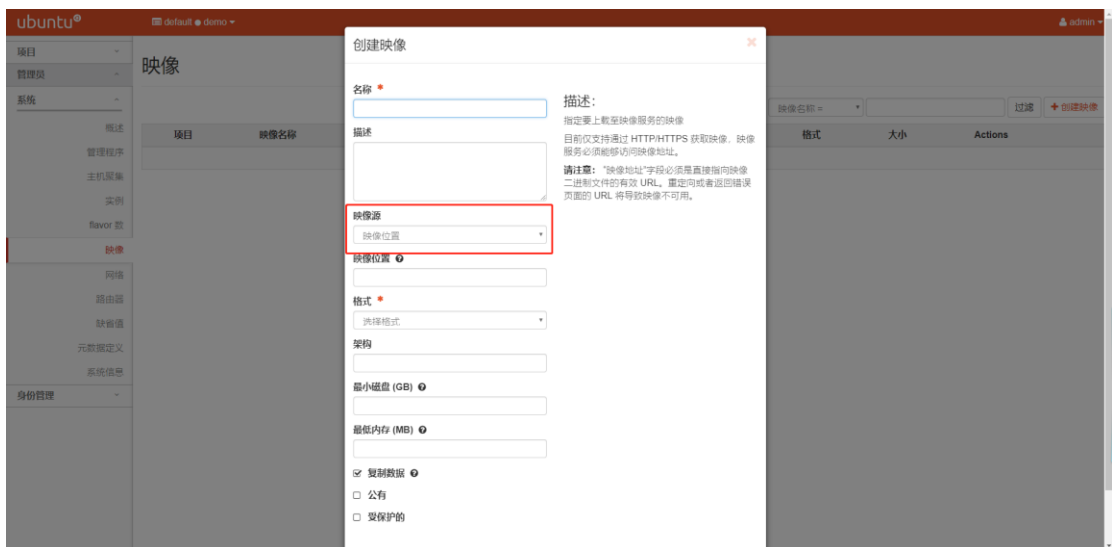
### 4. 上传 cirros 镜像

点击左侧导航栏管理员栏目中的映像选项，进入镜像管理界面

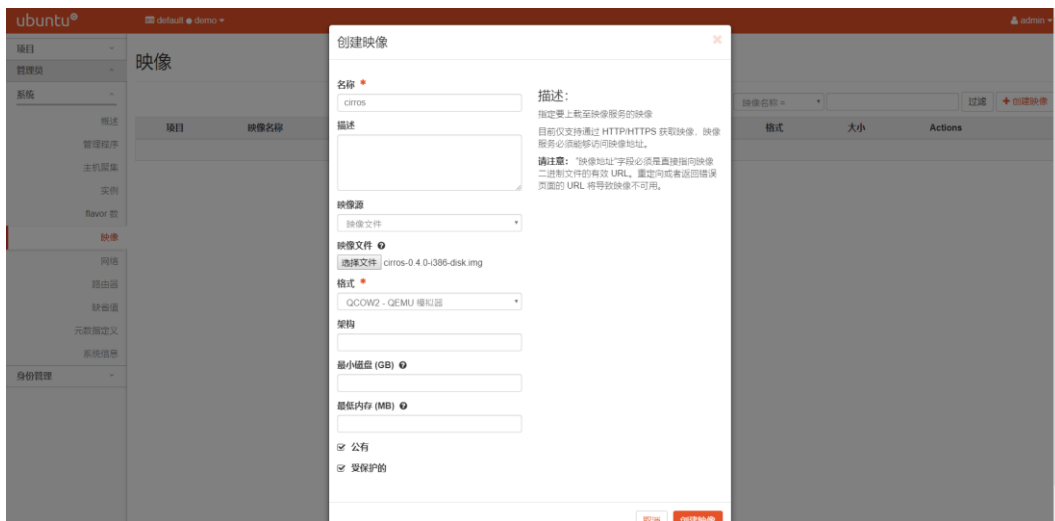
The screenshot shows the OpenStack dashboard interface. On the left sidebar, the 'Admin' (管理员) menu is expanded, and 'Images' (映像) is highlighted. The main content area displays the 'Images' management page. At the top, there is a search bar and a '+ Create Image' (创建映像) button. Below this is a table with columns: 'Project' (项目), 'Image Name' (映像名称), 'Type' (类型), 'Status' (状态), 'Public' (公有), 'Protected' (受保护的), 'Format' (格式), 'Size' (大小), and 'Actions'.

项目	映像名称	类型	状态	公有	受保护的	格式	大小	Actions
No items to display.								

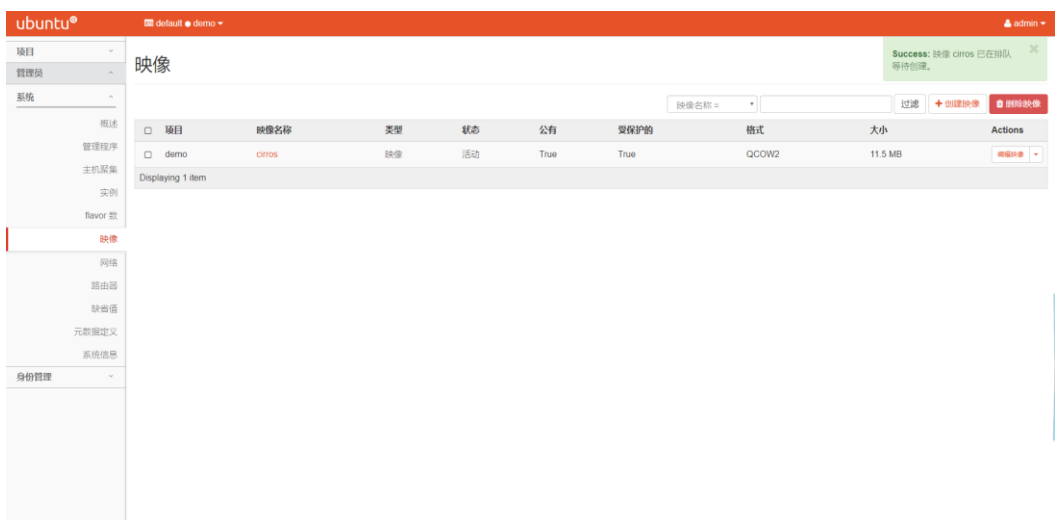
点击创建映像按钮，弹出添加映像的对话框，更改映像源的来源，从“映像位置”改为“映像文件”。



