chapitre6:

1- En tant qu'utilisateur operator1 du projet finance, créez un conteneur d'objet appelé customer-dropbox.

Sur workstation, approvisionnez le fichier d'environnement d'identité de l'utilisateur operator1 dans le projet finance.

Utilisez la commande openstack container create pour créer le conteneur.

#source ~/operator1-finance-rc

# openstack container create customer-dropbox

2- Téléchargez statement-20200710.zip dans customer-dropbox. Configurez statement-20200710.zip de sorte qu’il expire après 2 minutes. Vérifiez que l’objet n’est pas disponible après la date d’expiration.

2.1- Utilisez la commande openstack object create pour télécharger /home/student/statement-20200710.zip dans customer-dropbox.

# openstack object create --name statement-20200710.zip customer-dropbox /home/student/statement-20200710.zip

2.2- Utilisez la commande swift post pour définir l’expiration de statement-20200710.zip sur 120 secondes à partir de maintenant.

#swift post customer-dropbox statement-20200710.zip -H "X-Delete-After:120"

2.3: Utilisez la commande swift stat pour afficher l’heure d’expiration de l’objet statement-20200710.zip.

# swift stat customer-dropbox statement-20200710.zip

Notez que la valeur X-Delete-After a été automatiquement convertie en valeur X-Delete-At , qui est une date et une heure absolues au format d’horodatage Unix.

2.4: Convertissez la valeur X-Delete-At au format datetime standard.

# date -d @1594352552 (cette valeur est la valeur de X-Delete-At)

2.5- L’objet statement-20200710.zip doit être indisponible après l’heure indiquée à l’étape précédente.

#openstack object show customer-dropbox statement-20200710.zip

3- Même si l’objet statement-20200710.zip n’est pas disponible, il ne peut pas être supprimé immédiatement. Affichez les journaux Swift sur controller0.

3.1- Connectez-vous à controller0 en tant que heat-admin, puis devenez root.

**# ssh heat-admin@controller0**

**# sudo -i**

3.2: Utilisez la commande grep pour rechercher des messages object-expirer dans /var/log/containers/swift/swift.log.

# grep object-expirer /var/log/containers/swift/swift.log

Le démon object-expirer vérifie régulièrement les objets expirés dans le compte masqué .expiring\_objects.

3.3- Lorsqu’un objet est arrivé à expiration, il finit par être supprimé, mais cela peut prendre plusieurs minutes. Utilisez la commande grep pour rechercher la suppression de l’objet statement-20200710.zip. Déconnectez-vous lorsque vous avez terminé.

# grep DELETE /var/log/containers/swift/swift.log

=============================================

chapitre7

**tenant networks and provider networks**

The primary difference between tenant networks and provider networks revolves around who provisions them. Provider networks are created by the OpenStack administrator on behalf of tenants and can be dedicated to a particular tenant, shared by a subset of tenants (see [RBAC for networks](https://specs.openstack.org/openstack/neutron-specs/specs/liberty/rbac-networks.html)) or shared by all tenants. On the other hand, tenant networks are created by tenants for use by their instances and cannot be shared (based upon default policy settings).

To deploy an instance for public access there are two methods; use a provider network, or use a tenant network that has a routed connection to an external network.

Use a provider network if you have no need for flexibility, or the networking services offered by OpenStack. A tenant network gives you the flexibility to change firewall rules as required, or change the network layout of an application without requiring a network administrator.

Provider networks can be of type local, flat, VLAN, GRE, VXLAN, or GENEVE.

OpenStack requires at least one external network and this is created during overcloud deployment. An external network is a provider network that is marked as external. By marking a provider network as external, it can be connected to an OpenStack router as a gateway network. This configuration uses Network Address Translation (NAT) by default.

Lorsque vous utilisez tenant networks, vous pouvez choisir n'importe quelle plage de sous-réseaux et n'importe quelle configuration DHCP sans affecter aucun autre utilisateur ou projet cloud. Toutefois, lorsque vous utilisez des provider networks, vous êtes contraint de travailler dans la configuration d'infrastructure existante. Par exemple, la plage de réseau à utiliser dans le sous-réseau associé est déterminée par les administrateurs réseau. Si vous configurez une plage réseau qui ne correspond pas, votre trafic réseau ne pourra pas être acheminé et aucune communication vers ou depuis l'instance ne fonctionnera.

### **Managing Networks Using the CLI**

External, provider, and tenant networks can be created using the CLI, however you must have administrative privileges to create a provider network.

Use the openstack network create command to create the provider network. The external, shared, provider-network-type, and provider-physical-network options are required.

The --share option allows all projects to use the virtual network. The --external option specifies that the virtual network is external. Not all provider networks are external; the --internal option creates an internal provider network. The --provider-network-type and --provider-physical-network options connect the flat virtual network to the flat physical network.

### **Hébergement d’instances publiques sur des réseaux tenants**

Pour qu’une instance d’un réseau tenant soit disponible publiquement, plusieurs exigences pour les routeurs et les adresses IP flottante doivent être satisfaites :

* Le réseau tenant doit être connecté à un routeur.
* Le routeur doit disposer d’un réseau externe configuré à l’aide de l’option --external-gateway.
* Le réseau externe doit disposer d’un pool d’allocation d’adresses IP à partir duquel créer des adresses IP flottantes.
* Une adresse IP flottante doit être ajoutée à l'instance.

### **Présentation des adresses IP flottantes**

Dans la terminologie OpenStack, une adresse IP flottante est une adresse IP allouée depuis un pool pour un réseau marqué comme *externe*. Le pool d’allocation d'adresses IP flottantes est créé en même temps que le réseau externe. Une adresse IP flottante est une adresse IP routable qui est publiquement accessible. Les adresses IP flottantes autorisent la communication entre le réseau externe et l'instance. Les utilisateurs cloud peuvent associer une adresse IP flottante à une instance après qu'elle a été lancée. Une fois qu'une adresse IP flottante est associée à une instance, les utilisateurs cloud peuvent la gérer à la volée. Ils peuvent, par exemple, dissocier l'adresse IP flottante et en associer une nouvelle. Le service de mise en réseau OpenStack met automatiquement à jour les entrées associées, comme les règles de routage, les ports et les règles Netfilter.

