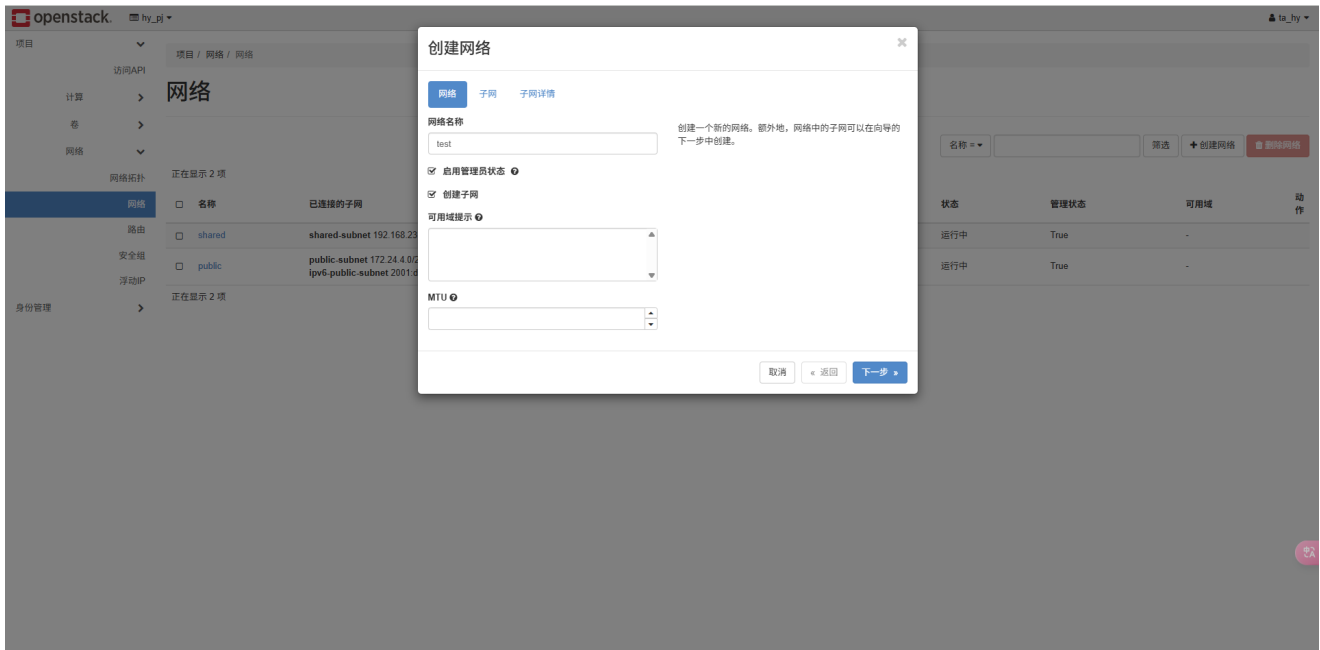


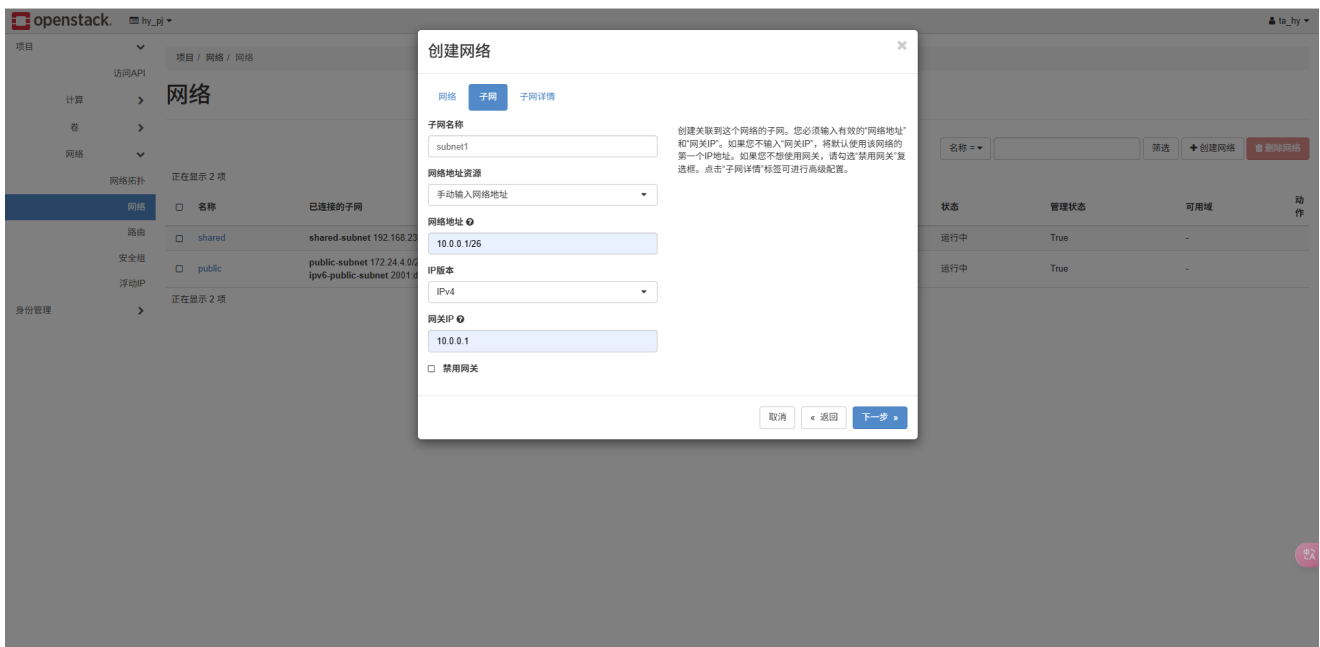
# 4.16实验课—Openstack多子网配置

## 1.