输入映像的详细信息，并上传群文件中的 cirros 镜像。注意镜像格式为 QEMU

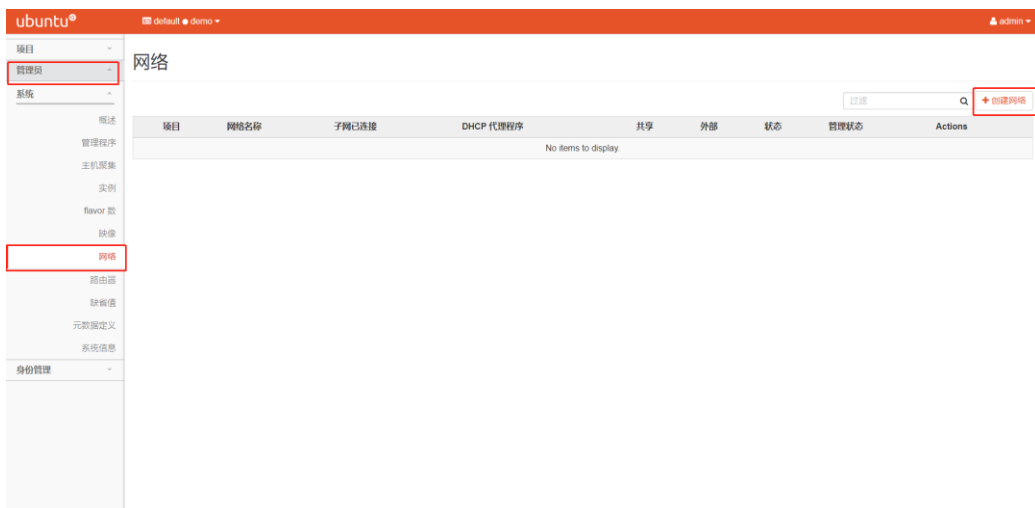


点击创建镜像，可以看到映像列表中多了一条 cirros 镜像的记录

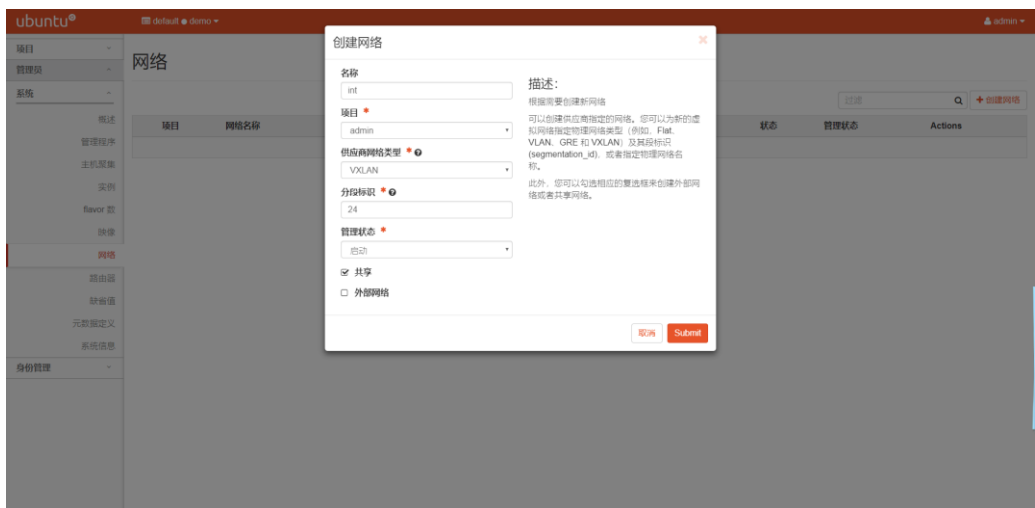


## 5. 创建一个内部网络

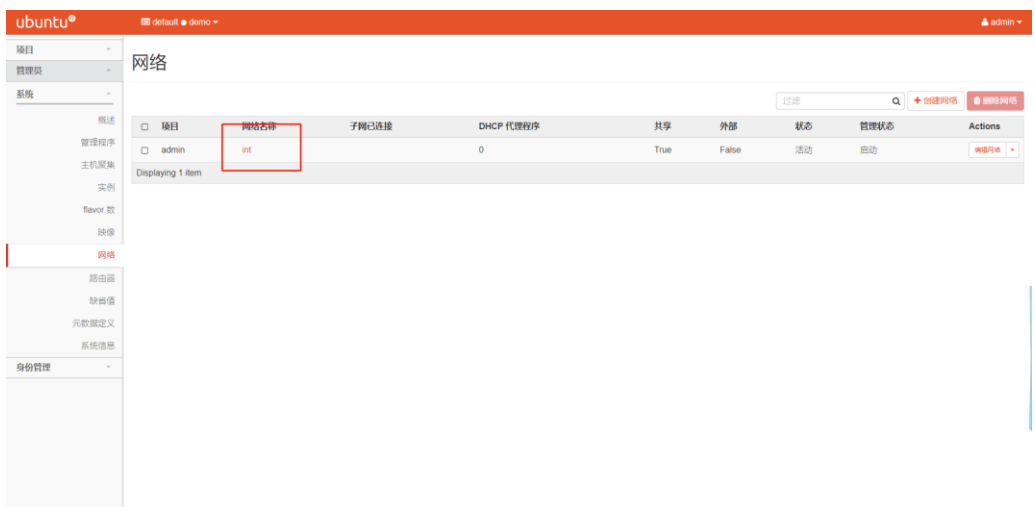
点击左侧导航栏管理员栏目中的网络选项，进入网络管理页面



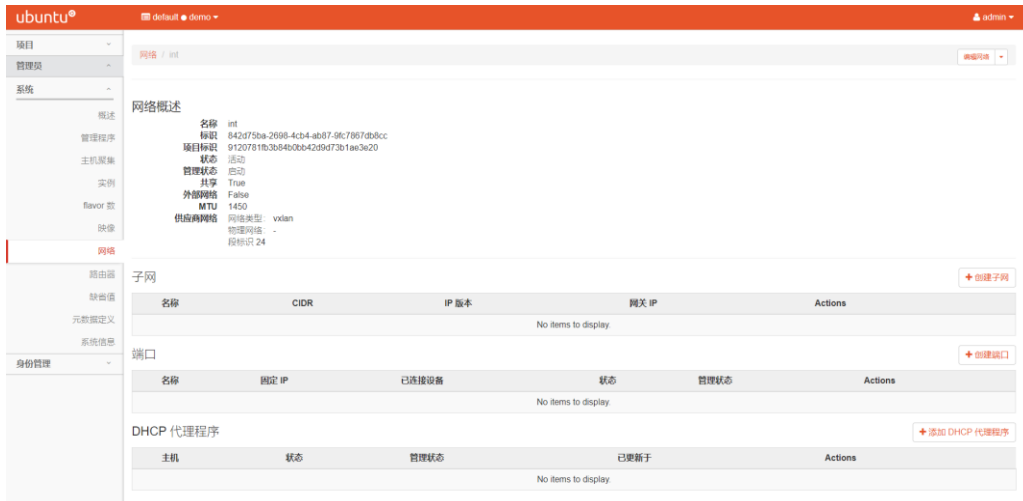
点击创建网络，弹出创建网络对话框，输入想要创建的网络的相关信息（注意这里我们所创建的内部网络，使用 VXLAN 网络类型，所以需要设置一个分段标识）



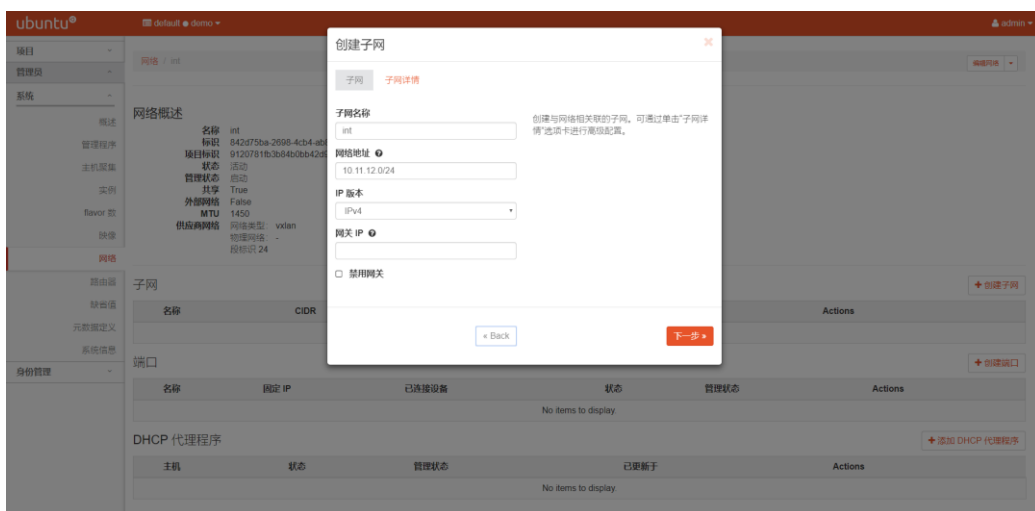
点击 Submit 提交信息，可以看到创建了一条 int 网络的记录



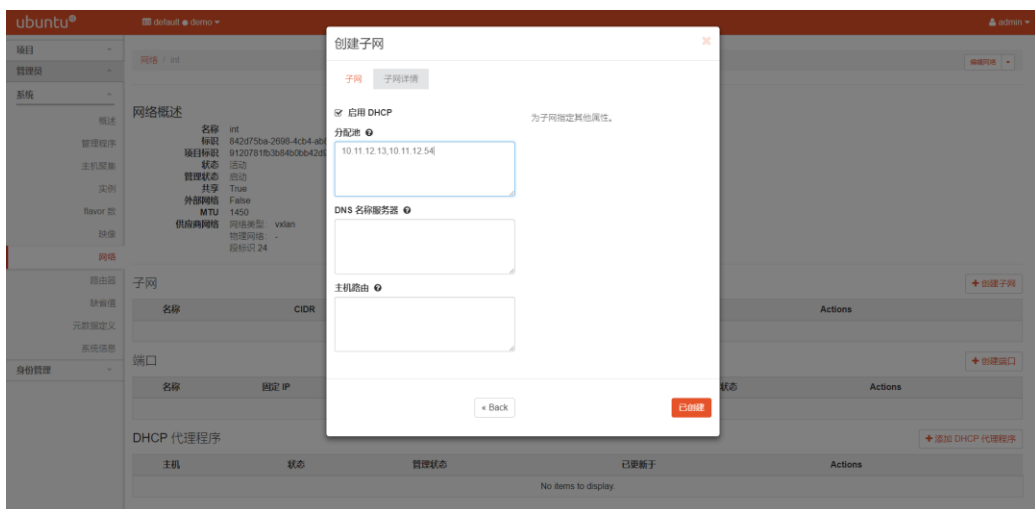
点击 int 记录中的网络名称“int”，进入网络的设置页面



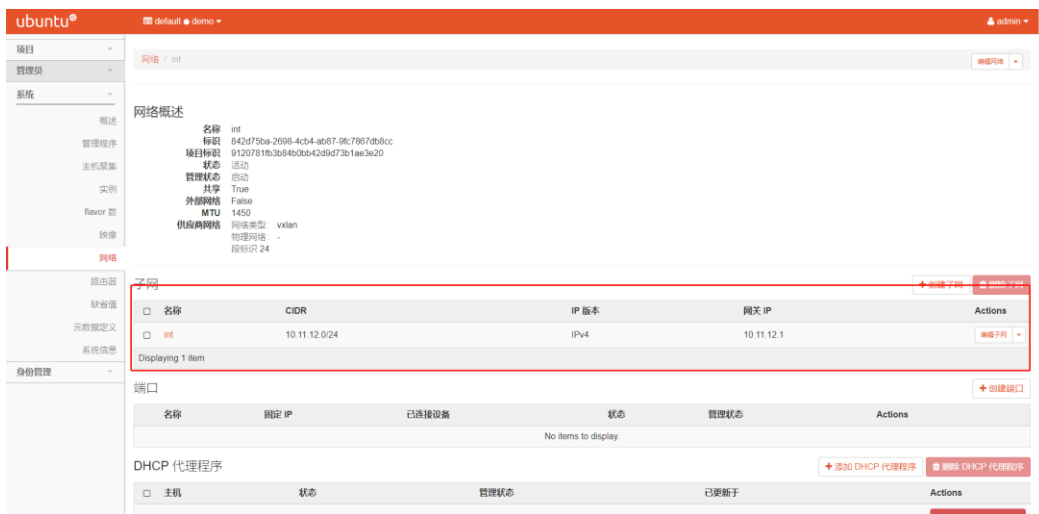
点击创建子网，弹出创建子网的对话框，这里我们可以创建任意的子网（即 10.0.0.0 -- 10.255.255.255，172.16.0.0 – 172.31.255.255，192.168.0.0 – 192.168.255.255 这三个内网网段），输入相关信息，注意这里网关 IP 如果留白，则会把对应网段的第一个 ip 地址设为网关，如 10.11.12.1，所以后面设置子网的分配池，注意冲突。



