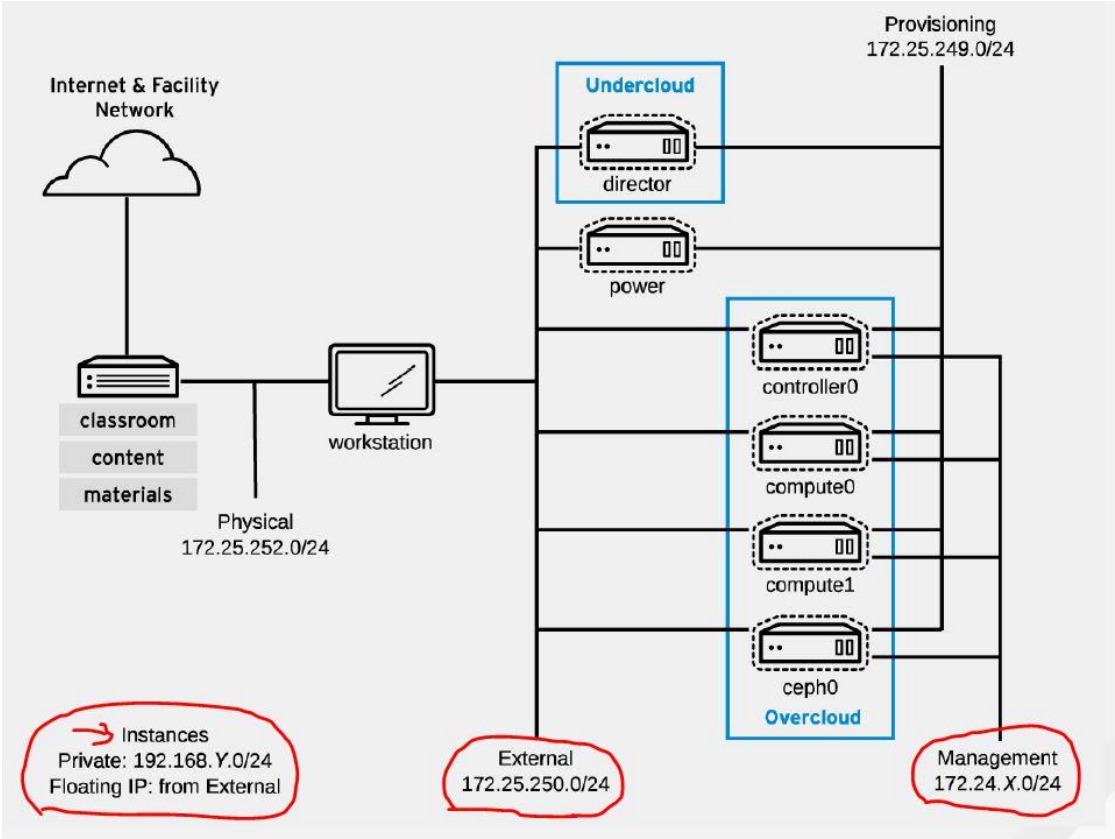
Openstack_v10(CL-210 课程)_01

2018 年 3 月 6 日
15:50



Openstack10 实验环境拓扑图

环境中包括一个 UnderCloud 云下平台和一个 Overcloud 云上平台

UnderCloud 中包括一个 director（一台部署主机），通过该主机部署所有 Overcloud 中的 Openstack 环境。

Director 本身也是一台 all in one 的 Openstack。

通过**真实主机**执行如下命令可以远程该主机：

```
# ssh stack@director
```

OverCloud 是 Openstack 云平台：

Controller0 是云控制台节点（Web 登陆界面在该节点）

Compute0 和 compute1 是计算节点（所有虚拟机运行在计算节点）

通过在 **director** 主机执行如下命令可以远程这些主机：

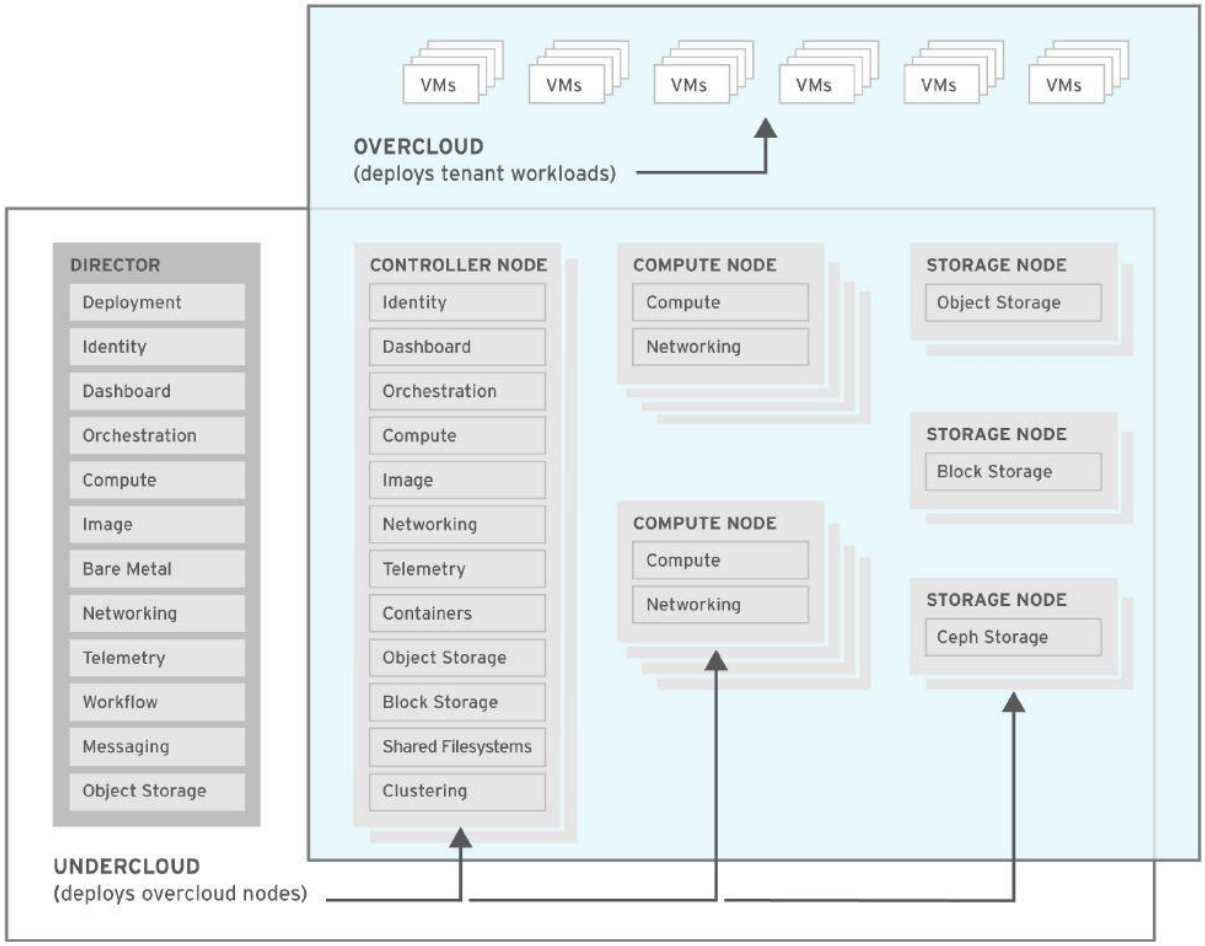
```
# ssh controller0                    //默认登陆账户为 heat-admin
# ssh compute0
# ssh compute1
# ssh ceph0
```

classroom 主机为讲师机，为环境中提供共享服务（如 DNS，DHCP 等），实验过程中此主机必须开启。

power 电源管理虚拟机，通过该主机可以远程给 OverCloud 中的主机开机或关机，实验过程中此主机必须开启。

workstation 是有图形的一台客户端主机（帐户名：student，密码：student）（账户：root，密码：redhat）。

组件拓扑图：


VMs
VMs
VMs
VMs
VMs
VMs
OVERCLOUD
(deploys tenant workloads)
DIRECTOR
Deployment
Identity
Dashboard
Orchestration
Compute
Image
Bare Metal
Networking
Telemetry
Workflow
Messaging
Object Storage
CONTROLLER NODE
Identity
Dashboard
Orchestration
Compute
Image
Networking
Telemetry
Containers
Object Storage
Block Storage
Shared Filesystems
Clustering
COMPUTE NODE
Compute
Networking
COMPUTE NODE
Compute
Networking
STORAGE NODE
Object Storage
STORAGE NODE
Block Storage
STORAGE NODE
Ceph Storage
UNDERCLOUD
(deploys overcloud nodes)

