

90 minutes avec OpenStack

Laurent Foucher

email: <u>u03@u03.fr</u> blog: <u>https://blog.u03.fr/</u> git : <u>https://github.com/U03</u>

Version : 2019/11/11

90 Minutes avec OpenStack – 2019/11/11

A propos de ce livre

J'ai écrit ce livre comme une introduction à OpenStack en français, car il est important pour moi de disposer de ressources en français, même s'il faut bien l'admettre la « langue officielle » de l'informatique est l'anglais.

Il est probable qu'il vous faille plus de 90 minutes pour lire ce livre, en particulier si vous reproduisez les différents exemples, réalisez les quelques exercices et surtout expérimentez par vous même.

Ce livre vous est offert sous la licence 'Creative Commons' :

« Attribution - Pas d'Utilisation Commerciale - Partage dans les Mêmes Conditions 3.0 France »



Vous trouverez plus d'informations sur cette licence sur le site de 'Creative Commons France' :

https://creativecommons.org/licenses/by-nc-sa/3.0/fr/

Table des matières

Présentation de OpenStack	6
Architecture de OpenStack	6
Installation de DevStack	8
Premier contact avec OpenStack	11
Les projets et les utilisateurs	11
L'API et l'interface en ligne de commande	11
Les images	12
Les volumes	12
Les clés SSH	13
Les quotas et les gabarits	14
Les réseaux	15
Connexion au Dashboard	16
Notre première instance	18
A savoir avant de lancer notre première instance	18
Lancement de notre première instance	18
Association d'une adresse IP flottante	23
Modification du groupe de sécurité	25
Connexion à l'instance	30
Affichage des informations sur l'instance	
Suppression des éléments créés	32
Utilisation des images	32
Téléchargement de l'image Ubuntu « Cloud Image »	32
Contrôle de l'intégrité de l'image téléchargée	33
Chargement de l'image dans Glance	33
Vérification de la disponibilité de l'image	34
Lancement d'une instance en ligne de commandes	34
Création d'un groupe de sécurité	35
Lancement de l'instance	36

	Ajout d'une adresse IP flottante à notre instance	38
	Connexion à notre instance	39
Les vo	lumes de données	40
	Création d'un volume de données via le Dashboard	40
	Préparation (partitionnement et formatage) du volume	43
	Détachement du volume via le Dashboard	44
	Suppression d'un volume via le Dashboard	44
	Création et attachement d'un volume en ligne de commande	45
	Détachement et suppression d'un volume en ligne de commande	46
L'orche	estration	47
L'orche	estration avec Heat	47
	Les ressources	47
Une p	remière pile	49
	Utilisation de la pile	50
	Suppression de la pile	54
Autres	s types de ressources de Heat	55
	Création d'une groupe de sécurité et d'une adresse IP flottante	55
	Utilisation d'une adresse IP flottante	56
	Création d'un volume à partir d'une image	57
	Attachement d'un volume de données à une instance	57
Utilisa	tion des piles à l'aide du CLI openstack	59
	Lancement d'une pile	59
	Mise à jour d'une pile	60
	Destruction d'une pile	61
	Lancement d'une pile avec un fichier de paramètres	61
Créati	on d'infrastructures réseau	63
	Création des réseaux et sous-réseaux	65
	Attachement du serveur au réseau	67
	Attachement à plusieurs réseaux	67
Le sto	ckage d'objets avec Swift	69

Création d'un conteneur Swift et chargement d'objets depuis le Dashboard	69
Manipulation des conteneurs et objets en ligne de commande	71
Stockage et récupération d'un fichier	72
Ajout de propriétés	72
Terraform	74
Installation et initialisation	74
Éléments d'une configuration	75
Une première configuration	76
Création d'un volume système et d'un groupe de sécurité :	78
Création et utilisation de réseaux	80
Script d'initialisation d'instance simple	81
Script d'initialisation d'instance complexe (template)	82
Utilisation directe des API	83
Obtention d'un Token et de la liste des endpoints	83
Utilisation de l'API Cinder : Obtenir la liste des volumes	85
Utilisation de l'API Cinder : Création d'un volume	87
Utilisation de l'API Compute: Création d'une instance	89
Utilisation du client CLI openstack en mode debug	91
Conclusion	93

Présentation de OpenStack

OpenStack est un ensemble de logiciels qui permettent de déployer des infrastructures virtuelles (Infrastructure As A Service).

Chaque infrastructure (projet ou tenant) est composée de serveurs virtuels (instances ou serveurs) qui fournissent la puissance de traitement, utilisent du stockage (disques virtuels, fichiers) et s'intègrent dans des réseaux virtuels (sous-réseaux, routeurs, pare-feu).

Les projets sont indépendants les uns des autres, les adresses IP à l'intérieur d'un projet sont privées, elles sont choisies librement par l'administrateur du projet, elles ne sont accessibles ni de l'extérieur, ni des autres projets.

L'administrateur du système OpenStack définit des adresses IP spécifiques qui sont routables depuis l'extérieur du Cloud, ces adresses sont appelées 'adresses IP flottantes'. Les administrateurs des différents projets peuvent demander que des adresses leur soient allouées pour rendre accessibles des instances depuis l'extérieur de leur projet.

L'administration et l'utilisation de OpenStack se fait à travers d'une collection d'API (Application Programming Interface) qui donnent à OpenStack sa puissance et sa souplesse. Ces API sont utilisables grâce à une Interface en Ligne de Commande (CLI), grâce à une interface web spécifique appelée Dashboard, ou grâce à tout programme ou script capable d'appeler les API directement.

Architecture de OpenStack

Il y a 4 groupes de services :

- Control
- Network
- Storage
- Compute

L'élément Control fait fonctionner les API, l'interface web, la base de données (MySQL, MariaDB ou PostgreSQL), et le bus de messages (RabbitMQ, Qpid ou ActiveMQ).

Les principaux éléments constituant OpenStack sont les suivants :

- Dashboard : L'interface web, elle utilise l'API d'OpenStack, elle est développée sur le framework 'Horizon' (ce qui lui vaut d'être parfois appelée 'Horizon')
- Keystone : Composant de gestion des identités, il gère l'authentification mais également il assure la partie catalogue : pour les points d'accès (endpoints) aux autres API, les projets, utilisateurs, rôles et tous les composants. Keystone est la pierre angulaire du système, tous les composants doivent appartenir à un projet (également appelé tenant ou propriétaire)
- Nova : gère les instances (serveurs virtuels)

- Glance : gère les images qui permettent de lancer les instances, il s'agit d'images de disques avec l'OS préinstallé et non d'images de DVD d'installation. Ces images sont rendues anonymes (suppression des adresses MAC, clés SSH...) la personnalisation est faite par le script 'cloud-init' lors du boot de l'instance
- Neutron : Gestion du réseau virtuel (SDN : Software Defined Network)
- Cinder : Gestionnaire de stockage en mode blocs, il permet de créer des volumes (des disques durs virtuels) qui seront utilisables par les instances
- Swift : Gestionnaire de stockage d'objets, il permet de stocker des objets (fichiers) qui seront accessibles via une URL, avec ou sans authentification
- Ceilometer : Métrologie
- Heat : Orchestration (automatisation du déploiement des architectures).

L'orchestration est un point important dans les infrastructures Cloud, elle permet de déployer des infrastructures de façon automatique, mais surtout de les redéployer autant de fois que nécessaire, en particulier pour réaliser la mise en production des applications.

La CI/CD (Intégration Continue / Déploiement Continu) est une méthode et un ensemble d'outils qui permettent à partir d'un référentiel (de programmes, de paramètres) d'automatiser la création (intégration) d'artefacts (paquetages livrables), puis d'automatiser le déploiement de ces artefacts vers différents environnements (intégration, qualification, recette, pré-production, production).

L'orchestration et la CI/CD associées aux technologies du Cloud rendent les infrastructures « jetables » puisque facilement déployables, que ce soient les serveurs, les logiciels installés, le paramétrage du système d'exploitation et des logiciels associés, et même l'application.

Ainsi seules données sont conservées de façon perenne.

Installation de DevStack

DevStack est une distribution simplifiée d'OpenStack qui fait fonctionner les éléments nécessaires à un environnement de développement sur un serveur unique.

La distribution DevStack déploie les éléments principaux d'OpenStack, elle est destinée à la formation et au développement, elle est déconseillée en production.

Il est conseillé d'avoir au moins 16Go de RAM disponibles sur la machine cible.

Nous commençons par préparer le serveur physique en installant le système d'exploitation « Ubuntu Server 18.04.3 LTS » (ou la dernière version LTS disponible sur le site de Ubuntu):

<u>https://www.ubuntu.com/download/server</u>

LTS signifie Long Term Support afin de différencier les versions de production qui sont supportées durant 5 ans minimum, des versions dites de développement qui servent à mettre au point la release LTS suivante.

Les versions de développement contiennent des versions plus récentes des produits par rapport aux versions LTS mais elles ne disposent d'aucun support et ne sont pas nécessairement stables.

Une installation par défaut de Ubuntu est suffisante, ne pas oublier d'ajouter l'option 'SSH Server' afin de pouvoir accéder à votre serveur en SSH. La procédure d'installation de Ubuntu vous propose de créer un compte utilisateur, créez un compte nommé 'stack' avec le mot de passe de votre choix (ce compte aura droit d'utiliser la commande 'sudo' en spécifiant son mot de passe),

L'installation de DevStack est simple, elle ne nécessite que l'installation du client Git, les autres packages nécessaires seront installés automatiquement. Un fichier de configuration minimal permet de procéder au déploiement de DevStack.

Nous réalisons les étapes suivantes :

- Modification du fichier sudoers pour autoriser l'utilisateur stack à passer des commandes à la place de root sans avoir à spécifier de mot de passe.
- Mise à jour de l'index des packages de apt, puis installation du package git
- Clone du repository Git de DevStack
- Création du fichier de configuration permettant l'installation de DevStack. Modifiez le paramètre HOST_IP pour mettre l'adresse IP du serveur sur lequel vous procédez à l'installation. Tous les mots de passe auront pour valeur 'topsecret'.
- Le paramètre VOLUME_BACKING_FILE_SIZE permet de spécifier l'espace disque qui sera alloué à Cinder pour stocker les différents volumes. L'allocation est faite en mode 'thin provisionning', l'espace utilisé sur le disque physique dépend de l'utilisation réelle qui est faite par Cinder.
- Installation de DevStack grâce à la commande './stack.sh' (la commande './unstack.sh' permettra de procéder à la désinstallation si nécessaire).

stack@microwave:~\$ echo "stack ALL=(ALL) NOPASSWD: ALL" | sudo tee -a /etc/sudoers

stack@microwave:~\$ sudo apt-get update stack@microwave:~\$ sudo apt-get -y install git stack@microwave:~\$ git clone https://git.openstack.org/openstack-dev/devstack stack@microwave:~\$ cd devstack stack@microwave:~/devstack\$ cat << EOF > local.conf [[local|localrc]] HOST IP=192.168.1.211 GIT_BASE=http://git.openstack.org ADMIN_PASSWORD=topsecret DATABASE_PASSWORD=\\$ADMIN_PASSWORD RABBIT PASSWORD=\\$ADMIN PASSWORD SERVICE PASSWORD=\\$ADMIN PASSWORD # Activation de Swift sans replication enable_service s-proxy s-object s-container s-account SWIFT_HASH=66a3d6b56c1f479c8b4e70ab5c2000f5 SWIFT_REPLICAS=1 SWIFT_DATA_DIR=\$DEST/data/swift $\ensuremath{\texttt{\#}}$ Activation du composant <code>HEAT</code> et de l'onglet <code>HEAT</code> dans le dashboard enable_plugin heat https://git.openstack.org/openstack/heat enable_plugin heat-dashboard <u>https://git.openstack.org/openstack/heat-dashboard</u> # Agrandissement volume pour Cinder VOLUME_BACKING_FILE_SIZE=100G EOF stack@microwave:~/devstack\$./stack.sh [.../...] ____ DevStack Component Timing (times are in seconds) run process 53 test_with_retry apt-get-update 2 359 osc 26 1100 wait_for_service git_timed 107 dbsync pip_install 982 736 apt-get 848 Unaccounted time Total runtime 4217 This is your host IP address: 192.168.1.211 This is your host IPv6 address: ::1 Horizon is now available at http://192.168.1.211/dashboard Keystone is serving at http://192.168.1.211/identity/ The dashboard Keystone is serving at http://192.168.1.211/identity/ The dashboard Keystone is serving at http://192.168.1.211/identity/ The dashboard Keystone is serving at http://dashboard K

L'installation de DevStack dure jusqu'à 1 heure en fonction de la puissance de votre serveur mais aussi de la vitesse de votre connexion à Internet pour télécharger les éléments nécessaires.

Une fois DevStack installée vous pouvez autoriser vos futures machines virtuelles à se connecter à Internet (et à votre réseau privé) en paramétrant le NAT comme ceci :

stack@microwave:~\$ ip addr
1: lo: <loopback,up,lower_up> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000</loopback,up,lower_up>
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
inet6 ::1/128 scope host
valid_lft forever preferred_lft forever
2: enol: <broadcast,multicast,up,lower_up> mtu 1500 qdisc mq state UP group default qlen 1000</broadcast,multicast,up,lower_up>
link/ether 94:18:82:38:17:30 brd ff:ff:ff:ff:ff:ff
inet <mark>192.168.1.211</mark> /24 brd 192.168.1.255 scope global dynamic enol
valid_lft 73654sec preferred_lft 73654sec
inet6 fe80::9618:82ff:fe38:1730/64 scope link
valid_lft forever preferred_lft forever
stack@microwave:~\$ sudo iptables -t nat -A POSTROUTING -o <mark>eno1</mark> -j MASQUERADE

Il arrive que le réseau privé créé défaut créé par DevStack ne comporte pas de DNS, vous pouvez utiliser ceux de Google, ou de votre réseau privé comme ceci :

stack@microwave:~\$	source dev	vstack/open	rc demo demo				
stack@microwave:~\$	openstack	subnet set	private-subnet	dns-nameserver	8.8.8.8 -	-dns-nameserver	8.8.4.4

Premier contact avec OpenStack

Les projets et les utilisateurs

Toutes les ressources doivent appartenir à un projet, un utilisateur ne peut utiliser les ressources d'un projet que si un rôle lui a été attribué sur ce projet.

Certaines ressources ne sont pas liées à un projet, mais à un utilisateur, c'est le cas par exemple des clés SSH que nous verrons un peu plus loin.

L'API et l'interface en ligne de commande

Les différents composants de OpenStack déploient des API (Application Programing Interface) avec lesquelles il est possible de communiquer en utilisant le protocole TCP via des points d'entrée (endpoint) catalogués dans KeyStone.

Un utilitaire en ligne de commande (CLI – Command Line Interface) est fourni pour s'interfacer avec les composants OpenStack de façon simple. Il est également téléchargeable séparément pour pouvoir gérer son infrastructure depuis une machine distante.

Ces API sont également utilisables par des utilitaires d'automatisation (orchestration) qui permettent de déployer des infrastructures à la demande de façon reproductible.

On l'a dit, Keystone est le point d'entrée du système, il gère l'authentification ainsi que le catalogue des différents projets (tenants) et endpoints.

Pour interagir avec OpenStack il faut disposer du endpoint de KeyStone, d'un nom d'utilisateur, d'un mot de passe et du nom d'un projet sur lequel on souhaite travailler (ou du nom du projet d'admin).

Après authentification KeyStone va fournir un jeton d'authentification (token), ainsi que la liste des endpoints des autres API parmi lesquels on peut choisir le endpoint nécessaire (par exemple Glance pour la gestion des images...) Ce jeton permet ensuite de s'authentifier auprès des autres services, un jeton a une durée de vie limitée qui est donnée par KeyStone en même temps que la valeur du jeton.

Le client CLI de OpenStack est appelé 'openstack', il utilise des variables d'environnement ou des paramètres en ligne de commande pour se connecter à l'infrastructure OpenStack. Cette commande permet de passer des commandes à tous les composants de OpenStack.

DevStack est fourni avec script appelé 'openrc' qui positionne les variables nécessaires, il prend 2 paramètres facultatifs, le premier est le nom d'utilisateur ('demo' par défaut), le second est le nom du projet ('demo' par défaut).

Ci-dessous nous appelons le script openre pour nous placer avec le user demo dans le projet demo, dans les variables positionnées par le script on note l'URL d'accès KeyStone qui est le point d'entrée du système :

stack@microwave:~\$ cd ~/devstack
stack@microwave:~/devstack\$ source openrc demo demo
stack@microwave:~/devstack\$ set | grep -P "^OS_"
OS_AUTH_TYPE=password
OS_AUTH_URL=http://192.168.1.211/identity
OS_CACERT=
OS_IDENTITY_API_VERSION=3

OS_PASSWORD=topsecret OS_PROJECT_DOMAIN_ID=default OS_PROJECT_NAME=demo OS_REGION_NAME=RegionOne OS_TENANT_NAME=demo OS_USERNAME=demo OS_USER_DOMAIN_ID=default OS_VOLUME_API_VERSION=2

La syntaxe générale de la commande 'openstack' est :

openstack objet opération paramètres:

Maintenant que les variables d'environnement nécessaires sont positionnées nous pouvons appeler l'utilitaire 'openstack' pour avoir la liste des projets sur lesquels l'utilisateur a des droits :

<pre>stack@microwave:~\$ openstack project</pre>	t	list
ID	+- 	Name
+	+-	+
8eb8bd6dcd2447b2bc8224b4ceb64672	L	demo
8feed042b5504d3fa074fca9ae4694f5	L	invisible_to_admin
+	+-	+

Les images

L'installation d'un serveur physique est réalisée en installant le système d'exploitation sur un volume (disque dur) en démarrant le serveur sur un DVD d'installation.

La création d'une instance (serveur virtuel) se fait en utilisant une image spécifique préparée à partir d'un disque sur lequel un système d'exploitation a été installé.

Les éditeurs de systèmes d'exploitation fournissent ces images 'Cloud Ready' au même titre que les images de média d'installation.

Les images 'Cloud Ready' sont anonymisées (suppression des adresses MAC des cartes réseau, des clés SSH) et un script spécifique appelé 'cloud-init' est installé dessus, il prendra en charge le paramétrage des instances lors de leur démarrage.

Les volumes

Les volumes sont des disques durs virtuels (on parle de stockage en mode 'blocs'), ils sont créés par exemple lors de la création d'une instance pour y placer le système d'exploitation présent sur l'image utilisée.

Dans le Cloud l'infrastructure est virtuelle, habituellement elle est également éphémère car détruite et redéployée par l'Orchestration à chaque fois que cela est nécessaire (par exemple pour installer une mise à jour de l'application ou du système d'exploitation).

C'est pourquoi on déploie le système d'exploitation sur un volume qui ne contiendra aucune donnée, il sera détruit et recréé à chaque mise en production en même temps que l'instance qui l'utilisait.

On crée séparément des volumes sur lesquels on va stocker les données de façon pérenne, lors des mises en production ces volumes seront détachés des instances auxquelles ils étaient connectés, ils seront ensuite reconnectés aux instances créées par l'Orchestration lors du déploiement de la mise en production.

Les clés SSH

L'utilisation de mots de passe pour se connecter à un serveur n'est pas sécurisée, lors de la création d'une instance on spécifie une clé publique qui sera ajoutée par cloud-init aux clés autorisées de l'utilisateur par défaut pour permettre la connexion en SSH à l'instance.

Le nom de l'utilisateur par défaut est le nom de la distribution (centos pour CentOS, debian pour Debian et ubuntu pour Ubuntu)

Il existe deux possibilités concernant les clés SSH :

- Utiliser un bi-clé existant, et ajouter la clé publique dans la base de données de OpenStack
- Faire tirer un bi-clé par la commande OpenStack, la clé publique sera automatiquement ajoutée dans la base de donnés de OpenStack, la clé privée associée devra être conservée pour être utilisée lors de la connexion à l'instance

Les clés SSH de type DSA ne sont pas utilisables avec OpenStack, en effet ce type de clés est reconnu comme non sûr. Pour pouvoir utiliser une clé existante elle doit être obligatoirement de type RSA.

Utilisation d'une clé existante

La commande openstack keypair create avec le paramètre --public-key permet d'ajouter une clé publique SSH sans la base de données OpenStack.

La clé appartient à utilisateur, elle n'appartient pas à un projet en particulier, elle est utilisable pour tous les projets de l'utilisateur. Les autres utilisateurs qui ont accès aux mêmes projets que vous n'ont pas accès à vos clés.

stack@microwav stack@microwav stack@microwav	ve:~\$ cd ve:~/devs ve:~/devs	devstack/ tack\$ source openrc demo tack\$ openstack keypair createpuł	blic-key
+ Field	+ Value		+
<pre> fingerprint 71:1d:81:03:3c:46:97:b8:e8:88:eb:e2:b2:15:6c:f name Foucher Laurent RSA user_id 7436b216d6ee4a1b809c723579c0e2bc + </pre>			
stack@microwav	ve:~/devs	tack\$ openstack user show 7436b216d (.6ee4a1b80
Field		, Value	
domain_id email		/ default demo@example.com	7
enabled id		True 7436b216d6ee4a1b809c723579c0e2bc	i I
name options		demo {}	
password_exp +	oires_at	None +	+

Pour information un bi-clé SSH de type RSA s'obtient comme ceci :

stack@microwave:~\$ ssh-keygen -t rsa -b 4096 -C "Ma cle SSH" Generating public/private rsa key pair. Enter file in which to save the key (/home/stack/.ssh/id_rsa): Enter passphrase (empty for no passphrase): Enter same passphrase again: Your identification has been saved in /home/stack/.ssh/id_rsa. Your public key has been saved in /home/stack/.ssh/id_rsa.pub. The key fingerprint is:

```
      SHA256:MvDIi026TPR+ACWZn5DTlvfyLLBGtMyog54UPX3I118 Ma cle SSH

      The key's randomart image is:

      +---[RSA 4096]----+

      |
      =

      |
      B + 0

      |
      B + 0

      |
      B+*.0..

      |
      ...

      |
      ...

      |
      ...

      |
      ...

      |
      ...

      |
      ...

      |
      ...

      |
      ...

      |
      ...

      |
      ...

      |
      ...

      |
      ...

      |
      ...

      |
      ...

      |
      ...

      |
      ...

      |
      ...

      |
      ...

      |
      ...

      |
      ...

      |
      ...

      |
      ...

      |
      ...

      |
      ...

      |
      ...

      |
      ...

      |
      ...

      |
      ...

      |
      ....
    </tr
```

Création d'un bi-clé par la commande OpenStack

La commande openstack keypair create avec le paramètre --private-key permet de tirer un bi-clé et d'ajouter la clé publique SSH dans la base de données OpenStack. La clé privée est sauvegardée dans le fichier spécifié en paramètre, il faut faire très attention en manipulant ce fichier, il est à traiter avec la même prudence qu'un mot de passe en clair. En particulier attention aux droits sur le fichier qui sont ceux par défaut de l'utilisateur (umask) et sont trop permissifs :



Les quotas et les gabarits

A chaque projet sont associés des quotas qui définissent la quantité maximale de chaque type de ressources qui peuvent être allouées par le projet :

- Instances: le nombre de serveurs virtuels
- vCPU : le nombre de processeurs virtuels
- RAM : la mémoire vive totale qui peut être allouée aux différentes instances
- Adresses IP flottantes (adresses IP accessibles depuis l'extérieur du projet
- Groupes de sécurité : les pare-feu placés en frontal des instances pour filtrer les flux entrants et sortants des instances
- Volumes : les disques durs virtuels qui permettent de stocker le système d'exploitation ou les données des instances (en nombre et en taille totale)

Les administrateurs des projets peuvent commander des instances de serveurs virtuels parmi différents modèles (vCPU, RAM...) appelés gabarits (ou flavor en anglais dans le dashboard et le CLI). Seul l'administrateur OpenStack peut définir ces gabarits.

5	st	acl	k @1	microwave:~	\$	openst	ta	ck fla	v	or list		,		,
1		ID	ļ	Name		RAM		Disk		Ephemeral	VCPUs		Is Public	+
1		1		m1.tiny	1	512		1		0	, 1	+-	True	+
	Ι.	2	Т	m1.small	L	2048	1	20	T	0	1	L	True	L
		3	I.	m1.medium	L	4096	I.	40	T	0	2	L	True	I.
		4	Т	m1.large	L	8192	1	80	T	0	4	L	True	L
		42	Т	m1.nano	L	64	1	0	T	0	1	L	True	l
		5	1	m1.xlarge	L.	16384	1	160	T.	0	8	L.	True	I.

84	m1.micro	1	128	0	0	1	True	
c1	cirros256	I.	256	0	0	1	True	1
d1	ds512M	I.	512	5	0	1	True	- I
d2	ds1G	I.	1024	10	0	1	True	1
d3	ds2G	I.	2048	10	0	2	True	1
d4	ds4G	I.	4096	20	0	4	True	- I
+	+	+-	+-	+	+	+-		+

Les réseaux

OpenStack permet de construire des infrastructures complètes, y compris au niveau réseau.