单个网络下多子网创建

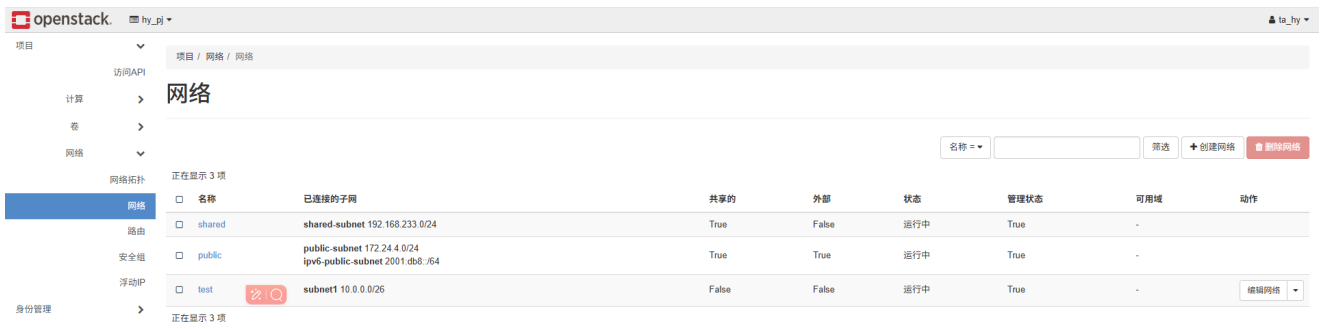
点击网络—网络—创建网络，网络名称test。



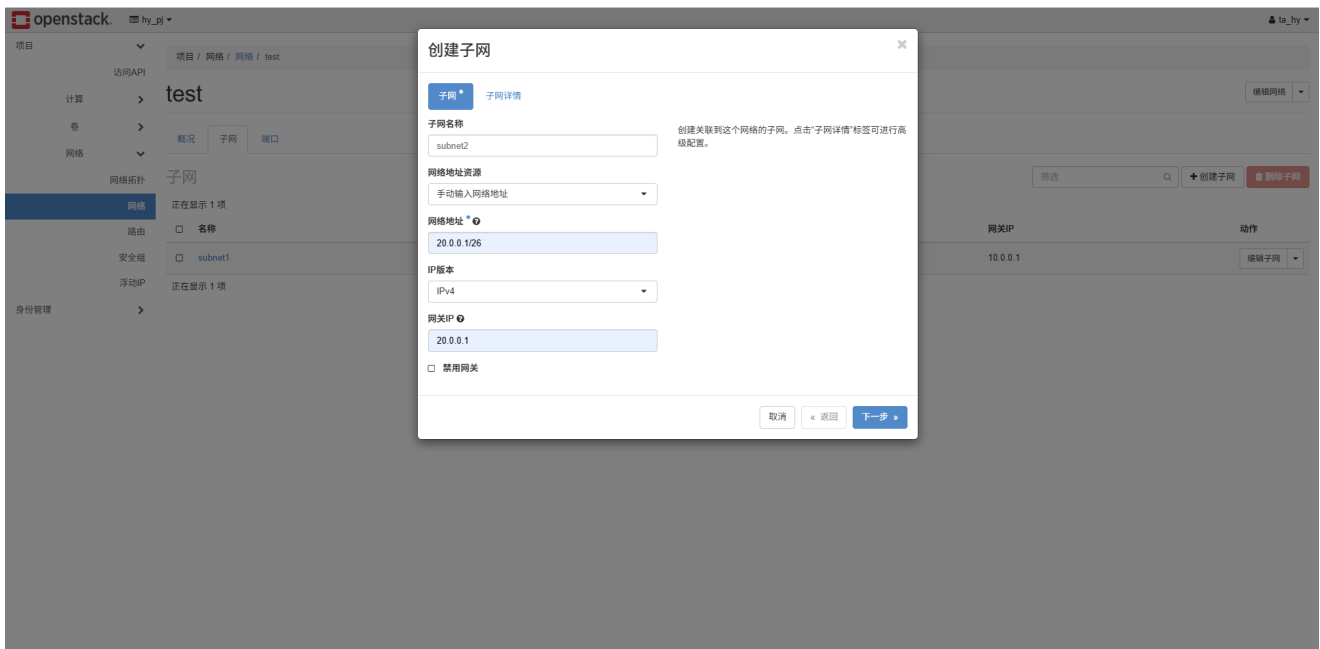
点击子网，指定子网名称，网络地址10.0.0.1/26，网关10.0.0.1。