Quand une adresse IP flottante est dissociée, elle devient disponible dans le pool d'adresses IP flottantes allouées au réseau externe et peut être attachée à une autre instance.

Les adresses IP flottantes offrent les avantages suivants :

* Exposition d'un service exécuté dans une instance : pour un serveur web, par exemple.
* Gestion de groupes de sécurité pour créer une gestion de l'accès au réseau évoluée. Les utilisateurs cloud peuvent allouer des adresses IP flottantes à un pool de serveurs de base de données et créer des règles pour limiter l'accès au réseau de réplication.
* Les adresses IP flottantes peuvent être associées à des instances et en être dissociées, le tout de manière dynamique, de sorte que les utilisateurs cloud peuvent, en quelques secondes, fournir l'accès à un service public et le retirer.
* Une solution haute disponibilité par le biais d'une interaction par programme avec l'API du service de mise en réseau OpenStack pour associer et dissocier des adresses IP flottantes.

lab:

## **Guided Exercise: Managing Routers and Floating IPs**

In this exercise, you will create and configure a router using the command line and the Dashboard, and create a set of floating IP addresses and allocate them to an instance.

**Outcomes**

You should be able to:

* Create and manage a router.
* Verify the connectivity of the router.
* Create and manage floating IP addresses.
* Verify the external connectivity of an instance.

As the student user on the workstation machine, use the lab command to prepare your system for this exercise.

This command ensures that all resources required for the exercise are present.

[student@workstation ~]$ **lab public-routers start**

**Procedure 7.2. Instructions**

This guided exercise is in two parts. In part 1 you use the CLI, and in part 2 you use the Dashboard.

1. As the developer1 user in the finance project, create a new router and verify the resources. Use the values in the following table:

| **Setting** | **Router** |
| --- | --- |
| Router Name | finance-router1 |
| External network | provider-datacentre |
| Network subnet | finance-subnet3 |

* 1. On workstation, source the identity environment file for the developer1 user in the finance project.  
     [student@workstation ~]$ **source ~/developer1-finance-rc**

Use the openstack router create command to create the finance-router1 router.  
[student@workstation ~(developer1-finance)]$ **openstack router create \**

> **finance-router1**

+-------------------------+------------------------------------------------------+

| Field | Value |

+-------------------------+------------------------------------------------------+

| admin\_state\_up | UP |

| availability\_zone\_hints | None |

| availability\_zones | None |

| created\_at | 2020-07-14T16:00:45Z |

| description | |

| external\_gateway\_info | null |

| flavor\_id | None |

| id | 6ea98144-c243-41cc-8064-c8281821c6d0 |

| location | cloud='', project.domain\_id=, |

| | project.domain\_name='Example', |

| | project.id='f76fd09fa0b14a678b5b61f9e3ec3c87', |

| | project.name='finance', region\_name='regionOne', |

| | zone= |

| name | finance-router1 |

| project\_id | f76fd09fa0b14a678b5b61f9e3ec3c87 |

| revision\_number | 0 |

| routes | |

| status | ACTIVE |

| tags | |

| updated\_at | 2020-07-14T16:00:45Z |

* 1. +-------------------------+------------------------------------------------------+

List the available routers to confirm that the finance-router1 has been created.  
[student@workstation ~(developer1-finance)]$ **openstack router list**

+--------------------+-----------------+--------+-------+---------------------+

| ID | Name | Status | State | Project |

+--------------------+-----------------+--------+-------+---------------------+

| 6ea98144-c243-41cc | finance-router1 | ACTIVE | UP | f76fd09fa0b14a678b5 |

| -8064-c8281821c6d0 | | | | b61f9e3ec3c87 |

* 1. +--------------------+-----------------+--------+-------+---------------------+

Connect the router to the finance-subnet3 subnet.  
[student@workstation ~(developer1-finance)]$ **openstack router add subnet \**

* 1. > **finance-router1 finance-subnet3**

Set the external network provider-datacentre as the gateway for the router.  
[student@workstation ~(developer1-finance)]$ **openstack router set \**

> **--external-gateway provider-datacentre \**

* 1. > **finance-router1**

1. Test the routing from finance-server1.

Retrieve the console URL for the finance-server1 instance.  
[student@workstation ~(developer1-finance)]$ **openstack console url show \**

> **finance-server1**

+-------+------------------------------------------------------------------------+

| Field | Value |

+-------+------------------------------------------------------------------------+

| type | novnc |

| url | http://172.25.250.50:6080/vnc\_auto.html?path=%3Ftoken%3Df541d3ea- |

| | af9b-4e6c-9881-a9604a9e87e1 |

* 1. +-------+------------------------------------------------------------------------+
  2. Open a browser and navigate to the console URL previously obtained.
  3. Log in to finance-server1 using root as the user name and redhat as the password.

Use the ping command to verify workstation server connectivity.  
[root@finance-server1 ~]# **ping -c3 workstation**

PING workstation (172.25.250.9) 56(84) bytes of data.