En créant des routeurs et des réseaux on crée des infrastructures qui se rapprochent des infrastructures 'legacy' (héritées).

Les réseaux correspondent à des réseaux physiques, un réseau est composé d'au moins un sousréseau.

Les sous-réseaux dans Neutron correspondent à des sous-réseaux au niveau IP, leur définition est composée des éléments suivants :

- La plage d'adresse, elle est définie en notation CIDR : adresse de réseau + longueur de masque (par exemple : 192.168.1.0/24)
- Une plage d'adresse DHCP permettant d'attribuer dynamiquement des adresses aux instances
- La passerelle par défaut du sous-réseau
- Les DNS qui seront utilisés par les serveurs

Des routeurs permettent d'interconnecter les réseaux entre eux.

Des ports, qui correspondent à des prises réseaux (carte réseau de serveur, port de routeur...)

Nous verrons comment construire des réseaux dans la partie consacrée à l'orchestration Heat.

Connexion au Dashboard

On pourrait traduire 'Dashboard' par 'Tableau de bord', il est également appelé 'Horizon'. Il s'agit de l'interface graphique de OpenStack.

Le Dashboard est utilisable avec n'importe quel navigateur moderne. Par défaut lors de l'installation de DevStack les utilisateurs 'demo' et 'admin' sont créés avec le mot de passe choisi lors de l'installation (il est rappelé à la fin de l'installation de DevStack en même temps que l'URL d'accès au Dashboard).

Log in User Name Password	
Connect	

Lors de la connexion au Dashboard un premier écran s'affiche avec les différents quotas dont nous avons parlé. A gauche du bandeau supérieur une liste déroulante permet de passer d'un projet à l'autre parmi les différents projets sur lesquels nous avons des droits. A droite est rappelé quel utilisateur est connecté, un menu déroulant permet de modifier certains réglages (en particulier la langue d'affichage du Dashboard et le fuseau horaire).

Aperçu de l'instanc	e - OpenSile X +					1	2	D	×
<>> ♂ ŵ	④ 192.168.1.21	1/dashboard/project/	🖂 🕁	Q, Rechercher	*	11		đ	≡
openstack.	🕅 demo 🕶						4	demo	
Projet Acc Compute	Projets : v demo imvisible_to_admin v v u v	te / Vus d'ensemble						2 C o Co Co	
Vue d'ensi	emble								
Inst	ances Synthè	se des Quotas							
In	nages Compute								
Paires d	e clês								
Groupes de se	weurs								
Volumes	>	Instances Utilisé 0 sur 10	VCPUs Utilisé 0 su	r 20	RA Utilisé 00cte	M. At sur!	50Ga		1
Réseau	> Volume								
Orchestration	>								
Stockage d'objet	>								
Identite	>	Volumes	Instantanés du	volume	Stockage di	e volu	mes		
		Utilisé 0 sur 10	Utilisé 0 su	ir 10	Utilisé 00ctet	sur ti	000Ga		
	Réseau								
			(* C		'				

Nous utilisons l'option «Paramètres»du menu déroulant de droite pour change la langue de l'interface et nous placer dans le fuseau horaire de notre choix. Il est possible de se placer en « UTC » (Temps Universel) ou « UTC+1 : France ».

Souvent les serveurs sont réglés sur le fuseau horaire UTC, pour des raisons de commodité dans les grandes infrastructures Cloud qui s'étalent sur plusieurs fuseaux horaires, dans ce cas choisir UTC dans le Dashboard permet de rester cohérent avec l'heure des serveurs.

Paramètres de l'utilis	ateur 👾 🕅 🛄						
opensta	ck. 🔳 dem	•		4	demo		
roje! Ientité aramétres	> > ~	Paramètres / Paramètres de l'utilisateur Paramètres de l'u	tilisate	Succès (Paramètres sauvegardés.	X		
Paramètres d Changer le m	e l'utilisateur	Paramètres de l'utilisate	ur				
		Langue *					
		Français (fr)	~	Description :			
		Fuseau horaire	Modilier les parametres du dashboard.				
		UTC +01:00 : heure : France	~				
		Nombre d'éléments par page * 🛛					
		20					
		Lignes de journalisation par instance*	0				
		35	-				

Notre première instance

A savoir avant de lancer notre première instance

Nous allons lancer notre première instance grâce au Dashboard, mais auparavant nous allons discuter d'un point essentiel sur les volumes et sur l'infrastructure d'OpenStack.

Nous avons dit qu'OpenStack est (entre autres) composé de nœuds (serveurs) de type « Compute » (ce sont ceux sur lesquels s'exécutent nos instances) et de nœuds de type « Storage » (ce sont ceux qui ont en charge le stockage des volumes de données).

Il possible de lancer une instance en laissant OpenStack créer un volume pour le système d'exploitation directement sur l'espace de stockage du nœud « compute ». Cependant en cas de panne du nœud (ou si l'administrateur décide de l'arrêter pour une mise à jour par exemple) alors l'instance et les données présentes sur ce volume sont perdues...

La bonne façon de faire est de créer un volume à partir de l'image du système d'exploitation que nous souhaitons utiliser, ce volume sera utilisé pour lancer notre instance.

Pour nos premiers tests nous allons utiliser une image de système d'exploitation spécifiquement faite pour les tests, elle est légère, ne nécessite pas beaucoup de ressources, n'a pas de nombreuses fonctionnalités et ne doit pas être utilisée en production pour des raisons de sécurité. Son nom est 'Cirros', elle a été téléchargée depuis Internet et ajoutée à OpenStack par la procédure d'installation de DevStack.

Lancement de notre première instance

Dans le Dashboard nous affichons l'onglet « Projet \rightarrow Compute \rightarrow Images », ici nous voyons l'image de Cirros. A droite de l'image 'cirros' nous avons un menu déroulant et nous choisissons « Créer le volume »

Images - OpenStac	k Dashboard X	+					-	- 0	×
€ → ♂ @	① 192.16	68.1.211/dashboard/project/images	⊡ ☆	Q Rec	hercher	4	H/	•	E
openstack.	📾 demo 🕶							a d	emo =
Projet	Pro	vjet / Compute / Images							
Compute Vue d'ensi	emble								
Inst	Q.	Cliquer na poor les filtres qu'is recharch	o plein texto.	×	+ Créer une in	mage 🔲	Supprin	nor les inv	1981
le	nagea Affic	hage de 1 élément							
Paires d	e clés 🛛	Nom *	Туре	Statut	Visibilité	Protégée			
Groupes de se	nveura 🖸	> cirros-0.4.0-x86_64-disk	Image	Active	Publique	Non		Démarrer	•
Volumes	> Affic	hage de 1 élément							
Réseau	>								
Orchestration	>								
Stockage d'objet	>								
Identité	>								

Nous appelons notre volume « premier_volume », la source est déjà renseignée puisque nous appelons cette fonction depuis l'écran « Images », nous laissons la taille à 1Go (c'est largement

suffisant pour Cirros), les paramètres « Type » et « Zone de disponibilité » doivent être laissés à leur valeur par défaut, cliquons sur « Créer le volume ».

) → C @	① 192.168.1.211/dashboa	rd/project/	/images	🖻	10	Q, Rechercher	Ŧ	111	۲
openstack.	III damo 🕶								a der
Créer le volume									ж
	Les Volumes son	t des blocs	de périphériq	ue qui peuv	ervt. être	attachés à des instances.			0
Derans du Volume	Nom					Quota des volumes et instanta	inės (Gio)		
	premier_volume					(1000 Max)			
	Desertation								
	Description					0%			
						0 Utilization actualle.			
	Utiliser une ima	ge comme	source			1 Ajouté			
	cirros-0.4. 0-x86	i_64-disk (12.13 Mo)			999 Restant Quota de volume			
	Туре		Taille (Gio	•] •		(10 Max)			
	lvmdriver-1	4	1	15	2				
	Zone de disponi	ibilité *				10%			
	nove				7	0 Utilisation actuelle			
						1 Ajouté			
						9 Restant Description de Type de Volum	0 75		
						lymdriver-1			

Nous affichons l'onglet « Projet \rightarrow Volumes \rightarrow Volumes », nous voyons le volume que nous venons de créer, il fait 1Go, son statut est « disponible » cela signifie qu'il n'est attaché à aucune instance, il est « amorçable » car il a été créé à partir d'une image de système d'exploitation, ça veux dire qu'on peut s'en servir pour lancer une instance.

Volumes - OpenStack Dashbo	* × +					- 🗆 X
← → ♂ ☆	(i) 192.168.1.211/dashboard/proj	ect/volumes/		🖾 🔍 Rechercher		<u>⊻</u> II\ © © ≡
🗖 openstack. 📼 demo	•					🛔 demo 👻
Projet V	Projet / Volumes / Volumes					
Compute >	Volumes					
Volumes 🗸 Volumes			Filtrer	Q. + Créer le volume	≓ Accepter le transfert	Supprimer les volumes
Instantanés Groupes	Affichage de 1 élément	n Taille Statut	Groupe Type	Attaché à Zone de disponibil	ité Amorçable Chiffré	Actions
Groupe d'instantanés	premier_volume -	1Gio disponible(s)	- lvmdriver-1	nova	Oui Non	Modifier le volume 👻
Rěseau > Orchestration > Stockage ďobjet > Identité >	Affichage de 1 élément					

Nous allons pouvoir créer l'instance en nous servant de ce volume, nous allons dans l'onglet « Projet \rightarrow Compute \rightarrow Instances » et nous cliquons sur « Lancer une instance » :

Instances - OpenStack Dashi	* × +											-	Ď	: 3 6
) > C @	@ 192,168.1.211/	dashboard/project/ins	stances/			🖾	☆	Q, Rechercher			Ŧ	111	۵ 🕸	Ξ
openstack. 🛲 den	no •												å den	• *
nojel 🗸	Projet / Compute	/ Instances												
Compute 🗸	Instance	S												
Vue d'ensemble					ID de	- Finstance -	••			Filtrer	A Lan	icer une	instanc	
Images	Nom de l'instance	Nom de l'image	Adresse IP Ga	barit Pa	aire de clés	Statut	Zone	de disponibilité	Táche	Ètat de l'alime	entation	Age	Actio	ns
Paires de clés					Aucun éle	ément à affi	cher							
Groupes de serveurs														
Volumes >														
Rēseau >														
Orchestration >														

L'écran de lancement d'une instance s'affiche, il est constitué de plusieurs onglets, les onglets qui doivent être remplis sont suivis d'une étoile « * », le bouton « Lancer l'Instance » reste en grisé tant que tous les onglets requis ne sont pas complétés.

Nous entrons « premiere_instance » dans le champ « Nom de l'instance » et nous cliquons sur l'onglet « Source » :

a instances - Opens				Carl Inc.			- 14 - 19	2005 MI 2005		-
⊖⇒ ୯ ଢ	③ 192.168.1.211/d	ashboard/project/instances/	… ☺ ☆	Q Rechercher			ŤΙ		4	
openstack.	🕅 demo =								å dem	R
Projet	Lancer Instance					×				
Acc	Details *	Veuillez fournir le nom d'hôte initial de d'instances. Augmenter le nombre pou	l'instance, la zone de disponibilité ir créer plusieurs instances avec le	où elle sera déploy s mêrnes paramètre	ée ainsi que le nombre es.	0				
Var den	Source *	Nom de l'Instance *		Total de	es instances 0 Max)					
In	Gabarit.*				-		a La	icer uni	instanc	i
	Réseaux *	Description			10%		etation	Age	Action	
Pares	Ports reseaux	Zone de disponibilité		0 Utilis	ation actuelle					
Graupes de s	Groupes de séculté	nova		9 Rest	ant					
Voumes	Paire de clès	Nombre *								
Réseau	Configuration	1		 Image: A set of the set of the						
Orchestration	Groupes de serveurs									
Stockage d'objet	Scheduler Hints									
identiké	Métadonnées									
	¥ Annuler		(B)	drug Suscent 1	A Lawrence Section	-				

Dans le champ « Sélectionner le source de démarrage » nous choisissons « Volume », dans la zone inférieure de l'onglet la liste des volumes amorçables disponibles s'affiche, nous cliquons sur la flèche à droite du volume « premier_volume » :

openstack.	Lancer Instance		× dem
4 Acc	Détails	La source d'une instance est le modèle qui a servi à créer l'instance. Vous pouvez utilise d'instance, un volume ou un instantané de volume (si activé). Vous pouvez également ch persistent en créant un nouveau volume.	r une image, un instantané oisir d'utilisar du stockage
Compute	Source *	Sélectionnez la source de démarrage. Supprimer le volume ap	orès terminaison de
Van d'en	Gabarit *	Volume Volume	
. In	Réseaux*	Oui Non	Cancer une instan
	Ports réseaux	Alloué Nom Description Taille Type Zone de dispos	ntation Age Activ
Paires	Groupes de sécurté	nom beschpton fanne type zone te dispo	in the second seco
Groupes de s		Sélectionner un élément depuis les éléments disponibles ci-de	sabua
Mahaman	Paire de clés	V Disponible(s)	Sälectionner un
	Configuration	Comparison of the second to exclusive state to be	Source and the second s
Research	Groupes de serveurs	Cuquer to pour les mines ou la récolución pien sexue.	
Orchestration		Nom Description Taille Type Zone de di	isponibilité
Stockage d'objet	Scheduler Hints	> premier_volume 1 Go qcow2 nova	*
	Mitadonnées		

Le volume « premier_volume » passe de la zone « Disponibles » à « Alloué », on clique sur l'onglet « Gabarit » :

Instances - OpenSt	ack Dashbor 🗙	+ x			=	٥	×
← → ♂ @	(1)	192.168.1.211/dashboard/project/instances/ 😇 🏠 🔍 Rechercher		ע ו	i\	۲	≡
projet	Lancer Ins	tance	×			👗 dem	o • ^
Acc	Détails	La source d'une instance est le modèle qui a servi à créer l'instance. Vous pouvez utiliser une image, un instantané d'instance, un volume ou un instantané de volume (si activé). Vous pouvez également choisir d'utiliser du stockage persistent en créant un nouveau volume	0				
Compute	Source	Sélectionnez la source de démarrage. Supprimer le volume après terminaison de					
Vue d'en	Gabarit *	Volume Volume Volum Non	1				
Ins	Réseaux *			Ca La	cer une	instanc	e
Datas	Ports résea	Alloué Nom Description Taille Type Zone de disponibilité	'n	itation	Age	Action	IS
Faires Groupes de s	Groupes de	> premier_volume 1 Go qcow2 nova	Ī				
Volumes	Configuratio	on ✓ Disponible(s) Sélectionner	un				
Réseau	Groupes de	Cliquer ici pour les filtres ou la recherche plein texte.	ĸ				
Orchestration		Nom Description Taille Type Zone de disponibilité					
Stockage d'objet	Scheduler	lints Aucun élément disponible					
Identité	Métadonné	85					
	× Annuler	<retour sutvant=""> A Lancer Instance</retour>					
							~

La liste des gabarits s'affiche, on sélectionne le gabarit « m1.tiny » qui est le plus petit gabarit avec 1vCPU et 512Mo de mémoire :

-							500M (1979)	(Castrone)			NA 153	M 2020	
⇒ C'û	③ 192.168.1.211/d	ashboard/pro	ject/insta	snces/			🖂 🕁	Q Rechercher			¥ ∎		\$
openstack.	Lancer Instance									×			6 dem
et Acc	Détails	Les s	jabarits s UČ	iont en place	pour gére	r la taile de la	capacité de stockage	e, de mémoire et de	e calcul d'une instanc	0			
Compute	Source		Nom	VCPUS	RAM	Total Disque	Disque Racine	Disque Éphómère	Publique 🖨				
Vut d'et	Gabarit Réseaux.*	>	m1.tin V	1	512 Mo	1 Go	1 Go	0 Go	Oui	*	🕰 Lan	per une s	nstanc
	Ports réseaux	∨ 0	Disponil	ble(s) 💷					Selectio	oner un	ntation	Age	Action
Paires	Groupes de sécurité	Q,	Cirque	r ici phur lies	filtres ou l	a recharche più	in teste.			×			
Groupes de s	Paire de clés		Nom	VCPUS	RAM	Total Disque	Disque Racine	Disque Éphémère	Publique				
Réseau	Configuration	>	m1 nano	15	64 Mo	1 Go	1 Go	0 Go	Oui	٠			
Orchestration	Graupes de serveurs Scheduler Hints	>	m1.micro	10 1 0	128 Mo	1 Go	1 Go	0 Go	Oui	٠			
Stockage d'objet	Métadonnées	>	cirros256	i i	256 Mo	1 Go	1 Go	0 Go	Oui	٠			
		>	ds512M	1	612 Mo	5 Go	5 Go	0 Go	Oui	*			

Il faut choisir dans l'onglet 'Réseaux' à quel réseau attacher notre instance, nous choisissons 'private':

instances - OpenSt	uccussion A								1999 - 1999 1999 - 1999		1.0
∈)⇒ ୯ ଢ	③ 192.168.1.211/d	ashboard/project/instances/		… ⊠ ☆	Q Rechercher			$\overline{\mathbf{x}}$	M/ (11) @	1
openstack.	🕅 demo =									å de	mp:
Projet	Lancer Instance						×				
Acc	Détails	Les réseaux fournis V Alloué	sent les canaux de communica	ion des instances da Sélectionner des	ns le cloud. : réseaux à partir de l	a liste fournie c	i-dessous.				
Vur d'en	Source	Résea	u Sous-réseaux associé	s Partagé	Admin State	Statut					
10	Gabarit	¢1 > private	private-subnet	Non	Haut	Active	*	a 11	nicer un	e instac	ice.
	Réseaux		18					ntation	Age.	Acti	ons
Paras	Ports réseaux	✓ Disponible(s	0		Sélectio	inner au moins	un réseau				
Groupes de s	Groupes de sécurité	Q. Cliquerici pr	iur les filtres ou la récherche ph	in Teide.			×				
Matument	Paire de clês	Réseau	Sous-réseaux associés	Partagé	Admin State	Statut					
Réseau	Configuration	> shared	shared-subnet	Oui	Haut	Active	*				
Orchestration	Groupes de serveurs										
Stockage d'objet	Scheduler Hints										
dentité	Mitadonnées										
	× Annuler			(R	tour Suivant >	• Lancer	Instance				

Si on clique sur l'onglet « Paire de clés » on voit que comme une seule clé est disponible elle a été sélectionnée par défaut (c'est aussi le cas d'autres éléments sur d'autres onglets). Il est déjà possible de lancer l'instance, nous la lançons en cliquant sur 'Lancer Instance' :

Instances - OpenSt	ack Dashbox X 🕂							D	2
) → ଫ @	③ 192.168.1.211/d	lashboard/project/instances/	🖾 🕁	Q Rechercher		± 1	IN (D)	۲	
openstack.	III demo +							& der	
nojat	Lancer Instance				×				
Acc	Détails	Une paire de clés vous perme sélectionner une paire de clés	t de vous connecter en SSH à votre instan existante, en importer une ou en générer	ce nouvellement créée. Vous pouvez une nouvelle.	0				
Vur d'en	Source	+ Créer une paire de clés	L'importer une paire de clés						
	Gabartt	Alloué				• La	icer unr	instan	
	Réseaux	Affichage de 1 élément Nom	Туре			ntation	Age	Actio	n
Paren	Ports réseaux	> LEO	sab		-				
Groupes de s	Groupes de sécurité				<u> </u>				
Mahamana	Paire de clés	Amonage de 1 element							
Réseau	Configuration	V Disponible(s)		Sélect	ionner un				
Orchestration	Grannes de serveurs	Q Cliquer ici pour les filtr	es ou la recherche plein texte.		×				
Stockage d'objet	Construction of Construction	Affichage de 0 élément							
entite	Scheduler Hirits	Nom	Туре						
	Métadonnées		Aucun élément à afficher						
		Affichage de 0 élément							
	* Annuler		(R	etour Suivant)	stance				

L'instance que nous venons de créer apparaît dans l'onglet « Projet \rightarrow Compute \rightarrow Instances », l'instance a une adresse IP qui lui a été allouée, cette adresse IP est une adresse IP privée, elle n'est pas accessible depuis l'extérieur de notre projet OpenStack.

Instances - OpenStack Das	hbo × +	ý.											5	s I	o i	ж
(←) → ♂ @	① 192	168.1.21	1/dashbo	ard/project/instances/			©	0	Q, Recher	cher		5	Ł 111\		4	Ξ
🖸 openstack. 🔳 🕫	amo 🕶														å dem	• •
Projet Accès API	Projet	/ Compu	te / Insta	ices												
Compute 🗸	Inst	tanc	es													
Vue d'ensemble				ID de l'instance = •				Filtre	Lancer	une insta	nce 🛛 🗈 Siggin	iner les, inst	ances.	Ph	is d'ac	horis
Images Paires de clés Groupes de serveurs	Affichag	ge de 1 élé om de nstance	Nom de Fimage	Adresse IP	Gabarit	Paire de clés	Statut		Zone de disponibilité	Tàche	État de l'alimentation	Age	Action	15		
Volumes >		emiere_i tance	-	10.0.0.56, fdac 50a1 b51d 0.1616 3eff fec4 508c	m1 tiny	LFO	Active	di.	nova	Aucun	En fonctionnement	0 minute	Crée	r un in	stantar	é i
Réseau > Orchestration > Stockage d'objet >	Affichag	ge de 1 élé	iment													
ldentité >																

Association d'une adresse IP flottante

Le Cloud OpenStack dispose d'un certain nombre d'adresses IP flottantes, l'administrateur d'un projet peut demander à se faire allouer des adresses IP flottantes (dans la limite des quotas du projet), quand une adresse IP flottante est allouée au projet l'administrateur peut l'associer à une instance.

Nous allons attribuer une adresse IP flottante à l'instance afin qu'elle soit accessible en dehors de notre projet. Nous choisissons « Allouer une adresse IP flottante » dans le menu déroulant en face de notre instance :

Instances - OpenSt	tack Dashb	o X	+											55	. I	D	2
⊢)→ ଫ @		(i) 19	92.168.1.2	11/dashbo	ard/project/instances/			6	D 🖸	Q, Recher	cher		3	Ł IIIN		4	
openstack.	a dem es API	• 0														iemo •	
Compute	~	Ins	stanc	es													
Vue d'ens	semble tances				ID de Tinstance = •				Filtre	r & Lance	une insta	nce 🛛 🛚 Supp	nner les mit	lancès	Plu	is d'ac	ti
h	mages	Affich	rage de 1 él	lément													
Paires d Groupes de se	de clés erveura	0	Nom de l'instance	Nom de l'image	Adresse IP	Gabarit	Paire de clés	Statut		Zone de disponibilité	Täche	État de l'alimentation	Age	Action	s		
Volumes	>	0	premiere_j ostance	÷.	10.0.0.56, fdac 50a1 b51d 0.1816 3eff.fec4 508d	m1.tiny	LFO	Active	÷.	nova	Aucun	En fonctionnement	0 minute	Crée	r un ins	stantar	
Réseau Orchestration Stockage d'objet dentité	> > >	Affich	rage de 1 él	lément									Associer un Atlacher fin Editer finst Atlacher le v Détacher le Mettre à jou	e adress terface sterface soce solume Volume r les mét	e IP Bo adonné	ittanle ies	
													Éditer les g	roupes d	e sécul	nté rité de	1

Pour le moment nous n'avons pas d'IP flottante allouée à notre projet. Nous commençons par demander l'allocation d'une IP flottante en cliquant sur le « + » en face de « Aucune IP flottante allouée » :

Instances - OpenStack Dashb-	× +								5	2	Ď	*
← → ♂ @	① 192.	168.1.211/dashboard/project/instances/		🖂 🕁	Q, Rei	hercher		4	2 111		۲	H
🗖 openstack. 🔳 demo							-				å der	
Projet 🗸	Projet	Gérer les Associations d'IP flott	ant	les		×						
ACCED AP1	Inst	Adresse IP *		Sélectionner Dadresse IP mie	your south	iter faire						
Compute 🗸 🗸	11131	Aucune adresse IP flottante alloué 👻 🕇		correspondre à l'instance ou a	au port séle	ctionné(e).						
Vus d'ensemble		Port à associer *										
(Instances)		premiere_instance: 10.0.0.56					T Street	Permit les les		P	tus d'a	
Images	Affichag											
Pairon de clés	- 34				Annula	Appropriat	10	Ase	Actio			
Groupes de serveurs	- rh				2 minute	Construction	motation	1.90				
Weiturnen 👂	C per	meser i 10.0.0.56. amce idac.5001.651d.0.1616.3eff.fec4.506d	((11)	f tiny LFO Active of	nova	Aucun Br	i nctionnement	0 minute	Cré	er un k	nutantz	iné 🛛

Nous demandons l'allocation d'une adresse IP dans le réseau « public » et nous cliquons sur « Allocation d'IP » :

Instances - OpenStack Dashb	× T				
(←) → ♂ @	① 19	2.168.1.211/dashboard/project/instances/	🖂 合	Q Rechercher	¥ ⊪\ © ¢ ≡
Copenstack. maam	9.72. S				🛔 demo -
Projet 🛩	Devis	Allouer une IP flottante		×	
Accès API	(B	Pool *			
Compute 😽 😽	Ins	public •	Description :		
Vue d'ensemble		Description	Allouer une IP flottante depuis	un peol d'IP	
Instances (Quotas de projet		Discourses to Million and Plus d'action
Images	Afficin		P nottantes		
Paren de clés					de Age Actions
Groupes de serveurs			An	Allocation (CP)	mentation
Witames >	0	conce tooic pear tooic was to perfined peop	4		discomment 0 minute Créer un instantané

Une adresse IP flottante nous est allouée, nous pouvons maintenant l'associer à notre instance en cliquant sur « Associer » :

Instances - OpenStack Dashb	×	£						=	Ó	×:
€ → ♂ @	① 193	2.168.1.211/dashboard/project/instances/		🖾 🕁	Q Rechercher		± 1		3 🕸	≡
🖸 openstack. 🔳 dam	9.72 J								å de	me.*
Projet 👻	Proje	Gérer les Associations d'IP flo	ottar	ntes		Suc	coès : IP flottante 172.24.4 ués.	38	1	×
(Receipt Party)	Ins	Adresse IP *		Selectionner Ladresse IP que v	ous souhaiter faire					
Compute 👻	1110	172.24.4.38 •	+	correspondre à l'instance ou au	port sélectionne(e).	- 8				
Vus d'ensemble		Port à associer				10				
Instances		premiere_instance: 10.0.0.56					E Supervisi les Malarce		Plus d'a	ections
Images	Affich									
Parez de clés	0				Annuler Associ	de	e Age Act	iona		
Graupes de serveura			_			me	entation			
Witarian >	C #	nimere_i 10.0.0.56, namce Ndac.5041.651d.0.1616.3eff.fec4	508d	m1 timy LFO Active of	nova Aucun	En	onement 0 minute C	vier ur	willand	ané -

L'adresse IP flottante qui a été associée apparaît maintenant en face de notre instance :

Instances - OpenStack Dashb-	× +											17	ſ	3 %
⊖ → ♂ @	192.168.1.21	1/dashboard	d/project/instances/?action=row_u	pdate8tab	le	©	10	Q, Recher	cher		$\overline{\tau}$	111		8
openstack. 📾 demi	•													iemo =
Projet V Accès API	Projet / Compu	te / Instance	8											
Compute 🗸	Instanc	es												
Vue d'ensemble			ID de l'instance = •				Filtre	t Lance	er une inst	ance B Sop	ömer les inst	lances	P	us d'ac
Images	Affichage de 1 élé	iment												
Paires de clés Groupes de serveurs	Nom de Finstance	Nom de A l'image	dresse IP	Gabarit	Paire de clés	Statut		Zone de disponibilité	Táche	État de l'alimentation	Age	Actio	กร	
Volumes >	D premiere_i nstance	- 1 R	0.0.0.56 <mark>172.24.4.38.</mark> tac 50a1 b51d 0.1816.3eft.fec4.508d	m1.tiny	LFO	Active	ie s	nova	Aucun	En fonctionnement	7 minutes	Cré	er un ir	istantar
Réseau > Orchestration >	Affichage de 1 élé	iment												

Si après avoir dissocié une adresse IP flottante d'une instance vous la libérez elle sera rendue au pool créé par les administrateurs du système et il n'y a aucun moyen de vous la réattribuer.