点击下一步后，可以选择设置子网的分配池，即可用的 IP 集合



点击已创建，我们可以看到成功创建了一个子网

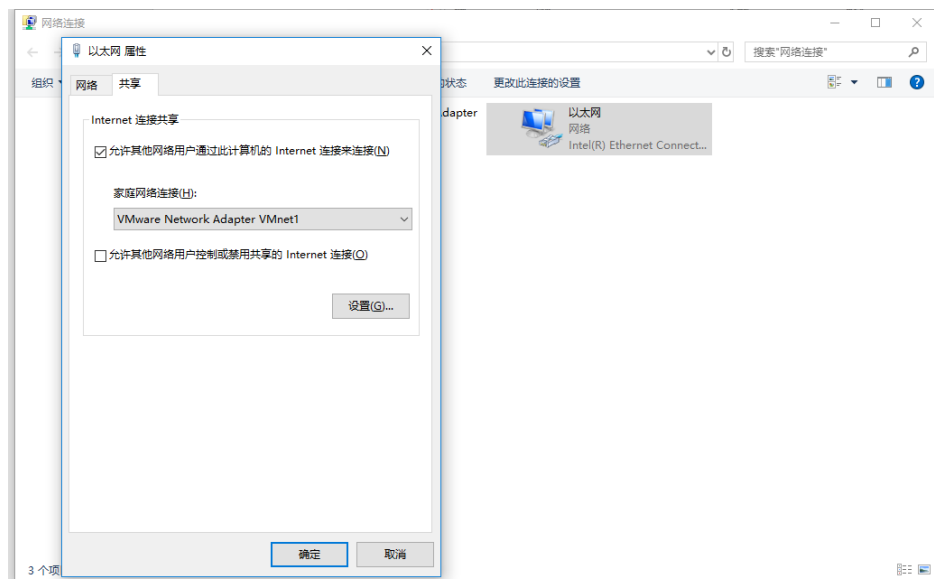


## 6. 创建一个外部网络

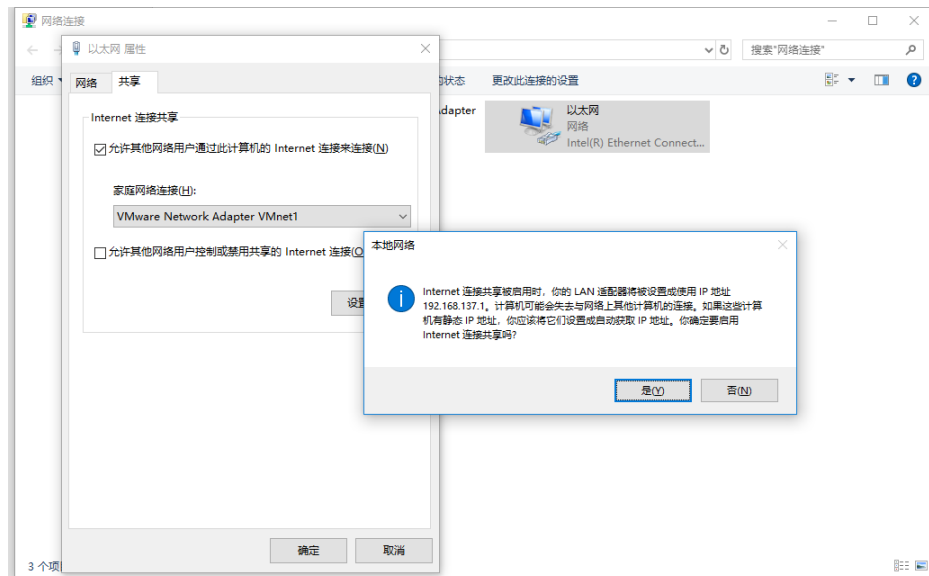
这里我们分两种情况

- (1) 仅主机网络网卡作为外部网络的网卡

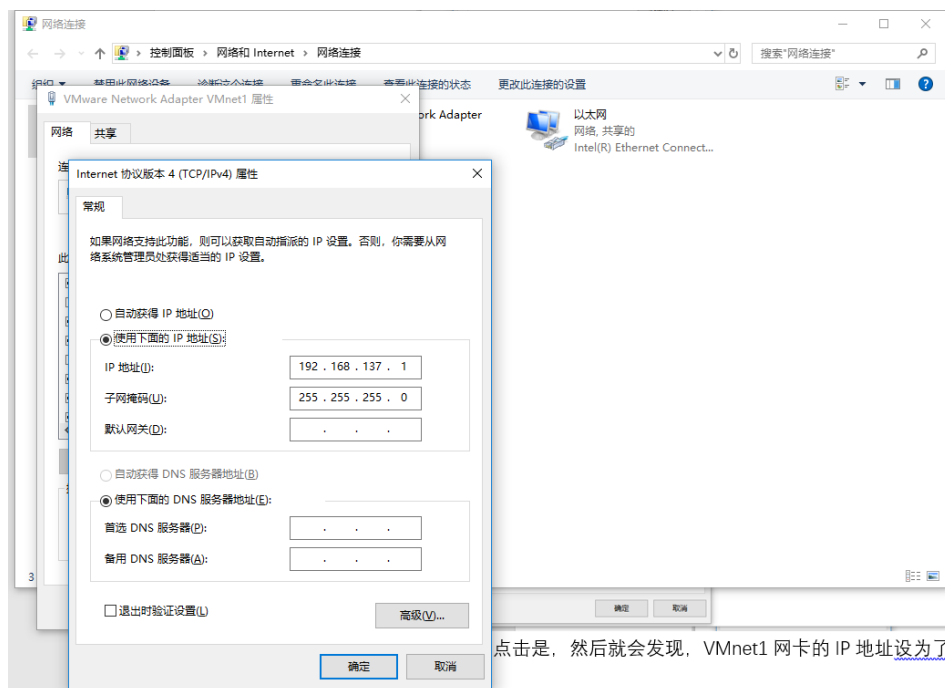
这种情况下，为了 Openstack 里的实例能够访问互联网，需要将宿主机的网络共享给 Vmware 的虚拟网卡 VMnet1



如上图所示，点击确认后，会弹出如下对话框



这里我们点击是，然后就会发现，VMnet1 网卡的 IP 地址设为了 192.168.137.1

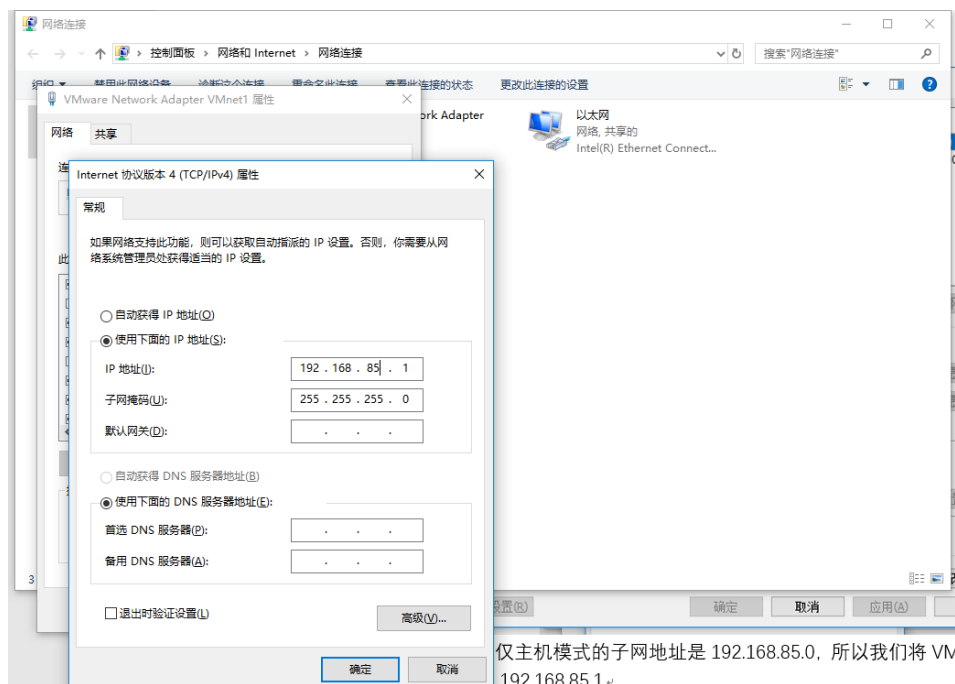


点击是，然后就会发现，VMnet1 网卡的 IP 地址设为了

这时候，我们需要通过 VMware workstation 的虚拟网络编辑器，找到原来设置的仅主机网络的网段。



可以看到仅主机模式的子网地址是 192.168.85.0，所以我们将 VMnet1 的网卡 ip 地址改为 192.168.85.1，并确定保存。

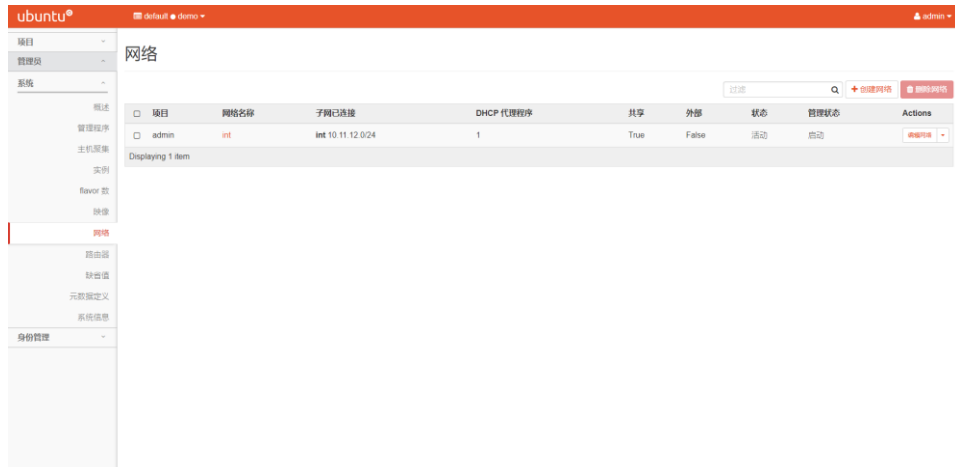


另外，我们还需要在 VMware workstation 的虚拟网络编辑器中，将仅主机网络的 dhcp 服务关闭。

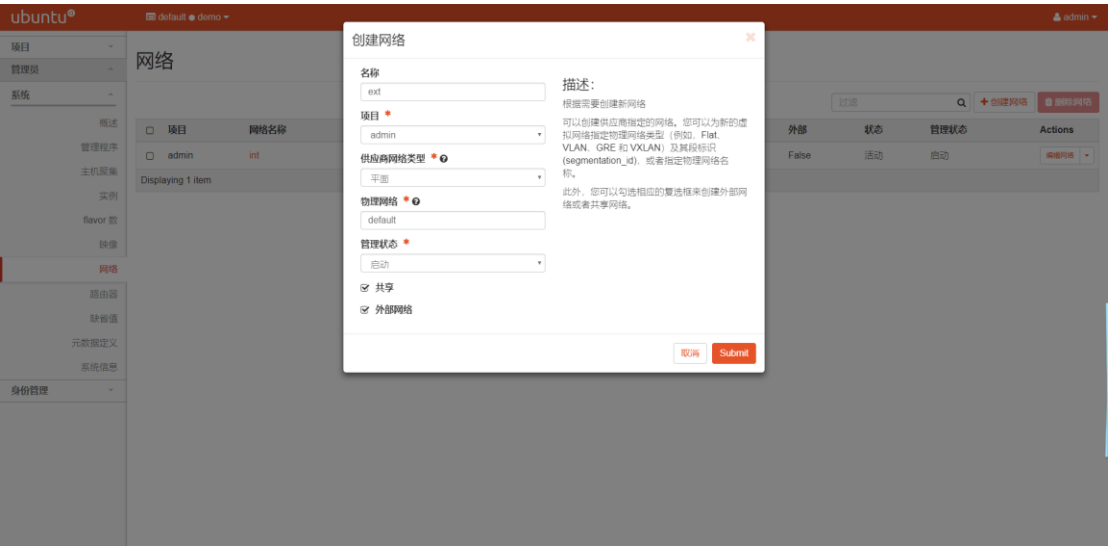




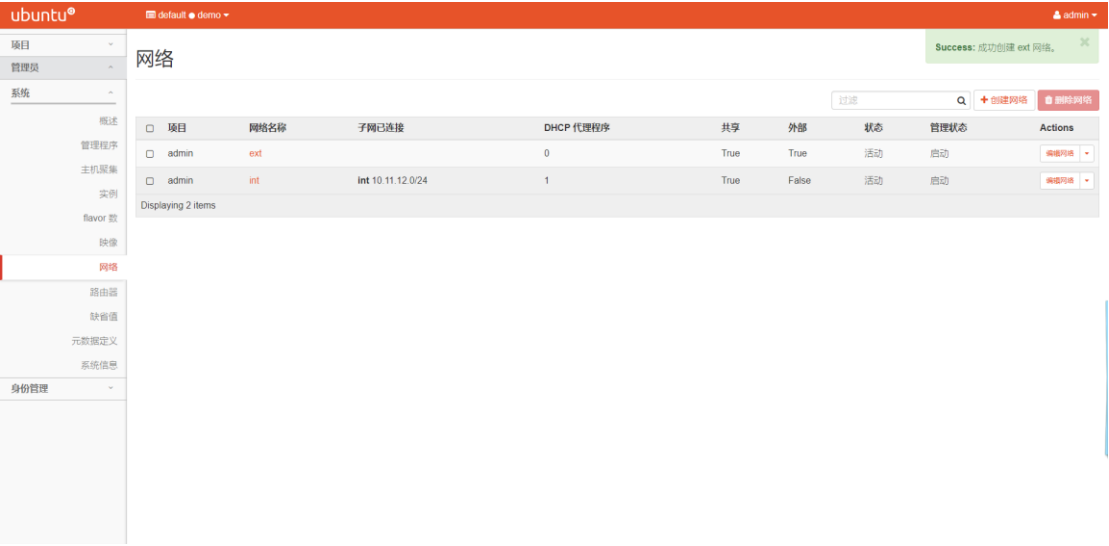
到此，虚拟机的网络设置已经完成，我们登入 Openstack 的 Dashboard 网站，点击左侧导航栏管理员栏目中的网络选项，进入网络设置页面