64 bytes from workstation.lab.example.com (172.25.250.9): icmp\_seq=1 ttl=63 time=2.48 ms

64 bytes from workstation.lab.example.com (172.25.250.9): icmp\_seq=2 ttl=63 time=2.82 ms

64 bytes from workstation.lab.example.com (172.25.250.9): icmp\_seq=3 ttl=63 time=1.31 ms

--- workstation ping statistics ---

3 packets transmitted, 3 received, 0% packet loss, time 9ms

* 1. rtt min/avg/max/mdev = 1.310/2.203/2.817/0.646 ms
  2. Log out from finance-server1.  
     [root@finance-server1 ~]# **logout**

1. In the workstation terminal, associate a floating IP address to the finance-server1 instance and verify it.

Use the openstack floating ip create command to create one floating IP address in the provider-datacentre external network.  
**Note**Your floating IP address may differ from the one shown below.  
[student@workstation ~(developer1-finance)]$ **openstack floating ip create \**

> **provider-datacentre**

+---------------------+----------------------------------------------------------+

| Field | Value |

+---------------------+----------------------------------------------------------+

| created\_at | 2020-07-14T16:48:40Z |

| description | |

| dns\_domain | |

| dns\_name | |

| fixed\_ip\_address | None |

| floating\_ip\_address | 172.25.250.125 |

| floating\_network\_id | ef95203b-7c9f-46c0-b328-e51aa7729798 |

| id | cda15c6c-3325-49f5-a05a-740761aec953 |

| location | Munch({'cloud': '', 'region\_name': 'regionOne', 'zone': |

| | None, 'project': Munch({'id': |

| | 'f76fd09fa0b14a678b5b61f9e3ec3c87', 'name': 'finance', |

| | 'domain\_id': None, 'domain\_name': 'Example'})}) |

| name | 172.25.250.125 |

| port\_details | None |

| port\_id | None |

| project\_id | f76fd09fa0b14a678b5b61f9e3ec3c87 |

| qos\_policy\_id | None |

| revision\_number | 0 |

| router\_id | None |

| status | DOWN |

| subnet\_id | None |

| tags | [] |

| updated\_at | 2020-07-14T16:48:40Z |

* 1. +---------------------+----------------------------------------------------------+

Associate the floating IP address with the finance-server1 instance.  
[student@workstation ~(developer1-finance)]$ **openstack server add floating ip \**

* 1. > **finance-server1 172.25.250.*125***

Verify that the floating IP address has been attached to the finance-server1 instance.  
[student@workstation ~(developer1-finance)]$ **openstack server list \**

> **-c Name -c Networks**

+-----------------+------------------------------------------------+

| Name | Networks |

+-----------------+------------------------------------------------+

| finance-server2 | finance-network4=192.168.4.68 |

| finance-server1 | finance-network3=192.168.3.151, 172.25.250.125 |

* 1. +-----------------+------------------------------------------------+

Now you can log in using SSH to the finance-server1 instance from workstation.  
[student@workstation ~(developer1-finance)]$ **ssh -i .ssh/example-keypair \**

> **cloud-user@172.25.250.125**

Activate the web console with: systemctl enable --now cockpit.socket

This system is not registered to Red Hat Insights. See https://cloud.redhat.com/

To register this system, run: insights-client --register

Last login: Tue Jul 14 13:11:09 2020 from 172.25.250.9

* 1. [cloud-user@finance-server1 ~]$

Log out from finance-server1.  
[cloud-user@finance-server1 ~]$ **logout**

Connection to 172.25.250.125 closed.

* 1. [student@workstation ~(developer1-finance)]$

1. Disassociate the floating IP address from the finance-server1 instance and release the floating IP address.

Disassociate the 172.25.250.*N* floating IP address from the finance-server1 instance.  
**Note**Remember to use the floating IP address value from your output.  
[student@workstation ~(developer1-finance)]$ **openstack server remove floating ip \**

* 1. > **finance-server1 172.25.250.125**

Release the 172.25.250.*N* floating IP address.  
[student@workstation ~(developer1-finance)]$ **openstack floating ip delete \**

* 1. > **172.25.250.125**
  2. List the available floating IP addresses to confirm the deletion of 172.25.250.*N*.  
     [student@workstation ~(developer1-finance)]$ **openstack floating ip list**

1. Disconnect and remove the finance-router1 router.

Disconnect the router from the finance-subnet3 subnet.  
[student@workstation ~(developer1-finance)]$ **openstack router remove subnet \**

* 1. > **finance-router1 finance-subnet3**

Remove the external provider-datacentre network as the gateway for the router.  
[student@workstation ~(developer1-finance)]$ **openstack router unset \**

* 1. > **--external-gateway finance-router1**

Delete the finance-router1 router.  
[student@workstation ~(developer1-finance)]$ **openstack router delete \**

* 1. > **finance-router1**
  2. Verify that the finance-router1 router has been deleted.  
     [student@workstation ~(developer1-finance)]$ **openstack router list  
     Note**Use the Dashboard to perform the following steps.

1. In the Dashboard as developer1, create a router and then verify the resources. Use the values in the following table:

| **Setting** | **Router** |
| --- | --- |
| Router name | finance-router2 |
| External network | provider-datacentre |
| Network subnet | finance-subnet4 |

* 1. Log in to the Dashboard at http://dashboard.overcloud.example.com using Example as the domain, developer1 as the user, and redhat as the password. Confirm that the selected project is finance.
  2. Navigate to Project → Network → Routers and click Create Router to create a new router.
  3. Enter finance-router2 as the router name. From the External Network list, select provider-datacentre and then click Create Router.
  4. Click the name of the router, finance-router2, to access its details. Click the Interfaces tab to manage the interfaces for the router.
  5. Click Add Interface to add a new interface. From the Subnet list, select finance-network4: 192.168.4.0/24 (finance-subnet4) as the subnet and click Submit.
  6. To verify the resources have been created, navigate to Project → Network → Network Topology.