Ceci peut être un problème si jamais vous avez référencé cette adresse à l'extérieur (DNS, Firewall, applications tierces..)

Modification du groupe de sécurité

Les groupes de sécurité sont des pare-feu qui se placent devant les instances pour filtrer les flux réseau entrants et sortants. Le pare-feu par défaut autorise tous les flux sortants de l'instance, et bloque tous les flux entrants.

L'administrateur du projet peut créer des groupes de sécurité en fonction de ses besoins (dans la limite des quotas de son projet)

Nous allons modifier le groupe de sécurité par défaut afin d'autoriser les connexions SSH (protocole TCP port 22) vers notre instance.

Les groupes de sécurité sont dans l'onglet « Projet \rightarrow Réseau \rightarrow Groupes de sécurité », pour le moment nous n'avons que le groupe « default » <u>qu'il ne faut surtout pas modifier</u> :

000	0	102.16	R 4 044 Jul	achthe and fereningt frame with a		O Pacharathan		L In	m		
o v c w	W	126.10	0.1.211/0	autonaro/budentsermitA*0	© u	- sectorcher		X == 1	ω.		1
openstack	, 🕅 den	10.7								demo 🕇	8
Projet	~										
,	ccas API	Pro	jet / Rése	au / Groupes de sécurité							
	1	Gr	oun	es de sécurité	5						
Compute	>	G	oup	es de securite							
Volumes	>										
Réseau	×			Filter Q	+ Créer un g	roupe de sécurité	Supprimer les t	Groupes	de St	curthé	ĺ
Topologie	du réseau	Affic	hage de 1 e	élément							
	Réseaux		Nom	ID du groupe de sécurité		Description	(Actions			
	Routeurs	0	default	4545dc36-f071-43c5-b902-a7b	5fe35b586	Default secu	arity group	Gérer I	es Ré	gles	
Groupes d	e sécurité	Affici	hage de 1 e	élément							

Nous allons créer un nouveau groupe de sécurité en cliquant sur 'Créer un groupe de sécurité', nous l'appelons 'ssh_externe' et nous validons la création en cliquant sur le bouton en bas à droite :

		- 4	(-			
opensta	ICK. m demo -			_			demo *
Projet	Créer un groupe de sécurité			×			
	Nom *	Description					
1000	ssh_externe	Description		87 E			
Malamer	Description	Les groupes de sec filtrage IP qui sont a	unte sont des ensembles de appliqués aux paramètres de l	interface			-
Reseau		réseau de la machin sécurité créé, yous	ne virtuelle. Une fois le groupe pouvez lui ajouter des règles.	de	10000		
Topole							
					lions		
					Serer In	n Røgle	6
			for second second second second second	1			

Suite à la création du groupe de sécurité nous arrivons directement sur l'écran de gestion des règles :

Gérer les rè	gles du gro	oupe de si × +				<u>7</u> 7		×
↔ ∀ ↔ ↔		192.168.1.211/dashboard/project/security_g	⊌ ☆	Q Rechercher	$\overline{\mathbf{T}}$	111	•	Ξ
🖸 opensta	ck. 🛛	🔳 demo 🕶					🛔 de	emo • ^
Projet	Ajo	uter une règle			ж			
	Règle	e *						
Compute	Règ	gle TCP personnalisée -	Description	1:				
Volumes	Desci	ription 🛛	Les règles définiss instances associé	ent quel trafic est autorisé es au groupe de sécurité. U	pour les Jn groupe de			
Réseau	Aut	toriser le SSH depuis l'extérieur	sécurité se compo	se de trois parties principa	les : àgle désiré			
Topolo	91)		ou utiliser des règi	es personnalisées. Les opt	tions sont	pprim		ilės
	Direc	tion	Règle ICMP perso	nnalisée.	imansee ou	DE LA COL		
	Ent	rée 🔹	Ouvrir Port / Plag UDP, vous pouvez	je de Ports : Pour les règle choisir d'ouvrir soit un seul	es TCP et port, soit			
Groupe	Ouvri	ir.*	une plage de ports ports", vous aurez	. En sélectionnant l'option l'espace pour fournir le port	"plage de t de début et	ns		
	Por	t 🔹	le port de fin pour l ICMP, vous devez	a plage concernée. Pour le en revanche fournir un type	es règles ICMP et un			
Orchestratio	or Port*	0	code dans les esp	aces proposés.		oprime	er une Rég	le
Stockage d'o	b 22		Distant : Vous dev par cette règle. Vo	ez spécifier la source du tr us pouvez faire cela soit so	afic autorisé ous la forme			
Identité	Dista	nt * 0	d'un bloc d'adresse groupe source (gro	es IP (CIDR), soit sous la fo upe de sécurité). Choisir u	orme d'un n groupe de	oprime	er une Rég	le
	CID	R 🔹	sécurité comme se de ce groupe à ace	ource autorise n'importe qu céder à n'importe quelle au	elle instance tre instance			
	CIDR	*0	au travers de cette	rêgle.				
	0.0	.0.0/0						
	-							
				Annuler	Ajouter			
				3ł				~

Gerer les règles du	groupe de s	~ page								199	- u	
-)→ C @	(i) 19	92.168.1.211/das)	board/project/	security_(:	… ⊚	\$ Q	Rechercher		$\underline{\star}$	10	•	¢ 3
openstack.	🕅 demo	•										demo •
Projet	ės API	Projet / Réseau	/ Groupes de s	écurité / Géré	ir les règle	s du groupe	d					
Compute	,	Gérer le sécurité	es règle : ssh	es du exter	grou ne (4	ipe d 43f65	e 28c-					
Réseau		6e66-43	324-94	ab-ae	26e	97f06	668)					
Topologie du	réseau						+ Ajou	ter une règle	Sup	işinin Martin	er læs R	áglés
R	éseaux outeurs	Affichage de 2 élé	ments									the strength
Groupes de s IP flo	ecurite) Itantes	Direction	Type de protocole (EtherType)	Protocole IP	Plage de ports	Préfixe IP distante	de sécurité distant	Description	Action	15		
Orchestration	>	🗆 Sortie	IPv4	Tous	Tous	0.0.0.0/0		a	Sup	primi	r une R	ègle
Stockage d'objet	>	Sortie	IPv6	Tous	Tous	::/0	18	a :	Sup	prime	r une Re	ègle
dentité	>	Affichage de 2 élé	ments									

En cliquant sur « Ajouter une règle » nous allons ajouter la règle dont nous avons besoin pour autoriser les connexions SSH, nous créons une « Règle TCP personnalisée , en « Entrée», pour le « Port » numéro « 22 » (SSH), quelle que soit l'origine « CIDR 0.0.0/0) :

Gérer les règi	es du groupe de 🗁 🗙 🕂					- 27	2	0	×
-)→ C' @	① 192.168.1.211/dashboard/project/sec	urity_g	… ⊠ ☆	Q, Rechercher	$\underline{+}$	111		œ	≣
openstac	k. 🕅 demo +						- 0	å den	•
Projet	Ajouter une règle				×	l			
Course to	Régle *								
Compute	Régle TCP personnalisée	٠	Descriptio	n :					
Volumes	Description O		Les régles définis instances associ	sent quel trafic est autorisé po les au groupe de sécunté. Un	ur les groupe de				
Réseau Tispologi	Autoriser le SSH depuis l'extérieur		Règle : Vous pou ou utiliser des règ	ose de trois parties principales ivez spécifier le modèle de règi les personnalisées. Les option polisiée. Bèrte LIPP Personna	le désiré na sont				
	Direction		Règle ICMP perso	nnalisée. nnalisée.	ROBE OU				
	Entrée		Ouvrir Port / Pla	ge de Ports : Pour les règles	TCP et				
Graupe	Ouvrir *		une plage de port ports", yous aures	s En sélectionnant l'option "pla respace pour fournir le port de	age de e début et	ns (
	Port	•	le port de fin pour	la plage concernée. Pour les r	ègles MR et un				
Orchestration	Port* O		code dans les es	paces proposés.			in un	e Regil	1
Stacksge d'ob	22		Distant : Vous de par cette règle V	vez spécifier la source du trafic ous pouvez faire cela soit sous	autorisé la forme				
Mantha	Distant * 😡		d'un bloc d'adress groupe source (gr	es IP (CIDR), soit sous la form oupe de sécurité). Choisir un g	ne d'un proupe de	iprim	un lain	r Régli	
ADDETUKE	CIDR	•	sécurité comme s	cource autorise n'importe quelle	instance				
	CIDR" 0		au travers de cett	e règle.	Trana to e				
	0.0.0.0/0								
				Annules	Aputer				

Une fois la modification validée les différentes règles qui composent le groupe de sécurité s'affichent :

) → œ w	W 19	2.168.1.211/das	npoard/project/	security_(;	6	2 12 0	6, flechercha	R.C.	± ≡/ 0	n e	-
openstack.	🕅 demo 🕯	4								a der	no •
Projet	¥ és API	Projet / Résea	u / Groupes de	sécurité / Gér	er les règi	es du group	ed	iccès (Règle ajour /TORISER IP+4 2)	tée avec succés 2/tcp from 0.0.0	0/0	
Compute Volumes Réseau		Gérer le sécurite 6e66-4	es règl é : ssh_ 324-94	es du _exter ab-ae	gro ne (26e	upe c 43f6 97f0	le 528c- 668)				
Topologie du Ri	réseau Iseaux	Affichage de 3 él	éments				+ A	jouter une règle	C Supprime	les Ragi	1
Groupes de s IP flo	écunté	Direction	Type de protocole (EtherType)	Protocole IP	Plage de ports	Préfixe IP distante	Groupe de sécurité distant	Description	Actions		
Orchestration	>	Sortie	IPv4	Tous	Tous	0.0.0.0/0	141	54) -	Supprimer	una Régl	
Stockage d'objet	>	D Sortie	IPv6	Tous	Tous	:::/0	(#)	1991	Supprimer	une Régi	
denote.	ć	Entrée	IPv4	TCP	22 (SSH)	0 0 0 0/0	-	Autoriser le SSH depuis Fextérieur	Supprimer	une Régl	
		Affichage de 3 él	éments								

Il faut désormais attribuer le groupe de sécurité que nous venons de créer à notre instance afin que vous puissions nous connecter. Ceci se fait depuis l'écran « Projet \rightarrow Compute \rightarrow Instances », en cliquant sur 'éditer les groupes de sécurité' dans le menu déroulant en face de notre instance :

Instances - OpenStack Das	hbe × +												57.		×
€) → ♂ @	① 192.168.1.3	211/dashbo	ard/project/instances/				94	Q B	echercher			± 1	n 🗉		-
🖸 openstack. 🔳 📾	ema •													demo	
Accès API Compute 🗸	Instand	es													
Vue d'ensemble	í.		ID de l'instance = •				Filtre	• Lar	icer une in	stance Sur	gennar les inst	ances	Plus	d'actio	its .
Images Paires de clés Groupes de serveurs	Affichage de 1 é	Nom de l'image	Adresse IP	Gabarit	Paire de clès	Statut	Z	one de sponibilité	Tàche	État de l'alimentation	Age	Action	5		
Volumes > Réssau >	promiere_i natance	cirros-0. 4.0-x86 64-dis k	10.0.0.56, 172.24.4.38, fdac 50a1 b51d 0.1816 3eff fec4 508d	m1.tiny	LFO	Active	a' n	Na	Aucun	En fonctionnement	28 minutes Dissocier ur	Crée	r un inst	lantanê Bante	P
Orchestration > Stockage d'objet > Idontité >	Affichage de 1 é	lement									Attacher finth Détacher finth Éditer finsta Attacher le vi Détacher le vi Mettre à jour Éditer les gr Console Voir le journa	erface derface nice olume Volume les mét oupes d oupes d	adonnéi a sécurr a sécurr	es té de po	at

Les groupes de sécurité disponibles mais non associés à l'instance apparaissent dans la colonne de gauche, ceux associés à l'instance dans la colonne de droite. Cliquer sur le '+' ou le '-' en face d'un groupe le fait passer d'une colonne à l'autre :

Instances - OpenStack Dashi	н= × +						
€ → ୯ ŵ	① 192.16	8.1.211/dashboard/project/instances/	··· 🗵 🕁	Q Rechercher	± 1	n 🗊 🔹	≡
🖸 openstack. 🔳 dan	10.**					🛓 de	mo •
Projet 🛩	Projet /	Éditer l'instance		×			
Compute 😽	Insta	Information * Groupes de sécunté					
Var densemble		Ajout ou retrait de groupes de sécurité à cette inst	ance à partir de la liste des groupes de séc	unté disponibles.			
Instances		Avertissement: Si vous modifiez ici les groupes d	le sécurité, la modification sera appliquée à ofaces et elucions annues de sécurité par	toutes les interfaces de		Plus g'artis	115.*
Images	Affichage	les groupes de sécurité de port".	eraces er broseors (konhes de secritie be	port onnaes plator Earler			
Parea de ciés	D Non	Tous les groupes de sécurité	Q Groupes de sécurité associ	iés à l'instance Q	dion Age Action	•22	
Groupes de serveurs	-			. Filte int			
Volumes 🗦	- perm	ssh_externe	default	1.0	AND TRACK		
Réseau >	. HOTOM	Personal States of States		,) ter	ment 20 minutes Cres	un instantane	
Orchestration >	Affichage i			Annuler Enregistrer			
Stockage d'objet 🔹 🗲				_			
Identité >							

Nous associons le groupe 'ssh_externe' à l'instance et nous enregistrons:

 C mo Projet Comparts Comparts Information* Cotopes de sécurité de groupes de sécurité de port. Comparts Comparts Information* Cotopes de sécurité de port. Comparts Comparts) (2) ≡ ▲ demo •
Openstack mitisme * Projet Éditer l'instance Accia API Éditer l'instance Compute Information * Oroupes de sécurité à cette instance à partir de la liste des groupes de sécurité disponibles. Avertissement: Si vous modifier ici les groupes de sécurité la modification sera appliqué à toutes les interfaces de finistance. Si cette instance posside plusieurs interfaces et plusieurs groupes de sécurité par port, utilisez plutét "Editer les groupes de sécurité de port". Planes: de clés Nor Planes: de clés Nor Tous les groupes de sécurité (Filter Q) Groupes de sécurité associés à l'instance Q Instance Actione	▲ demo •
Projet Éditer l'instance Accis APP Accis APP Accis APP Information* Croupes de sécurité Compute Information* Croupes de sécurité Vuit d'ensamble Ajout ou retrait de groupes de sécurité à crite instance à partir de la liste des groupes de sécurité disponibles. Avertissement: Si vous modifiez ici les groupes de sécurité à crite instance possède plusieurs interfaces et plusieurs groupes de sécurité par port, utilisez plut d'Editer les groupes de sécurité de port. Pairres de clés Michaine Pairres de clés Tous les groupes de sécurité (Editer Q) Groupes de sécurité à souciés à l'instance Q Instance Q	e factoria +
Compute Instri Croupes de sécurité Van d'ensemble Ajout ou retrait de groupes de sécurité à cette instance à partir de la liste des groupes de sécurité disponibles. Ajout ou retrait de groupes de sécurité à cette instance à partir de la liste des groupes de sécurité disponibles. Instance Avertissement: Si vous modifiez ici les groupes de sécurité disponibles. Instance sous modifiez ici les groupes de sécurité disponibles. Instance sous modifiez ici les groupes de sécurité disponibles. Instance sous modifiez ici les groupes de sécurité disponibles. Instance sous modifiez ici les groupes de sécurité disponibles. Instance sous modifiez ici les groupes de sécurité disponibles. Instance sous modifiez ici les groupes de sécurité disponibles. Instance sous modifiez ici les groupes de sécurité disponibles. Instance sous modifiez ici les groupes de sécurité disponibles. Instance sous modifiez ici les groupes de sécurité disponibles. Instance sous modifiez ici les groupes de sécurité disponibles. Instance sous modifiez ici les groupes de sécurité disponibles. Instance sous modifiez ici les groupes de sécurité disponibles. Instance sous modifiez ici les groupes de sécurité disponibles. Paires de clés Nort Tous les groupes de sécurité proteine sous ce sécurité disponibles. Instance sous de sécurité disponibles. Instance sous de sécurité disponibles.	factions +
Vur demaentále Ajout ou retrat de groupes de sécurité à cette instance à partir de la liste des groupes de sécurité disponibles. Ajout ou retrat de groupes de sécurité à la modification sera appliquée à toutes les interfaces de finistance. Avertissement: Si vous modifiez ici les groupes de sécurité disponibles. Image: Africhage Africhage Africhage Age: Actions Pares de clés Nor Tous les groupes de sécurité (riber Q) Groupes de sécurité associés à l'instance Q Image: Age: Actions	e Carlines +
Instance Avertissement: Si vous modifiez ici les groupes de sécurité, la modification sera appliquée à toutes les interfaces de finstance. Si cette instance posside plusieurs interfaces et plusieurs groupes de sécurité par port, utilisez plutôt "Editer Pares de clés Nor Tous les groupes de sécurité Groupes de sécurité associés à l'instance Q Adde Actione	e factions -
Insance d'une insance possed p	
Planes de clés Not Tous les groupes de sécurité filler Q Groupes de sécurité associés à l'Instance Q atation Age Actions	
- rim	
Groupes de serveurs	
Naturnes > Aucun groupe de sécurité trauvé:	ann an the second second
Résau: > mail	tuntane -
Orchestration > Attichage	
Stockage diobyet >	
Annuer Exemptore	

Connexion à l'instance

Nous pouvons désormais nous connecter en SSH à notre instance depuis notre serveur DevStack en utilisant l'adresse IP flottante de notre instance et le user « cirros », l'adresse IP flottante est affichée en face de l'instance dans le Dashboard :

```
stack@microwave:~$ ssh cirros@172.24.4.38
The authenticity of host '172.24.4.38 (172.24.4.38)' can't be established.
ECDSA key fingerprint is SHA256:Lq/GLmCuTBJrUqJ7GpB/q8sz6VqGfrqQbZ0kIzSpp3o.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.24.4.38' (ECDSA) to the list of known hosts.
$ uname -a
Linux premiere-instance 4.4.0-28-generic #47-Ubuntu SMP Fri Jun 24 10:09:13 UTC 2016 x86_64 GNU/Linux
```

Comme nous avons lancé instance en spécifiant notre clé SSH nous avons pu nous connecter sans avoir à spécifier de mot de passe. Si nous avons un problème avec notre clé SSH nous pouvons utiliser le mot passe « cubswin:) » ou « gocubsgo » (à partir de cirros v0.4)



Notez que le nom de notre instance est 'premiere_instance' et que le hostname de l'instance est 'premiere-instance' car le caractère '_' est interdit dans les entrées DNS (et donc dans les hostnames)

Affichage des informations sur l'instance

Depuis la liste des instances dans le Dashboard (projet \rightarrow compute \rightarrow instances) nous pouvons afficher les informations sur une instance, sa log système ainsi que sa console (écran / clavier).