Director 主机是一台 All in One 的 Openstack，该主机安装了 Openstack 所需的所有组建：

Identity(Keystone）用户认证与授权组件；

Dashboard(Horizon) Web 控制台组件；

Orchestration(Heat) 使用模板批量部署虚拟机的组件；

Compute(Nova) 计算组件，提供虚拟机的管理功能；

Networking(Neutron) 网络管理组件；

Image(Glance): 虚拟机镜像管理组件；

Telemetry 测量计费组件；

Bare Metal(Ironic) 裸机部署，允许将 openstack 组件安装在裸机上；

Messaging(RabbitMQ) 消息队列组件（为 Openstack 所有组件之间提供通信平台）；

Object Storage（Swift） 对象存储组件。

director 基于 TripleO 项目（openstack on Openstack)

Controller 节点（Openstack 的控制端，是 Openstack 的用户入口，安装了与 Director 类似的组件，但不是必须都要安装在该主机）

Compute 节点（通过安装 Nova 组件管理虚拟机，为整个 Openstack 提供计算能力）

为了提供更好的性能，计算节点仅需要安装 Nova 和 Neutron 组件即可。

实验前比做的环境检查工作：

 # ssh statck@director

[stack@director ~]$ openstack compute service list

```
+----+----------------------+-----------------------------------+------------+-------------+-----------+-----------------------------------+
| ID | Binary               | Host                              | Zone       | Status      | State     | Updated At                        |
+----+----------------------+-----------------------------------+------------+-------------+-----------+-----------------------------------+
|  1 | nova-cert            | director.lab.example.com          | internal | enabled   | up        | 2018-03-05T23:01:18.000000 |
|  2 | nova-scheduler | director.lab.example.com          | internal | enabled   | up        | 2018-03-05T23:01:18.000000 |
|  3 | nova-conductor | director.lab.example.com          | internal | enabled   | up        | 2018-03-05T23:01:16.000000 |
|  4 | nova-compute   | director.lab.example.com          | nova       | enabled   |down     | 2018-03-05T23:01:20.000000 |
+----+----------------------+-----------------------------------+------------+-------------+-----------+-----------------------------------+
```

每次环境重置后，nova-compute 服务有可能未启动（不是每次）。

使用如下命令，重启 nova-compute 服务即可解决该问题：

[stack@director ~]$ sudo openstack-service restart nova-compute

```
+----+----------------------+-----------------------------------+------------+-------------+-----------+
| ID | Binary               | Host                              | Zone       | Status      | State     |
+----+----------------------+-----------------------------------+------------+-------------+-----------+
|  1 | nova-cert            | director.lab.example.com          | internal | enabled   | up        |
|  2 | nova-scheduler | director.lab.example.com          | internal | enabled   | up        |
|  3 | nova-conductor | director.lab.example.com          | internal | enabled   | up        |
|  4 | nova-compute   | director.lab.example.com          | nova       | enabled   |up       |
+----+----------------------+-----------------------------------+------------+-------------+-----------+
```

或者是使用 systemctl 命令也可以，速度会更快，因为该命令更底层：

[stack@director ~]$ sudo systemctl status openstack-nova-compute.service

提示：如果 nova-compute 服务未启动，不要继续后面的实验！

启动实验环境虚拟机：

检查是否所有实验虚拟机都已经正常开启。

使用 rht-vmctl 可以快速启动虚拟机：

#rht-vmctl start all

使用该命令后 controller0、compute0、ceph0 依然未启动。

此时可以使用 rht-vmctl start 虚拟机名称，启动虚拟机（如:rht-vmctl start controller0)

或者通过在 director 主机执行如下命令亦可，如：

[stack@director ~]$ openstack server start overcloud-cephstorage-0

[stack@director ~]$ openstack server list -c ID -c Name -c Status

```
+--------------------------------------+-------------------------+---------+
| ID                                   |          Name           | Status  |
+--------------------------------------+-------------------------+---------+
| c3cc04ff-3a5e-47e9-afad-09e417ab47c4 | overcloud-compute-0     | SHUTOFF |
| 2799c626-db04-4d63-b875-a96006a02de9 | overcloud-cephstorage-0 | ACTIVE  |
| 9d03a91b-96cc-441e-af96-6e7343e6db92 | overcloud-controller-0  | ACTIVE  |
+--------------------------------------+-------------------------+---------+
```