1. Create and associate a floating IP address with the finance-server2 instance.
   1. Navigate to Project → Network → Floating IPs and click Allocate IP To Project.
   2. Ensure that provider-datacentre is set as the Pool, and then click Allocate IP.
   3. Click Associate in the row of the floating IP address. When the Manage Floating IP Associations window opens, select finance-server2: 192.168.4.N from the Port to be associated list. Click Associate.
   4. Navigate to Project → Compute → Instances
   5. Review the IP Address column and notice the IP address for finance-server2.
2. Disassociate the floating IP address from the finance-server2 instance and release the floating IP.
   1. Navigate to Project → Compute → Instances, and click the arrow next to the Create Snapshot button for the row labeled finance-server2.
   2. Select Disassociate Floating IP to disassociate the floating IP address from the instance. In the Disassociate Floating IP window, click Disassociate.
   3. Navigate to Project → Network → Floating IPs, and click the arrow next to the Associate button for the floating IP address row.
   4. Select Release Floating IP to return the floating IP address to the pool. In the Confirm Release Floating IP window, click Release Floating IP.
3. Disconnect and delete the finance-router2 router.
   1. Navigate to Project → Network → Routers.
   2. Click the name of the router, finance-router2, to access its details. Click the Interfaces tab to manage the interfaces for the router.
   3. Click Delete Interface to remove the interface. In the Confirm Delete Interface window, click Delete Interface.
   4. Click Clear Gateway to remove the external provider-datacentre network as the gateway for the router. In the Confirm Clear Gateway window, click Clear Gateway.
   5. Click the arrow next to the Set Gateway button for the finance-router2 row, and then click Delete Router. In the Confirm Delete Router window, click Delete Router.
   6. Log out of the Dashboard by clicking the developer1 menu in the upper-right corner, and then choose Sign out.

**Finish**

On the workstation machine, use the lab command to complete this exercise. This is important to ensure that resources from previous exercises do not impact upcoming exercises.

==============================================

## **Lab: Configuring Resources to Launch an Instance with Public Access**

**Performance Checklist**

In this lab, you will create a public network, as well as a router. You will create some floating IP addresses, a key pair, and various security group rules. You will launch an instance with public access.

**Outcomes**

You should be able to:

* Create a public network and its associated subnet.
* Create and configure a router.
* Set security group rules in a security group.
* Create a key pair and a floating IP.
* Launch an instance with public access.

Confirm that the workstation and overcloud virtual machines are started.

As the student user on the workstation machine, use the lab command to prepare your system for this lab.

This command ensures that all resources required for the exercise are present.

[student@workstation ~]$ **lab public-review start**

**Procedure 7.5. Instructions**

1. As the operator1 user, create the provider-datacentre external network and the provider-subnet-172.25.250 subnet. Use the following specifications:

| **Option** | **Value** |
| --- | --- |
| Name | provider-datacentre |
| Shared | Yes |
| External | Yes |
| Provider network type | flat |
| Provider physical network name | datacentre |
| Subnet name | provider-subnet-172.25.250 |
| Network subnet | 172.25.250.0/24 |
| Network gateway | 172.25.250.254 |
| DNS name server | 172.25.250.254 |
| Allocation pool | 172.25.250.101,172.25.250.189 |
| DHCP | Disabled |

* 1. On workstation, source the identity environment file for the operator1 user.  
     [student@workstation ~]$ **source operator1-production-rc**

Create an external network named provider-datacentre.  
[student@workstation ~(operator1-production)]$ **openstack network create \**

> **--external \**

> **--share \**

> **--provider-network-type flat \**

> **--provider-physical-network datacentre \**

> **provider-datacentre**

+---------------------------+----------------------------------------------------+

| Field | Value |

+---------------------------+----------------------------------------------------+

| admin\_state\_up | UP |

| availability\_zone\_hints | |

| availability\_zones | |

| created\_at | 2020-07-16T21:46:14Z |

| description | |

| dns\_domain | |

| id | 8cd2ef04-5735-481d-ac28-30441636be3c |

| ipv4\_address\_scope | None |

| ipv6\_address\_scope | None |

| is\_default | False |

| is\_vlan\_transparent | None |

| location | cloud='', project.domain\_id=, |

| | project.domain\_name='Example', |

| | project.id='44fe99b84dcd4a6b8026bf4203241a42', |

| | project.name='production', |

| | region\_name='regionOne', zone= |

| mtu | 1500 |

| name | provider-datacentre |

| port\_security\_enabled | True |

| project\_id | 44fe99b84dcd4a6b8026bf4203241a42 |

| provider:network\_type | flat |

| provider:physical\_network | datacentre |

| provider:segmentation\_id | None |

| qos\_policy\_id | None |

| revision\_number | 1 |

| router:external | External |

| segments | None |

| shared | True |

| status | ACTIVE |

| subnets | |

| tags | |

| updated\_at | 2020-07-16T21:46:14Z |

* 1. +---------------------------+----------------------------------------------------+

Create the provider-subnet-172.25.250 subnet for the external network with an allocation pool of 172.25.250.101-172.25.250.189. Disable DHCP services for the subnet and use 172.25.250.254 as the gateway as well as the DNS name server.  
[student@workstation ~(operator1-production)]$ **openstack subnet create \**