En cliquant sur le nom du serveur, nous affichons 4 onglets (Vue d'ensemble, Journal, Console, Log des actions) :

premiere_instance	- OpenSt	idixi 🕂				- 57	s [1	×
€ → ୯ û	٢	192.168.1.211/dashboard/pr	oject/instances/ 🛛 💀 🔂	Q, Rechercher	$^{\pm}$	11		8	Ξ
openstack.	🕅 den	no *					A d	mo	•
Projet	¥ és API	Projet / Compute / Instan	ces / premiere_instance						
Compute	~	premiere_in	stance		Créer	un ins	itantarié		3
Vue d'ens	emble	Vue d'ensemble Inter	aces Journal Console	Log des actions					
inst It Pairos d	tances mages de clés	Nom ID Description ID du Projet Statut	premiere_instance 0a40946a-6141-40c7-a7d8-6ca3dd - 5cd4212611664857a9e941a7b5d3 Active	158371 Se13					
Groupes de se	erveurs	Verrouillé Zone de disponibilité	False						
Volumes	>	Créé Age	13 octobre 2019 19:16 47 minutes						
Réseau	>	Spécifications							
Orchestration	>	New Av Coloria	123322						
Stockage d'objet	>	ID du gabarit RAM	1 512Mo						
Identité	>	VCPUs Disque	1 VCPU 1Ge						
		Adresses IP							
		private	10.0.0.56, stac 50a1 b51d:0.f816	3eff.fec4.508d, 172.24.4.38					
		Groupes de sécurité							

L'onglet « Journal » donne la log système de l'instance (sur écran cliquer 'Voir le journal complet' permet d'afficher le journal complet et pas seulement les 35 dernières lignes) :

	e - OpenStaci X Sta	- 0 ×
€) → ♂ @	① 192.168.1.211/dashboard/project/instances ···· 😇 🏠 🔍 Rech	ercher 👱 🛍 🗂 📽 🗏
openstack.	🕅 demo 🕶	🛓 demo 🖛
Projet	Projet / Compute / Instances / premiere_instance is API	
Compute	 premiere_instance 	Créer un instantané 🔹
Vue d'en	semble Vive d'ensemble Interfaces Journal Console Log des act	bhs
i.	Journal de la console de l'instance	
Paires	de clác	10
1. 101.0.0	Congreat on Jonuar 32	Go Voir le journal complet
Groupes de s	verveurs sahd hast keys === BEGIN SSH HOST KEY KEYS	Go Voir te journaî cômptet
Groupes de s Volumes	congusur au journal 35 congusur au jour	Gi Voir le journal complet Im/HEncjukocf 9mudyG5xxE3cn)t 381k/KQLn4m3 y4/IFcf/h1C7AEyCa139sPx8kTsunoCrspHduKX
Groupes de s Volumes Réseau	anours	Go Voir le journat complet m/HZncjskscF9hud9Q5scE3enkt3@EinKNQLnAn3 y+YTFcT/hLC7AEyCat39sPv8wTzundCrapHduKX
Groupes de s Volumes Réseau Orchestration	were said hoat keys mere Longusur du journal In san-ras ANAB3HERCIyc12AAAD0408AA80000HU2Lu67m0x2b102C3KCHU3 san-ras ANAB3HERCIyc12AAAD0408AA80400HU2Lu67m0x2b102C3+G8H5K12 san-ras ANAB3HERCIyc12AAAD0408AA80400HU2Lu67m0x2b102C3+G8H5K12 san-ras ANAB3HERCIyc12AAAD0408AA80400HU2Lu67m0x2b102C3+G8H5K12 san-ras ANAB3HERCIyc12AAAD0408AA80400HU2Lu67m0x2b102C3+G8H5K12 san-ras ANAB3HERCIyc12AA00408AA80400HU2Lu67m0x2b102C3+G8H5K12 san-ras ANAB3HERCIyc12AA00408AA80400HU2Lu67m0x2b102C3+G8H5K12 san-ras ANAB3HERCIyc12AA00408A80400HU2Lu67m0x2b102C3+G8H5K12 san-ras ANAB3HERCIyc12AAA00408AA80400HU2Lu67m0x2b102C3+G8H5K12 san-ras ANAB3HERCIyc12AA00408A80400HU2Lu67m0x2b102C3+G8H5K12 san-ras ANAB3HERCIyc12AAA00408A80400HU2Lu67m0x2b102C3+G8H5K12 san-ras ANAB3HERCIyc12AA004084A80400HU2Lu67m0x2b102C3+G8H5K12 san-ras ANAB3HERCIyc12AA004084A80400HU2Lu67m0x2b102C3+G8H5K12 san-ras ANAB3HERCIYC12AA004084A80400HU2Lu67m0x2b102C3+G8H5K12 san-ras ANAB3HERCIYC12AA004084A80400HU2Lu67m0x2b102C3+G8H5K12 san-ras ANAB3HERCIYC12AA00404A744 san-ras ANAB3HERCIYC12AA004084A7408 san-ras AnaB4040447448 san-ras AnaB404044748 san-ras ANAB3HERCIYC12AA004044A748 san-ras AnaB404044748 san-ras AnaB404044748	Go Voir le journai complet m/HEncjakkocf/9hudydSoxeE/enkt385LikUqLn4m3
Groupes de s Volumes Réseau Orchestration Stockage d'objet	we use Longusur du journal 35 were used du journal 35 were used du journal 35 seneurs 55 seneurs 56 <td>Go Voir le journat complet Im/HZmCjskGCF3hu49@5xxE3enAt38LikKQcnAe3 Voir le journat complet y=Y1FcT/hLC7AEyCal398PvMe1ZundCrspHdu4X Voir le journat complet i0e1:051d:0:f816:3eff:fec4:508d/64 Voir le journat complet</td>	Go Voir le journat complet Im/HZmCjskGCF3hu49@5xxE3enAt38LikKQcnAe3 Voir le journat complet y=Y1FcT/hLC7AEyCal398PvMe1ZundCrspHdu4X Voir le journat complet i0e1:051d:0:f816:3eff:fec4:508d/64 Voir le journat complet

L'onglet « Console » permet d'avoir l'écran et le clavier de l'instance :



Il peut être nécessaire de cliquer sur 'Cliquer ici pour ne voir que la console' afin que la console soit utilisable.

La disposition des touches du clavier peut ne pas être correcte (AZERTY/QWERTY)

Suppression des éléments créés

Depuis le Dashboard nous pouvons supprimer les éléments crées :

- L'instance depuis le menu « Projet → Compute → Instances » en cliquant sur le menu déroulant à droite de l'instance concernée et en sélectionnant « Supprimer l'instance ».
- Le volume associé de la même façon depuis le menu « Projet \rightarrow Volumes \rightarrow Volumes »
- Le groupe de sécurité depuis l'écran « Réseau → Groupes de sécurité »
- L'adresse IP flottante depuis l'écran « Réseau → IP flottantes »,choisir « Libérer l'IP flottante » dans le menu déroulant à droite de l'adresse

Utilisation des images

Les éditeurs de systèmes d'exploitation fournissent des images « cloud ready », contrairement aux images classiques qui sont des images de DVD d'installation de l'OS les images « Cloud Ready » sont des images de disques sur lesquels l'OS a été installé.

Téléchargement de l'image Ubuntu « Cloud Image »

Nous allons télécharger l'image cloud de Ubuntu 18.04, les images cloud de Ubuntu 16.04 sont disponibles à l'adresse suivante:

```
https://cloud-images.ubuntu.com/releases/18.04/release/
```

Il y a deux points auxquels il faut faire attention :

- Tout comme pour les serveurs réels, les images cloud sont dépendantes du processeur, nous allons prendre une image 'amd64' qui correspond aux architectures Intel/AMD en 64 bits
- Ensuite le format de l'image dépend de l'hyperviseur (le système de cloud) utilisé : HyperV (Microsoft Azure), OVF (VMware), QCOW2 (QEMU et KVM). DevStack est basé sur l'hyperviseur KVM, nous prenons donc l'image QCOW2

Nous allons prendre le fichier ubuntu-18.04-server-cloudimg-amd64.img (le libellé 'Cloud image for 64-bit computers (USB image)' pour cette image prête à confusion)



Contrôle de l'intégrité de l'image téléchargée

Une fois l'image téléchargée nous contrôlons la somme de contrôle du fichier téléchargé (les sommes de contrôle sont dans des fichiers SHA512SUMS et MD5SUMS sur la page de téléchargement).

Plusieurs algorithmes de hashage sont utilisés, nous pouvons utiliser sha512sum pour contrôler l'intégrité du fichier téléchargé, MD5 n'est plus considéré comme un algorithme sûr, mais c'est celui utilisé par Glance pour le contrôle d'intégrité (on retrouve cette somme de contrôle dans le champ checksum des commandes openstack image).

stack@microwave:~\$ **sha512sum ubuntu-18.04-server-cloudimg-amd64.img** <mark>e39393b4b347790b18c..4c0fc4be4cbcf084c56d</mark> ubuntu-18.04-server-cloudimg-amd64.img

stack@microwave:~\$ **md5sum ubuntu-18.04-server-cloudimg-amd64.img** <mark>33e7ble0d309877a3f6c27bc52dlbd68</mark> ubuntu-18.04-server-cloudimg-amd64.img

Chargement de l'image dans Glance

Nous chargeons l'image dans Glance qui est en charge du catalogage et du stockage des images dans OpenStack.

Tout utilisateur d'un projet peut charger une image dans Glance, elle ne sera visible que dans son projet (ce comportement est modifiable par l'administrateur).

L'administrateur du système OpenStack (ou un utilisateur disposant des droits nécessaires) peut charger une image dans Glance et la rendre publique c'est à dire visible depuis tous les projets.

Nous chargeons les variables d'environnement nécessaires pour nous authentifier comme user admin dans le projet admin et nous chargeons l'image dans Glance en la rendant publique :

<pre>stack@microwave:~\$ source devstack/openrc stack@microwave:~\$ openstack image create</pre>	admin admin public \ disk-format qcow2 \ container-format bare \ file ubuntu-18.04-server-cloudimg-amd64.img \ ubuntu-18.04
Field Value	,

+	-+			+	
checksum	<mark>33e7b1e0d309877a3f</mark>	5 <mark>c27bc52d1bd68</mark>		L. L	
container_format	bare			L. L	
created_at	2019-10-13T20:48:03	32		1	
disk format	qcow2			1	
file	/v2/images/7ab813dd	1-6937-4f96-8cd1-ed0ec64e8a	d5/file		
id	7ab813dd-6937-4f96-	-8cd1-ed0ec64e8ad5		1	
min disk	0			1	
min ram	0			1	
name	ubuntu-18.04				
owner	0302dd0df6e74588a2	79d139e452b219		1	
properties	os hash algo='sha5	2 ', os hash value=' <mark>e39393b</mark>	4b347790b1	18c4c0fc4be4cbcf084c56d', os hidden='False'	
protected	False				
schema	/v2/schemas/image			1	
size	344588288				
status	active			1	
tags	1			1	
updated at	2019-10-13T20:48:11	Z			
virtual size	None				
visibility	public			1	
+	-+			+	
stack@microwave:~\$	openstack image list				
+		·	++	ŧ	
ID		Name	Status		
+		·	++	+	
67d1e1b2-a9f8-45	1f-afd2-f4697da2b6c7	cirros-0.4.0-x86_64-disk	active		
7ab813dd-6937-4f	96-8cd1-ed0ec64e8ad5	ubuntu-18.04	active	1	
+		·	++	ŧ	



Certaines images sont compressées (en .zip ou .xz) il faut les décompresser avant de les charger dans Glance.

D'une manière générale Glance ne fait aucun contrôle sur l'image : charger une image avec les mauvais paramètres, charger une image compressée ou d'un format incorrect produira des erreurs inattendues au moment du lancement d'une instance utilisant cette image.

Vérification de la disponibilité de l'image

Nous vérifions que l'image est disponible dans le projet demo, on note que l'utilisateur demo n'a pas les autorisations nécessaires pour supprimer cette image car il s'agit d'une image publique :



Lancement d'une instance en ligne de commandes

Nous pouvons lancer une instance grâce à la commande openstack de la même manière que nous l'avons fait en utilisant précédemment en utilisant le Dashboard.

Nous commençons par créer le volume système que nous allons appeler ubuntu-18.04_vol à partir de l'image ubuntu-18.04, cette opération peut prendre une ou plusieurs minutes, et l'image passe par plusieurs états avant d'être disponible.

stack@microwave:~\$ source devstack/open	rc demo demo	
<pre>stack@microwave:~\$ openstack image list</pre>		
+	+	+
ID	Name	Status
+	+	+
1 67d1o1b2_>9f8_451f_>fd2_f4697d>2b6a7	airros-0 1 0-x86 61-disk	lactivo

7ab813dd-6937-4f96-8d	cd1-ed0ec	54e8ad5 ubuntu-18.04 active	
<pre>stack@microwave:~\$ open</pre>	nstack vol	ume createsize 10image ubuntu-18.04 ubuntu-18.04_vol	
+ Field	+ Value	/ I	
+	+	+	
allaChments	1 []		
availability_2011e	HUVA falso		
DOOLADIE	laise		
consistencygroup_ia	None 2010 10	12001-00-45 000000	
created_at	2019-10-	-13721:02:45.000000	
description	None		
encrypted	False		
1 10	eac2644.	-216/-410a-8024-a1e85844a8ac	
multiattach	False		
name	ubuntu	.8.04_vo1	
properties	1		
replication_status	None		
size	1 10		
snapshot_id	None		
source_volid	None		
status	creating	I I	
type	lvmdrive	er-1	
updated_at	None		
user_id	57d02b0	338d149a1aac96ce6757a3a14	
<pre>stack@microwave.~\$ one</pre>	+		
+			
Field +		Value	
attachments		[]	
availability_zone		nova	
bootable		false	
consistencygroup_id		None	
created_at		2019-10-13T21:02:45.000000	
description		None	
encrypted		False	
id		edc26441-2f67-410d-8b24-a1e85844d8dc	
multiattach		False	
name		ubuntu-18.04_vol	
os-vol-tenant-attr:te	enant_id	5cd4212611654857a9e941a7b5d35e13	
properties	_		
replication status		None	
size		3	
snapshot id		None	
source volid		None	
status		downloading	
type		lvmdriver-1	
updated at		2019-10-13T21:03:21.000000	
user id		57d02b0938d149a1aac96ce6757a3a14	
volume_image_metadata	a	{u'signature_verified': u'False'}	
<pre>+stack@microwave:~\$ oper</pre>	nstack vol	ume show ubuntu-18.04 vol	
+		//////////////////////////////////////	
+		vaiue	
attachments		[]	
availability zone		nova	
bootable		true	
consistencygroup id		None	
created at		2019-10-13T21:02:45.000000	
description		None	
encrypted		False	
id		edc26441-2f67-410d-8b24-a1e85844d8dc	
. multiattach		False	
name		ubuntu-18.04 vol	
os-vol-tenant-attr:te	enant id	5cd4212611654857a9e941a7b5d35e13	
properties			
replication status		None	
size		10	
snapshot id		None	
source volid		None	
status		available	
type		lvmdriver-1	
updated at		2019-10-13T21:04:42.000000	
user id		57d02b0938d149a1aac96ce6757a3a14	
volume image metadata	a	{u'checksum': u'33e7b1e0d309877a3f6c27bc52d1bd68', u'min ram': u'0'.	
		u'disk format': u'gcow2', u'image name': u'ubuntu-18.04',	
1		u'image id': u'7ab813dd-6937-4f96-8cd1-ed0ec64e8ad5'.	
I		u'signature verified': u'False', u'container format': u'bare',	
1		u'min_disk': u'0', u'size': u'344588288'}	
+		+	

Création d'un groupe de sécurité

Nous allons créer notre propre « security group » qui autorise tout le trafic en sortie (c'est le comportement par défaut) et autorise le port 22 (SSH) en entrée.

<pre>stack@microwave:~\$ +</pre>	<pre>openstack security group create bastion_ssh ++</pre>	
Field	Value	
created_at description id project_id revision_number rules tags updated_at	<pre>2019-10-13T21:12:20Z bastion_ssh 9116395f-92dc-47c8-bd38-9424f6b7f88f bastion_ssh 5cd4212611654857a9e941a7b5d35e13 1 direction='egress', ethertype='IPv6' direction='egress', ethertype='IPv4' [] 2019-10-13T21:12:20Z 4</pre>	
stack@microwave:~\$	openstack security group rule createingressr protocol to	remote-ip 0.0.0.0/0 \ cpdst-port 22 bastion_ssh
+ Field	++ Value	_
<pre>+</pre>	<pre>// 2019-10-13T21:12:34Z // // // // // // // // // // // // //</pre>	<i>t</i>
<pre>created_at created_at description id name project_id revision_number rules tags updated_at</pre>	<pre>+</pre>	

Lancement de l'instance

Pour lancer notre instance nous avons besoin des éléments suivantes :

- Le volume système que nous venons de créer
- Un gabarit (flavor), par exemple m1.small
- Une clé publique pour pouvoir nous connecter en SSH à l'instance, elle est déjà disponible
- Le groupe de sécurité qui autorise tout le trafic en sortie et le port 22 en entrée
- Le réseau privé auquel attacher l'instance

stack@microwave:~\$ openstack volume list	C													
+ ID	+ Name		+ Status	Size	++ Attached to	<u>_</u>								
	ubuntu-18.04_	vol	available	10		-								
' stack@microwave:~\$ openstack security gu	coup list	L			,		<i></i>							
	Name	Des	cription		Project		' Tags +							
, 4545dc36-f071-43c5-b902-a7b6fe35b586 9116395f-92dc-47c8-bd38-9424f6b7f88f	default <mark>bastion_ssh</mark>	Defa basi	ault security tion_ssh	y group	5cd421261165 5cd421261165	54857a9e941a7b5d35e13 54857a9e941a7b5d35e13	, [] []							
<pre>stack@microwave:~\$ openstack flavor list ++++++</pre>														
--	--------------------------	----------------	-------------	----------------	--------	----------	---------------	---------	---------------------	--------------	----	--	--	--
++ ID Nar	ne	+ RAM	+ Disk	-+ Ephem	eral	+	-+ Is	Public	+					
++		+	+	-+		+	-+		+					
1 m1.	tiny	512	1	1	0	1	Tru	е	I.					
2 <mark>m1.</mark> 3 m1	. <u>small</u> medium	2048 4096	20 40	1	0	1 1	Tru	e	1					
4 m1.	large	8192	80	İ	0	4	Tru	e	İ					
42 m1.	nano	64	1	1	0	1	Tru	е	l .					
5 m1.	.xlarge	16384	160		0	8	Tru	e	1					
04 mii. c1 cii	rros256	256	1 1		0	1 1	Tru	e	1					
d1 ds	512M	512	5	Ì	0	1	Tru	e	l					
d2 ds:	l G	1024	10	1	0	1	Tru	е	l.					
d3 ds2 d4 ds2	2G	2048		1	0	2	Tru	e	1					
++		+	+	-+		+	-+		+					
stack@mic:	cowave:~\$	opensta	ack key	ypair li	st									
++ Name 1	 Tingerpr	 int						+						
++								+						
<mark>LFO</mark> 8	Bc:56:76	:56:d9:de	e:50:8	1:8b:4d:	6c:b1	:ee:55:1	b2:ae	I.						
++ stack@mic:	rowave:~:	s opensta	ack ne	twork li	st			+						
+					+	+-					-+			
ID Name Subnets														
0e6e2f0c-c4cc-4eca-8428-8117864b6e4a <mark>private</mark> 10b8dba4-4613-4462-b8f4-500bd6c7fc84 12ceb0b7-9ed6-4bfd-9160-ab44b4f868b2 public b3ee49a7-80b9-4898-8f53-a58cb1e3908c														
12ceb0b7-9ed6-4bfd-9160-ab44b4f868b2 public b3ee49a7-80b9-4898-8f53-a58cb1e3908c 38f2e146-43e0-4f93-b3db-87fc25a4be19 shared 29ea72f8-7ac3-4d2a-a2f8-e4b2f2946718														
3812e146-43e0-4f93-b3db-87fc25a4be19 shared 29ea72f8-7ac3-4d2a-a2f8-e4b2f2946718 +														
stack@microwave:~\$ openstack server createvolume ubuntu-18.04_vol \ security-group bastion ssh \														
security-group basion_ssh \ flavor m1.small \														
flavor m1.small \ key-name "LFO"														
rey-name "Lrow" (network private) ubuntn-18 04 inst														
ubuntu-18.04_inst														
Field			I.	Value						1				
+			+							-+				
OS-DCF:0	diskConf: NZ.avail:	ig bilitu		MANUAL										
05-EXI-2	STS:powei	r state	20110	NOSTATE										
OS-EXT-S	STS:task	state	i	schedul	ing					i				
OS-EXT-S	STS:vm_st	tate	1	buildin	g					1				
OS-SRV-U	JSG:laund	ched_at	- 1	None										
US-SRV-U accessTi	JSG:LeIII. Pv4	inaled_a		None										
accessIi	Pv6		i i							1				
addresse	95		1							1				
adminPas	5 <i>5</i>			YdPdfs8	6022C									
coniig_0 created	arive			2019-10	-1.372	1:23:44	7							
flavor			i	m1.smal	1 (2)		-			i				
hostId			1							1				
id				01e719a	0-abd	4-44dd	99e1-2	7076de3	dce5					
lmage kev name	-			L.FO										
name	-		i	ubuntu-	18.04	inst				i				
progress	3		1	0	-	-				1				
project_	id			5cd4212	61165	4857a9e.	941a7b	5d35e13						
propert. security	z arouns			name='9	11639	5f-92dc	-47c8-	bd38-94	24f6b7f88f '					
status			i	BUILD						i				
updated			1	2019-10	-13T2	1:23:44	Ζ			1				
user_id	attacher	-		57d02b0	938d1	49alaac.	96ce67	57a3a14						
+		بر 	 +							-+				
stack@mici	rowave:~{	\$ opensta	ack se	rver sho	w ubu	ntu-18.	04_ins	t						
+			+	Valuo							+			
+			+								+			
OS-DCF:	diskConfi	ig	1	MANUAL							I			
OS-EXT-A	AZ:availa	ability_:	zone	nova							1			
OS-EXT-S	STS:powei	r_state		Running										
05-EAT-2 05-EXT-2	STS:vm si	_state	1	active							1			
OS-SRV-USG:launched at 2019-10-13T21:23:57.000000														
OS-SRV-USG:terminated_at None														
access1Pv4														
accessIi	2V6		1	nrivota	=fdaa	·50=1.h	512.0.	f816.3~	ff.ferR.Jodi	5. 10 0 0 12	1			
config (drive			PIIVALE	LUAC	.JuaiiD	JIU:0:	1010.Je	11.1000.2003	, 10.0.0.43				
created			Í	2019-10	-13T2	1:23:43	Ζ				I			
flavor			1	m1.smal	1 (2)						1			
hostId			1	b3ce8aa	16af2.	2651ac4	ab1291	280fd72	7ff9c016c906	58dcfcc861a7	5			
1 1d			1	<i>uie/19a</i>	u-abd	4-44dd	<i>yye1-2</i>	/U/6de3	uces		l.			
1mage														
image														

0	1	
5cd4212611654857a9e941a7b5d35e13	1	
	I.	
name='bastion_ssh'	I	
ACTIVE	I. I.	
2019-10-13T21:23:58Z	I	
57d02b0938d149a1aac96ce6757a3a14	I. I.	
id='edc26441-2f67-410d-8b24-a1e85844d8dc'	I. I. I. I. I. I. I. I. I. I. I. I. I. I	
+	+	
	<pre>0 5cd4212611654857a9e941a7b5d35e13 1 name='bastion_ssh' ACTIVE 2019-10-13T21:23:58Z 57d02b0938d149a1aac96ce6757a3a14 id='edc26441-2f67-410d-8b24-a1e85844d8dc'</pre>	0 5cd4212611654857a9e941a7b5d35e13 name='bastion_ssh' ACTIVE 2019-10-13T21:23:58Z 57d02b0938d149a1aac96ce6757a3a14 id='edc26441-2f67-410d-8b24-a1e85844d8dc'

Le lancement d'une instance tout comme la création d'un volume n'est pas immédiat, la commande de création est lancée puis exécutée de façon asynchrone par les différents composants de Openstack. La commande openstack server show permet de savoir quand l'instance est fonctionnelle (« running »)

Ajout d'une adresse IP flottante à notre instance

Par défaut dans notre projet nous avons au minimum deux réseaux, un réseau public (externe à notre projet, il appartient au système OpenStack) et un réseau privé (interne à notre projet) :

st	ack@microwave:~\$ openstack network list												
	ID	Name	, Subnets										
 +-	0e6e2f0c-c4cc-4eca-8428-8117864b6e4a 12ceb0b7-9ed6-4bfd-9160-ab44b4f868b2 38f2e146-43e0-4f93-b3db-87fc25a4be19 	private <mark>public</mark> shared +	10b8dba4-4613-4462-b8f4-500bd6c7fc84 b3ee49a7-80b9-4898-8f53-a58cb1e3908c 29ea72f8-7ac3-4d2a-a2f8-e4b2f2946718										

Pour le moment notre instance a une adresse IP dans le réseau « private » :

stack@microwave:~\$ openstack :	server show ubuntu-18.04_inst	
+	-++	
Field	Value	
+	-++	
OS-DCF:diskConfig	MANUAL	
OS-EXT-AZ:availability_zone	nova	
OS-EXT-STS:power_state	Running	
OS-EXT-STS:task_state	None	
	/	
id	01e719a0-abd4-44dd-99e1-27076de3dce5	
name	ubuntu-18.04_inst	
	/	
addresses	<mark>private</mark> =fdac:50a1:b51d:0:f816:3eff:fec8:2ed5, <mark>10.0.0.43</mark>	
	/	
volumes_attached	id='edc26441-2f67-410d-8b24-a1e85844d8dc'	
+	++	

Nous créons une adresse IP flottante pour notre serveur, nous la créons dans le réseau « public » :

stack@microwave:~\$ ope	nstack floating i	p create public		
Field	/ Value		<i>+</i> 	
created_at	2019-10-14T18:4.	2:24Z	+	
description	l		I	
dns_domain	None		I	
dns_name	None		I	
fixed_ip_address	None		I	
floating_ip_address	<mark>172.24.4.105</mark>		I	
floating_network_id	12ceb0b7-9ed6-41	bfd-9160-ab44b4f868b2	I.	
id	168bd864-8489-40	d50-96dd-24bd93f316e4	I	
name	172.24.4.105		I	
port_details	None		I.	
port_id	None		I	
project_id	5cd421261165485	7a9e941a7b5d35e13	I	
qos_policy_id	None		I	
revision_number	0			
router_id	None		I	
status	DOWN			
subnet_id	None			
tags	[] 0010 10 147710 4	0.045		
updated_at	2019-10-14118:42	2:242	l.	
the stack of the s	+ flooting is	 	+	
+		/	+	+
ID		Floating IP Address	Fixed IP Address	Port
+ 168bd864-8489-4d50-9	 6dd-24bd93f316e4	+ 172.24.4.105	+ None	+ None

Nous attachons maintenant l'adresse IP flottante à notre instance :

<pre>stack@microwave:~\$ openstack server add stack@microwave:~\$ openstack floating ip '</pre>	floating ip ubuntu-1 p list	.8.04_inst	: 172.24.4.10)5						
ID	Floating IP Address	s Fixed	IP Address	Port						
168bd864-8489-4d50-96dd-24bd93f316e4	172.24.4.105	3db24397-cd2e-4cba-853a-e48ebcbd71e2								
stack@microwave:~\$ openstack server list	E		,		,	+ ,				
ID	Name Status Networks				Image	Flavor				
+ 01e719a0-abd4-44dd-99e1-27076de3dce5 +	ubuntu-18.04_inst	ACTIVE	private=10.	.0.0.43, <mark>172.24.4.105</mark>	/ /	m1.small 				

Connexion à notre instance

Nous pouvons désormais nous connecter à notre instance :

ubuntu@ubuntu-18:~\$ **uname -a** Linux ubuntu-18 4.15.0-65-generic #74-Ubuntu SMP Tue Sep 17 17:06:04 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux

Les volumes de données

Nous l'avons vu, le contenu du volume système, c'est à dire le système d'exploitation et les logiciels associés, sont « jetables », seules les données de l'application (bases de données, fichiers...) sont à conserver de façon perenne.