关闭实验环境虚拟机：

1. 先关闭 Overcloud 中的虚拟机（controller0,compute0,ceph0)
2. 再关闭其他虚拟机 rht-vmctl stop all

如果没有安装前面的要求顺序关闭虚拟机，就有可能导致虚拟机状态的不同步！

使用物理机使用 rht-vmctl 命令查看的状态与 director 主机使用 openstack server list 查看的状态不一致！
如果不一致，下面表格中的命令很重要！

rht-vmctl 状态	openstack 状态: **SHUTOFF**	openstack 状态: **ACTIVE**
DEFINED	openstack server start <node>	openstack server reboot <node>
RUNNING	nova reset-state --active <node>	无需任何操作

检查整个 OverCloud 环境的健康状态：

登陆 workstation 主机，执行 lab 脚本，执行如下命令，可以检查实验环境的健康状态：

[student@workstation ~]$ lab overcloud-health-check setup

Check the overcloud environment? (y|N)..... **Y**

Verifying overcloud nodes

- Retrieving state for overcloud-compute-0..... SUCCESS
- Retrieving state for overcloud-cephstorage-0..... SUCCESS
- Retrieving state for overcloud-controller-0..... SUCCESS
- Waiting for overcloud-compute-0 to be available..... SUCCESS

· Waiting for overcloud-cephstorage-0 to be available..... SUCCESS

- Waiting for overcloud-controller-0 to be available..... SUCCESS
- Verifying ceph0 access..... SUCCESS
- Starting ceph0 disk arrays and restarting ceph.target..... SUCCESS
- Checking RabbitMQ (5m timer)..... SUCCESS

- Ensuring the Downloads directory exists..... SUCCESS
· Ensuring OpenStack services are running, please wait..... SUCCESS

查看 Openstack 相关服务是否启动（在 controller0 执行）：

[heat-admin@overcloud-controller-0 ~]$ systemctl -t service list-units open* neutron*

使用脚本创建资源验证环境：

登陆 workstation 主机执行如下命令：

[student@workstation ~]$ lab deployment-review setup

该脚本会创建如下资源：

创建<production>项目，创建<operator1>用户,密码:redhat

创建 network 网络与 subnet 子网（provider-172.25.250）

创建 image 虚拟机镜像

创建 Flavor 主机箱

使用命令行手动创建资源：

登陆 workstation 主机执行如下命令：

[student@workstation ~]$ cat admin-rc **//该文件中有 Openstack 的账户与密码**

```
unset OS_SERVICE_TOKEN
export OS_AUTH_URL=http://172.25.250.50:5000/v2.0
export OS_PASSWORD=favmvajW7WEh7MwEnUKnB4MkB
export OS_REGION_NAME=regionOne
export OS_TENANT_NAME=admin
export OS_USERNAME=admin
```

```
export PS1='[\u@\h \W(admin-admin)]\$ '
[student@workstation ~]$ source admin-rc                    //加载账户文件

[student@workstation ~(admin-admin)]$ openstack project create --description Production production
//创建项目，项目名称为 production
[student@workstation ~(admin-admin)]$ openstack project list
+----------------------------------+-------------+
| ID                               | Name        |
+----------------------------------+-------------+
| 1febd350c5994e53a0d5aec224f30d96 | service     |
| de8ae5483fad465f8ef6caa8f30f4719 | admin       |
| ff8e4c69cd0f4167afc2fa1e7c8f46ac | production  |
+----------------------------------+-------------+
[student@workstation ~(admin-admin)]$ openstack project show production     //显示项目信息

[student@workstation ~(admin-admin)]$ openstack user list                   //显示用户列表
+----------------------------------+-----------+
| ID                               | Name      |
+----------------------------------+-----------+
| f599788fec954a3298edd2ecd1003352 | admin     |
| de3b2c86585f4fcca2a8de1df60348aa | neutron   |
| e5a6b8abe19d43a89185b269264e7c5e | heat      |
+----------------------------------+-----------+
[student@workstation ~(admin-admin)]$ openstack help user create
```

```
usage: openstack user create [-h] [-f {json,shell,table,value,yaml}]
                             [-c COLUMN] [--max-width <integer>] [--noindent]
                             [--prefix PREFIX] [--project <project>]
                             [--password <password>] [--password-prompt]
                             [--email <email-address>] [--enable | --disable]
                             [--or-show]
                             <name>
```