> **--subnet-range 172.25.250.0/24 \**

> **--gateway 172.25.250.254 \**

> **--dns-nameserver 172.25.250.254 \**

> **--allocation-pool start=172.25.250.101,end=172.25.250.189 \**

> **--no-dhcp \**

> **--network provider-datacentre \**

> **provider-subnet-172.25.250**

+-------------------+------------------------------------------------------------+

| Field | Value |

+-------------------+------------------------------------------------------------+

| allocation\_pools | 172.25.250.101-172.25.250.189 |

| cidr | 172.25.250.0/24 |

| created\_at | 2020-07-16T21:51:59Z |

| description | |

| dns\_nameservers | 172.25.250.254 |

| enable\_dhcp | False |

| gateway\_ip | 172.25.250.254 |

| host\_routes | |

| id | bc2e8cfb-f703-417f-af04-577f15cb8797 |

| ip\_version | 4 |

| ipv6\_address\_mode | None |

| ipv6\_ra\_mode | None |

| location | cloud='', project.domain\_id=, |

| | project.domain\_name='Example', |

| | project.id='44fe99b84dcd4a6b8026bf4203241a42', |

| | project.name='production', region\_name='regionOne', zone= |

| name | provider-subnet-172.25.250 |

| network\_id | 8cd2ef04-5735-481d-ac28-30441636be3c |

| prefix\_length | None |

| project\_id | 44fe99b84dcd4a6b8026bf4203241a42 |

| revision\_number | 0 |

| segment\_id | None |

| service\_types | |

| subnetpool\_id | None |

| tags | |

| updated\_at | 2020-07-16T21:51:59Z |

* 1. +-------------------+------------------------------------------------------------+

As the developer1 user, create the production-router1 router and connect it to the production-subnet1 project subnet, and set the external gateway to provider-datacentre.

1. Source the credentials for the developer1 user.  
   [student@workstation ~(operator1-production)]$ **source developer1-production-rc**

Create the production-router1 router.  
[student@workstation ~(developer1-production)]$ **openstack router create \**

> **production-router1**

+-------------------------+------------------------------------------------------+

| Field | Value |

+-------------------------+------------------------------------------------------+

| admin\_state\_up | UP |

| availability\_zone\_hints | None |

| availability\_zones | None |

| created\_at | 2020-07-16T22:20:38Z |

| description | |

| external\_gateway\_info | null |

| flavor\_id | None |

| id | 99910b9d-d6b4-41af-b688-aba948508401 |

| location | cloud='', project.domain\_id=, |

| | project.domain\_name='Example', |

| | project.id='44fe99b84dcd4a6b8026bf4203241a42', |

| | project.name='production', region\_name='regionOne', |

| | zone= |

| name | production-router1 |

| project\_id | 44fe99b84dcd4a6b8026bf4203241a42 |

| revision\_number | 0 |

| routes | |

| status | ACTIVE |

| tags | |

| updated\_at | 2020-07-16T22:20:38Z |

1. +-------------------------+------------------------------------------------------+

Connect the router to the production-subnet1 project subnet.  
[student@workstation ~(developer1-production)]$ **openstack router add \**

> **subnet production-router1 \**

1. > **production-subnet1**

Set the provider-datacentre as the external gateway to the router.  
[student@workstation ~(developer1-production)]$ **openstack router set \**

> **--external-gateway provider-datacentre \**

1. > **production-router1**

Create the security group production-secgroup1 and add the rules listed in the following table.

| **Option** | **Value** |
| --- | --- |
| Security group name | production-secgroup1 |
| Rules | TCP, port 22  TCP, port 80  ICMP |

Create the production-secgroup1 security group.  
[student@workstation ~(developer1-production)]$ **openstack security group \**

> **create production-secgroup1**

+-----------------+--------------------------------------------------------------+

| Field | Value |

+-----------------+--------------------------------------------------------------+

| created\_at | 2020-07-16T22:52:25Z |

| description | production-secgroup1 |

| id | 62affbaa-b52b-4d3b-b08b-29faabf24dd6 |

| location | cloud='', project.domain\_id=, project.domain\_name='Example', |

| | project.id='44fe99b84dcd4a6b8026bf4203241a42', |

| | project.name='production', region\_name='regionOne', zone= |

| name | production-secgroup1 |

| project\_id | 44fe99b84dcd4a6b8026bf4203241a42 |

| revision\_number | 1 |

| rules | created\_at='2020-07-16T22:52:25Z', direction='egress', |

| | ethertype='IPv6', id='94b4cd21-2bba-4451-9b96-5dd35077d108', |

| | updated\_at='2020-07-16T22:52:25Z' |

| | created\_at='2020-07-16T22:52:25Z', direction='egress', |

| | ethertype='IPv4', id='ad50140f-1fd8-4579-974f-340cb91b1a65', |

| | updated\_at='2020-07-16T22:52:25Z' |

| tags | [] |

| updated\_at | 2020-07-16T22:52:25Z |

1. +-----------------+--------------------------------------------------------------+

Add a rule in the production-secgroup1 security group to allow SSH access.  
[student@workstation ~(developer1-production)]$ **openstack security group rule \**

> **create --protocol tcp \**

> **--dst-port 22 \**

> **production-secgroup1**

+-------------------+------------------------------------------------------------+

| Field | Value |

+-------------------+------------------------------------------------------------+

| created\_at | 2020-07-16T22:54:42Z |

| description | |

| direction | ingress |

| ether\_type | IPv4 |

| id | c6a58eec-53fe-4f82-b68d-cc27a7e17131 |

| location | cloud='', project.domain\_id=, |

| | project.domain\_name='Example', |

| | project.id='44fe99b84dcd4a6b8026bf4203241a42', |

| | project.name='production', region\_name='regionOne', zone= |

| name | None |

| port\_range\_max | 22 |

| port\_range\_min | 22 |

| project\_id | 44fe99b84dcd4a6b8026bf4203241a42 |

| protocol | tcp |

| remote\_group\_id | None |

| remote\_ip\_prefix | 0.0.0.0/0 |

| revision\_number | 0 |

| security\_group\_id | 62affbaa-b52b-4d3b-b08b-29faabf24dd6 |

| tags | [] |

| updated\_at | 2020-07-16T22:54:42Z |

1. +-------------------+------------------------------------------------------------+

Add a security rule to allow HTTP connections using a default port of 80.  
[student@workstation ~(developer1-production)]$ **openstack security group \**