Le volume système est celui sur lequel a été démarré l'instance, nous l'avons créé à partir d'une image système mise à notre disposition, les logiciels systèmes peuvent être installés et paramétrés de façon automatique, nous verrons plus loin un exemple.

Nous créons un volume pour stocker des données, ce volume est créé de façon similaire au volume système (sauf qu'il est créé vide, et non à partir d'une image). Ce volume est ensuite attaché à une instance.

Création d'un volume de données via le Dashboard

Les volumes tout comme les instances appartiennent à un projet, il convient donc de se connecter avec le bon utilisateur et de se placer dans le bon projet.

Dans le menu « Projet \rightarrow Volumes \rightarrow Volumes » nous voyons le volume système de notre instance, son statut est « En cours d'utilisation », il est attaché à notre instance sur le 'device' /dev/vda

Volumes - OpenS	Stack Dashbo	. × +										 :	۵	×
(←) → ୯ @		① 192.168.1.211/d.	ishboard/projec	ct/volumes/			… ⊠ ☆	Q. Rechercher			$\underline{\Psi}$		۲	Ξ
openstack.	💷 dem	io •											å den	10 *
Projet	vás API	Projet / Volumes / Vol	umes											
Compute	>	Volumes												
Volumes	♥ Volumes					Fi	Rear Q	+ Créer le volum	e 🖴 Accepter le	transfert	B Supp	einner lien	volum	
inst (antanés Groupes	Affichage d⊭ 1 élément	Description	Taille Statut	Groupe	Туре	Attaché à	Zone de dispo	nibilité Amorçab	e Chiffré	Action	IS		
Groupe d'inst	antanés	ubuntu-16.04_val	5	10Gio En cours d'utilisation	-	lvmdriver-'	1 /dev/vda sur ubuntu-18.64_	inst nova	Oui	Non	Modi	tier le volu	ime	
Réseau Orchestration Stockage d'objet	> >	Affichage de 1 élément												
Identitë	>													

Nous cliquons sur « Créer le volume » pour créer un nouveau volume :

Nous mettons « donnees » comme nom de volume, dans le champ « Source du volume » nous mettons « Aucune source, volume vide », comme taille nous mettons 5 Go par exemple, puis nous cliquons sur « Créer le volume » :

Uolumes - OpenStack Da	shbox X +			÷: 0)
€) → ୯ ଘ	④ 192.168.1.21	1/dashboard/project/volumes/	😇 🟠 🔍 Rechercher	± in ⊡ ¢
🗖 openstack. 📼	dema +			🛔 demo.
Projet	Projet / Volumes /	Créer le volume	×	-
Actes AF Compute 3 Volumes Volume Instantani Groupe d'instantani Résmai 3 Orchestration 3 Stockage d'ubget 3 Stockage d'ubget 3	Affichage de 1. éélene Affichage de 1. éélene Nom Affichage de 1. éélene	Nom du volume donnees Description Source du volume Aucune source, volume vide Type hmdriver.1 Taille (Gio)* 5 Zone de disponibilité nova Groupe ④ Aucun groupe	Description : Les Volumes sont des princhérique de type bloc qui pouvent être attachés à des instances. Description de Type de Volume: Ivmdriver-1 Aucure description disponible. Totat des Totat des	Accepter le transfert Amorçable Chiffré Accions Ou Non Montfer le valume
			Annuler Créer le volume	

Le statut de notre nouveau volume est « disponible » :

U Volumes	s - Open	Stack Dashb	es) 🗙	+											 :		×
€ → C 6	ŵ		٢	192.168.1.211/d	ashboard/proj	ect/volu	mes/			··· 🗟 🖒 🔍	Rechercher			± 1		۲	=
openst	tack	den	no •													å der	10 *
Projet	A	v coks API	Proj	et / Volumes / Vo	lumes												1
Compu	ute	>	Vo	lumes													
Volume	ies	✓ Volumes								atrei Q	+ Créer le volume	# Accepter le tra	nsfert	B Supp	dittine the	võluin	ea (
	îns	tantanés Groupes	Affich	age de 2 éléments Nom	Description	Taille	Statut	Groupe	Туре	Attaché à	Zone de disponibilité	Amorçable	Chiffré	Action	5		
Group	pe d'ins	tantanés		donnees	51	5Gia	disponible(s)		lvmdriver	-1	nova	non	Non	Modif	ter le vol	ume	-
Résea	au	>	o	ubuntu-18.04_vol	5)	10Gia	En cours d'utilisation	8	lvmdriver	r-1 /dev/vda sur ubuntu-18.04_inst	nova	Oui	Non	Modif	fer te vol	ume	•
Orchestra Stockage o Identité	ration d'objet	> > >	Affich	lage de 2 éléments													

Dans la liste déroulante à droite du volume « donnees »nous sélectionnons « Gérer les attachements » :

🔲 Volumes - Op	penStack Dash	ben X	+													×
(←) → C* @		0	192.168.1.211/	lashboard/proj	ect/volu	mes/			⊡ ☆	Q, Rechercher			$\overline{\mathbf{A}}$		۲	=
openstac	k. 📼 de	mo •												14	demo	• ^
Projet	Accès API	Proj	jet / Volumes / Vi	alumes												
Compute	>	Vo	lumes													
Volumes	↓ Volumes							Filte	a a	+ Créer le volume	# Accepter le 1	ransfert	C Suppo	mer las v	olumes	
	Instantanés Groupes	Affich	nage de 2 éléments Nom	Description	Taille	Statut	Groupe	Туре	Attaché à	Zone de disponibilité	Amorçabl	e Chiffré	Actions			
Groupe d	instantanés	•	donnees	a.	5Gio	disponible(s)		lvmdriver-1		nova	non	Non	Modifie	r le volum	10 -	Ĩ
Réseau Orchestration	>	o	ubuntu-18.04_v ol	54	10Gio	En cours d'utilisation	2	lvmdriver-1	/dev/vda sur ubuntu-16,04_in st	nova	Oui	Étend Gérer	re un volur les attachi	ne. ements		
Stockage ďobj	et >	Affich	tage de 2 éléments	l.								Créer Chang Charg	un instant jer le type er dans l'i	ané de volume nage		
192.168.1.211/dashboard/	project/volum	tes/e4539f3	6-09b7-4198-afc4-67	66c5116e22/attac	th/							Créer	un transfe	rt.		

Dans la liste « Attacher à l'instance » nous sélectionnons notre instance « Ubuntu-18.04_inst », puis nous cliquons sur « Attacher le volume » :

Volumes - OpenStack Dashb	ies × +								- (i ×
(←) → ♂ @	192.168.1.211	1/dashboard/project/volum	ves/	🖾 🔓 🔍 Recher	cher			± ∥		\$ ≡
🗖 openstack. 📼 der	na +									i demo •
Projet -	Projet / Volumes /	Gérer les attac	chements de volume		×					
Compute >	Volumes	Instance	Périphérique	Actions						
Volumes 😽			Aucun élément à afficher.							
Vilumen					-	Accepter le tra	nsfert	B Soup		dariin.
instantanés	Affichage de 2 éléme									
Groupez	D Nom	Attacher à l'ins	tance		Abiline	Amorçable	Chiffré	Actions	6)}	
Groope d'Instantanés	D dommes	Attacher à l'instance • •	9 719a0-abd4-44dd-99e1-27076de3dce5)		•	non	Non	Modifi	ier 1e volun	16 .
Réseau >	D abarna-18.04					Our	Non	Modifi	ler té volun	- 01
Orchestration > Stickage d'objet >	Affichage de 2 éléme			Annuler Attacher le v	olume					

L'attachement du volume prend quelques secondes, une fois l'attachement terminé nous voyons qu'il est attaché en /dev/vdb :

U Volumes	- OpenS	itack Dashb	en X	÷													a	×
(←) → @ @	5		٢	192.168.1.211/d	ashboard/proji	ect/volü	mes/			··· 🖂 🕁	9,1	Rechercher			$\underline{\star}$	III\ (ت	Ξ
C opensta	ack.	💷 der	no •														👗 de	mo 🕶
Projet	he	v něs ADI	Proj	et / Volumes / Vo	lumes													
Comput	te	>	Vo	lumes														
Volume	s	✓ /olumes							. Fi	liter Q	•	• Créer le volume 🛛 🛱	Accepter le tra	nsfert	B Sup	primer	ties volum	189
	inst.	antanés Groupes	Affich	age de 2 éléments Nom	Description	Taille	Statut	Groupe	Туре	Attaché à		Zone de disponibilité	Amorçable	Chiffré	Actic	ns		
Group	e d'inst	antanés		donnees	51	5Gio	En cours dutilisation	•	lvmdriver-	1 /dew/vdb sur ubuntu-18.04_	inst	nova	non	Non	Mot	tifier le	volume	•
Réseau	u	>	۰	ubuntu-18.04_vnl	0	10Gio	En cours d'utilisation	-	lvmdriver-	1 /dev/vda sur ubuntu-18.04_i	inst	nova	Oui	Non	Mot	tifier le	volume	•
Orchestra Stockage d Identité	tion Fobjet	> > >	Affich	age de 2 éléments														

Préparation (partitionnement et formatage) du volume

Nous nous connectons à notre instance, nous voyons le device /dev/vbd:

stack@microwave:~\$ ssh ubuntu@172.24.4.105
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-65-generic x86_64)
ubuntu@ubuntu-18:~\$ ls -1 /dev/vd*
brw-rw---- 1 root disk 252, 0 Oct 13 21:24 /dev/vda
brw-rw---- 1 root disk 252, 14 Oct 13 21:24 /dev/vda1
brw-rw---- 1 root disk 252, 15 Oct 13 21:24 /dev/vda14
brw-rw---- 1 root disk 252, 16 Oct 15 19:08 /dev/vda

Afin de faciliter un éventuel agrandissement nous allons mettre notre disque en LVM, nous commençons par partitionner le disque avec la commande fdisk, nous voyons ensuite le device correspondant à notre partition (/dev/vdb1):

```
ubuntu@ubuntu-18:~$ sudo fdisk /dev/vdb
Welcome to fdisk (util-linux 2.31.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.
Device does not contain a recognized partition table. \Rightarrow LE V
Created a new DOS disklabel with disk identifier 0xabdc88e0.
                                                                                   ⇒ LE VOLUME N'EST PAS ENCORE PARTITIONNE
Command (m for help): n \Rightarrow n \text{ pour new}
Partition type
   p primary (0 primary, 0 extended, 4 free)
e extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1 → Partition 1
First sector (2048-10485759, default 2048): → Appuy
                                                                                    sur Entré
Last sector, +sectors or +size{K,M,G,T,P} (2048-10485759, default 10485759): <mark>→ Appuyer sur Entrée</mark>
Created a new partition 1 of type 'Linux' and of size 5 GiB.
Command (m for help): t \Rightarrow t pour type
Selected partition 1
Partition type (type L to list all types): 8e \Rightarrow 8e pour Linux LVM
Changed type of partition 'Linux' to 'Linux LVM'.
Command (m for help): p \Rightarrow p pour print (pour vérifier)
Disk /dev/vdb: 5 GiB, 5368709120 bytes, 10485760 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xabdc88e0
Device Boot Start End Sectors Size Id Type
<mark>/dev/vdb1 2048 10485759 10483712 5G 8e Linux LVM</mark>
Command (m for help): w \Rightarrow w pour writ
The partition table has been altered.
                                                          e, provoque la sortie du programme
Calling ioctl() to re-read partition table.
Syncing disks.
ubuntu@ubuntu-18:~$ 1s -1 /dev/vd*
brw-rw---- 1 root disk 252, 0 Oct 13 21:24 /dev/vda
brw-rw---- 1 root disk 252, 1 Oct 13 21:24 /dev/vda1
brw-rw---- 1 root disk 252, 14 Oct 13 21:24 /dev/vda14
brw-rw---- 1 root disk 252, 15 Oct 13 21:24 /dev/vda15
brw-rw---- 1 root disk 252, 16 Oct 15 19:14 /dev/vdb
brw-rw---- 1 root disk 252, 17 Oct 15 19:14 /dev/vdb
```

Nous créons un Physical Volume, un Logical Volume et nous le formatons en ext4 :

```
Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

Nous créons le point montage, modifions le fichier /etc/fstab, montons le système de fichiers et nous créons un fichier de test :

ubuntu@ubuntu-18:~\$ sudo mkdir /DONNEES			
ubuntu@ubuntu-18:~\$ sudo tee /etc/fstab << EOF /dev/mapper/PV_DONNEES-1v_donnees /DONNEES EOF	ext4	defaults	1 2
/dev/mapper/PV_DONNEES-lv_donnees /DONNEES	ext4	defaults	1 2
ubuntu@ubuntu-18:~\$ sudo mount /DONNEES			
ubuntu@ubuntu-18:~\$ df -h /DONNEES			
Filesystem Size Used Avail Use	% Mounted	on	
/dev/mapper/PV_DONNEES-lv_donnees 4.9G 20M 4.6G 1	<pre>% / DONNEES</pre>	5	
ubuntu@ubuntu-18:~\$ sudo tee /DONNEES/hello.txt << EOF Hello EOF			

Le partitionnement du volume et le formatage du volume logique provoquent la perte des données existantes. Les commandes correspondantes demandent confirmation s'il y a un risque que des données soient présentes.

Détachement du volume via le Dashboard

Depuis le Dashboard il est possible de détacher le volume de l'instance à laquelle il est attaché en utilisant le menu déroulant à droite du volume dans la liste des volumes.

Détacher un volume revient à débrancher un disque dur d'un serveur physique, s'il était encore monté par le système d'exploitation cela risquerai de provoquer une perte ou une corruption de données.

Cinder et Nova n'ont pas possibilité de savoir si le volume est monté par le système d'exploitation de l'instance, aussi il convient de vérifier avant de procéder au démontage.

Au cas où vous souhaiteriez démonter le volume, il faut démonter la partition et il est préférable de procéder à la désactivation du volume group :

```
ubuntu@ubuntu-18:~$ sudo umount /DONNEES
ubuntu@ubuntu-18:~$ sudo vgchange /dev/mapper/PV_DONNEES -an
0 logical volume(s) in volume group "PV_DONNEES" now active
```

Suppression d'un volume via le Dashboard

Depuis le Dashboard il est possible de supprimer un volume en utilisant le menu déroulant à droite du volume dans la liste des volumes, pour ceci il ne doit plus être attaché à aucun serveur.



La suppression d'un volume est irréversible, les données qu'il contenait sont définitivement perdues

Nous ne détruisons pas le volume de données que nous venons de créer car nous allons le

réutiliser plus loin.

Création et attachement d'un volume en ligne de commande

Il est également possible de créer un volume et de l'attacher à un serveur en ligne de commande.

On voit là aussi que la création d'un volume (même vide) n'est pas instantanée.



Le nom des volumes n'est pas unique, plusieurs volumes peuvent porter le même nom, si on passe la commande en utilisant le nom et qu'il n'est pas unique une erreur va se produire, dans ce cas il est obligatoire d'utiliser l'ID unique du volume.

<pre>stack@microwave:~\$ sou stack@microwave:~\$ ope</pre>	rce devstack/open nstack volume crea	rc demo demo atesize 1 donnee:	s_bis				
+ Field	+ Value		+				
<pre>+</pre>	<pre>////////////////////////////////////</pre>	1:36.000000 fea-9d63-16e91c5707, 1aac96ce6757a3a14	+ 				
<pre>stack@microwave:~\$ ope: +</pre>	nstack volume list	t +	+	-+	+		+
1D + 140b7ad2-0485-4fea-9 e4539f36-09b7-4198-a edc26441-2f67-410d-8.	d63-16e91c5707a8 fc4-6766c5116e22 b24-a1e85844d8dc	Name + <mark>donnees_bis</mark> donnees ubuntu-18.04_vol	Status + <mark>available</mark> in-use in-use	5120 1 5 10	Attached to ubuntu-18.04 Attached to ubuntu-18.04 Attached to ubuntu-18.04	4_inst or 4_inst or	n /dev/vdb n /dev/vda
+stack@microwave:~\$ op	enstack server lis	* st	+	+	+	,	
ID		 Name	Status	Network	s	Image	Flavor
01e719a0-abd4-44dd-9	9e1-27076de3dce5	<mark>ubuntu-18.04_inst</mark>	ACTIVE	private	=10.0.0.43, <mark>172.24.4.105</mark>	, ,	m1.small
<pre>t</pre>							
+		+ Name	++ Status S	Size A	ttached to		+
<pre>++ 140b7ad2-0485-4fea-9d63-16e91c5707a8 donnees_bis in-use 1 Attached to ubuntu-18.04_inst on /dev/vdc e4539f36-09b7-4198-afc4-6766c5116e22 donnees in-use 5 Attached to ubuntu-18.04_inst on /dev/vdb edc26441-2f67-410d-8b24-a1e85844d8dc ubuntu-18.04_vol in-use 10 Attached to ubuntu-18.04_inst on /dev/vda ++</pre>							

L'option '-f json' de la commande openstack permet d'afficher le résultat de la commande au format JSON au lieu d'afficher le résultat sous forme de tableau. Le format est humainement lisible et le résultat plus complet que sous forme de tableau :