**[student@workstation ~(admin-admin)]$ openstack user create --project production **

> --password redhat --email student@example.com operation1

//创建用户 operation1，密码:redhat，该用户输入 production 项目组

[student@workstation ~(admin-admin)]$ openstack user list **//查看账户列表**

[student@workstation ~(admin-admin)]$ cp admin-rc operation-rc **//拷贝环境变量并修改**

文件修改后，内容如下

[student@workstation ~(admin-admin)]$ cat operation-rc

```
unset OS_SERVICE_TOKEN
export OS_AUTH_URL=http://172.25.250.50:5000/v2.0
export OS_PASSWORD=redhat
export OS_REGION_NAME=regionOne
export OS_TENANT_NAME=production
export OS_USERNAME=operation1
export PS1='[\u@\h \W(operation-production)]\$ '
```

```
[student@workstation ~(admin-admin)]$ source operation-rc                    //加载新的账户信息

[student@workstation ~]$ openstack keypair --help
Command "keypair" matches:
  keypair create
  keypair delete
  keypair list
  keypair show


[student@workstation ~(operation-production)]$ openstack keypair create operation1-keypair1 > operation1-keypair1.key
//创建密钥，并导出到文件
[student@workstation ~(operation-production)]$ openstack keypair list
+---------------------------+-------------------------------------------------+
| Name                      |                   Fingerprint                   |
+---------------------------+-------------------------------------------------+
| operation1-keypair1  | 93:37:3b:4e:12:22:25:3d:b4:7b:4b:2a:77:a9:ec:8a |
+---------------------------+-------------------------------------------------+
[student@workstation ~(operation-production)]$ chmod 600 operation1-keypair1.key
//修改密钥权限


[student@workstation ~(operation-production)]$ openstack --help |grep security
[student@workstation ~(operation-production)]$ openstack  security group create production-ssh
```

//创建安全组

```
[student@workstation ~(operation-production)]$ openstack help security group rule create
usage: openstack security group rule create [-h]
                                            [-f {json,shell,table,value,yaml}]
                                            [-c COLUMN]
                                            [--max-width <integer>]
                                            [--noindent] [--prefix PREFIX]
                                            [--src-ip <ip-address> | --src-group <group>]
                                            [--dst-port <port-range>]
                                            [--icmp-type <icmp-type>]
                                            [--icmp-code <icmp-code>]
                                            [--protocol <protocol>]
                                            [--ingress | --egress]
                                            [--ethertype <ethertype>]
                                            [--project <project>]
                                            [--project-domain <project-domain>]
                                            <group>
[student@workstation ~(operation-production)]$ openstack  security group rule create \
> --protocol tcp --dst-port 22   production-ssh
```

//创建安全组规则，允许 ssh

```
[student@workstation ~(operation-production)]$ openstack  security group rule create \
> --protocol icmp   production-ssh
```

//创建安全组规则，允许 icmp

[student@workstation ~(operation-production)]$ openstack network list

//查看网络与子网

[student@workstation ~(operation-production)]$ openstack network create production-network1

//创建网络

[student@workstation ~(operation-production)]$ openstack help subnet create

```
usage: openstack subnet create [-h] [-f {json,shell,table,value,yaml}]
                               [-c COLUMN] [--max-width <integer>]
                               [--noindent] [--prefix PREFIX]
                               [--project <project>]
                               [--project-domain <project-domain>]
                               [--subnet-pool <subnet-pool> | --use-default-subnet-pool]
                               [--prefix-length <prefix-length>]
                               [--subnet-range <subnet-range>]
                               [--dhcp | --no-dhcp] [--gateway <gateway>]
                               [--ip-version {4,6}]
                               [--ipv6-ra-mode {dhcpv6-stateful,dhcpv6-stateless,slaac}]
                               [--ipv6-address-mode {dhcpv6-stateful,dhcpv6-stateless,slaac}]
                               --network <network>
                               [--allocation-pool start=<ip-address>,end=<ip-address>]
                               [--dns-nameserver <dns-nameserver>]
                               [--host-route destination=<subnet>,gateway=<ip-address>]
                               name
```