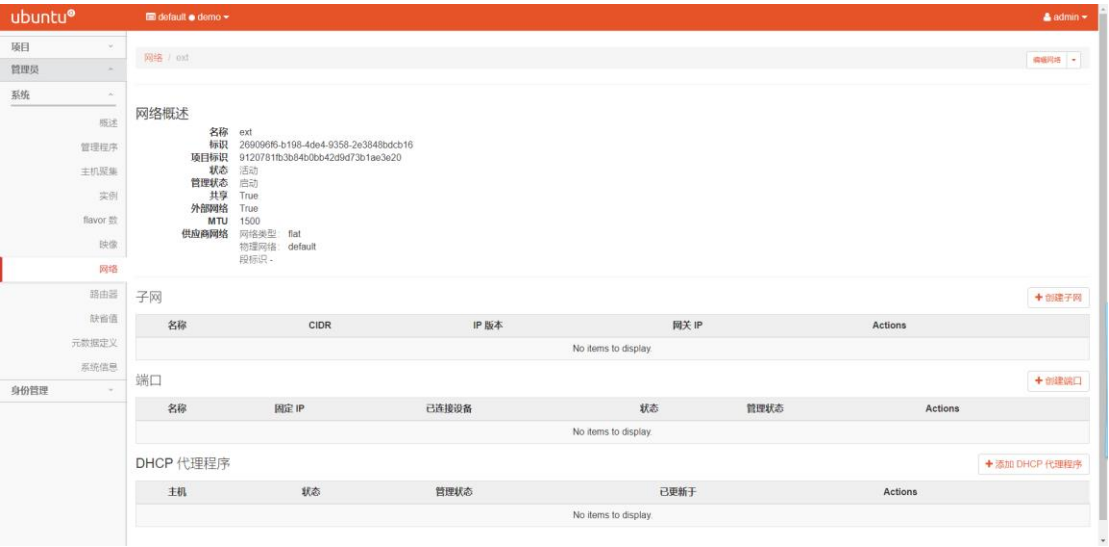
点击创建网络，我们来创建一个外部网络，注意这里的网络类型，我们选择平面，然后记得勾选外部网络的选项。



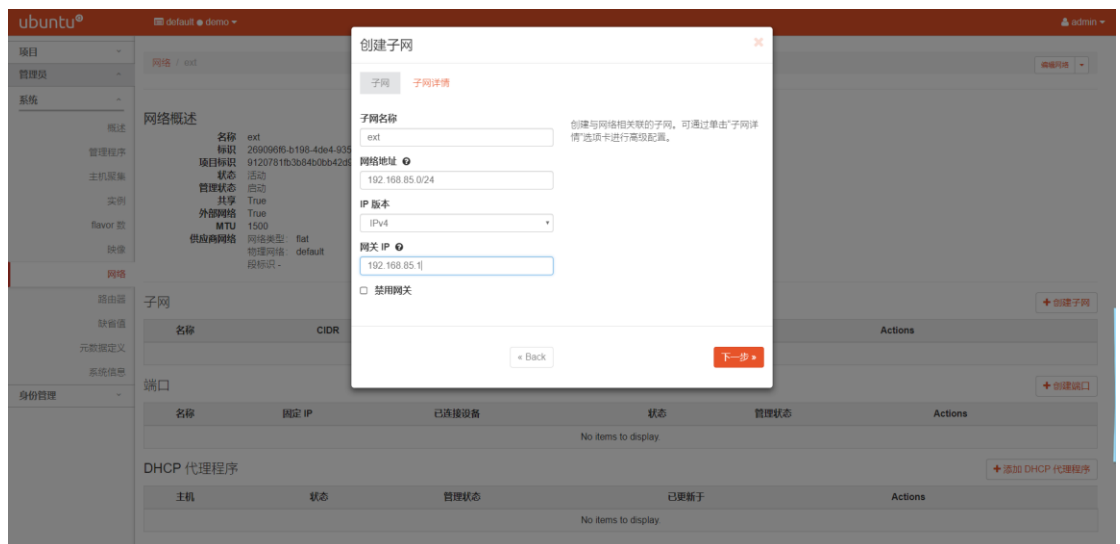
点击 Submit 保存，我们可以看到多了一条外部网络 ext 的记录



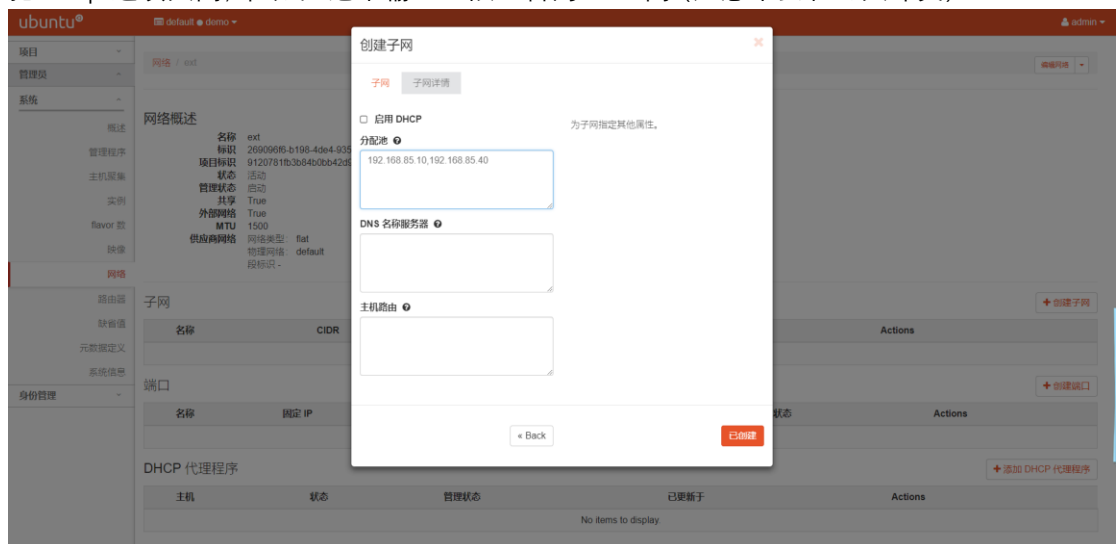
跟前面创建内部网络一样，我们点击 ext 外部网络这个记录的网络名称“ext”，进入网络的管理界面



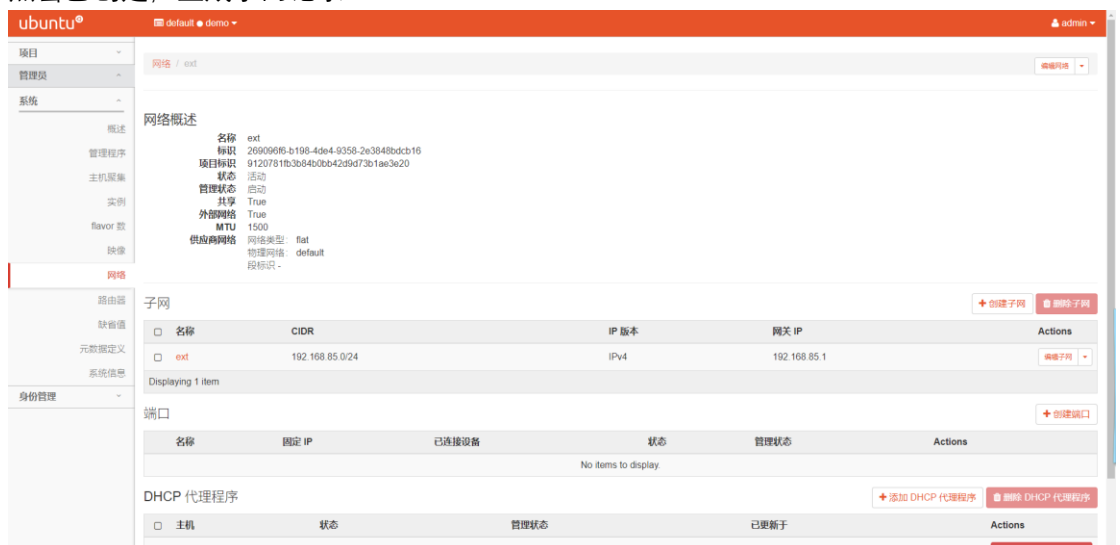
点击创建子网，弹出对话框，我们输入创建外部子网的对应信息，然后点击下一步。注意这里我们设置的子网网段要和网卡 VMnet1 的网段一样。