```
stack@microwave:~$ openstack volume show donnees_bis -f json
{
    "status": "in-use",
    "properties": {},
    "user_id": "57d02b0938d149a1aac96ce6757a3a14",
    "description": null,
    "availability_zone": "nova",
    "bootable": "false",
    "encrypted": false,
    "created at": "2019-10-15T19:21:36.000000",
```

"multiattach": false,
"updated at": "2019-10-15T19:25:18.000000",
"source volid": null,
"name": "donnees bis",
"snapshot id": null,
"consistencygroup id": null,
"replication status": null,
"os-vol-tenant-attr:tenant id": "5cd4212611654857a9e941a7b5d35e13",
"size": 1,
"type": "lvmdriver-1",
"id": "140b7ad2-0485-4fea-9d63-16e91c5707a8",
"attachments": [
f
"server_id": "01e719a0-abd4-44dd-99e1-27076de3dce5",
"attachment id": "30162df5-3788-4aff-a3b7-a743beec55a9",
"attached_at": "2019-10-15T19:25:17.000000",
"host name": "microwave",
"volume id": "140b7ad2-0485-4fea-9d63-16e91c5707a8",
"device": "/dev/vdc",
"id": "140b7ad2-0485-4fea-9d63-16e91c5707a8"
}
1

Détachement et suppression d'un volume en ligne de commande

Nous pouvons détacher le deuxième volume du serveur puis détruire le volume :

stack@microwave:~\$ openstack server remove volume ubuntu-18.04_inst donnees_bis stack@microwave:~\$ openstack volume delete donnees_bis stack@microwave:~\$ openstack volume list 							
ID	, Name	Status	Size	Attached to			
+ e4539f36-09b7-4198-afc4-6766c5116e22 edc26441-2f67-410d-8b24-ale85844d8dc +	donnees ubuntu-18.04_vol +	in-use in-use in-use	5 10	Attached to ubuntu-18.04_inst on /dev/vdb Attached to ubuntu-18.04_inst on /dev/vda			



Il vous est laissé à titre d'exercice de vérifier si le filesystem du volume 'donnees' a été démonté en vous connectant à l'instance puis de détacher le volume de l'instance (s'il n'a pas été démonté la commande de démontage et de désactivation du volumegroup est indiquée un peu plus haut).

<u>Ne détruisez pas le volume 'donnees' il va resservir</u>.

L'orchestration

L'orchestration est une des fonctionnalités puissantes du Cloud, elle permet de déployer des infrastructures de manière automatique et reproductible, elle se base sur un langage de description des infrastructures et un langage de template (modèles).

L'orchestration permet aussi d'assurer l'élasticité, c'est à dire de faire varier les infrastructures en fonction de la charge.

Différents systèmes d'orchestration existent, ceci grâce à l'ouverture des API OpenStack.

L'orchestration avec Heat

Heat est un des projets de OpenStack.

Heat permet de décrire les infrastructures sous forme de fichier texte, l'infrastructure peut comprendre la plupart des types de ressources (instances, volumes, routeurs, réseaux...)

La description d'une infrastructure pour Heat s'appelle un template (modèle). Déployer un template dans un tenant (éventuellement avec des paramètres) va créer une « pile » de ressources.

Heat est composé :

- d'un service en charge de synchroniser les opérations nécessaires à la création de l'infrastructure décrite
- d'une API
- d'une interface en ligne de commande qui s'intègre à la commande CLI openstack.

Un template de base est composé de 4 parties :

- Une description du template, à titre purement documentaire
- Un ensemble de paramètres qui permettent d'adapter le template lors de chaque déploiement et de créer une pile adaptée au client et à l'environnement, ces paramètres ont une valeur par défaut qui peut être écrasée lors de la création de la pile, soit depuis des champs de saisie dans le Dashboard, soit par des paramètres passés en ligne de commande lors de la création de la pile à l'aide du CLI openstack
- Un ensemble de ressources qui vont être créées et liées les unes aux autres (des volumes aux serveurs, des interfaces à des réseaux...)
- Des sorties qui vont nous permettre de récupérer des informations sur la pile créée (adresses IP, URL...), ces données sont décrites et mises en forme dans la pile

Les ressources

On l'a dit, une pile est composée de ressources, ce sont les objets que l'on peut créer dans OpenStack, la pile décrit également les liaisons entre ces ressources. Ces ressources sont définies par un type, par exemple une instance est de type OS::Nova::Server(OS car il s'agit d'un objet de OpenStack, Nova car Nova est le composant de OpenStack qui gère les serveurs), OS::Nova::KeyPair est une paire de clé. Il existe également des objets de type AWS:: pour assurer la compatibilité avec AWS, le cloud de Amazon.

Dans le Dashboard on retrouve les différents types d'objets disponibles ainsi que leur documentation succincte dans le menu « Projet \rightarrow Orchestration \rightarrow Type de ressources ».

Une première pile

Notre premier template va être simple, il va créer une instance à partir de l'image Ubuntu que nous avons chargée dans OpenStack, et un script bash va être utilisé pour installer deux packages logiciels et exécuter deux commandes.

Recopiez le template ci-dessous dans un fichier que vous appellerez premiere_stack.yaml, il est également disponible à cette adresse : <u>https://u03.fr/90openstack/premiere_stack.yaml</u>



On y retrouve les 4 parties (« description », « parameters », « ressources » et « outputs »).

Dans la section « ressources » nous n'avons qu'une ressource, il s'agit d'une instance de serveur, son identifiant est « instance_heat », grâce à cet identifiant, utilisable uniquement dans la définition de pile, la ressource pourra être référencée ailleurs dans la pile (dans la définition d'autres ressources ou dans la section « outputs »).

Le type de la ressource est « OS : : Nova : : Server », ses propriétés sont :

- image : le nom de l'image, ce nom provient du paramètre « image_instance »
- flavor : il s'agit du gabarit, ce nom provient du paramètre « gabarit instance »

90 Minutes avec OpenStack – 2019/11/11

- key_name : le nom de la paire de clé utilisée pour se connecter en SSH à l'instance, une clé avec ce nom doit être enregistrée dans le projet
- user_data : ce champ s'étend sur plusieurs lignes (grâce au « | », il s'agit du script qui va s'exécuter au moment du lancement de l'instance, ici ce champ est constant mais nous verrons qu'il est possible de modifier ce champ en fonction des valeurs des paramètres passés lors de la création de la pile.



La ligne **heat_template_version:** 2013–05–23 ne correspond pas à la version du template, mais à la version du langage de template utilisée, ce qui correspond à la version d'OpenStack pour laquelle le template a été initialement écrit. N'essayez donc pas d'y mettre la date du jour par exemple.

Utilisation de la pile

L'utilisation des piles se fait dans le Dashboard dans le menu « Projet \rightarrow Orchestration \rightarrow Piles », pour lancer une pile il faut cliquer sur le bouton « Lancer la pile » :



La source du modèle (le template) est premiere_stack.yaml, (disponible un peu plus haut dans le document ou à l'adresse <u>https://u03.fr/90openstack/premiere_stack.yaml</u>). Vous pouvez au choix utiliser une copie locale sur votre poste de travail (source du modèle 'Fichier' et cliquez sur « Parcourir »), soit utiliser directement le fichier via son URL. Le « champ environnement » permet de sélectionner un fichier de paramètres qui vont venir écraser les paramètres par défaut spécifiés dans le template, nous ne l'utilisons pas pour le moment. Cliquez sur « Suivant » :

ananchael				_			
opensiace	s millionno +						3am
jat .	Sélectionner un modèle			×			
Connector	Source du modèle *				I		
Gampute	URL *	Descriptio	on :				
Volumes	URL du modèle 🛛	Un modèle perm d'infrastructure, o	ermet d'automatiser le déploiement re, de services et d'applications.				
Resiau.	Assau https://u03.fr/90openstack/premiere_stack.yam/ Utiliser une des options de source de modèle dis				enou	nanser h	
Orthestration	Environment source	pile.	odelle a utilitaler fors de la creati	on de cette	tion	97	
	Fichier •						
Types de	Fichier d'environment Ø						
Versions	Parcourir Aucun fichier sélectionné						
Contraction of			Annuler	Suivant			
Stockage d'obje							

Nous allons nommer notre pile « ma_pile », nous tapons notre mot de passe dans le champ prévu à cet effet, et nous tapons le nom de notre clé SSH dans le champ « nom_de_cle » puis nous cliquons sur « Démarrer » :

DA CA	(i) 🖉 1921681211/dashbarred/periort/starks		Q Recharches	Ł	lin.	m		=
openstac	k. m dama +		~ netherine	Ŧ		0	å der	10.7
Projet	Lancer la pile			×	I			
Connector	Nom de la pile * Ø	_						
Compute	ma_pile	Descriptio	n :					
Volumes	Délai d'attente de création (minutes) * 9	Créer une nouvelle	e pile avec les valeurs fournies.					
Réseau	60				Preve	suals	er la p	
Orchestration	Annulation en cas d'échec O				ictio	ns		
	Mot de passe pour l'utilisateur "demo" * 0							
Types d	сорянскет Ø							
Version	gabarit_instance							
Générateu	m1.small							
Stockage d'ob	image_instance O							
dentită-	ubuntu-18.04							
	nom_de_cle * Ø							
	LFO							
	reseau_instance O							
	private							
			Annular					

La création de la pile va demander environ une minute :

Piles - OpenStack	Dashboard	i x +						050	: LL :	^
←) → ♂ @		④ 192.168.1,211/dashboard	/project/stacks/		⊡ ☆	Q. Rechercher	*	III\ C	D®	Ξ
openstack.	m den	na •							🛔 de	imo •
Projet	× ás API	Projet / Orchestration / Pile	85							
Compute	>	Piles								
Réseau	>	Nom de la stack = •		Filtrer	+ Lancer la pile	Prévisualiser la pile	Supprimet les plies	Plus	d'action	15 *
Orchestration	∨ Piles	Affichage de 1 élément	Créé		Mis à jour	Statut		Actions		
Types de res	source	O ma_ple	0 minute		lamais	Création terminée		Véniñer	la pile	٠
Versions du r Générateur de r Stockage d'objet Identité	modéle Nodéle	Affichage de 1 élément								

En cliquant sur « ma_pile » nous affichons par défaut la « topologie » de notre pile, elle est simple, il n'y a que notre serveur.

Nous voyons que l'écran « ma_pile » est composé de 5 onglets « Topologie » , « Vue d'ensemble », « Ressources », « Événements » et « Modèle ».

🖸 ma_pile - OpenSta	ck Dashboar 🗙 🕂	- ¤ ×
€)→ ଫ @	① 192.168.1.211/dashboard/project/stacks/stack/e1466d62-e1dc-411	¥ ⊪\ © @ ≡
openstack.	🕅 dema 🕶	🛔 demo 🖛
Projet	Projet / Orchestration / Piles / ma_pile	
Compute	> ma_pile	Vérifier la pile 🔹
Volumes Réseau Orchestration	Topologie Vue d'ensemble Ressources Événements Modèle Topologie ma_pile	
)	Pites Create Complete	
Types de rest Versions du r	nerále	
Générateur de n	noděle	
Stockage d'objet	,	
Identité.	,	

L'onglet « Vue d'ensemble » nous donne les informations essentielles sur la pile, on y retrouve la description présente dans la définition de la pile, l'état de la pile, les paramètres qui ont été passés et les sorties (Outputs) qui ont été définies dans la pile dans la section « outputs », nous retrouvons ici nos salutations :

🖸 ma_pile - OpenSta	ack Dashboar X 1+	- ¤ ×
<> → ♂ @	① 192.168.1.211/dashboard/project/stadks/stadk/e1466d62-e1dc-411	¥ ⊯\ © ≇ ≡
openstack.	🕅 demo 🕶	🛔 demo 👻
Projet Acco Compute Volumes	Projet / Orchestration / Pées / ma_pile Ma_pile Ma_pile Magnitude Magnit	Vérdier la pile 🔹
Réseau Orchestration	Vier densemble Norm ma_pile Description Notre premiere pile Heat	
Types de res Versions du r Générateur de r	Isource Créé 1 minute modèle Demlére mise à jour Jamais Statut Create_Complete : Stack CREATE completed successfully modèle Softies	
Stockage d'objet Identité	solutations La salutation du jour mello Moridiii	
	Paramètres de la pile OS::project_id 5cd4212611654857a9e941a7b5d35e13 reseau_instance printe OS::stack_id e1465d2e1dc-4112-b9te-6588b6c2181 OS::stack_mem ma_pie gabait_instance int anail image_instance ubuntu-18.04 nom_de_cte LFO	

L'onglet « Ressouces » donne les différentes ressources qui composent notre pile, ici nous n'avons qu'un serveur, dont le nom est « instance heat » dans la définition de notre pile :

🔲 ma_pile - OpenSta	ick Dashbo	× +				100	۵	×
€ → ଫ @		① 192.168.1.211/dashboard/project/stacks/stack/e1466d62-e1dc-	411 ···· 🖂 🏠	Q Rechercher	8	E III	0	=
🖸 openstack.	an den						🛔 den	10 * 1
Projet	× ès API	Projet / Orchestration / Piles / ma_pile						
Compute	>	ma_pile				Vérifie	la pile	•
Volumes	>	Topologie Vue d'ensemble Ressources Événements	Modèle					
Orchestration	~	Affichage de 1 nem						
	Piles	Ressource de la pile Ressource	Type de ressource de pil	e Date de mise à jour	Statut	Raisor	du stat	ut
Types de res	source	instance_heat 41/77035-a2ef-42ab-8e43-ac4128781150	OS Nova Server	3 minutes	Création terminée	state o	hanged	
Versions du r	nodèle	A5chage de 1 item						

Si nous cliquons sur l'identifiant en hexadécimal de notre instance nous allons nous retrouver sur l'écran de vue d'ensemble de notre instance, de la même manière que si vous avions cliqué dessus dans l'écran « Projet \rightarrow Compute \rightarrow Instances ».

On aperçoit au passage que le nom de notre instance est composé du nom de la pile, du nom de la ressource dans la pile et d'une chaîne de caractères aléatoires, nous aurions pu donner un nom à cette instance en utilisant l'attribut « name » dans la définition de la ressource :

ma_pile-instance_hee	t-eza6tt: × +				6	- 0	1	×
€) → ♂ @	(1) 192.168.1,2	11/dashboa	d/project/instances/41f77036-a2ef-42ab-8 🛛 😁 🔂 🗌	2 Rechercher	¥ #	n 🗊	8	Ξ
openstack.	🔟 dema 🕶					≜ d	emo 🕶	^
Projet Accês	Projet / Com	oute / Insta	ces / ma_pile-instance_heat-eza6					
Compute	↓ ma_pil	e-ins	tance_heat-eza6ttzivdv2		Créer un i	instantané		
Vue d'ensen	Vue d'ensemb	le Inter	aces Journal Console Log des actions					
lma	ges	Nom ID escription	ma_pile-instance_heat-eza6ttzivdv/2 41f77036-a2ef-42ab-8e43-ac4128761150 -					
Paires de l	ilés ID	du Projet Statut	5cd4212611654857a9e941a7b5d35e13 Active					
Groupes de servi	Zone de di	Verrouillé ponibilité	False nova					
Volumes	>	Créé Age	16 actabre 2019 19:28 4 minutes					
Réseau	>							

Cliquer sur l'onglet « console » nous affiche la console, on peut y voir une vache qui prononce une phrase amusante en anglais (cliquez dans la partie noire de l'écran puis tapez sur la touche espace si l'écran de la console est entré en veille) :

ma_pile-instance_heat	eza6ttzi × +	×
← → ♂ ☆	① 192.168.1.211/dashboard/project/instances/41f77036-a2ef-42ab-8 ♡ ☆	± II\ © © ≡
openstack.	l demo ▼	🛔 demo 👻 🏠
Projet Accès A	Projet / Compute / Instances / ma_pile-instance_heat-eza6	
Compute	ma_pile-instance_heat-eza6ttzivdv2	Créer un instantané 🔹
Vue d'ensemi	le Vue d'ensemble Interfaces Journal Console Log des actions es	
Imag	Console de l'instance	
Paires de cl Groupes de server	 Si la console ne répond plus aux entrées clavier, cliquez sur la barre d'état grise ci-dessous. <u>Cliquer ici pour ne voir que la console</u> Pour quitter le mode plein écran, cliquez sur le bouton retour du navigateur. 	
Volumes	Connected (unencrypted) to QEMU (instance-00000005)	Send CtrlAltDel
Réseau	>	
Orchestration		
Stockage d'objet	UDUNTU 18.04.3 LIS ma-pile-instance-neat-ezabttzivav2 ttyi > ma-nile-instance-heat-eza6ttzivdv2 login:	
Identité	<pre>> / It has long been an axiom of mine that \ the little things are infinitely the most important. Sir Arthur Conan Doyle, "A Case of Identity"</pre>	~

Il vous est laissé à titre d'exercice depuis le Dashboard d'attacher une adresse IP flottante à l'instance, d'autoriser la connexion SSH dans les groupes de sécurité et de vous connecter à l'instance avec l'utilisateur ubuntu, vous serez également accueilli par la vache.

Suppression de la pile

Dans le menu « Projet \rightarrow Orchestration \rightarrow Piles » il est possible de supprimer la pile en utilisant le menu déroulant à droite de la pile.



La suppression d'une pile entraîne la suppression de la totalité des éléments créés par la pile, il s'agit d'une opération irréversible.

Autres types de ressources de Heat

Nous avons créé une première pile qui comportait juste une instance de base et utilisait quelques paramètres.

D'autres types de ressources permettent de construire des infrastructures plus évoluées.

Création d'une groupe de sécurité et d'une adresse IP flottante

Nous allons voir comment créer un groupe de sécurité et l'attacher ainsi qu'une adresse IP flottante à notre instance.

Le groupe de sécurité de sécurité est créé en même temps que les règles (ici il y a une seule règle qui autorise les accès en SSH (tcp port 22), en entrée (ingress), depuis toutes les adresses IP (0.0.0.0/0).

Le groupe de sécurité est ensuite attachée à l'instance avec la propriété security_groups de l'instance gràace à get resource :



La gestion de l'adresse IP flottante est différente, on crée une adresse IP flottante (OS::Neutron::FloatingIP) dans le réseau 'public' et on crée une deuxième ressource qui est une association d'adresse IP flottante (OS::Nova::FloatingIPAssociation)

```
floating_ip_instance_heat:
   type: OS::Neutron::FloatingIP
   properties:
     floating_network: { get_param: reseau_public }
association_reverseproxy:
   type: OS::Nova::FloatingIPAssociation
   properties:
     floating_ip: { get_resource: floating_ip_instance_heat }
     server id: { get resource: instance heat }
```

L'adresse IP flottante du serveur va être attribuée par le système, nous pouvons modifier la section « outputs » pour afficher l'adresse IP qui a été attribuée :

```
outputs:
instance_heat_public_ip:
description: Adresse IP publique de l'instance Heat
value: { get_attr: [ floating_ip_instance_heat, floating_ip_address ] }
```

La pile complète est disponible à cette adresse :

• <u>https://u03.fr/90openstack/groupe_securite_ip.yaml</u>

Une fois la création de la pile terminée nous avons l'adresse IP flottante qui a été attribuée dans l'onglet « Vue d'ensemble de la pile » :

€ → @ @		① 192.168.1.211/dashboard/project/stacks/stack/289f62f0-03c2-43a0 🗵 🟠 🔍 Rechercher	4	III\	0	s	Ξ
openstack.	🕅 den	no •			🛔 de	imo •	,
Projet	× és API	Projet / Orchestration / Piles / ma_pile					
Compute	>	ma_pile		Vérifia	r la pile	•	
Volumes	>	Topologie Vue d'ensemble Ressources Événements Madéle					
Réseau	>	Nom ma_pile					
Orchestration	~	ID 2895270-0322-4339-6108-234893do411b Description Notre premiere pile Heat, avec une adresse IP flottante et une groupe de securite					
	Piles	Statut					
Types de res Versions du r	source nodéle	Créé 22 heures: 29 minutes Demilére mise à jour Jamais Statut Create_Complete : Stack: CREATE completed successfully					
Générateur de r	nodèle	Sorties					
Stockage d'objet	>	Instance_heat_public_lp Adresse IP publique de l'instance Heat					
Identité	>	172.24.4.144					

Nous pouvons désormais nous connecter en SSH à notre serveur en utilisant l'adresse IP flottante qui lui a été allouée :



L'inconvénient de créer l'adresse IP flottante dans la pile est que celle-ci est libérée lors de la destruction de la pile, et lors de la recréation de la pile une nouvelle adresse IP sera allouée.

Cela rend impossible la création d'entrées DNS qui pointent vers l'application.

C'est pour cela qu'on alloue les IP flottantes en dehors de la pile et qu'on les associe au serveur dans la pile.



н

Il vous est laissé à titre d'exercice de supprimer la pile.

Utilisation d'une adresse IP flottante

Ci-dessous la création de l'adresse IP flottante, la définition du paramètre dans la pile, et l'associaton de l'adresse IP flottante au serveur. Globalement on utilise 'get-param' au lieu d'un 'get-ressource'.

stack@microwave:~\$ openstack floating ip create public

Field	Value	
+	+	-+
created_at	2018-03-26T17:38:04Z	
description	I contraction of the second seco	
fixed_ip_address	None	
floating_ip_address	172.24.4.14	
floating_network_id	087bc8a1-65e2-4747-b96e-4dd421c542e5	
id	dd92bcbc-77a5-4624-bf3b-b02489da4ee1	
name	172.24.4.14	
port_id	None	
project_id	3f36cd33d14e468293caa7bf311c00de	
qos_policy_id	None	
revision_number	0	
router_id	None	
status	DOWN	
subnet_id	None None	
updated_at	2018-03-26T17:38:04Z	
+	+	+
parameters:		
id floating ip rever	seproxy:	
type: string		
description: ID de	la floating IP pour le reverse-proxy	
default: dd92bcbc-	77a5-4624-bf3b-b02489da4ee1	
resources:		
association reversep.	roxy:	
type: OS::Nova::Fl	oatingIPAssociation	
properties:		
floating ip: { g	et param: id floating ip reverseproxy ;	
server id: { get	resource: instance reverseproxy }	

Création d'un volume à partir d'une image

Comme nous l'avons vu, l'orchestration permet de déployer automatiquement des infrastructures, en particulier pour des mises à jour du système ou de l'application.

Pour ceci il est possible de redéployer des serveurs à partir d'une image système mise à jour. Tout comme nous l'avions fait à partir du Dashboard :



Le volume est ensuite une ressource utilisable dans la définition d'une instance, grâce à la propriété block_device_mapping de l'instance, « get_ressource » permettant de référencer la ressource correspondant au volume par son identifiant :

```
instance_heat:
   type: OS::Nova::Server
   properties:
    name: une_instance
   flavor: { get_param: instance_type_heat }
   block_device_mapping: [{ device_name: "vda", volume_id : { get_resource : instance_heat_volume } ]]
```

La pile est disponible à cette adresse : https://u03.fr/90openstack/volume_systeme.yaml



Il vous est laissé à titre d'exercice créer la pile, de vous connecter au serveur et de supprimer la pile.

Attachement d'un volume de données à une instance

Les volumes de données contiennent les données persistantes de l'application, lors du redéploiement de l'application il faut les ré-attacher aux instances concernées.

L'attachement du volume système dans l'exemple précédent se faisait en utilisant get_resource pour récupérer l'id du volume créé par la pile.

L'attachement est le même que pour le volume système, sauf que la valeur de l'id va être passée en paramètre :



Nous allons réutiliser le volume que nous avions créé précédemment et qui s'appelait « donnees », l'attachement se faisant par id du volume (car seuls les id sont uniques dans le système), nous obtenons son identifiant depuis le Dashboard ou en ligne de commande :

stack@microwave:~\$ source devstack/openrc demo demo stack@microwave:~\$ openstack volume list					
ID	Name	Status	Size	Attached to	
- <mark>e4539f36-09b7-4198-afc4-6766c5116e22</mark> edc26441-2f67-410d-8b24-ale85844d8dc +	/ donnees ubuntu-18.04_vol /	available in-use +	5 10	/ Attached to ubuntu-18.04_inst on /dev/vda +	-+

La pile complète est disponible à cette adresse : https://u03.fr/90openstack/volume_donnees.yaml

	O M 192.166.1.211/dashboard/project/stable/	••• 🕲 ជ 🛛 🔍 Rechercher	<u> </u>
openstack.	l demo 👻		🛔 demo
bjæt	Lancer la pile		×
Accès Compute Volumes Réseau Orchestration	Nom de la pile * • pile_donness Délai d'attente de création (minutes) * • &0 * Annulation en cas d'échec • Mot de passe pour l'utilisateur "demo" * •	Description : Créer une nouvelle pile avec les valeurs fournies.	a Actions
Versions du mo Générateur de mo Backage d'objet netaé	gabriti_instance m1.small Id_volume_donnees * 0 e4539056-09b7-4198-alc+6766c5116e22 image_instance 0 Volunte-18.04 nom_de_cle * 0 LFO reseau_instance 0 prints reseau_instance 0 prints public		

Nous avions créé un fichier de test dans le volume de données lors de sa création, nous le retrouvons dans notre instance :

ubuntu@pile-instance-heat-vfn4bx77nznj:~\$ df -h /DONNEES/ Filesystem Size Used Avail Use% Mounted on /dev/mapper/PV_DONNEES-lv_donnees 4.8G 10M 4.6G 1% /DONNEES ubuntu@pile-instance-heat-vfn4bx77nznj:~\$ cat /DONNEES/hello.txt Hello

Utilisation des piles à l'aide du CLI openstack

Toutes les opérations sous OpenStack sont effectuées grâce aux API, le Dashboard et le CLI OpenStack sont deux des moyens d'utiliser ces API, il est donc possible d'utiliser le CLI OpenStack pour gérer les piles.

Lancement d'une pile

Nous spécifions les différents paramètres sur la ligne de commande, la création d'une pile est asynchrone et prend un certain temps en fonction des ressources à créer :

stack@microwave:~\$ sour	rce devstack/openrc demo demo	
stack@microwave:~\$ oper	nstack stack createwait	\
	parameter nom_de_cle="LFO"	\
	parameter id_volume_donnees=" <mark>e4539f36-09b7-4198-afc4-6766c5116e22</mark> "	\
	<pre>-t https://u03.fr/90openstack/volume_donnees.yaml</pre>	١
	<pre>ma_pile_donnees</pre>	
2019-10-19 20:31:04Z [m	ma_pile_donnees]: CREATE_IN_PROGRESS	
2019-10-19 20:31:04Z [m	ma_pile_donnees.instance_heat_volume]: CREATE_IN_PROGRESS	
2019-10-19 20:31:06Z [m	ma_pile_donnees.floating_ip_instance_heat]: CREATE_IN_PROGRESS state changed	
2019-10-19 20:31:06Z [n	na_pile_donnees.instance_heat_security_group]: CREATE_IN_PROGRESS state changed	
2019-10-19 20:31:07Z [m	ma_pile_donnees.instance_heat_security_group]: CREATE_COMPLETE	
2019-10-19 20:31:09Z [n	na_pile_donnees.floating_ip_instance_heat]: CREATE_COMPLETE state changed	
2019-10-19 20:31:09Z [m	ma_pile_donnees.instance_heat_volume]: CREATE_COMPLETE state changed	
2019-10-19 20:31:09Z [m	ma_pile_donnees.instance_heat]: CREATE_IN_PROGRESS	
2019-10-19 20:31:38Z [m	na_pile_donnees.instance_heat]: CREATE_COMPLETE	
2019-10-19 20:31:38Z [n	ma_pile_donnees.association_reverseproxy]: CREATE_IN_PROGRESS state changed	
2019-10-19 20:31:45Z [m	ma_pile_donnees.association_reverseproxy]: CREATE_COMPLETE state changed	
2019-10-19 20:31:45Z [m	ma_pile_donnees]: CREATE_COMPLETE	
+	++	
Field	Value	
+	4.1325438-8de5-473e-b310-b54f42bd586f	
stack name	ma pile donnees	
description	Creation d'un volume systeme a partir d'une image et attachement a l'instance	
creation time	2019-10-19720:31:04Z	
updated time	None	
stack status	CREATE COMPLETE	
stack status reason	Stack CREATE completed successfully	
+	++	

Une fois la création de la pile terminée nous retrouvons les données présentes dans la section « outputs » de la définition de notre pile :

<pre>stack@microwave:~\$ openstack</pre>	k stack show ma_pile_donnees
++++++++	'alue
id 41 stack_name ma description Ci	1325438-8de5-473e-b310-b54f42bd586f a_pile_donnees reation d'un volume systeme a partir d'une image et attachement a l'instance
creation_time 20 updated_time NG stack_status Cf stack_status_reason Sf parameters 00 00 00 00 	019-10-19T20:31:04Z ione REATE_COMPLETE tack CREATE completed successfully SS::project_id: 5cd4212611654857a9e941a7b5d35e13 SS::stack_id: 41325438-8de5-473e-b310-b54f42bd586f SS::stack_name: ma_pile_donnees rabarit_instance: m1.small d_volume_donnees: e4539f36-09b7-4198-afc4-6766c5116e22 mage_instance: ubuntu-18.04 om_de_cle: LFO reseau_instance: private reseau_public: public
<mark>outputs</mark> - 	description: <mark>Adresse IP publique de l'instance Heat</mark> output_key: <mark>instance_heat_public_ip</mark> output_value: <mark>172.24.4.166</mark>
parent No disable_rollback Ti deletion_time No stack_user_project_id 14 capabilities [1 notification_topics [1 stack_owner No timeout_mins No tags No	ione I rue I ione I 8bdd00b6d314b2faf056b0b3ebc50f0 I I I J I ione I ione I ione I ione I

Il est possible de lister les différentes ressources qui composent notre pile, notez au passage comment il est possible de sélectionner les colonnes affichées:

stack@microwave:~\$ openstack st -c	<pre>ack resource list ma_pile_donnees \ resource_name -c physical_resource_id </pre>	-c resource_type -c resource_statu	s +
' resource_name +	/ physical_resource_id	' resource_type +	resource_status +
instance_heat instance_heat_volume floating_ip_instance_heat association_reverseproxy instance_heat_security_group +	2d8cad6c-e75d-44f1-acc9-48e61459b0a7 79f521fa-daca-4699-80a8-0ceaf8e941d5 b88cf79f-1a54-4971-9cd9-f2b7f50fa08f 1 9 503bfcc6-892b-45d3-ad43-ea033e7a5ff4	OS::Nova::Server OS::Cinder::Volume OS::Neutron::FloatingIP OS::Nova::FloatingIPAssociation OS::Neutron:SecurityGroup	CREATE_COMPLETE CREATE_COMPLETE CREATE_COMPLETE CREATE_COMPLETE CREATE_COMPLETE

Il est ensuite possible d'afficher une ressource particulière à l'aide de la commande ci-dessous :

stack@microwave:~\$ openstack stack resource show ma_pile_donnees instance_heat_volume

Vous pouvez aussi obtenir la liste des évènements liées à la pile :

stack@microwave:~\$ or	penstack stack event list ma_pile_donnees
2019-10-19 20:31:04Z	[ma_pile_donnees]: CREATE_IN_PROGRESS Stack CREATE started
2019-10-19 20:31:04Z	[ma_pile_donnees.instance_heat_volume]: CREATE_IN_PROGRESS
2019-10-19 20:31:06Z	[ma_pile_donnees.floating_ip_instance_heat]: CREATE_IN_PROGRESS state changed
2019-10-19 20:31:06Z	[ma_pile_donnees.instance_heat_security_group]: CREATE_IN_PROGRESS state changed
2019-10-19 20:31:07Z	[ma_pile_donnees.instance_heat_security_group]: CREATE_COMPLETE state changed
2019-10-19 20:31:09Z	[ma_pile_donnees.floating_ip_instance_heat]: CREATE_COMPLETE state changed
2019-10-19 20:31:09Z	[ma_pile_donnees.instance_heat_volume]: CREATE_COMPLETE state changed
2019-10-19 20:31:09Z	[ma_pile_donnees.instance_heat]: CREATE_IN_PROGRESS state changed
2019-10-19 20:31:38Z	[ma_pile_donnees.instance_heat]: CREATE_COMPLETE state changed
2019-10-19 20:31:38Z	[ma_pile_donnees.association_reverseproxy]: CREATE_IN_PROGRESS state changed
2019-10-19 20:31:452	[ma_pile_donnees.association_reverseproxy]: CREATE_COMPLETE state changed
2019-10-19 20.31.457	[ma_nile_donnees]. CREATE COMPLETE_Stack CREATE completed successfully

Mise à jour d'une pile

Il est possible de mettre à jour une pile (lorsque certains paramètres ou définitions ont été modifiés), lors de la mise à jour de la pile les ressources dont la définition a été modifiée sont détruits et recréés.

Nous ajoutons une règle dans le groupe de sécurité pour ajouter le port 80 :