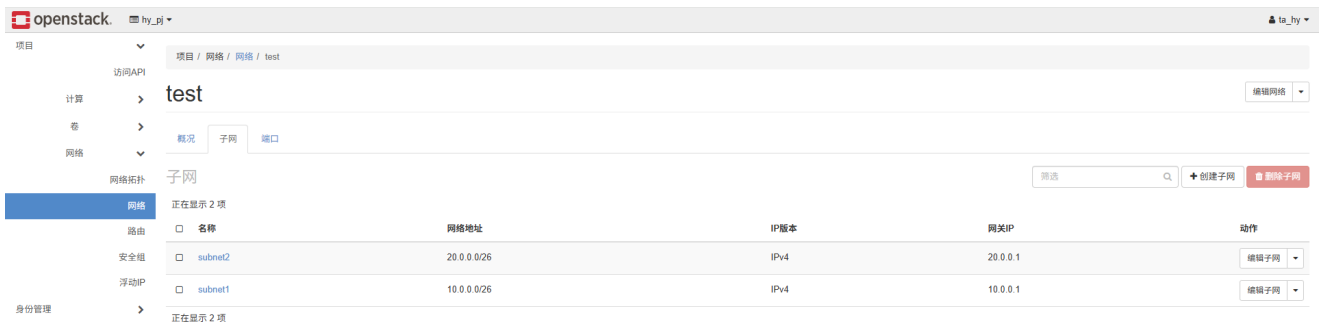
成功创建网络test，test包含一个子网subnet1。



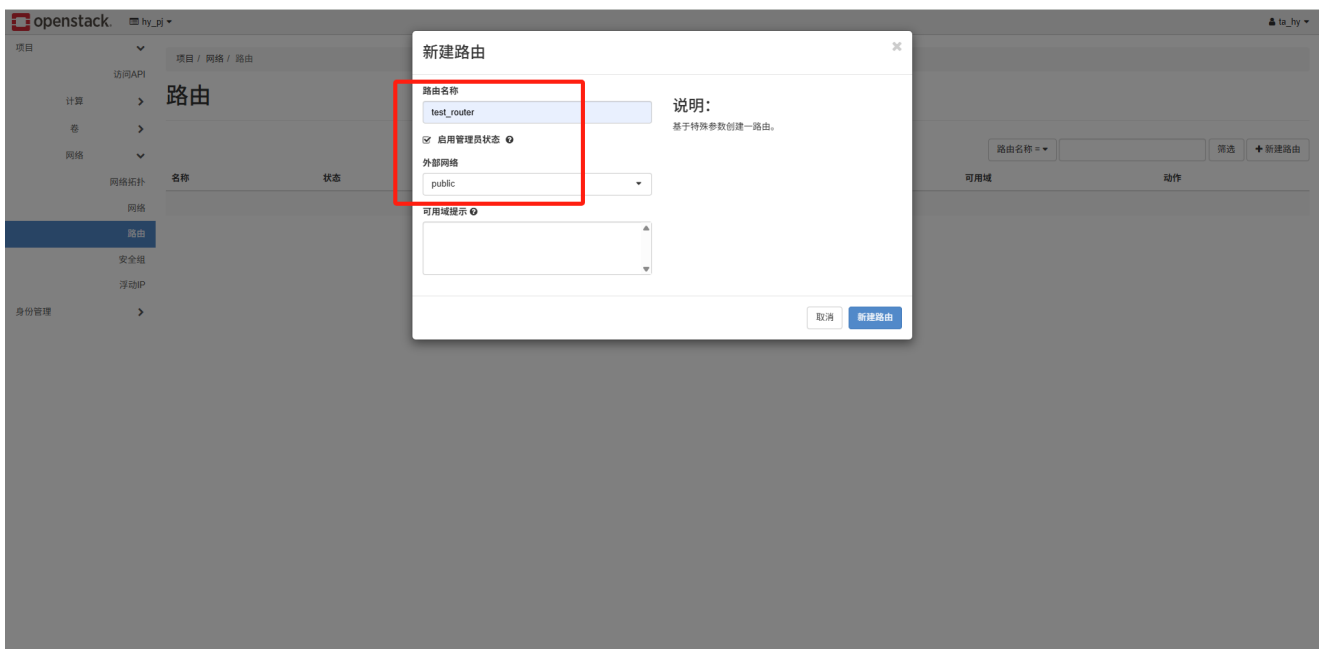