把 dhcp 选项关闭，在分配池中输入一段适合的 IP 区间（注意不要和网关冲突）

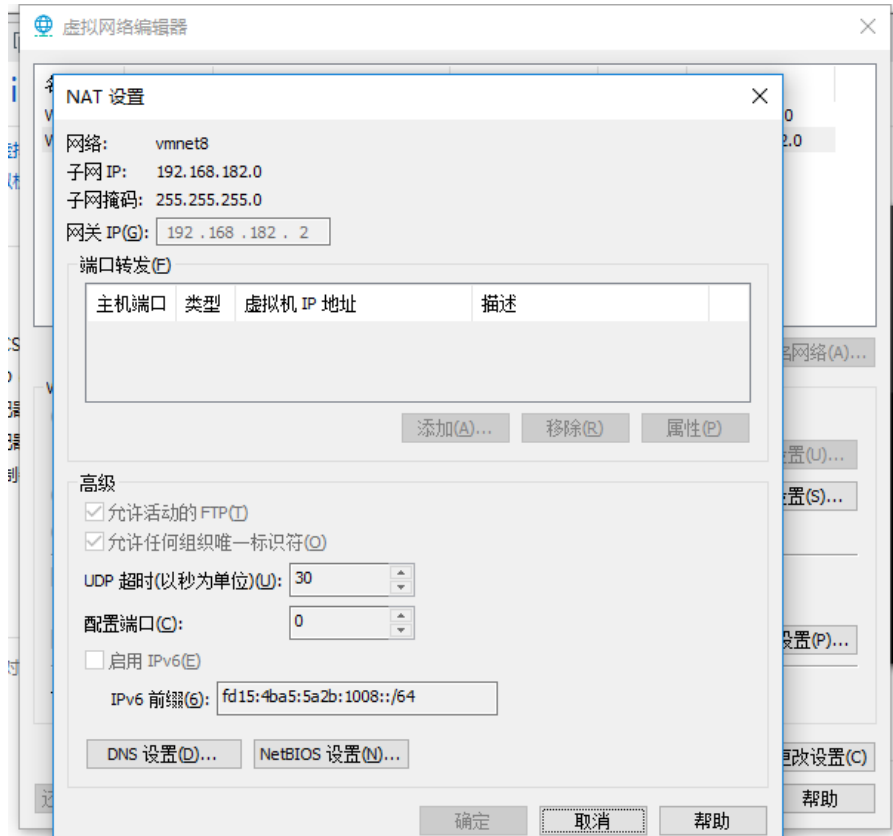


点击已创建，生成子网记录



(2) “NAT 网卡”作为外部网络网卡

如果以“NAT 网卡”作为外部网络的网卡的话，首先通过 VMware workstation 的虚拟网络编辑器，查看 NAT 网络的网关是什么

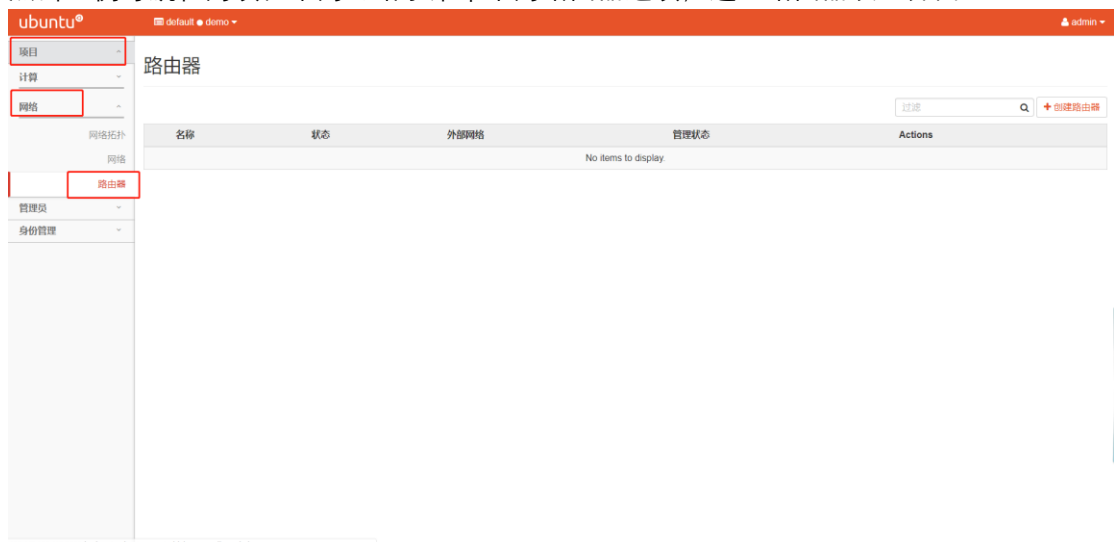


例如我这里的 NAT 的网关 IP 是 192.168.182.2，所以在上面创建外部网络时，设置的网关 IP 就应该为 192.168.182.2。

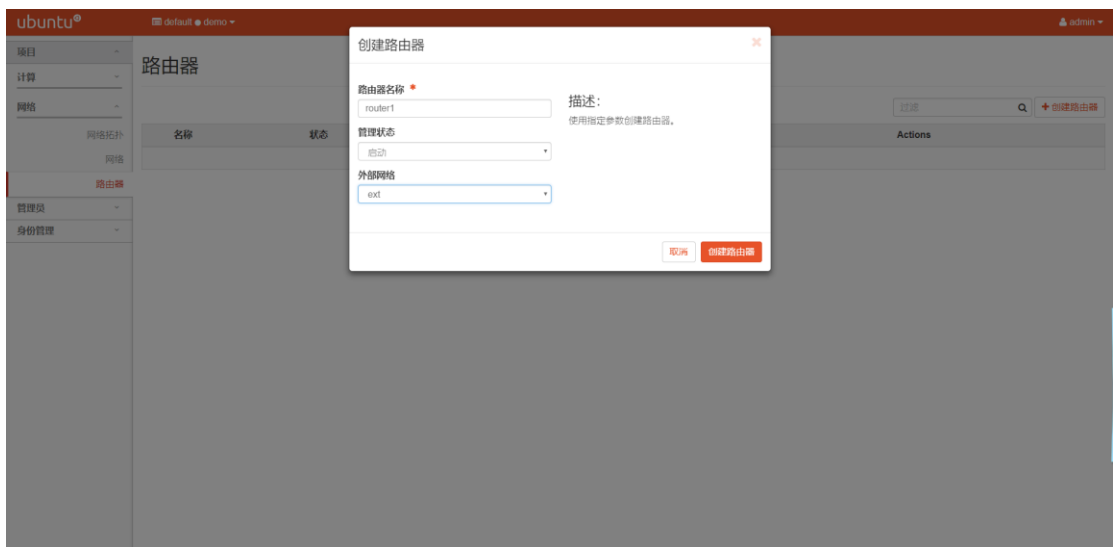
其他操作大同小异。注意创建外部网络不要勾选启用 dhcp 选项就好了，但是 vmware 的虚拟网络编辑器中的 NAT 网络需要勾选上 dhcp。