> **rule create --protocol tcp \**

> **--dst-port 80 \**

> **production-secgroup1**

+-------------------+------------------------------------------------------------+

| Field | Value |

+-------------------+------------------------------------------------------------+

| created\_at | 2020-07-16T23:00:07Z |

| description | |

| direction | ingress |

| ether\_type | IPv4 |

| id | dcee84ba-7845-46b1-957f-89f73427807e |

| location | cloud='', project.domain\_id=, |

| | project.domain\_name='Example', |

| | project.id='44fe99b84dcd4a6b8026bf4203241a42', |

| | project.name='production', region\_name='regionOne', zone= |

| name | None |

| port\_range\_max | 80 |

| port\_range\_min | 80 |

| project\_id | 44fe99b84dcd4a6b8026bf4203241a42 |

| protocol | tcp |

| remote\_group\_id | None |

| remote\_ip\_prefix | 0.0.0.0/0 |

| revision\_number | 0 |

| security\_group\_id | 62affbaa-b52b-4d3b-b08b-29faabf24dd6 |

| tags | [] |

| updated\_at | 2020-07-16T23:00:07Z |

1. +-------------------+------------------------------------------------------------+

Add a security rule to allow ICMP protocol.  
[student@workstation ~(developer1-production)]$ **openstack security group \**

> **rule create --protocol icmp \**

> **production-secgroup1**

+-------------------+------------------------------------------------------------+

| Field | Value |

+-------------------+------------------------------------------------------------+

| created\_at | 2020-07-16T23:31:39Z |

| description | |

| direction | ingress |

| ether\_type | IPv4 |

| id | 83cef166-8441-4a14-9110-d23b511be0e3 |

| location | cloud='', project.domain\_id=, |

| | project.domain\_name='Example', |

| | project.id='44fe99b84dcd4a6b8026bf4203241a42', |

| | project.name='production', region\_name='regionOne', zone= |

| name | None |

| port\_range\_max | None |

| port\_range\_min | None |

| project\_id | 44fe99b84dcd4a6b8026bf4203241a42 |

| protocol | icmp |

| remote\_group\_id | None |

| remote\_ip\_prefix | 0.0.0.0/0 |

| revision\_number | 0 |

| security\_group\_id | 62affbaa-b52b-4d3b-b08b-29faabf24dd6 |

| tags | [] |

| updated\_at | 2020-07-16T23:31:39Z |

1. +-------------------+------------------------------------------------------------+

Create the production-keypair1 key pair and save the private key to /home/student/Downloads/production-keypair1.pem. Use the chmod command to protect the private key file.

Create the production-keypair1 key pair and save the private key as /home/student/Downloads/production-keypair1.pem.  
[student@workstation ~(developer1-production)]$ **openstack keypair create \**

1. > **production-keypair1 > /home/student/Downloads/production-keypair1.pem**

Use the chmod command with a mode of 600 to protect the private key.  
[student@workstation ~(developer1-production)]$ **chmod 600 \**

1. > **/home/student/Downloads/production-keypair1.pem**

Create a random floating IP address in the provider-datacentre network.

Create a floating IP address in the provider-datacentre network. Your floating IP may differ from the output in the example below.  
[student@workstation ~(developer1-production)]$ **openstack floating ip \**

> **create provider-datacentre**

+---------------------+----------------------------------------------------------+

| Field | Value |

+---------------------+----------------------------------------------------------+

| created\_at | 2020-07-16T23:47:11Z |

| description | |

| dns\_domain | |

| dns\_name | |

| fixed\_ip\_address | None |

| floating\_ip\_address | 172.25.250.154 |

| floating\_network\_id | 8cd2ef04-5735-481d-ac28-30441636be3c |

| id | 883b0b3b-301d-4e64-ab59-1b7235900c99 |

| location | Munch({'cloud': '', 'region\_name': 'regionOne', 'zone': |

| | None, 'project': Munch({'id': |

| | '44fe99b84dcd4a6b8026bf4203241a42', 'name': |

| | 'production', 'domain\_id': None, 'domain\_name': |

| | 'Example'})}) |

| name | 172.25.250.154 |

| port\_details | None |

| port\_id | None |

| project\_id | 44fe99b84dcd4a6b8026bf4203241a42 |

| qos\_policy\_id | None |

| revision\_number | 0 |

| router\_id | None |

| status | DOWN |

| subnet\_id | None |

| tags | [] |

| updated\_at | 2020-07-16T23:47:11Z |

1. +---------------------+----------------------------------------------------------+

Create an instance named production-server9 using the following values:

| **Type** | **Value** |
| --- | --- |
| Instance name | production-server9 |
| Image | rhel8-web |
| Flavor | default |
| Key pair | production-keypair1 |
| Project network | production-network1 |
| Security group | production-secgroup1 |
| Floating IP address | 172.25.250.*154* |

Create the production-server9 instance using the resources from the table.  
[student@workstation ~(developer1-production)]$ **openstack server create \**

> **--image rhel8-web \**

> **--flavor default \**

> **--nic net-id=production-network1 \**

> **--security-group production-secgroup1 \**

> **--key-name production-keypair1 \**

> **--wait \**

> **production-server9**

+-----------------------------+--------------------------------------------------+

| Field | Value |

+-----------------------------+--------------------------------------------------+

| OS-DCF:diskConfig | MANUAL |

| OS-EXT-AZ:availability\_zone | nova |

| OS-EXT-STS:power\_state | Running |

| OS-EXT-STS:task\_state | None |

| OS-EXT-STS:vm\_state | active |

| OS-SRV-USG:launched\_at | 2020-07-17T00:26:49.000000 |

| OS-SRV-USG:terminated\_at | None |

| accessIPv4 | |

| accessIPv6 | |

| addresses | production-network1=192.168.1.127 |

| adminPass | uKe6kd2KhgUs |

| config\_drive | |

| created | 2020-07-17T00:26:30Z |

| flavor | default (8ff1eecb-15b4-4f32-bb99-4dec999a78c7) |

| hostId | 7c7a22df30865c0a78f42eed89c70fc942e0eca55f0cea89 |

| | d4d62908 |

| id | 7988da99-f71a-4a1f-9586-23c445177fb7 |

| image | rhel8-web (642e5661-0754-41bf-9906-7d34eda53050) |

| key\_name | production-keypair1 |

| name | production-server9 |

| progress | 0 |

| project\_id | 44fe99b84dcd4a6b8026bf4203241a42 |

| properties | |

| security\_groups | name='**production-secgroup1**' |

| status | ACTIVE |

| updated | 2020-07-17T00:26:49Z |

| user\_id | b3bf77638121e0abd2a5e89fa85f6649e76fd45d45a1470b |

| | ce0adcce1a5c3570 |

| volumes\_attached | |

1. +-----------------------------+--------------------------------------------------+