在网络test下创建子网subnet2，网络地址指定为20.0.0.1/26，网关IP为20.0.0.1。



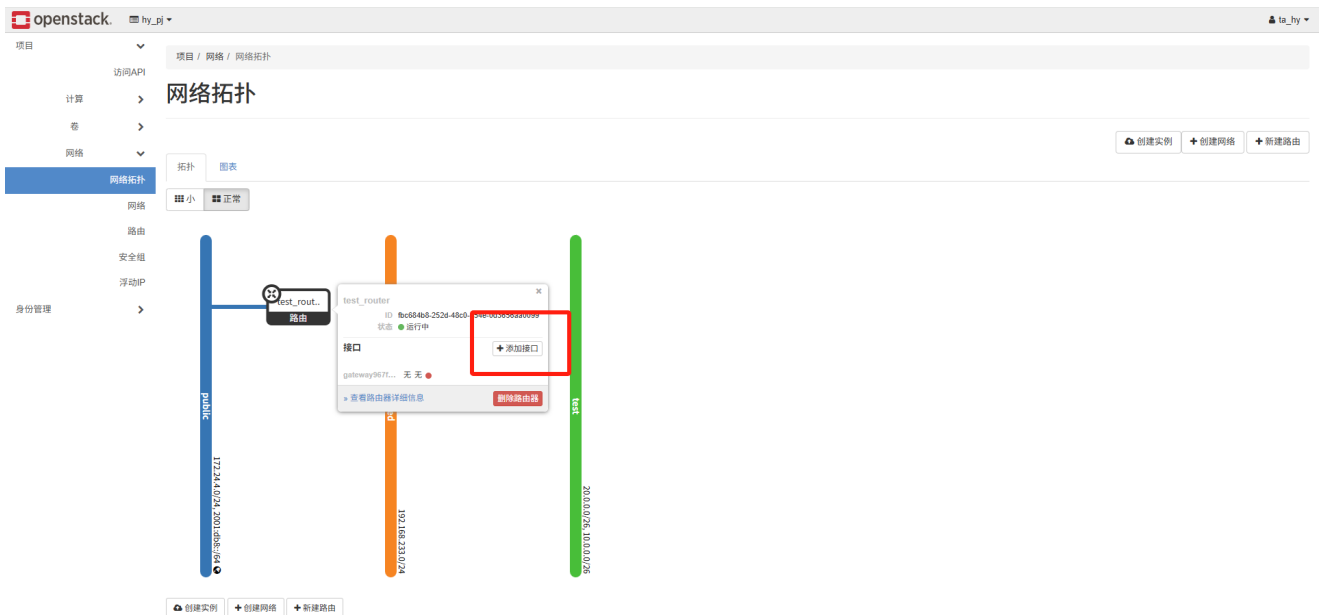
完成后可以得到以下结果，这样我们就完成了单个网络下的多子网创建。接下来进行路由设置。