## 7. 创建一个路由器

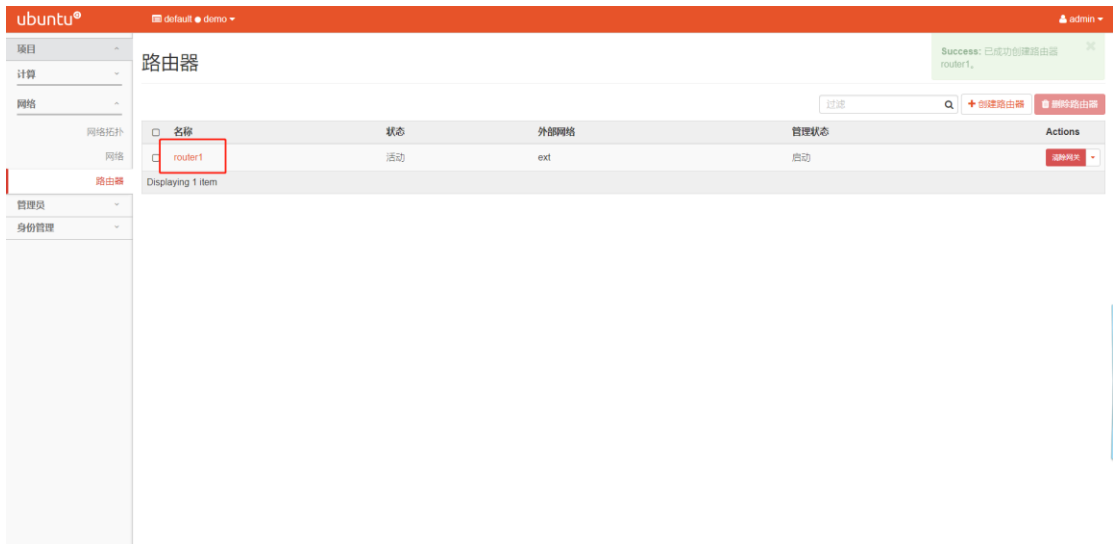
点击左侧导航栏的项目中的网络子菜单中的路由器选项，进入路由器设置界面



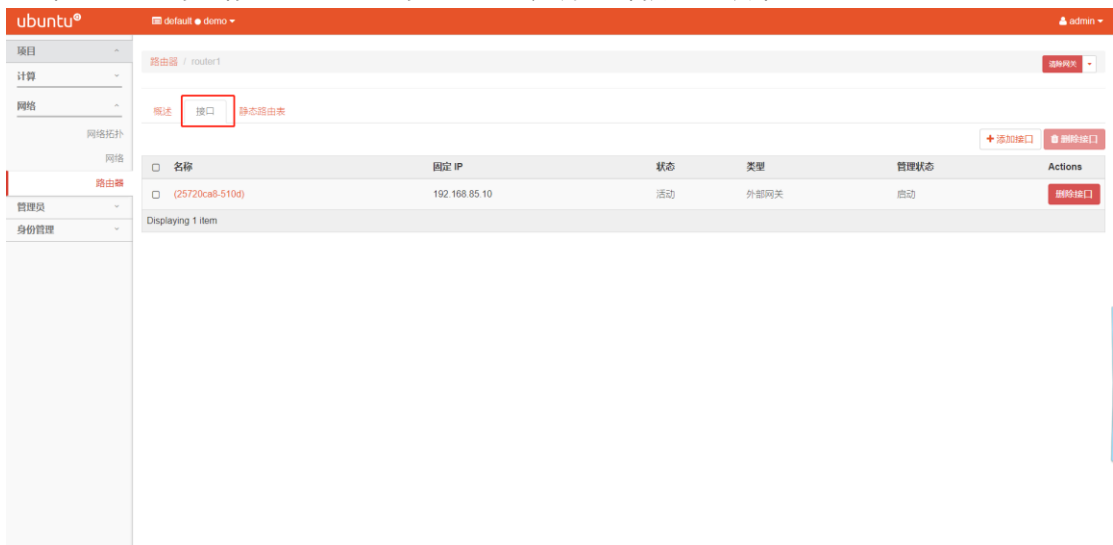
点击创建路由器，弹出对话框，输入路由器信息，外部网络选择 ext



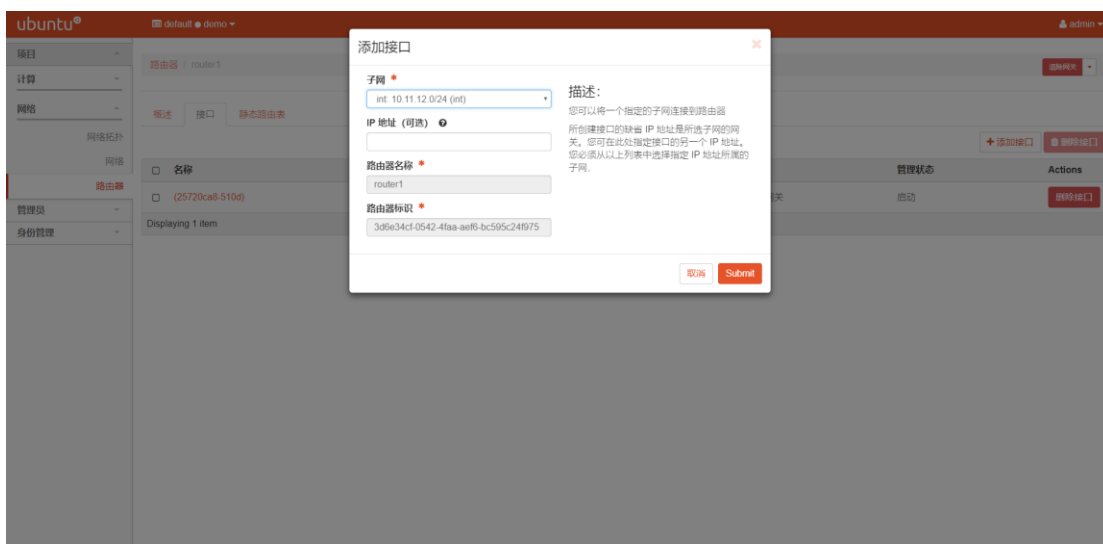
点击创建路由器，可以看到 router1 的路由器记录生成了



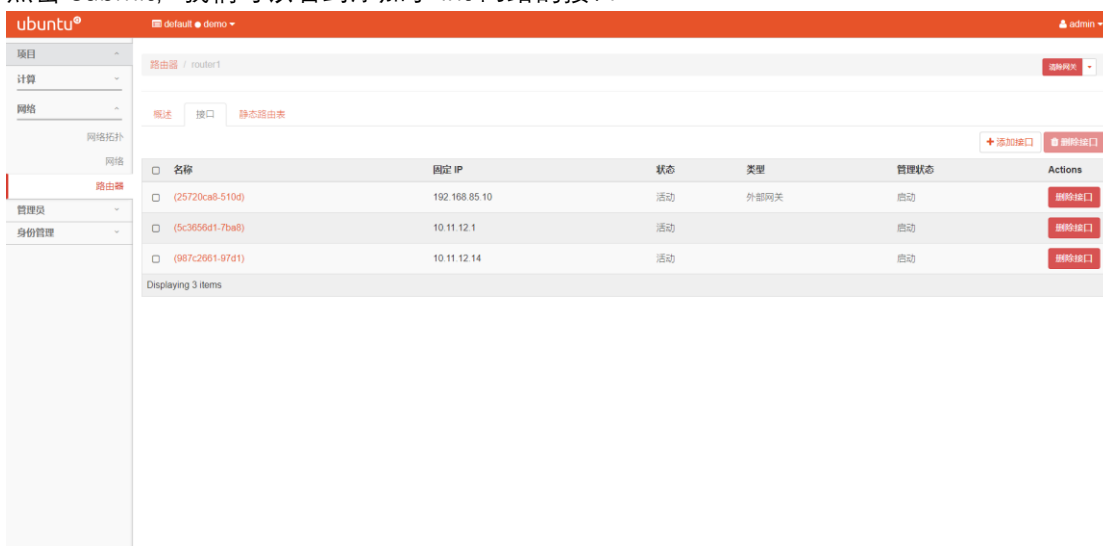
点击 router1 的名称进入路由器的设置界面，并选择接口选项卡



点击添加接口，我们将内部网络 int 加入到路由器中



点击 Submit, 我们可以看到添加了 int 网络的接口



至此, 网络配置暂时完成, 大家需要留意各接口的状态是否为活动。

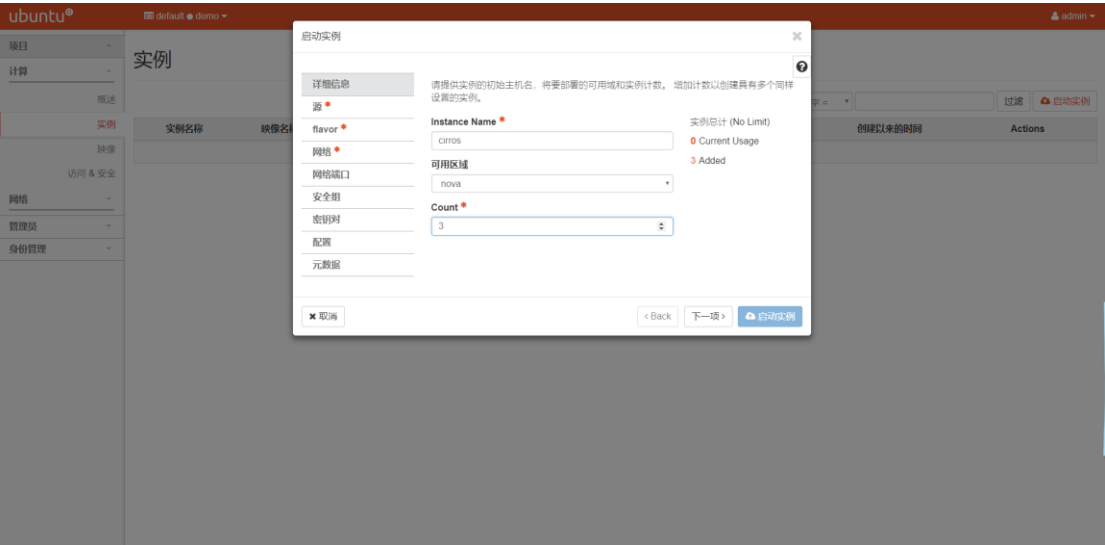
## 8. 创建实例

点击左侧导航栏项目中的计算子菜单中的实例选项, 进入实例管理界面

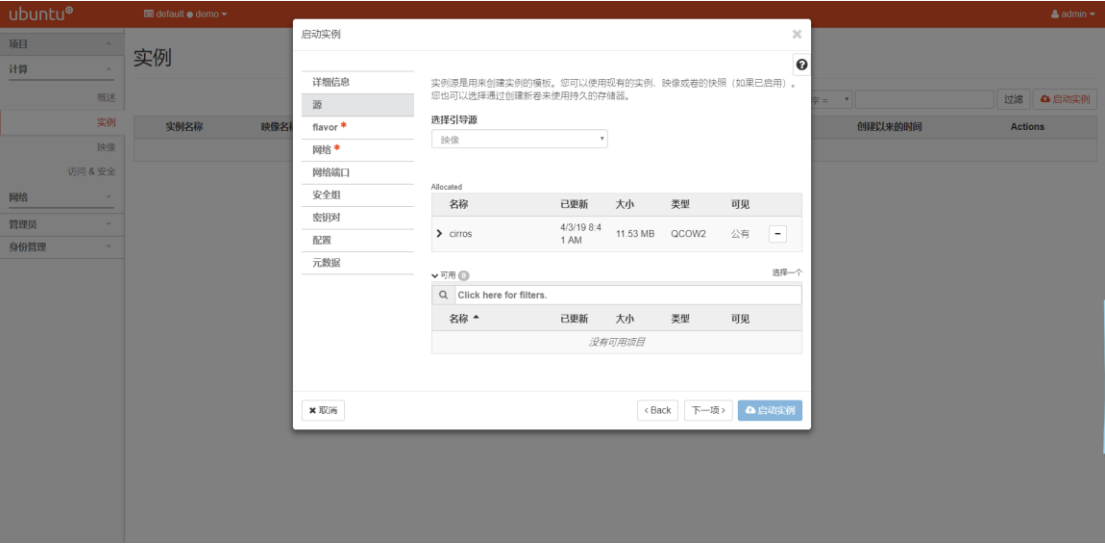


点击启动实例, 弹出对话框, 填写实例的信息, 在这里我们选择创建 3 个实例。注意,

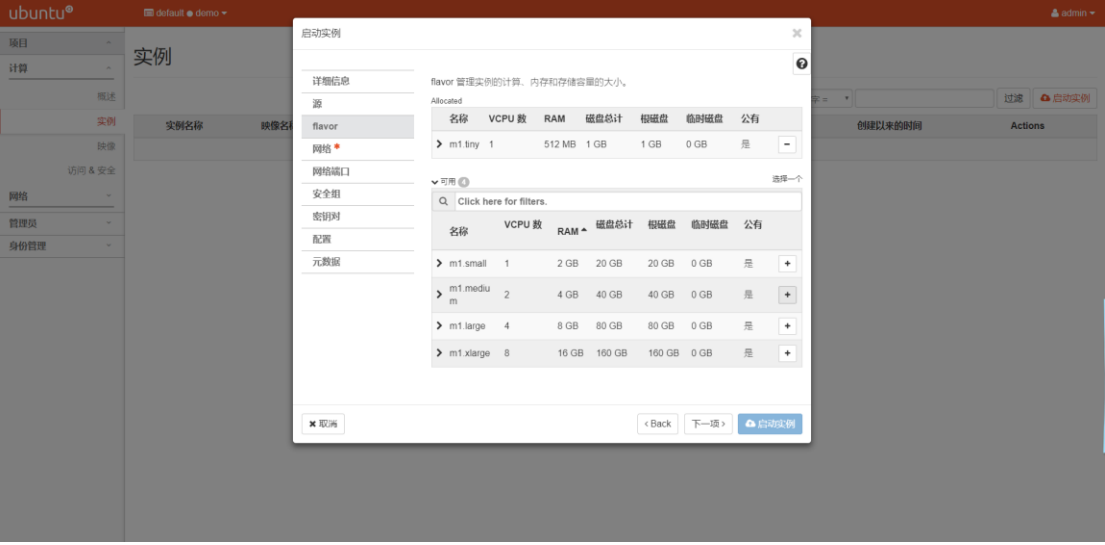
同学们的虚拟机的内存最好设置为 4g，不然可能无法开启实例。



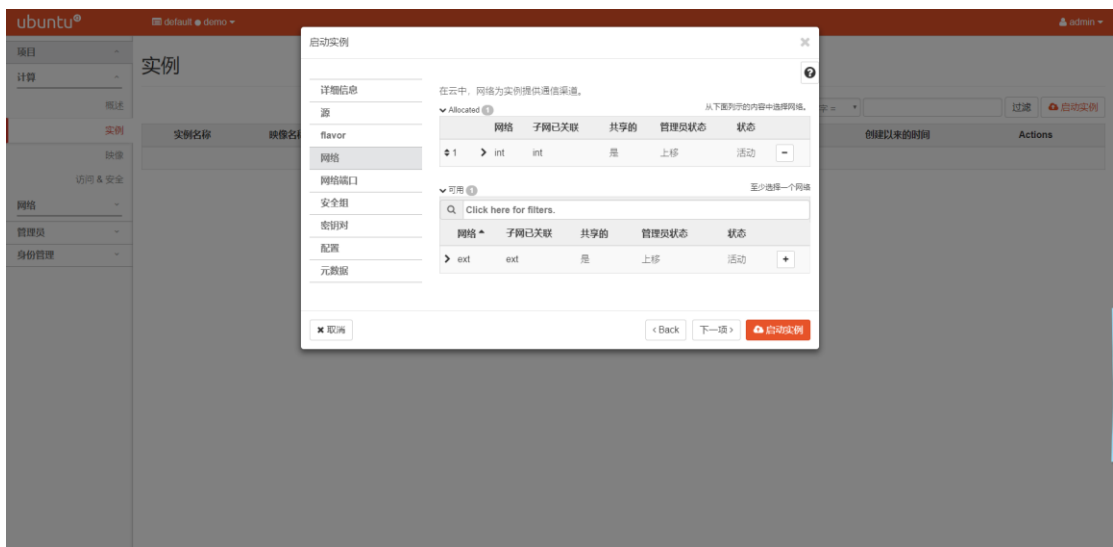
点击下一步，选择 cirros 镜像（按对应镜像的“+”按钮）



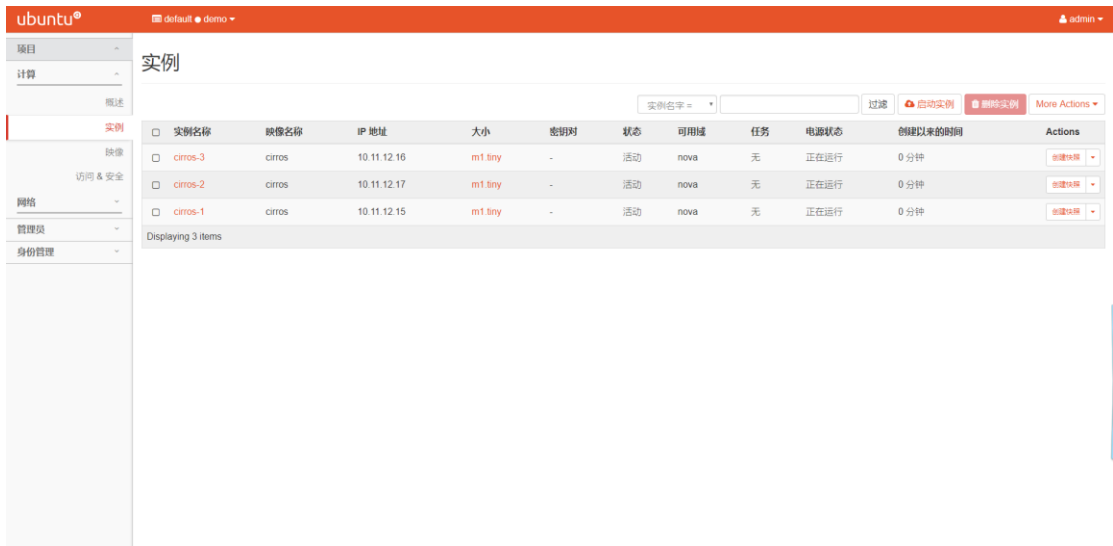
点击下一步，选择实例配置，我们选择最小一档即可



点击下一步，选择网络，这里我们选择 int 网络即可

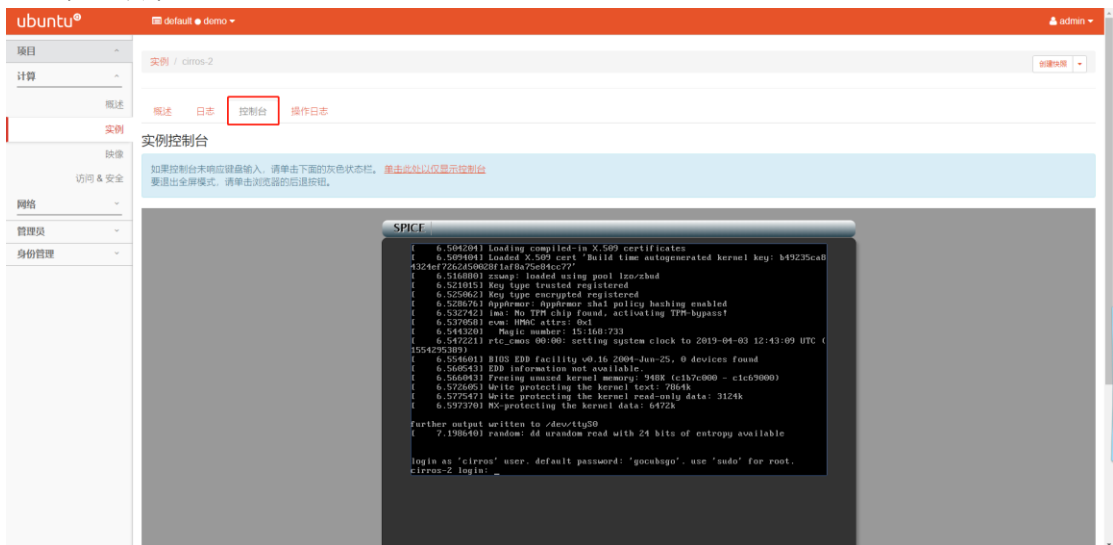


点击启动实例，实例将会被创建



## 9. 测试实例在内部网络中是否能相互 ping 通

点击三个实例中的随便一个的实例名称（如“cirros-2”），进入实例的管理界面，并点击控制台选项卡。





```
SPICE
[ 6.516880] zswap: loaded using pool lzo/zbud
[ 6.521015] Key type trusted registered
[ 6.525062] Key type encrypted registered
[ 6.528676] AppArmor: AppArmor sha1 policy hashing enabled
[ 6.532742] ima: No TPM chip found, activating TPM-bypass!
[ 6.537058] evm: HMAC attrs: 0x1
[ 6.544320] Magic number: 15:168:733
[ 6.547221] rtc_cmos 00:00: setting system clock to 2019-04-03 12:43:09 UTC (
1554295389)
[ 6.554601] BIOS EDD facility v0.16 2004-Jun-25, 0 devices found
[ 6.560543] EDD information not available.
[ 6.566043] Freeing unused kernel memory: 948K (c1b7c000 - c1c69000)
[ 6.572605] Write protecting the kernel text: 7864k
[ 6.577547] Write protecting the kernel read-only data: 3124k
[ 6.597370] NX-protecting the kernel data: 6472k

further output written to /dev/ttyS0
[ 7.198640] random: dd urandom read with 24 bits of entropy available

