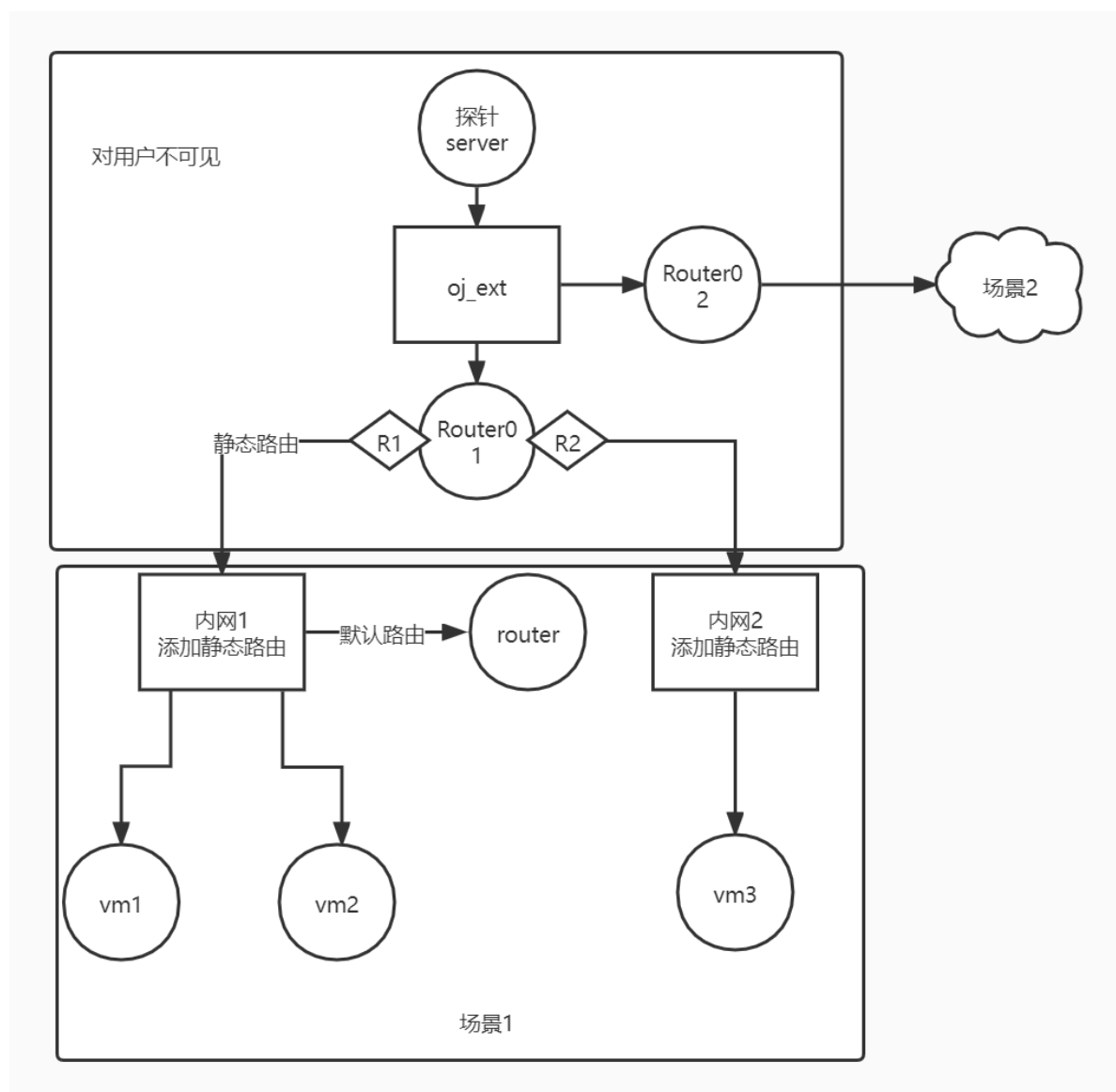


1. 拓扑图



操作步骤:

1. 每创建一个场景需要创建一个对应的路由器，比如场景1，对应的Router01
2. 创建内部网络1，添加静态路由，到 探针server 的下一跳 给 R1接口IP地址 例如: 10.100.7.146/32,192.168.1.220
3. 创建内部网络1时需要开启DHCP功能(给虚拟机推送静态路由)
4. 每个内部网络预留一个IP地址，本次示例为192.168.1.220；外部网络，每个场景占用一个外部网络IP地址，实现 SNAT功能。
5. Router01 和 内网网络1相连接，并配置对应IP地址，例如: 192.168.1.220(不能和真实网络冲突)，Router01连接oj_ext网络，并配置oj_ext 分配的地址是外部网关，实现SNAT功能。

方案优势:

1. 针对探针 server 容易扩展，后期 探针 server如果部署在 openstack环境之外也可以适用。

- 2. 安全性，客户端通过静态路由只能访问到 探针 server 和 互联路由器SNAT接口地址，别的外网机器访问不到。
- 3. 简单易用，方便后期维护。

2. 测试步骤

1. 创建路由器Router01

注意点: 这里的IP地址，尽量不要和默认网关的IP地址冲突

Router01连接外部网络，充当外部网络网关

新建路由 ✕

路由名称

router01

☒ 启用管理员状态 ⓘ

外部网络

oj_ext

☒ 启用SNAT

可用域提示 ⓘ

nova

说明:

基于特殊参数创建一路由。
仅当设置了外部网络时，启用SNAT才会生效。

取消

新建路由

<input type="checkbox"/>	Name	Fixed IPs	Status	Type	Admin State	Actions
<input type="checkbox"/>	(4e646f50-2735)	• 192.168.1.220	运行中	内部接口	UP	删除接口
<input type="checkbox"/>	(e190cf16-1f50)	• 10.100.7.102	运行中	外部网关	UP	删除接口

正在显示 2 条

2. 创建内网网络01

项目 / 网络 / 网络 / test1

test1

概况子网端口

子网

正在显示 1 项

☐ Name

☐ subnet

正在显示 1 项

编辑子网

子网 *

子网详情

☒ 激活DHCP

为子网指定扩展属性

分配地址池

192.168.1.2,192.168.1.254

DNS服务器

主机路由

10.100.7.137/32,192.168.1.220

取消

« 返回

保存

3. Router01连接内部网络

增加接口

子网 *

test1: 192.168.1.0/24 (subnet)

IP地址(可选)

192.168.1.220

说明:

您可以将一个指定的子网连接到路由器

这里如果你不指定一个IP地址,则会使用被选定子网的网关地址作为路由器上新建接口的IP地址。如果网关IP地址已经被使用,你必须使用选定子网的其它地址。

取消

提交

3. 创建虚拟机client和server

正在显示 20 项 | [后页](#) »

<input type="checkbox"/>	Project	Host	Name	Image Name	IP Address	Flavor	Status	Task	Power State	Age	Actions
<input type="checkbox"/>	admin	controller	c-2	cirros	192.168.1.152	mx.6c-0.5g-20g	运行	无	运行中	7 minutes	救援云主机
<input type="checkbox"/>	admin	compute46	c-1	cirros	192.168.1.165	mx.6c-0.5g-20g	运行	无	运行中	7 minutes	救援云主机
<input type="checkbox"/>	admin	compute46	server	cirros	10.100.7.137	mx.6c-0.5g-20g	运行	无	运行中	8 minutes	救援云主机

```
$ ls
$
$ route -n
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0          192.168.1.1    0.0.0.0         UG    0      0      0 eth0
10.100.7.137     192.168.1.220 255.255.255.255 UGH    0      0      0 eth0
169.254.169.254 192.168.1.2    255.255.255.255 UGH    0      0      0 eth0
192.168.1.0      0.0.0.0        255.255.255.0   U     0      0      0 eth0
$ _
```

```
$ ping 10.100.7.137
PING 10.100.7.137 (10.100.7.137): 56 data bytes
64 bytes from 10.100.7.137: seq=0 ttl=63 time=1.546 ms
64 bytes from 10.100.7.137: seq=0 ttl=63 time=2.109 ms (DUP!)
64 bytes from 10.100.7.137: seq=1 ttl=63 time=0.860 ms
64 bytes from 10.100.7.137: seq=1 ttl=63 time=1.552 ms (DUP!)
^C
```

```
$ ping 10.100.7.254
PING 10.100.7.254 (10.100.7.254): 56 data bytes
^C
--- 10.100.7.254 ping statistics ---
7 packets transmitted, 0 packets received, 100% packet loss
$
```