Associate the floating IP address with the instance.  
[student@workstation ~(developer1-production)]$ **openstack server \**

> **add floating ip \**

1. > **production-server9 172.25.250.*154***

From workstation, verify that the instance responds to web server HTTP requests. Verify that you can connect to the instance over SSH as the user cloud-user with the SSH private key in /home/student/Downloads/production-keypair1.pem. From the instance, verify that you can ping the external server materials.example.com.

From workstation, verify that the instance responds to HTTP requests.  
[student@workstation ~(developer1-production)]$ **curl http://172.25.250.*154***

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">

<head>

<title>Test Page for the Apache HTTP Server on Red Hat Enterprise Linux</title>

1. *...output omitted...*

Log in to the instance with SSH using the SSH private key.  
[student@workstation ~(developer1-production)]$ **ssh \**

> **-i /home/student/Downloads/production-keypair1.pem \**

> **cloud-user@172.25.250.*154***

Activate the web console with: systemctl enable --now cockpit.socket

This system is not registered to Red Hat Insights. See https://cloud.redhat.com/

To register this system, run: insights-client --register

Last login: Fri Jul 17 10:58:45 2020 from 172.25.250.9

1. [cloud-user@production-server9 ~]$

From the instance, ping the materials.example.com server.  
[cloud-user@production-server9 ~]$ **ping -c3 materials.example.com**

PING materials.example.com (172.25.254.254) 56(84) bytes of data.

64 bytes from classroom.example.com: icmp\_seq=1 ttl=62 time=1.76 ms

64 bytes from classroom.example.com: icmp\_seq=2 ttl=62 time=1.50 ms

64 bytes from classroom.example.com: icmp\_seq=3 ttl=62 time=0.926 ms

--- materials.example.com ping statistics ---

3 packets transmitted, 3 received, 0% packet loss, time 5ms

1. rtt min/avg/max/mdev = 0.926/1.395/1.762/0.348 ms

Exit from the instance.  
[cloud-user@production-server9 ~]$ **exit**

1. [student@workstation ~(developer1-production)]$

Using the Dashboard look at the network, subnet, router, security group and its rules, key pair, floating IP, and the instance created in the previous steps.  
**=========================================**

## **Exercice guidé: Personnalisation d’une instance au démarrage avec cloud-init**

Dans cet exercice, vous allez personnaliser deux instances, à l'aide des fonctions et fonctionnalités de cloud-init. Vous vous connecterez aux instances pour confirmer que cloud-init fonctionne et qu’il est en cours d’exécution, et vérifier que cloud-init a bien personnalisé les deux instances.

Résultats

Vous devez pouvoir personnaliser une instance à l’aide d’un fichier ou d’un script cloud-config et contrôler les opérations cloud-init en vérifiant le fichier /var/log/cloud-init.log ou que la personnalisation demandée s'est produite.

Vérifiez que workstation et les machines virtuelles de l'overcloud sont démarrées.

Connectez-vous à workstation en tant qu’utilisateur student avec le mot de passe student.

Sur workstation, exécutez la commande lab cloudapps-cloudinit start.

[student@workstation ~]$

Procédure 8.1. Instructions

1. Personnalisez une instance au moyen de Dashboard. Connectez-vous à Dashboard en utilisant Example comme domaine, developer1 comme utilisateur et redhat comme mot de passe. Cliquez sur le menu Project dans le coin supérieur droit de la fenêtre pour vous assurer que le projet actuel est bien finance. Lancez une instance nommée finance-server1 avec l'image rhel8, le gabarit default, le réseau finance-network1, le groupe de sécurité default et la paire de clés example-keypair. Créez un script de personnalisation qui inclut « Hello world! » dans l'instance /root/hello.txt.
   1. Sur workstation, ouvrez un navigateur Web et accédez à http://dashboard.overcloud.example.com. Connectez-vous à Dashboard en utilisant Example comme domaine, developer1 comme utilisateur et redhat comme mot de passe.
   2. Accédez à Compute → Instances, puis cliquez sur Launch Instance.
   3. Dans l'onglet Details, saisissez finance-server1 comme nom d'instance (Instance Name).
   4. Dans l'onglet Source, choisissez Image dans la liste Select Boot Source. Définissez l'option Create New Volume sur No. Dans la section Available, cliquez sur la flèche vers le haut de l'image rhel8-web.
   5. Dans l’onglet Flavor, dans la section Available, cliquez sur la flèche vers le haut pour le gabarit default.
   6. Dans l’onglet Networks, dans la section Available, cliquez sur la flèche vers le haut pour le réseau finance-network1.
   7. Dans l'onglet Security Groups, assurez-vous que le groupe de sécurité default a été sélectionné.
   8. Dans l'onglet Key Pair, vérifiez que la paire de clés example-keypair a été sélectionnée.