login as 'cirros' user, default password: 'gocubsgo'. use 'sudo' for root.
cirros-2 login: cirros[ 133.580537] random: nonblocking pool is initialized

Password:
$ _
```

这时我们先查看 cirros-2 的网卡情况

```
SPICE

login as 'cirros' user, default password: 'gocubsgo'. use 'sudo' for root.
cirros-2 login: cirros[ 133.580537] random: nonblocking pool is initialized

Password:
$ ifconfig
eth0      Link encap:Ethernet  HWaddr FA:16:3E:19:CF:37
          inet addr:10.11.12.17  Bcast:10.11.12.255  Mask:255.255.255.0
          inet6 addr: fe80::f816:3eff:fe19:cf37/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1450  Metric:1
          RX packets:103 errors:0 dropped:0 overruns:0 frame:0
          TX packets:106 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:11450 (11.1 KiB)  TX bytes:9546 (9.3 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

$ _
```

可以看到跟前面创建实例时自动分配的 IP 是一致的，这时候我们尝试 ping 一下 cirros-3 和 cirros-1，它们对应的 ip 地址

```
SPICE

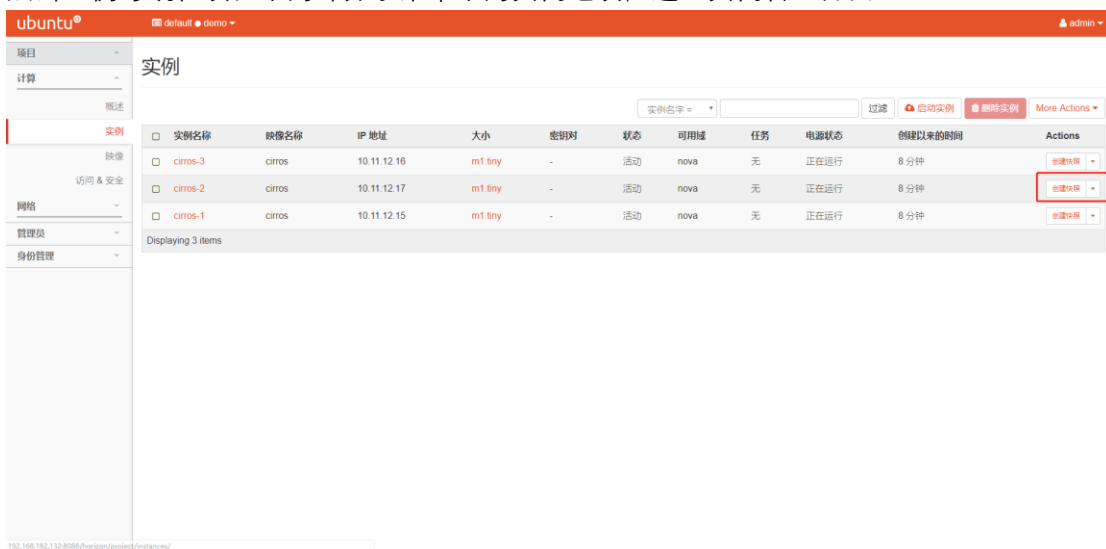
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

$ ping 10.11.12.15
PING 10.11.12.15 (10.11.12.15): 56 data bytes
64 bytes from 10.11.12.15: seq=0 ttl=64 time=1.534 ms
64 bytes from 10.11.12.15: seq=1 ttl=64 time=0.494 ms
64 bytes from 10.11.12.15: seq=2 ttl=64 time=1.212 ms
64 bytes from 10.11.12.15: seq=3 ttl=64 time=0.409 ms
^C
--- 10.11.12.15 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.409/0.912/1.534 ms
$ ping 10.11.12.16
PING 10.11.12.16 (10.11.12.16): 56 data bytes
64 bytes from 10.11.12.16: seq=0 ttl=64 time=1.243 ms
64 bytes from 10.11.12.16: seq=1 ttl=64 time=0.572 ms
64 bytes from 10.11.12.16: seq=2 ttl=64 time=0.302 ms
64 bytes from 10.11.12.16: seq=3 ttl=64 time=0.361 ms
^C
--- 10.11.12.16 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.302/0.619/1.243 ms
$
```