```
instance_heat_security_group:
type: 0S::Neutron::SecurityGroup
properties:
    rules:
        - remote_ip_prefix: 0.0.0.0/0
        protocol: tcp
        port_range_min: 22
        port_range_max: 22
        direction: ingress
        - remote_ip_prefix: 0.0.0.0/0
        protocol: tcp
        port_range_min: 80
        port_range_min: 80
        direction: ingress
```

La pile mise à jour est disponible à cette adresse :

https://u03.fr/90openstack/volume_donnees_update.yaml

Nous mettons à jour la pile :

stack@microwave:~\$ oper	stack stack update	wait	\
		parameter nom_de_cle=" <mark>LFO</mark> "	Λ
		parameter id_volume_donnees="e4539f36-09b7-4198-afc4-6766c5116e22"	Λ
		<pre>-t https://u03.fr/90openstack/volume_donnees_update.yaml</pre>	Λ
		ma_pile_donnees	
2019-10-19 20:41:08Z [I	na_pile_donnees]: Ui	PDATE_IN_PROGRESS Stack UPDATE started	
2019-10-19 20:41:10Z [I	na_pile_donnees.ins	ance_heat_security_group]: UPDATE_IN_PROGRESS state changed	
2019-10-19 20:41:12Z [I	na_pile_donnees.ins	ance_heat_security_group]: UPDATE_COMPLETE state changed	
2019-10-19 20:41:13Z [I	na_pile_donnees]: Ui	PDATE_COMPLETE Stack UPDATE completed successfully	
+	+	+	
Field	Value		
+	·	+	

<pre> id stack_name description creation_time updated_time stack_status stack_status_reason +</pre>	41325438-8de5-43 ma_pile_donnees Creation d'un vo 2019-10-19720:31 2019-10-19720:43 UPDATE_COMPLETE Stack UPDATE con +	73e-b310-b54f42bd50 blume systeme a par 1:04Z 1:08Z mpleted successful 	86f rtir d'une image en ly	t attachement a l'insta.	 nce 	
ID		Stack Name	Stack Status	Creation Time	Updated Time	+
+ 41325438-8de5-473e-b. +	310-b54f42bd586f	ma_pile_donnees	+ UPDATE_COMPLETE +	2019-10-19T20:31:04Z +		+ + +

Vérifions la bonne mise à jour du groupe de sécurité :

stack@microwave:~\$ openstack :	tack resource list ma_pile_donnees \		
,	-c resource_name -c physical_resource_io	d -c resource_type -c resource_stat	tus
resource_name	/ physical_resource_id	/ resource_type	resource_status +t
instance_heat instance_heat_volume floating_ip_instance_heat association_reverseproxy instance_heat_security_group +	<pre>/ 2d8cad6c-e75d-44f1-acc9-48e61459b0a7 / 79f521fa-daca-4699-80a8-0ceaf8e941d5 b88cf79f-1a54-4971-9cd9-f2b7f50fa08f 19 > 503bfcc6-892b-45d3-ad43-ea033e7a5ff4 </pre>	OS::Nova::Server OS::Cinder::Volume OS::Neutron::FloatingIP OS::Nova::FloatingIPAssociation OS::Neutron::SecurityGroup	CREATE_COMPLETE CREATE_COMPLETE CREATE_COMPLETE CREATE_COMPLETE UPDATE_COMPLETE +
<pre>stack@microwave:~\$ openstack ;</pre>	ecurity group show 503bfcc6-892b-45d3-ad	43-ea033e7a5ff4	
++++++		<i>+</i> 	
created_at 2019-10-1 description id 503bfcc6- name ma_pile_d project_id 5cd421261. revision_number 9 rules direction direction direction direction	<pre>////////////////////////////////////</pre>	7tcgwak x='22', port_range_min='22', x='80', port_range_min='80',	
	PT20:41:11Z		

Destruction d'une pile

Nous pouvons détruite une pile en ligne de commande, une confirmation est demandée car la destruction d'un pile peut provoquer la destruction d'un grand nombre de ressources, le paramètre « -y » permet d'outrepasser la confirmation. La destruction d'une pile est irréversible même si elle n'est pas immédiate :

```
stack@microwave:~$ source devstack/openrc demo demo
stack@microwave:~$ openstack stack delete ma_pile_donnees --wait
Are you sure you want to delete this stack(s) [y/N]? y
2019-10-19 20:49:26Z [ma_pile_donnees]: DELETE_IN_PROGRESS Stack DELETE started
2019-10-19 20:49:26Z [ma_pile_donnees.association_reverseproxy]: DELETE_IN_PROGRESS state changed
2019-10-19 20:49:28Z [ma_pile_donnees.association_reverseproxy]: DELETE_IN_PROGRESS state changed
2019-10-19 20:49:28Z [ma_pile_donnees.instance_heat]: DELETE_IN_PROGRESS state changed
2019-10-19 20:49:28Z [ma_pile_donnees.instance_heat]: DELETE_IN_PROGRESS state changed
2019-10-19 20:49:38Z [ma_pile_donnees.instance_heat]: DELETE_COMPLETE state changed
2019-10-19 20:49:38Z [ma_pile_donnees.instance_heat]: DELETE_IN_PROGRESS state changed
2019-10-19 20:49:38Z [ma_pile_donnees.instance_heat]: DELETE_IN_PROGRESS state changed
2019-10-19 20:49:38Z [ma_pile_donnees.instance_heat]: DELETE_IN_PROGRESS state changed
2019-10-19 20:49:38Z [ma_pile_donnees.instance_heat_security_group]: DELETE_IN_PROGRESS state changed
2019-10-19 20:49:38Z [ma_pile_donnees.instance_heat_security_group]: DELETE_IN_PROGRESS state changed
2019-10-19 20:49:38Z [ma_pile_donnees.instance_heat_security_group]: DELETE_COMPLETE state changed
2019-10-19 20:49:38Z [ma_pile_donnees.instance_heat_security_group]: DELETE_SIN_PROGRESS state changed
2019-10-19 20:49:49Z [ma_pile_donnees.instance_heat_security_group]: DELETE_COMPLETE state changed
2019-10-19 20:49:40Z [ma_pile_donnees.instance_heat_security_group]: DELETE_SIN_PROGRESS state changed
2019-10-19 20:49:40Z [ma_pile_donnees.instance_heat_security_group]: DELETE_SIN_PROGRESS state changed
2019-10-19 20:49:40Z [ma_pile_donnees.instance_heat_security_group]: DELETE_SIN_PROGRESS state changed
2019-10-19 20:49:40Z [ma_pile_donnees.instance_heat_security_group]: DELETE_SIN_PROGRESS state changed
2019-10-19 20:49:40Z [ma_pile_donnees.instance_heat_security_group]: DELETE_SIN_PROGRESS state changed
2019-10-19 20:49:40Z [ma_pile_donnees.instance_heat_s
```

Lancement d'une pile avec un fichier de paramètres

Nous créons fichier de paramètres que nous appelons env.yaml, adaptez le fichier à votre environnement (nom de la clé et ID de votre volume de données) :

parameter_defaults: nom de cle: LFO

nom_de_cle: LFO id_volume_donnees: e4539f36-09b7-4198-afc4-6766c5116e22

Nous utilisons le paramètre « -e » qui permet de spécifier un fichier d'environnement, le paramètre « --wait » permet d'attendre la fin de la création de la pile et de suivre sa progression :

stack@microwave:~\$ ope	enstack stack createwait	١
	-e env.yaml -t https://u03.fr/90openstack/volume_donnees.yam	11 \
	ma_pile_donnees	
2019-10-20 12:07:08Z	[ma_pile_donnees]: CREATE_IN_PROGRESS Stack CREATE started	
2019-10-20 12:07:08Z	[ma_pile_donnees.floating_ip_instance_heat]: CREATE_IN_PROGRESS state changed	
2019-10-20 12:07:09Z	[ma_pile_donnees.instance_heat_security_group]: CREATE_IN_PROGRESS state changed	1
2019-10-20 12:07:10Z	[ma_pile_donnees.instance_heat_security_group]: CREATE_COMPLETE state changed	
2019-10-20 12:07:10Z	[ma_pile_donnees.instance_heat_volume]: CREATE_IN_PROGRESS	
2019-10-20 12:07:11Z	[ma_pile_donnees.floating_ip_instance_heat]: CREATE_COMPLETE state changed	
2019-10-20 12:07:15z	[ma_pile_donnees.instance_heat_volume]: CREATE_COMPLETE state changed	
2019-10-20 12:07:15z	[ma_pile_donnees.instance_heat]: CREATE_IN_PROGRESS state changed	
2019-10-20 12:07:42Z	[ma_pile_donnees.instance_heat]: CREATE_COMPLETE	
2019-10-20 12:07:422	[ma_pile_donnees.association_reverseproxy]: CREATE_IN_PROGRESS	
2019-10-20 12:07:47Z	[ma_pile_donnees.association_reverseproxy]: CREATE_COMPLETE state changed	
2019-10-20 12:07:47Z	[ma_pile_donnees]: CREATE_COMPLETE Stack CREATE completed successfully	
+	-+	+
Field	Value	1
+	-+	·+
id	1b8ac52a-51af-43de-b40b-532bd377eda3	
stack_name	ma_pile_donnees	
description	Creation d'un volume systeme a partir d'une image et attachement a l'instance	
creation_time	2019-10-20712:07:072	
updated_time	None	
stack_status	CREATE_COMPLETE	
stack_status_reason	Stack CREATE completed successfully	1
+	-+	·+

Le paramètre « -c » sur les commandes de type 'show' permet de n'afficher que les champs listés :



Création d'infrastructures réseau

Nous avons dit que l'infrastructure réseau est constituée de :

- réseaux
- sous-réseaux
- routeurs
- ports (qui sont généralement créés de façon implicite, sur les serveurs, les routeurs...)

Ci-dessous une architecture réseau typique d'une application 'Cloud' avec 4 réseaux :



La pile complète est disponible à cette adresse :

<u>https://u03.fr/90openstack/4_reseaux.yaml</u>

Il va nous falloir deux adresses IP flottantes (une pour le bastion SSH de votre tenant, et une pour le reverse-proxy). Nous avons normalement une adresse IP allouée mais elle est déjà associée à un serveur, nous en allouons deux nouvelles :

stack@microwave:~\$ oper	nstack floating in	o list -c ID -c 'Float:	ing IP Address' -c	'Fixed IP Address'
ID		Floating IP Address	Fixed IP Address	•
168bd864-8489-4d50-96	5dd-24bd93f316e4	172.24.4.105	10.0.0.43	•
stack@microwave:~\$ oper	nstack floating ip	o create public -c 'flo	pating_ip_address'	-c id
Field	Value		• 	
floating_ip_address id	172.24.4.239 3fe5d827-a1aa-47	7d9-a73f-cb22cdaecc27	• 	
stack@microwave:~\$ oper	nstack floating ip	o create public -c 'flo	- pating_ip_address' -	-c id
Field	Value		• 	
floating_ip_address id	172.24.4.123 311cbd22-c4f4-41	L90-b342-d0d233906ccc	' 	
stack@microwave:~\$ oper	nstack floating ig	o list -c ID -c 'Float:	ing IP Address' -c	'Fixed IP Address'

+	Floating IP Address	Fixed IP Address
168bd864-8489-4d50-96dd-24bd93f316e4	172.24.4.105	10.0.0.43
311cbd22-c4f4-4190-b342-d0d233906ccc	172.24.4.123	None
3fe5d827-alaa-47d9-a73f-cb22cdaecc27	172.24.4.239	None

Nous pouvons désormais instancier notre pile :

stack@microwave:~\$ ope	nstack stack createwait	\
	parameter nom_de_cle=" <mark>LFO</mark> "	١
	parameter id_floating_ip_bastion=" <mark>3fe5d827-a1aa-47d9-a73f-cb22cdaecc27</mark> "	١
	parameter id_floating_ip_reverseproxy=" <mark>311cbd22-c4f4-4190-b342-d0d233906ccc</mark> "	١
	-t https://u03.fr/90openstack/4_reseaux.yaml	
	ma_pile_4_reseaux	
2019-10-20 12:46:39Z [ma_pile_4_reseaux]: CREATE_IN_PROGRESS Stack CREATE started	
2019-10-20 12:46:40Z [ma_pile_4_reseaux.instance_reverseproxy_security_group]: CREATE_IN_PROGRESS state changed	
2019-10-20 12:46:412 [ma_pile_4_reseaux.instance_reverseproxy_security_group]: CREATE_COMPLETE state changed	
2019-10-20 12:46:412 [ma_pile_4_reseaux.reverseproxy_network]: CREATE_IN_PROGRESS state changed	
2019-10-20 12:46:422 [ma_pile_4_reseaux.reverseproxy_network]: CREATE_COMPLETE state changed	
2019-10-20 12:46:422 [ma_pile_4_reseaux.router]: CREATE_IN_PROGRESS state changed	
2019-10-20 12:46:422 [ma_pile_4_reseaux.reverseproxy_subnet]: CREATE_IN_PROGRESS state changed	
2019-10-20 12:46:43Z [ma_pile_4_reseaux.instance_bastion_security_group]: CREATE_IN_PROGRESS state changed	
2019-10-20 12:46:44Z [ma_pile_4_reseaux.instance_database_security_group]: CREATE_IN_PROGRESS state changed	
2019-10-20 12:46:45Z [ma_pile_4_reseaux.reverseproxy_subnet]: CREATE_COMPLETE state changed	
2019-10-20 12:46:45Z [ma_pile_4_reseaux.instance_bastion_volume]: CREATE_IN_PROGRESS state changed	
2019-10-20 12:46:46Z [ma_pile_4_reseaux.instance_database_security_group]: CREATE_COMPLETE state changed	
2019-10-20 12:46:46Z [ma_pile_4_reseaux.instance_application_security_group]: CREATE_IN_PROGRESS state changed	
2019-10-20 12:46:47Z [ma_pile_4_reseaux.database_network]: CREATE_IN_PROGRESS state changed	
2019-10-20 12:46:48Z [ma_pile_4_reseaux.instance_reverseproxy_volume]: CREATE_IN_PROGRESS state changed	
/		
2019-10-20 12:50:02Z [ma_pile_4_reseaux.association_bastion]: CREATE_IN_PROGRESS	
2019-10-20 12:50:05Z [ma_pile_4_reseaux.instance_reverseproxy]: CREATE_COMPLETE	
2019-10-20 12:50:06Z [ma_pile_4_reseaux.association_reverseproxy]: CREATE_IN_PROGRESS	
2019-10-20 12:50:28Z [ma_pile_4_reseaux.association_bastion]: CREATE_COMPLETE state changed	
2019-10-20 12:50:422 [ma_pile_4_reseaux.association_reverseproxy]: CREATE_COMPLETE state changed	
2019-10-20 12:50:422 [ma_pile_4_reseaux]: CREATE_COMPLETE Stack CREATE completed successfully	
+	*	
Fleld	I value I	
, id		
l stack name	ma nile 4 reseaux	
description		
1	Nous construisons 4 instances de serveur.	
	- Une instance bastion SSH	
	En pré-reguis vous devez disposer des ID OpenStack des éléments suivants:	
	- L'adresse IP flottante qui sera attachée au reverse-proxy	
	- L'adresse IP flottante qui sera attachée au basion SSH	
creation time	2019-10-20112:46:38Z	
updated time	None	
stack status	CREATE COMPLETE	
stack status reason	Stack CREATE completed successfully	
+	+	



Attention même si la pile est finie de créer il peut falloir plusieurs minutes pour que les différents packages s'installent sur les différentes instances, cela dépend de la puissance de votre machine DevStack.

A l'issue de la création de la pile (et de l'initialisation de scripts sur les différents serveurs) si vous tapez l'adresse IP du reverse-proxy dans votre navigateur web la page suivante, ceci signifie que vous avez atteint le reverse-proxy, que celui-ci a pu relayer votre requête au serveur d'application et que le serveur d'application a interrogé le serveur de bases de données :

stack@microwave:~\$ or	venstack server list -c Name -c Networks	,
Name	Networks	
<mark>reverseproxy</mark> bastion application database ubuntu-18.04_inst	reverseproxy=10.99.2.194, 172.24.4.123 bastion=10.99.1.125, 172.24.4.239 application=10.99.3.13 database=10.99.4.13 private=fdac:50a1:b51d:0:f816:3eff:fec8:2ed5, 10.0.0.43, 172.24.4.105	+

172.24.4.123/	× +		- 🗆 ×
€ → ୯ ŵ	① 172.24.4.123	😇 🏠 🔍 Rechercher	¥ ⊪ © @ ∃
dept_no dept_na	me		
d009 Customer Se	rvice		
d005 Developmen	t		
d002 Finance			
d003 Human Reso	urces		
d001 Marketing			
d004 Production			
d006 Quality Man	agement		
d008 Research			
1922			

Création des réseaux et sous-réseaux

Le réseau 'public' est celui fourni par le Cloud, nous allons créer les 4 autres, chacun des réseaux est composé d'au moins un sous-réseau, pour chacun des sous-réseaux il faut choisir les paramètres suivants :

- Plage d'adresses IP (notation CIDR)
- Plage d'adresses IP pour attribuer dynamiquement les adresses IP en DHCP (notation : adresses IP début et fin)
- Adresse IP de la passerelle (cette adresse sera utilisée par le routeur lors de l'attachement du sous-réseau au routeur)
- Le ou les serveurs DNS qui seront utilisés par les serveurs connectés au sous-réseau

Dans le choix des adresses IP il faut garder à l'esprit qu'il s'agit de réseaux privés propres au projet, elles ne doivent être uniques qu'à l'intérieur du projet mais il faut éviter qu'elles entrent en collision avec des réseaux utilisés par les utilisateurs de votre application.

Pour chacun des réseaux nous commençons par créer le réseau (objet OS::Neutron::Net) puis le sous-réseau (OS::Neutron::Subnet).

La liaison entre le sous-réseau et le réseau se fait par la propriété network_id du sous-réseau, en utilisant get ressource pour obtenir l'ID du réseau.

La propriété allocation_pools donne la ou les plages d'allocation par le serveur DHCP, la propriété cidr donne la plage d'adresses du réseau, gateway_ip est l'adresse IP choisie pour être celle de la passerelle par défaut du réseau.

```
bastion_network:
  type: 0S::Neutron::Net
  properties:
    name: bastion
bastion_subnet:
  type: 0S::Neutron::Subnet
  properties:
    network_id: { get_resource: bastion_network }
    allocation_pools:
    - {start: 10.99.1.100, end: 10.99.1.250}
    cidr: 10.99.1.0/24
    dns_nameservers: { get_param: dns }
    gatewayCréation du routeur
```

Nous créons le routeur et nous le rattachons au réseau public (external_gateway_info), ainsi le routeur routera vers le réseau public tout le trafic qui n'est pas destiné à un des réseau qu'il gère, il fera également de la translation d'adresse pour masquer les adresses privées mais surtout présenter une adresse routable sur le réseau public :

```
router:
  type: 05::Neutron::Router
  properties:
    name: routeur
    external_gateway_info:
        network: { get_param: public_network }
```

Nous voyons ci-dessous l'adresse IP publique qui sera utilisée par le routeur pour faire de la translation d'adresse (python -m json.tool permet de mettre en forme du format JSON) :



Les interfaces sont créées comme ceci pour connecter les réseaux au routeur : :



Nous voyons ci-dessous les informations qui correspondent aux différentes interfaces du réseau une fois que le pile a été créée, nous retrouvons nos 4 réseaux et les 4 adresses que nous avons attribué à chacune des passerelles (python -m json.tool permet de mettre en forme du format JSON) :



Attachement du serveur au réseau

La propriété networks est un tableau de réseaux auxquels doit être attaché le serveur. Il est possible au passage de fixer l'adresse IP du serveur, si fixed_ip n'est pas spécifié alors une adresse sera automatiquement allouée à partir des plages d'adresses DHCP des sous-réseaux concernés. :

```
instance_database:
    properties:
    networks:
        - network: { get_resource: database_network }
        fixed_ip: 10.99.4.13
```

1

Techniquement le serveur est configuré en DHCP, et donc quand une adresse IP est fixée il s'agit d'une réservation DHCP effectuée au niveau de Neutron, et non d'une adresse IP configurée de façon statique dans les fichiers de configuration de l'interface.

Si vous fixez une adresse IP il vous est conseillé de ne pas le faire dans la plage d'adresse que vous avez assigné au DHCP lors de la création du sous-réseau (allocation pools).

Lorsque que nous regardons les paramètres réseau d'un de nos serveurs nous retrouvons les propriétés que nous avons définies pour notre sous-réseau :

ubuntu@bastion:	~\$ ip addr						
[]							
2: ens3: <broal< td=""><td>CAST,MULTICAST,U</td><td>JP,LOWER_UP> mtu</td><td>1450 qd:</td><td>isc pi</td><td>fifo_fas</td><td>t sta</td><td>te UP group default qlen 1000</td></broal<>	CAST,MULTICAST,U	JP,LOWER_UP> mtu	1450 qd:	isc pi	fifo_fas	t sta	te UP group default qlen 1000
link/ether	fa:16:3e:04:bb:3	32 brd ff:ff:ff:f	ff:ff:ff				
inet <mark>10.99.</mark>	1.13/24 brd 10.9	9.1.255 scope gl	obal ens	s3			
valid_lf	ft forever prefer	red_lft forever					
inet6 fe80:	:f816:3eff:fe04	bb32/64 scope li	nk				
valid_lf	ft forever prefer	red_lft forever					
ubuntu@bastion:	~\$ netstat -rn						
Kernel IP routi	ng table						
Destination	Gateway	Genmask	Flags	MSS	Window	irtt	Iface
0.0.0.0	10.99.1.254	0.0.0.0	UG	0	0	0	ens3
10.99.1.0	0.0.0.0	255.255.255.0	U	0	0	0	ens3
ubuntu@bastion:	~\$ nslookup u03.	fr					
Server:	<mark>8.8.8.8</mark>						
Address:	8.8.8.8#53						
Non-authoritati	ve answer:						
Name: u03.fr							
Address: 37.187	7.126.87						

Attachement à plusieurs réseaux

Il est possible de spécifier plusieurs réseaux en spécifiant plusieurs entrées network sous la propriété networks.

C'est par exemple utilisable pour un bastion ou un reverse-proxy afin d'isoler le réseau public et le réseau privé :



Le stockage d'objets avec Swift

Swift est un composant de OpenStack qui permet de stocker des objets (fichiers) qui seront accessibles en HTTP/HTTPS :

- de façon publique sans authentification
- à des utilisateurs authentifiés par Keystone et autorisés
- à des utilisateurs disposant d'une URL temporaire donnant des droits spécifiques pour une période de temps limitée.

Chaque projet peut créer des containers qui vont permettre de stocker des objets, à l'intérieur de chaque container il n'y a pas de réelle structure arborescente, les objets sont donc tous stockés au même niveau.

Création d'un conteneur Swift et chargement d'objets depuis le Dashboard

Le Dashboard permet de gérer les conteneurs Swift (création, chargement de fichiers, destruction).

(=) (=) (= http://192.166	8.1.211/de:	hboard/project/containers/	+ C Recherche	–
Conteneurs - OpenStack D	a× 📑			
openstack.	🔳 den	10 ¥		🛔 demo 👻
Projet	♥ és API	Projet / Stockage d'objet / Conteneurs		
Compute	>	Conteneurs		
Volumes	>	(10.000)		
Réseau	>	+ Conteneur	Ø Sélectionner un conteneu	ir à parcourir.
Orchestration	>	Q Cliquer ici pour les filtres. x		
Stockage d'objet	~	Aucun élément à afficher.		
Conb	eneurs			
Identité	>			
				🥹 🔩 100% 👻

En cliquant sur le bouton « +Conteneur » nous affichons l'écran de création de conteneur, nous l'appelons « photos », nous déclarons le conteneur public.



Swift ne gère pas les répertoires mais le Dashboard permet de simuler la création d'un répertoire (bouton « +Dossier »), ceci est simulé en chargeant un fichier à 0 octets avec le nom du pseudodossier.



Il vous est laissé à titre d'exercice à la fin du chapitre la création et l'utilisation de répertoires depuis le Dashboard.

Le conteneur est en accès public, il se présente sous la forme d'un répertoire accessible en http, il est possible d'obtenir l'URL du conteneur en cliquant avec le bouton de droite sur « Lien » en face de « Accès public » :

Ca (=) 👩 http://192.16	8.1.211/das	hboard/project/containers/container/photos		+ C	Rechercher		5	- 6	1 × 200
Conteneurs - OpenStack D	a× 📑								
🖸 openstack.	🗐 den	10 *						4	demo 🕶
Projet	~	Projet / Stockage d'objet / Conteneurs				Succès : Contene	ur photos	créé	ж
Compute	sės API	Conteneurs							
Volumes	>	+ Conteneur	photos						
Orchestration	>	Q Cliquer ici pour les filtres. 🗙	Q Plane Islance to Ohers			×	1	+ Dossier	
Stockage d'objet	¥	photos 🗴	Affichage de 0 élément	Tail	le				
Cont	enours	Nombre d'objets: 0		Aucun élé	ment å afficher.				
Identité	>	Date de création: Mar 7, 2018 & Accès publique: Linn	Affichage de 0 élément						
								6 • 1	00% +

Conteneurs - OpenStack I	68.1.211/de Da × [shboard/project/container	s/container/photos	+ C Rechercher	× □ × ₽ ₪ ☆ ₪ ⊕
openstack.	📾 der	no *			🛔 dumo 👻
Projet	↔ ces AFI	Projet / Stockage	Charger le fichier dans : photos	×	
Compute	>	Contene	Fichier C\TEMP\20140717_210108.JPG Parcourir	Remarque : Les délimiteurs (/) sont autorisés dans le nom de fichier afin de placer le nouveau fichier dans un deseire aui sear créé lors du charaement du Schier (dans	
Volumes	>	A Contractor	Nom du fichier	un des dossiers en profondeur).	
Résimu	>		lumieres		
Orchestration	>	Q			× ± + Donsier
Stockage d'objet	×	photos		X Annuler 2. Charger un fichier	
.Con	teneurs	Nombre d'objets: Taille:	0 0 octets	Aucun élément é aificher	t.
13					💷 🔍 100% 👻

Une fois chargé le fichier apparaît dans la liste, il est possible de le télécharger, d'obtenir les informations sur le fichier ou de le supprimer :

(=) (=) http://192.16(8.1.211/des	hboard/project/containers/container/photos		+ C Rechercher	- ロ × ター 公会回 @
Conteneurs - OpenStack D	a× 💽				
openstack.	🔳 dem	• •			🛔 demo 👻
Projet	•	Projet / Stockage d'objet / Conteneurs			
Compute	>	Conteneurs			
Volumes	>	+ Conteneur	photos		
Orchestration	>	Q //terme lat some for differen	Q. Pflance for an effective		× ± +Dossier
Stockage d'objet	×	photos	Affichage de 1 élément		
Cont	eneurs	Nombra d'objete: 1	□ Nom ▲	Taille	
Identité	×	Taille: 5.01 Mo Date de création: Mar 7, 2018	Iumieres	5.01 Mo	Télécharger 👻
		☑ Accès publique: Lien	Affichage de 1 élément		
					🞯 🔍 100% 👻 🖉

Le lien obtenu pour le conteneur est:

٠

• http://192.168.1.211:8080/v1/AUTH_3f36cd33d14e468293caa7bf311c00de/photos

Le nom du fichier donné lors du chargement est 'lumieres', l'URL du fichier est donc :

http://192.168.1.211:8080/v1/AUTH_3f36cd33d14e468293caa7bf311c00de/photos/lumieres



Manipulation des conteneurs et objets en ligne de commande

Il est possible de lister les conteneurs et leur contenu à l'aide de l'utilitaire en ligne de commande :

stack@microwav	e:~\$ source devstack/openrc demo demo
++	e. o openstack container fist
Name	
++	
photos	
stack@microwav	$e:\sim \$$ openstack container show photos
Field	/ Value
account	/ AUTH 3f36cd33d14e468293caa7bf311c00de
bytes_used	5249087
container	photos
object_count	1
read_acl	.r:*,.rlistings
+	-+

Il est également possible de lister le contenu des conteneurs ainsi que les informations sur un objet à l'aide de l'utilitaire en ligne de commande :

<pre>stack@microwave:~\$ ++ Name ++ l lumieres </pre>	openstack object list photos
++ stack@microwave:~\$	openstack object show photos lumieres
Field ++	Value
account container content-length content-type etag last-modified object properties	AUTH_3f36cd33d14e468293caa7bf311c00de photos 5249087 application/octet-stream f66cal814ff66b249ee973b01139207e Wed, 07 Mar 2018 21:10:44 GMT lumieres Orig-Filename='20140717_210108.JPG'

Stockage et récupération d'un fichier

La commande « <code>openstack object create » permet charger un objet dans Swift, on note que le champ « etag » est en fait le hash md5 du fichier :</code>



La commande « openstack object save» permet télécharger un objet présent dans Swift, on note que par défaut la commande sauve l'objet avec le nom sous lequel il a été stocké dans Swift, il est possible d'utiliser le paramètre « --file » pour spécifier le nom (et le chemin) du fichier sous lequel l'objet doit être sauvegardé localement :