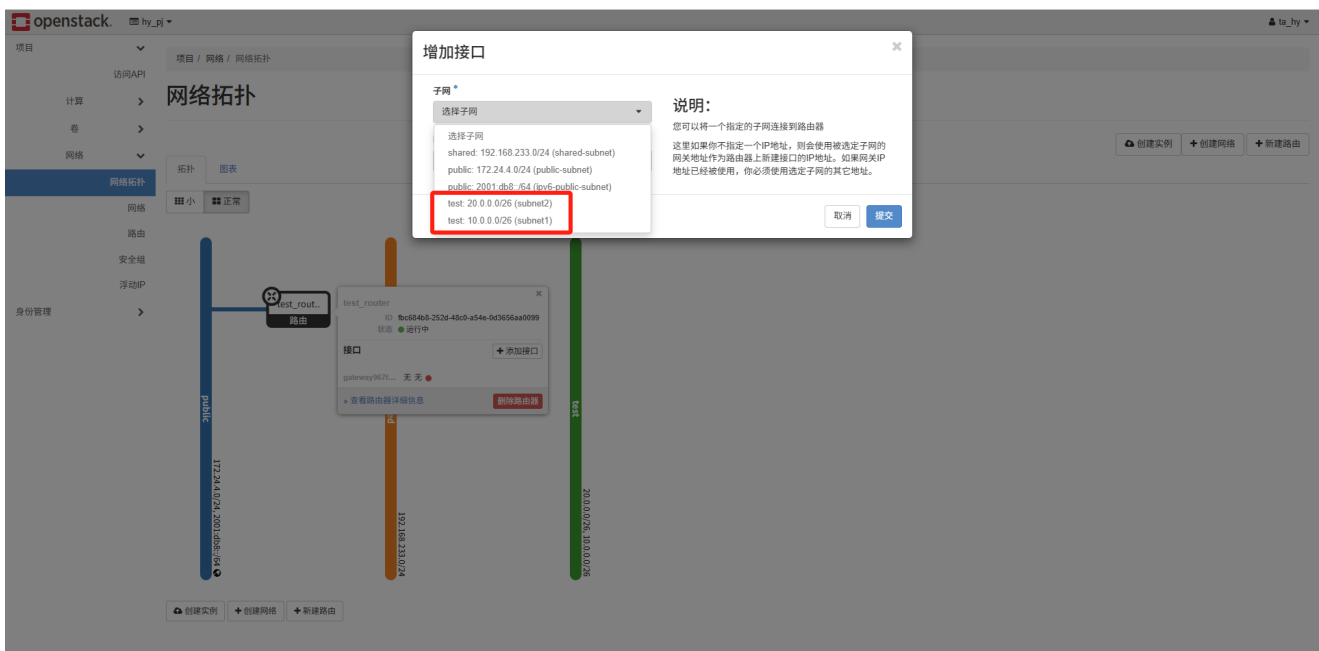
点击网络—路由—新建路由



在网络拓扑中添加多个接口



子网分别选择subnet1和subnet2。



此时我们就完成了单网络下多子网的配置。

## 2.多子网下多实例的创建与互通

我们分别在subnet1和subnet2下创建新的实例，并测试不同子网间实例的连通性。

openstack

hy\_计

项目

访问API

计算

概况

实例

镜像

密钥对

卷

网络

身份管理

项目 / 计算 / 实例

实例

实例 ID =

筛选

创建实例

删除实例

更多操作

正在显示 3 项

实例名称	镜像名称	IP 地址	实例类型	密钥对	状态	可用域	任务	电源状态	时间	动作	
subnet2_0	cidros-0.6.3-x86_64-disk	20.0.0.42	ciros256	-	运行	ud	nova	无	运行中	1 分钟	创建快照
subnet1_1	cidros-0.6.3-x86_64-disk	20.0.0.52	ciros256	-	运行	ud	nova	无	运行中	1 分钟	创建快照
subnet1_0	cidros-0.6.3-x86_64-disk	20.0.0.39	ciros256	-	运行	ud	nova	无	运行中	2 分钟	创建快照

正在显示 3 项

按照上一次实验的教程，我们可以创建三个独立的实例。逻辑上，我们把subnet1\_0，subnet1\_1定义为子网1中的实例，subnet2\_0定义为子网2中的实例。由于创建实例时默认按照最新的子网分配IP地址，我们需要在动作中找到连接接口，分离接口。指定子网1的实例为新的IP地址。