**[student@workstation ~(operation-production)]$ openstack subnet create **

**> --subnet-range 192.168.0.0/24 **

```
>  --dhcp \
>  --dns-nameserver 172.25.250.254 \
>  --network production-network1 \
>  production-subnet1
```
//在 production-network1 网络中创建一个子网

```
[student@workstation ~(operation-production)]$ openstack router create production-router1
```
//创建路由器

```
[student@workstation ~(operation-production)]$ neutron --help |grep route
[student@workstation ~(operation-production)]$ neutron router-gateway-set production-router1 provider-172.25.250
```
//将路有器外网的接口，连接到环境提前准备好的外部网络中

```
[student@workstation ~(operation-production)]$ openstack help router add subnet
usage: openstack router add subnet [-h] <router> <subnet>
[student@workstation ~(operation-production)]$ openstack router add subnet production-router1 production-subnet1
```
//为路有器连接内外接口，连接 production-subnet1 这个网络

//此时，可以通过登陆 Web 控制台，看看网络拓扑图（用户 operation1,密码:redhat)

//Project-->Network-->Network Topology

[student@workstation ~(operation-production)]$ openstack help floating ip create

```
usage: openstack floating ip create [-h] [-f {json,shell,table,value,yaml}]
                                    [-c COLUMN] [--max-width <integer>]
                                    [--noindent] [--prefix PREFIX]
                                    [--subnet <subnet>] [--port <port>]
                                    [--floating-ip-address <floating-ip-address>]
                                    [--fixed-ip-address <fixed-ip-address>]
                                    <network>
```

[student@workstation ~(operation-production)]$ openstack network list

```
+--------------------------------------+-----------------------+--------------------------------------+
| ID                                   | Name                  | Subnets                              |
+--------------------------------------+-----------------------+--------------------------------------+
| 213cda9b-5df3-4c53-abeb-4995f565564a | provider-172.25.250   | d344e1ee-90af-4ebb-85fd-8b53c807710f |
| 6aa0085c-645c-4df7-9892-2e5f4f68fd77 | production-network1   | 6532fe51-ca0a-4aff-ad96-e0f82ea7de23 |
+--------------------------------------+-----------------------+--------------------------------------+
```

[student@workstation ~(operation-production)]$ openstack floating ip create provider-172.25.250

//创建浮动 IP

[student@workstation ~(operation-production)]$ openstack security group list

[student@workstation ~(operation-production)]$ openstack image list

[student@workstation ~(operation-production)]$ openstack network list

[student@workstation ~(operation-production)]$ openstack flavor list

[student@workstation ~(operation-production)]$ openstack keypair list

```
[student@workstation ~(operation-production)]$ openstack help server create
usage: openstack server create [-h] [-f {json,shell,table,value,yaml}]
                               [-c COLUMN] [--max-width <integer>]
                               [--noindent] [--prefix PREFIX]
                               (--image <image> | --volume <volume>) --flavor
                               <flavor>
                               [--security-group <security-group-name>]
                               [--key-name <key-name>]
                               [--property <key=value>]
                               [--file <dest-filename=source-filename>]
                               [--user-data <user-data>]
                               [--availability-zone <zone-name>]
                               [--block-device-mapping <dev-name=mapping>]
                               [--nic <net-id=net-uuid,v4-fixed-ip=ip-addr,v6-fixed-ip=ip-addr,port-id=port-uuid>]
                               [--hint <key=value>]
                               [--config-drive <config-drive-volume>|True]
                               [--min <count>] [--max <count>] [--wait]
                               <server-name>
[student@workstation ~(operation-production)]$ openstack  server create \
> --image rhel7 \
> --flavor m1.web \
> --security-group production-ssh \
> --nic net-id=production-network1 \
> --key-name operation1-keypair1 \
```

```
> --wait  production-web1
```
//创建虚拟机实例

```
[student@workstation ~(operation-production)]$ openstack  server show production-web1
```
//查看虚拟机状态

```
[student@workstation ~(operation-production)]$ openstack  server add floating ip production-web1 172.25.250.N
```
//为虚拟主机分配浮动 IP（浮动 IP 可以连接外网）

```
[student@workstation ~(operation-production)]$ ping 172.25.250.N
```
//ping 浮动 IP，检查虚拟机连通性

```
[student@workstation ~(operation-production)]$ ssh -i operation1-keypair1.key cloud-user@172.25.250.N
```
//通过浮动 IP，远程虚拟机

登陆 workstation 主机执行如下命令，可以清除前面的实验：

```
[student@workstation ~]$ lab deployment-review cleanup
```