Dans l’onglet Configuration, ajoutez le contenu suivant au champ Customization Script, puis cliquez sur Launch Instance :  
 #!/bin/sh

* 1. echo 'Hello world!' > /root/hello.txt

1. Une fois l'instance active, attachez une adresse IP flottante à l'instance finance-server1.
   1. Une fois que le statut de l'instance passe à Active, attachez-y une adresse IP flottante. Dans le menu Actions de l’instance finance-server1, cliquez sur Associate Floating IP.  
       Section IP Address, cliquez sur + pour créer une nouvelle adresse IP flottante. Cliquez sur Allocate IP, puis sur Associate.
   2. Déconnectez-vous de Dashboard.
2. Sur workstation, approvisionnez le fichier developer1-finance-rc et créez un fichier user-data, appelé install\_httpd, pour personnaliser l’instance. Le script va installer le serveur Web et activer le service.
3.   
   [student@workstation ~]$
4.   
    Ce script va installer et activer le service du serveur Web.
5. Personnalisez une instance à l'aide de la ligne de commande. En tant qu'utilisateur developer1, lancez une instance nommée finance-server2 avec l'image rhel8-web, le gabarit default, le réseau finance-network1, le groupe de sécurité default et la paire de clés example-keypair. Incluez le script user-data /home/student/install\_httpd.
   1. Lancez une instance en utilisant l'option --user-data pour effectuer la personnalisation.
   2.   
      [student@workstation ~(developer1-finance)]$
   3.   
       Vérifiez que le statut de l'instance finance-server2 est active.
   4.   
      [student@workstation ~(developer1-finance)]$
6.   
    Lorsque le statut de l'instance est active, créez une adresse IP flottante et attachez-la à l'instance. Vérifiez que l'adresse IP flottante a été attribuée. Notez que les adresses IP sont fournies à partir d’un pool et sont différentes à chaque fois.
7.   
   [student@workstation ~(developer1-finance)]$
8.   
    Connectez-vous à finance-server1 pour vérifier que le script de personnalisation cloud-init a créé le fichier /root/hello.txt.
   1. Connectez-vous à finance-server1 avec ssh à l'aide de la clé privée example-keypair.
   2.   
      [student@workstation ~(developer1-finance)]$
   3.   
       Examinez /var/log/cloud-init.log pour vérifier que cloud-init s’est exécuté.
   4.   
      [cloud-user@finance-server1 ~]$
   5.   
       Assurez-vous que le fichier /root/hello.txt existe et que son contenu est correct.
   6.   
      [cloud-user@finance-server1 ~]$
   7.   
       Déconnectez-vous de finance-server1.
   8.   
      [cloud-user@finance-server1 ~]$
9.   
    Connectez-vous à finance-server2 pour vérifier que le script user-data /home/student/install\_httpd a installé et activé le service httpd.
   1. Déterminez l'adresse IP flottante allouée à l'instance finance-server2.
   2.   
      [student@workstation ~(developer1-finance)]$
   3.   
       À l'aide de ssh et de la clé privée example-keypair, connectez-vous à finance-server2.
   4.   
      **[student@workstation ~(developer1-finance)]$**
   5. **  
       Examinez /var/log/cloud-init.log pour vérifier que cloud-init s’est exécuté.**
   6. **  
      [cloud-user@finance-server2 ~]$**
   7. **  
       Vérifiez que httpd fonctionne.**
   8. **  
      [cloud-user@finance-server2 ~]$**
   9. **  
       Sur workstation, utilisez la commande curl pour accéder à http://172.25.250.106. La connexion doit être établie.**
   10. **  
       [student@workstation ~(developer1-finance)]$**

****

===============================================

QUIZ

## 1-How many entries are in the undercloud catalog?

10

11

12 \*

13

2- Examine the inventory after introspection is complete. What is the size of the "/dev/vda" disk on the "compute01" node?

42949672960

53687091200 \*

64424509440

59

3- What MAC address did introspection find for the "ens4" interface on the "compute02" node?

2c:c2:60:01:02:06

2c:c2:60:1f:6d:0e

2c:c2:60:45:c5:66

2c:c2:60:60:a6:f1 \*

4-Examine the "/etc/neutron/plugins/ml2/ml2\_conf.ini" file inside the "neutron\_api" container on the overcloud controller. What is the value of the geneve "vni\_ranges"?

1:100

1:1000

1:65535

1:65536 \*

5-Which Neutron service plug-ins are configured in overcloud?

octavia,qos,router,trunk

lbaasv2,ovn-router,trunk

qos,router,trunk

qos,ovn-router,trunk,segments \*

6-How many overcloud catalog entries does the "openstack catalog list" command report?

12 \*

13

14

15

7- How many Docker containers are started by Pacemaker on the overcloud controller?

3

7 \*

8

10

8-How many Nova containers run on the controller node?

0

1

4

8 \*

9-How much memory is reserved for the hypervisor's operating system? Hint: Check "reserved\_host\_memory\_mb" in the "/etc/nova/nova.conf" file:

undefined

1024

2048

4096 \*

10- How many connections are allowed to the Galera server?

1024

4096 \*

16384

65535

========================================

## **Quiz: Describing Multi-site Overcloud Deployments**

|  | | |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1-The control plane contains which type of node?  **Controller nodes**

2- Which storage component is used in the hyperconverged infrastructure? **ceph**

3-In a multisite installation, what is the maximum recommended round-trip time between the compute nodes and the control plane? **100ms**

4-In which Nova cell are flavors stored? **API cell**

5- Can nodes be members of multiple host aggregates? **True**

====================================================

## **Quiz: Configuring the Placement Service**

1- Which statement is true about the Placement service?

It tracks the inventory and usage of resource providers, which can be a compute node, a shared storage pool, or an IP allocation pool.

2- Which statement is true about resource providers?

They are entities that provide an inventory of classes of resources, such as disk or RAM

3-Which statement is true about traits?

Traits are qualitative characteristics of resource providers.

4-The Placement service is the same as a host aggregate. (True or False)

False