openstack

hy\_ji

项目

访问API

计算

概况

实例

实例名称

镜像名称

IP 地址

实例类型

密钥对

状态

可用域

任务

电源状态

时间

动作

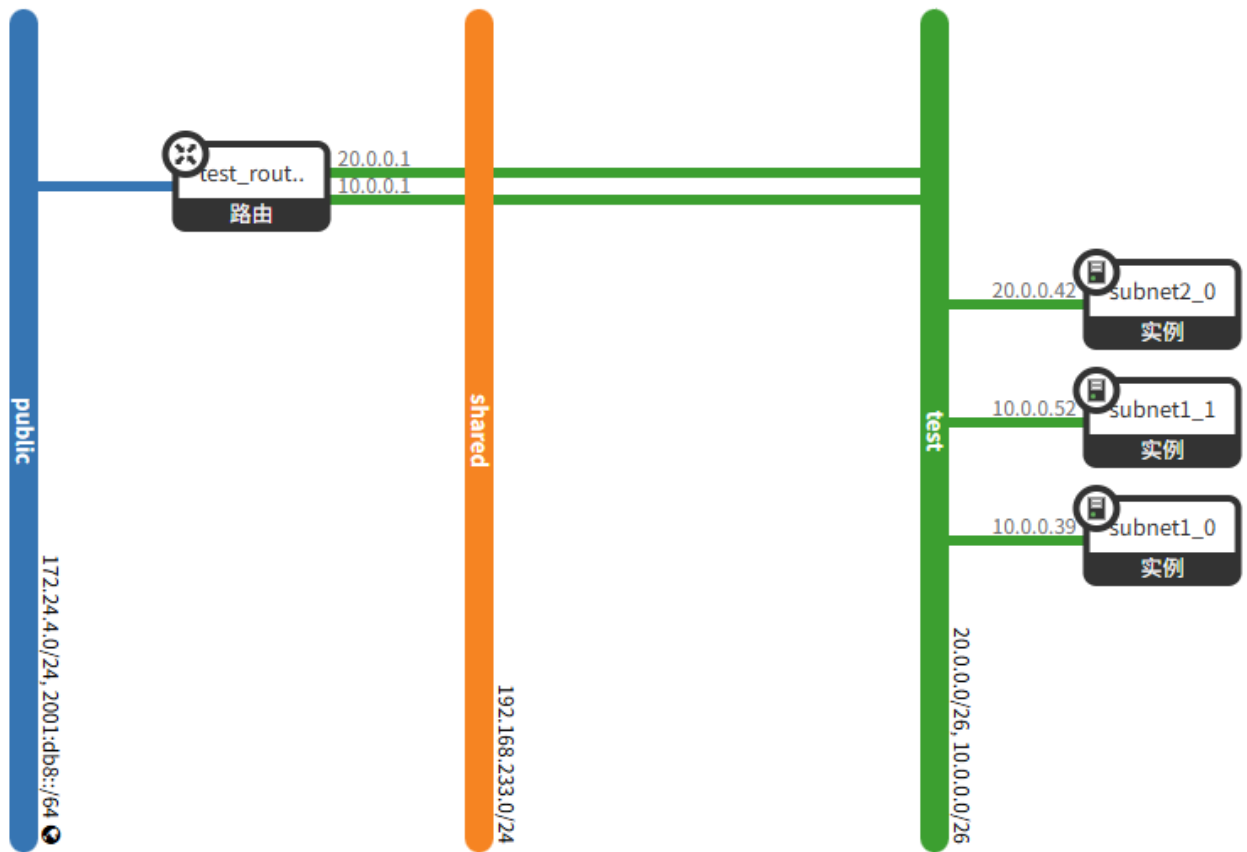
正在显示 3 项

密钥对	<input type="checkbox"/>	实例名称	镜像名称	IP 地址	实例类型	密钥对	状态	可用域	任务	电源状态	时间	动作	
主机组	<input type="checkbox"/>	subnet2_0	ciros-0.6.3-x86_64-disk	20.0.0.42	ciros256	-	运行	ud	nova	无	运行中	12 分钟	创建快照
卷	<input type="checkbox"/>	subnet1_1	ciros-0.6.3-x86_64-disk	10.0.0.52	ciros256	-	运行	ud	nova	无	运行中	13 分钟	创建快照
网络	<input type="checkbox"/>	subnet1_0	ciros-0.6.3-x86_64-disk	10.0.0.39	ciros256	-	运行	ud	nova	无	运行中	14 分钟	创建快照

身份管理

正在显示 3 项

此时的网络拓扑为



接下来，我们在subnet1\_0中分别测试它对subnet1\_1和subnet2\_0的连通性。

The screenshot shows a network topology diagram on the left and a terminal window on the right. The diagram is identical to the one above. The terminal window shows the output of the `ifconfig` command for the `eth1` interface, followed by ping tests from `subnet1_0` to `subnet1_1` and `subnet2_0`. The ping results show successful connections with 0% packet loss.

```
ifconfig eth1:10.0.0.39 Bcast:10.0.0.63 Mask:255.255.192
RX packets:7 errors:0 dropped:0 overruns:0 frame:0