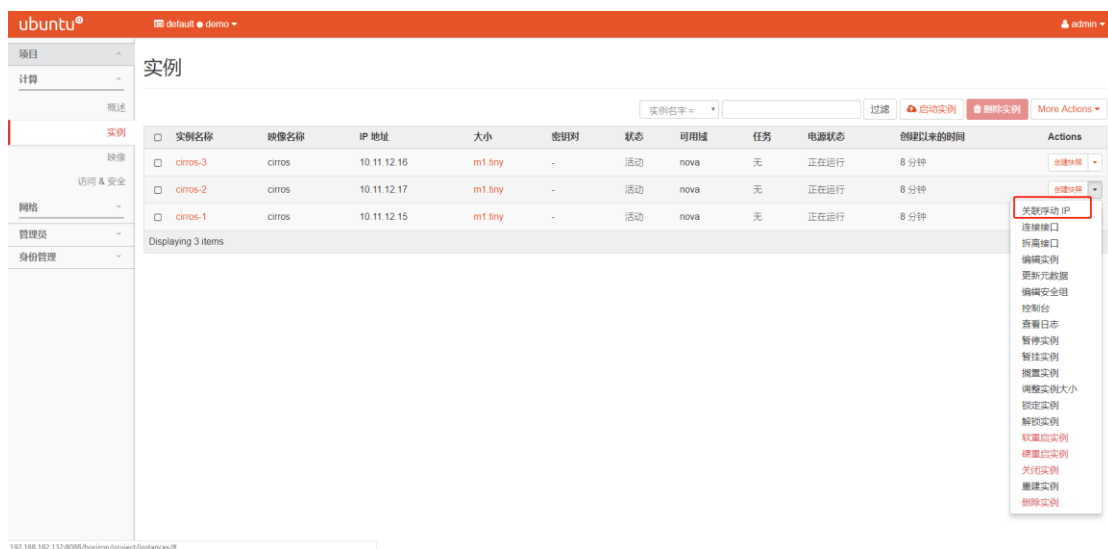
发现是可以相互 ping 通的，证明内部网络设置成功。

## 10. 给 cirros-2 绑定浮动 IP，查看是否能访问外网

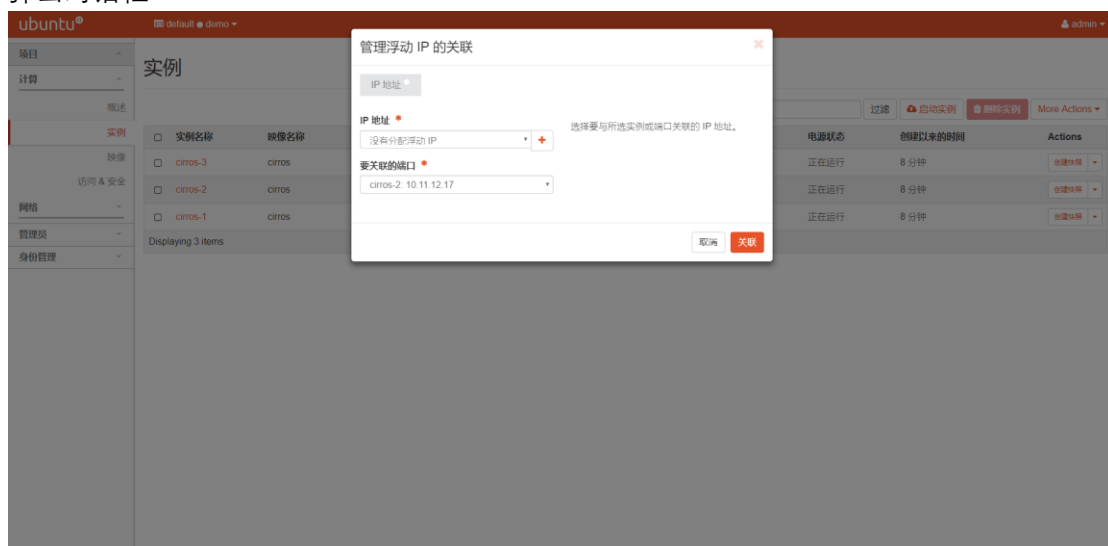
点击左侧导航栏项目中的计算子菜单中的实例选项，进入实例管理界面



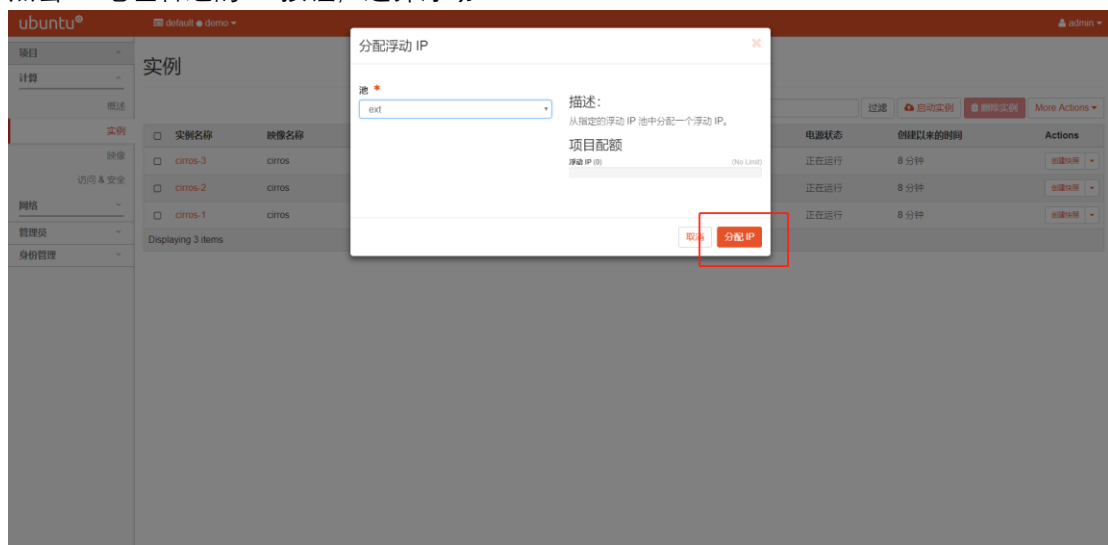
在 cirros-2 记录的右边有一个创建快照选项的右边有一个下拉按钮，点击它，选择关联浮动 IP



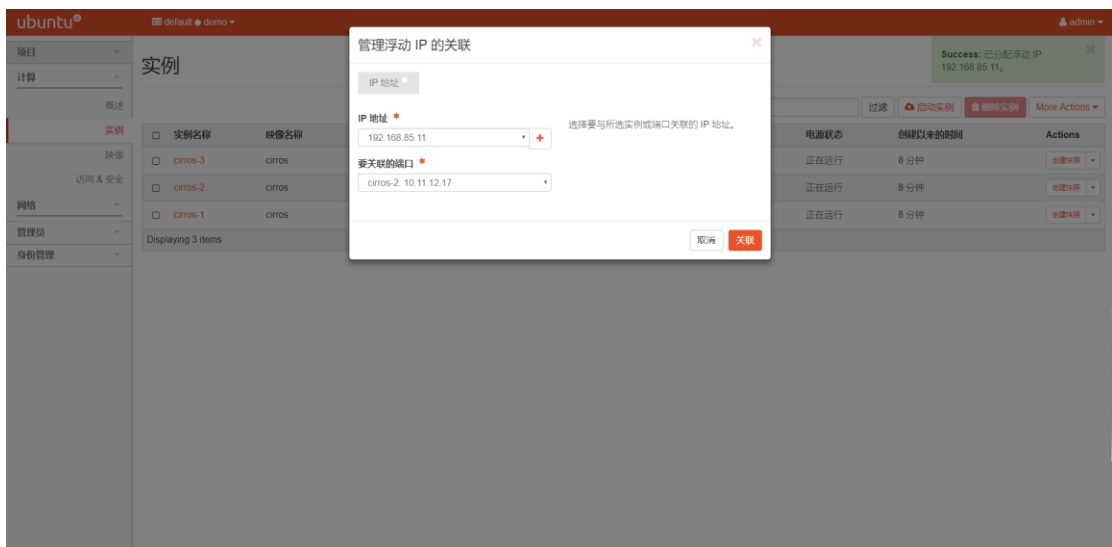
弹出对话框



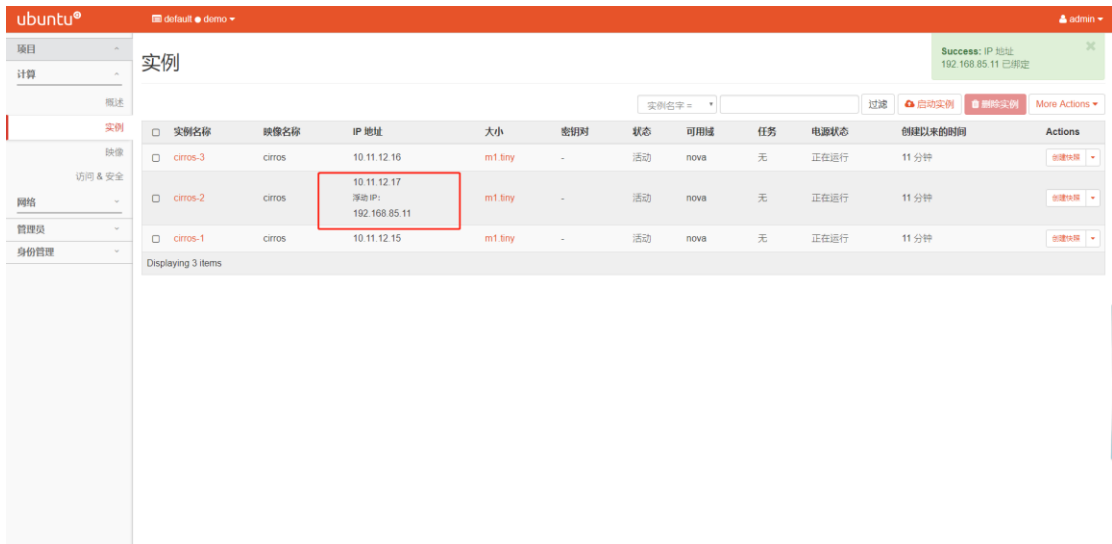
点击 IP 地址右边的“+”按钮，选择浮动 IP



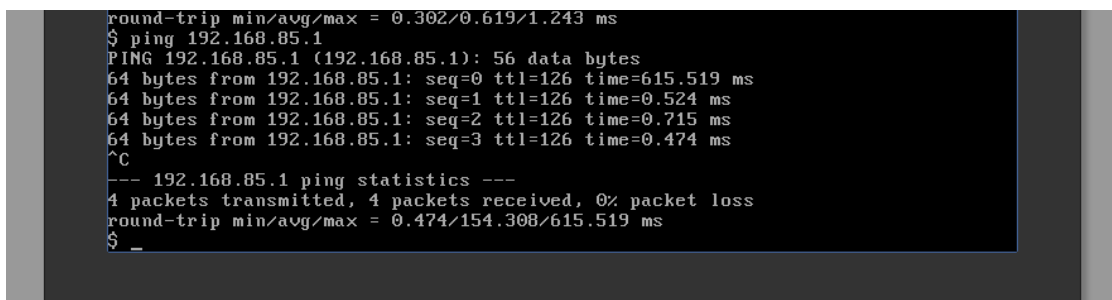
点击分配 IP，即会自动分配一个外部网络网段的 IP 地址



点击关联，即关联成功



再次点击 cirros-2 记录的实例名称，进入控制台，这时我们来尝试 ping 一下外部网络的网关 192.168.85.1



是可以 ping 通的，说明浮动 IP 绑定生效，接下来我们从宿主机 ping 一下百度，获取百度网站的具体 ip 地址

```
命令提示符
Microsoft Windows [版本 10.0.17134.648]
(c) 2018 Microsoft Corporation。保留所有权利。

C:\Users\Administrator>ping baidu.com

正在 Ping baidu.com [220.181.57.216] 具有 32 字节的数据:
来自 220.181.57.216 的回复: 字节=32 时间=41ms TTL=45
来自 220.181.57.216 的回复: 字节=32 时间=41ms TTL=45
来自 220.181.57.216 的回复: 字节=32 时间=41ms TTL=45
来自 220.181.57.216 的回复: 字节=32 时间=41ms TTL=45

220.181.57.216 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 = 41ms, 最长 = 41ms, 平均 = 41ms

C:\Users\Administrator>
```

可以看到是 220.181.57.216，我们在 cirros-2 的控制台中尝试 ping 这个 ip 地址

```
round-trip min/avg/max = 0.177/134.366/615.315 ms
$ ping 220.181.57.216
PING 220.181.57.216 (220.181.57.216): 56 data bytes
64 bytes from 220.181.57.216: seq=0 ttl=42 time=44.147 ms
64 bytes from 220.181.57.216: seq=1 ttl=42 time=41.963 ms
64 bytes from 220.181.57.216: seq=2 ttl=42 time=42.377 ms
64 bytes from 220.181.57.216: seq=3 ttl=42 time=42.177 ms
^C
--- 220.181.57.216 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 41.963/42.666/44.147 ms
$ _
```

可以看到是 ping 通了，证明 cirros-2 可以访问外网。

如果发现虚拟机重启后，或者电脑重启后，Openstack 里的实例访问不了外网了，很大可能是网络共享失效了。大家重新把宿主机的网络共享给 VMnet1 网卡，并且修改 VMnet1 网卡的 ip 地址即可。

## 11. 剩下的同学们自己自由发挥哈