TX packets:16 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:1214 (1.1 KiB) TX bytes:1620 (1.5 KiB)

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 carrier:0
collisions:0 txqueuelen:1000
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

$ ping 10.0.0.52
PING 10.0.0.52 (10.0.0.52) 56(84) bytes of data:
64 bytes from 10.0.0.52: icmp_seq=1 ttl=64 time=3.27 ms
64 bytes from 10.0.0.52: icmp_seq=2 ttl=64 time=0.877 ms
64 bytes from 10.0.0.52: icmp_seq=3 ttl=64 time=0.353 ms
--- 10.0.0.52 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2032ms
rtt min/avg/max/mdev = 0.353/1.500/3.270/1.268 ms

$ ping 20.0.0.42
PING 20.0.0.42 (20.0.0.42) 56(84) bytes of data:
64 bytes from 20.0.0.42: icmp_seq=1 ttl=63 time=3.36 ms
64 bytes from 20.0.0.42: icmp_seq=2 ttl=63 time=1.87 ms
64 bytes from 20.0.0.42: icmp_seq=3 ttl=63 time=0.410 ms
64 bytes from 20.0.0.42: icmp_seq=4 ttl=63 time=0.335 ms
--- 20.0.0.42 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/mdev = 0.335/1.494/3.361/1.239 ms
```

除了ping命令，这里我们也可以测试以下不同实例之间是否可以使用ssh，scp命令。

# 作业

1. 配置网络及路由，创建实例，截图网络拓扑，实例之间的连通性
2. 解释连接过程中的消息的传递过程
3. 多个网络间的实例如何互通，解释设计方案，截图网络拓扑。