```
stack@microwave:~$ openstack object save photos lumieres
stack@microwave:~$ file lumieres
lumieres: JPEG image data, Exif standard: [TIFF image data, little-endian, direntries=12, height=3456, manufacturer=SAMSUNG,
model=EK-GC100, orientation=upper-left, xresolution=208, yresolution=216, resolutionunit=2, software=GC100XXBMC4,
datetime=2014:07:17 21:01:07, width=4608], baseline, precision 8, 4608x3456, frames 3
stack@microwave:-$ penstack object save photos lumieres --file /tmp/lumieres.jpg
stack@microwave:-$ ls -1 /tmp/lumieres.jpg
-rw-rw-r-- 1 stack stack 5249087 Mar 10 22:36 /tmp/lumieres.jpg
```

Ajout de propriétés

Lors de l'ajout de l'objet via le Dashboard une propriété avait été ajoutée automatiquement (le nom réel du fichier initial), ce que n'a pas fait la commande « openstack object create » :

stack@microwave:~\$	openstack object show photos lumieres
++ Field	
++	+
account	AUTH_3f36cd33d14e468293caa7bf311c00de
container	photos
content-length	5249087
content-type	application/octet-stream
etag	f66ca1814ff66b249ee973b01139207e
------------------	--
last-modified	Wed, 07 Mar 2018 21:10:44 GMT
object	lumieres
properties	Orig-Filename='20140717 210108.JPG'
+	-+
stack@microwave:	$\cdot \hat{s}$ openstack object show photos takeshiba
+	-+
Field	Value
+	-+
account	AUTH_3f36cd33d14e468293caa7bf311c00de
container	photos
content-length	2357220
content-type	image/jpeg
etag	584175f6b846a13f34477d08d1ed9a0c
last-modified	Sat, 10 Mar 2018 21:27:25 GMT
object	takeshiba.jpg
+	-+

Il est possible d'ajouter les propriétés de son choix à un objet sous la forme de paires « clé=valeur »:

<pre>stack@microwave:~ stack@microwave:~</pre>	\$ openstack object set photos takeshiba.jpg \$ openstack object show photos takeshiba.jpg	-property annee=2014 \ -property categorie=vacances \ -property lieu='Tokyo'	
+ Field	+	+	
account container content-length content-type etag last-modified object	AUTH_3f36cd33d14e468293caa7bf311c00de photos 2357220 image/jpeg 584175f6b846a13f34477d08d1ed9a0c Sat, 10 Mar 2018 21:55:17 GMT takeshiba.jpg		
properties +	Annee='2014', Categorie='vacances', Lieu=' +	<u>okyo'</u> +	

La commande « set » efface les propriétés déjà présentes sur un objet, supposons que nous désirions modifier la propriété 'annee' pour mettre '2015', il ne suffit pas de spécifier la propriété 'annee', sinon toutes les autres propriétés seront supprimées.

<pre>stack@microwave: stack@microwave:</pre>	openstack object set photos tak openstack object show photos tak	əshiba.jpg keshiba.jp
+	Value Value	+ +
account container content-length content-type etag last-modified	AUTH_3f36cd33d14e468293caa7bf31 photos 2357220 image/jpeg 584175f6b846a13f34477d08d1ed9a0 Sat. 10 Mar 2018 21:56:52 GMT	1c00de c
object <mark>properties</mark> +	takeshiba.jpg Annee='2015' +	



La commande « unset » permet d'effacer une ou plusieurs propriétés, il vous est laissé à titre d'exercice l'ajout et la suppression de propriétés sur un objet présent dans Swift.

Terraform

On l'a vu, les API OpenStack sont très puissantes, elles sont utilisables dans la plupart des langages de programmation et donc de nombreux logiciels ont été développés pour tirer partie de OpenStack et de ses API.

Terraform permet de faire de l'orchestration comme Heat, il permet d'utiliser de nombreuses solutions de Cloud, dont OpenStack :

https://www.terraform.io/docs/providers/openstack/

Même si le format des fichiers est différent entre Heat et Terraform la logique est la même, en effet Heat et Terraform sont tributaires des API OpenStack et de la façon dont elles s'articulent.

Les descriptions d'infrastructures dans Terraform s'appellent 'configurations'.

Appliquer la configuration (terraform apply) va effectuer les actions nécessaires pour réaliser l'infrastructure décrite dans la configuration, soit en la construisant (lors du premier appel à terraform apply), soit en mettant à jour l'infrastructure (lors appels suivants à terraform apply). Pour ceci Terraform va bâtir un '*plan*' des opérations à réaliser.

Un appel à terraform destroy va permettre de détruire l'infrastructure construite.

Installation et initialisation

Terraform est téléchargeable à cette adresse : https://www.terraform.io/downloads.html

Terraform est disponible pour Windows et Linux ('i386', 'amd64' et processeurs 'arm'). Il se présente sous la forme d'un fichier '.zip' contenant un unique programme exécutable en commande ligne.

Les processeurs 'arm' sont utilisés sur la plupart des téléphones portables, des tablettes, de certains équipements réseaux et ordinateurs 'monocarte' tels que les Raspberry Pi

Dans cette exemple nous utilisons un Raspberry Pi (qui fait partie de mes machines de tests), il s'agit d'une mini machine qui coûte environ 40 euros.

https://blog.u03.fr/la-paillasse-de-linformaticien/

Téléchargez la version qui correspond à votre plateforme, dézippez le fichier et placez-le dans un répertoire vide que vous appelez 'tf' (par exemple).

<pre>foucher@raspberrypi:~/tf \$ wget https://releases.hashicorp.com/terraform/0.12.12/terraform_0.12.12_linux_arm.zip 2019-10-20 14:36:39 https://releases.hashicorp.com/terraform/0.12.12/terraform_0.12.12_linux_arm.zip Resolving releases.hashicorp.com (releases.hashicorp.com) 151.101.1.183, 151.101.65.183, 151.101.129.183, Connecting to releases.hashicorp.com (releases.hashicorp.com) 151.101.1.183 :443 connected. HTTP request sent, awaiting response 200 0K Length: 15115877 (14M) [application/zip] Saving to: `terraform_0.12.12_linux_arm.zip'</pre>					
terraform_0.12.12_linux_arm.zip 100%					
[=====>] 14.42M 790KB/s in 19s					
2019-10-20 14:36:59 (785 KB/s) - `terraform_0.12.12_linux_arm.zip' saved [15115877/15115877]					
foucher@raspberrypi:~/tf \$ unzip terraform_0.12.12_linux_arm.zip Archive: terraform_0.12.12_linux_arm.zip					

90 Minutes avec OpenStack – 2019/11/11

Dans le répertoire 'tf' créez un fichier que vous appellerez 'provider.tf' (par exemple, mais le nom doit finir par '.tf'):

```
provider "openstack" {
    user_name = "demo"
    tenant_name = "demo"
    password = "topsecret"
    auth_url = "http://192.168.1.211/identity"
    region = "RegionOne"
```

Lancez la commande 'terraform init', Terraform va charger tous les fichiers '.tf' présents dans le répertoire (ici uniquement le fichier 'provider.tf') et il va télécharger depuis le site de l'éditeur le plugin nécessaire pour utiliser OpenStack :

```
foucher@raspberrypi:~/tf $ ./terraform init
Initializing the backend ...
Initializing provider plugins...
- Checking for available provider plugins...
- Downloading plugin for provider "openstack" (terraform-providers/openstack) 1.23.0...
The following providers do not have any version constraints in configuration,
so the latest version was installed.
To prevent automatic upgrades to new major versions that may contain breaking
changes, it is recommended to add version = "..." constraints to the
corresponding provider blocks in configuration, with the constraint strings
suggested below.
* provider.openstack: version = "~> 1.23"
Terraform has been successfully initialized!
You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.
If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

Éléments d'une configuration

Nous avons déjà vu le provider, il permet à Terraform de récupérer le bon plugin et permet de lui fournir les paramètres associés (les mêmes que dans les variables d'environnement positionnées par le script openro).



Terraform est multi-provider, il permet par exemple de déployer à la fois de l'OpenStack, de l'AWS et du VMware...

Dans Terraform les variables (variable) sont l'équivalent des paramètres dans Heat, elle permettent d'adapter la configuration à l'environnement dans lequel elle est déployée.

Les ressources (resource) sont les éléments de l'infrastructure, comme dans Heat.

Les sources de données (data) sont des constructions syntaxiques qui permettent de récupérer l'ID unique d'éléments préexistants dans OpenStack à partir de différents paramètres (par exemple récupérer l'ID d'une image pour construire un volume de boot...)

Une première configuration

Nous allons faire une première configuration qui classiquement va créer une instance à partir d'une image.

Nous créons un fichier que nous appelons premiere_configuration.tf qui est également disponible à l'adresse https://u03.fr/90openstack/premiere_configuration.tf :



Vous avons 4 éléments variable qui correspondent à nos paramètres, il y a une description et il est possible de mettre une valeur par défaut.

La valeur des paramètres est récupérée par la construction syntaxique suivante :

"\${var.image instance}"

Comme dans Heat il est possible de passer un fichier avec la valeur des paramètres. Nous créons le fichier terraform.tfvars avec la valeur de nos éléments variable (nous conservons la valeur par défaut pour les variables reseau_prive et image_instance) :

nom_de_cle = "LFO" gabarit_instance = "ds1G"

La commande terraform plan nous permet de visualiser les opérations nécessaires à la construction de l'infrastructure décrite dans la configuration :



+ image_id	= (known after apply)
+ image_name	= " <mark>ubuntu-18.04</mark> "
+ key_pair	= " <mark>LFO</mark> "
+ name	= "instance_terraform"
+ power_state	= "active"
+ region	= (known after apply)
+ security_groups	= (known after apply)
+ stop_before_destro	by = false
+ network {	
+ access_network	k = false
+ fixed_ip_v4	= (known after apply)
+ fixed_ip_v6	= (known after apply)
+ floating_ip	= (known after apply)
+ mac	= (known after apply)
+ name	= "private"
+ port	= (known after apply)
+ uuid	= (known after apply)
}	
}	
Plan: 1 to add, 0 to chang	ge, 0 to destroy.
Note: You didn't specify a	an "-out" parameter to save this plan, so Terraform
can't guarantee that exact	tly these actions will be performed if
"terraform apply" is subse	equently run.

La commande terraform apply va déployer la configuration, une confirmation est demandée :



Nous retrouvons notre serveur en utilisant la commande CLI ${\tt openstack}$:

<pre>stack@microwave:~\$ source devstack/openrc demo demo stack@microwave:~\$ openstack server show instance_terraform</pre>				
stack@microwave:~\$ openstack se	erver show instance_terraform			
+ Field	Value			
<pre>> Jeld > > OS-DCF:diskConfig OS-EXT-AZ:availability_zone OS-EXT-STS:power_state OS-EXT-STS:tosk_state OS-SRV-USG:launched_at OS-SRV-USG:terminated_at accessIPv4 accessIPv4 accessIPv6 addresses config_drive created flavor hostId id image key_name name progress project_id properties security_groups status vedted</pre>	MANUAL nova Running None active 2019-10-20T15:58:54.000000 None private=fdac:50a1:b51d:0:f816:3eff:fe23:e552, 10.0.0.24 2019-10-20T15:58:422 ds1G (d2) b3ce8aa16af22651ac4ab1291280fd727ff9c016c9068dcfcc861a75 c8e4a067-d449-42ba-b5cf-ada95f2ff248 ubuntu-18.04 (7ab813dd-6937-4f96-8cd1-ed0ec64e8ad5) LFO instance_terraform 0 5cd4212611654857a9e941a7b5d35e13 name='default' ACTIVE 2010 10 20m15.58.557			
user_id	57d02b0938d149a1aac96ce6757a3a14			
volumes_attached +				

La commande terraform destroy va permettre de détruire l'infrastructure :

```
foucher@raspberrypi:~/tf $ ./terraform destroy
openstack_compute_instance_v2.instance_terraform: Refreshing state... [id=c8e4a067-d449-42ba-b5cf-ada95f2ff248]
An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:

    destroy

Terraform will perform the following actions:
   # openstack_compute_instance_v2.instance_terraform will be destroyed
resource "openstack_compute_instance_v2" "instance_terraform" {
    - access_ip_v4 = "10.0.0.24" -> null
    - access_ip_v6 = "[fdac:50a1:b51d:0:f816:3eff:fe23:e552]" -> null
         - access_ip_v6 = "[fdac:50a1:b51d:0:f816:3eff:fe23:e552]" -> nui
- all_metadata = {} -> null
- availability_zone = "nova" -> null
- flavor_id = "d2" -> null
- flavor_name = "ds1G" -> null
- force_delete = false -> null
- id = "c8e4a067-d49-42ba-b5cf-ada95f2ff248" -> null
- image_id = "7ab813dd-6937-4f96-8cd1-ed0ec64e8ad5" -> null
- image_name = "ubuntu-18.04" -> null
- key_pair = "LFO" -> null
- name = "instance_terraform" -> null
- power_state = "active" -> null
- security_groups = {
- "default",
              - "default",
] -> null
          - stop_before_destroy = false -> null
           - network {
                 - access_network = false -> null
- fixed_ip_v4 = "10.0.0.24" -> null
- fixed_ip_v6 = "[fdac:50a1:b51d:0:f816:3eff:fe23:e552]" -> null
- mac = "fa:16:3e:23:e5:52" -> null
- name = "private" -> null
- uuid = "0e6e2f0c-c4cc-4eca-8428-8117864b6e4a" -> null
                    - access network = false -> null
              ł
Plan: 0 to add, 0 to change, 1 to destroy.
Do you really want to destroy all resources?
   Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.
   Enter a value: yes
openstack_compute_instance_v2.instance_terraform: Destroying... [id=c8e4a067-d449-42ba-b5cf-ada95f2ff248]
openstack_compute_instance_v2.instance_terraform: Still destroying... [id=c8e4a067-d449-42ba-b5cf-ada95f2ff248, 10s elapsed]
openstack_compute_instance_v2.instance_terraform: Destruction complete after 10s
Destroy complete! Resources: 1 destroyed.
```

Création d'un volume système et d'un groupe de sécurité :

Nous allons déployer une infrastructure qui comprend les éléments suivants :

- Un volume de boot créé à partir d'une image
- Une instance qui démarre sur ce volume en utilisant un script de démarrage (user data)
- Un groupe de sécurité autorisant le trafic SSH

Dans les piles Heat il est possible de créer un volume de boot à partir du nom de l'image ou de son ID unique. L'unicité des noms d'images n'est pas garantie par Glance et lorsque deux images portent le même nom cela provoque l'échec de création de la pile Heat.

Afin de se prémunir contre ce genre d'échecs Terraform utilise l'ID de l'image au lieu de son nom, une source de donnés (data) permet d'interroger la catalogue Glance pour récupérer l'ID à partir du nom de l'image passé dans une variable.

```
variable "image_instance" {
  description = "Nom de l'image"
  default = "ubuntu-18.04"
}
data "openstack images image v2" "image boot" {
```

```
name = "${var.image_instance}"
most_recent = true
```

Il est ensuite possible de créer le volume à partir de cette image.

Le volume_type est également récupéré à partir d'une variable (il s'agit habituellement de lvmdriver-1 pour DevStack).

L'ID de l'image est récupéré à partir de la source de données grâce à la construction suivante :

• "\${data.openstack_images_image_v2.image_boot.id}"

```
resource "openstack_blockstorage_volume_v2" "volume_boot_terraform" {
    name = "volume_terraform"
    description = "Le volume de boot"
    size = 10
    volume_type = "${var.type_volume}"
    image_id = "${data.openstack_images_image_v2.image_boot.id}"
```

Il faut créer le groupe de sécurité puis lui attacher les règles de sécurité, ici celle qui autorise le trafic SSH (tcp port 22) :

```
resource "openstack_networking_secgroup_v2" "secgroup_tf" {
    name = "secgroup_ssh"
    description = "Autoriser le flux SSH"
}
resource "openstack_networking_secgroup_rule_v2" "secgrouprule_secgroup_tf_ssh" {
    direction = "ingress"
    ethertype = "IPv4"
    protocol = "tcp"
    port_range_min = 22
    port_range_max = 22
    remote_ip_prefix = "0.0.0.0/0"
    security_group_id = "$(openstack_networking_secgroup_v2.secgroup_tf.id)"
```

Nous pouvons désormais créer notre serveur en y attachant les différents éléments que nous avons construits. Le script exécuté au moment du lancement du serveur (user_data) est recopié depuis un fichier, contrairement aux piles Heat dans lesquelles il est inclus dans le fichier de la pile :

```
resource "openstack_compute_instance_v2" "instance_terraform" {
    name = "instance_terraform"
    flavor_name = "${var.nom_de_cle}"
    key_pair = "${var.nom_de_cle}"
    security_groups = ["${openstack_networking_secgroup_v2.secgroup_tf.name}"]
    user_data = "${file("configuration_evoluee_user_data.sh")}"
    block_device {
        uuid = "${openstack_blockstorage_volume_v2.volume_boot_terraform.id}"
        source_type = "volume"
        destination_type = "volume"
        boot_index = 0
        delete_on_termination = false
    }
}
```

Le fichier configuration_evoluee_user_data.sh provient d'une de nos piles Heat précédentes:

#!/bin/bash -v
apt-get update
apt-get install -y cowsay fortune
/usr/games/fortune | /usr/games/cowsay
/usr/games/fortune | /usr/games/cowsay > /dev/tty1
cat << EOF >> /home/ubuntu/.bashrc
/usr/games/fortune | /usr/games/cowsay
EOF

Les fichiers pour cette configuration sont disponibles à cette adresse :

- <u>https://u03.fr/90openstack/configuration_evoluee.tf</u>
- https://u03.fr/90openstack/configuration_evoluee_user_data.sh

Le fichier terraform.tfvars avec la valeur de nos éléments variable contiendra:

```
nom_de_cle = "LFO"
gabarit_instance = "ds1G"
reseau_public = "public"
type_volume = "lvmdriver-1"
```

Vous devez effacer le fichier premiere_configuration.tf, sinon Terraform va également tenter de déployer le serveur décrit dedans.



Il vous est laissé à titre d'exercice d'afficher le plan, de déployer la configuration, de vous connecter à l'instance avec l'utilisateur ubuntu (vous serez également accueilli par la vache) puis de détruire la configuration.

Note : N'oubliez pas qu'il faut un peu de temps au serveur pour se lancer et exécuter son script de démarrage.

Création et utilisation de réseaux

Tout comme avec Heat on commence par créer un réseau, puis on y attache un ou plusieurs sousréseaux.

Nous voyons dans l'exemple ci-dessous que le DNS est une variable, il est en effet important que le DNS soit adapté à notre infrastructure, sinon de nombreuses fonctionnalités risquent de ne pas fonctionner dans les instances attachées au réseau.

Les différents paramètres sont les mêmes que pour Heat, en particulier les plages d'adresses IP dynamiques pour DHCP (allocation pools):

```
variable "dns" {
   type = "list"
   description = "Adresse IP du serveur DNS"
   }

resource "openstack_networking_network_v2" "net_bastion" {
   name = "net_bastion"
   admin_state_up = "true"
   }

resource "openstack_networking_subnet_v2" "subnet_bastion" {
   name = "subnet_bastion"
   network_id = "$(openstack_networking_network_v2.net_bastion.id)"
   cidr = "10.99.1.0/24"
   allocation_pools = {
     start = "10.99.1.200"
     end = "10.99.1.254"
     ip_version = 4
   }
}
```



La variable DNS est une liste de serveurs DNS.

Il est nécessaire de spécifier type = "list" dans la déclaration de la variable.

Dans le fichier .tfvars la valeur de la varible se défini comme ceci :

dns = ["8.8.8.8", "8.8.4.4"]

Une fois les différents réseaux définis il est possible de créer le routeur pour les relier entre eux et les relier au réseau public. Là encore, le nom du réseau public doit correspondre à l'infrastructure et il s'agit donc d'une variable pour rendre notre configuration Terraform partageable.



Notez comment le serveur est rattaché à son réseau, ici une adresse fixe lui est donnée :



Script d'initialisation d'instance simple

Terraform dispose de la possibilité de faire exécuter un script au moment du lancement de l'instance (ce qui est normal, les user data étant une fonctionnalité de OpenStack).

Le cas le plus simple est l'inclusion d'un fichier externe qui sera exécuté au lancement de l'instance :

Le fichier cowsay.sh est un script shell classique :

```
#!/bin/bash -v
apt-get update
apt-get install -y cowsay fortune
/usr/games/fortune | /usr/games/cowsay
cat << EOF >> /home/ubuntu/.bashrc
/usr/games/fortune | /usr/games/cowsay
EOF
```

Script d'initialisation d'instance complexe (template)

Terraform dispose d'un système de templates qui permet, entre autres, de personnaliser le script d'initialisation des instances en remplaçant des motifs dans un fichier modèle

Un élément 'data template_file' est utilisé pour désigner le fichier modèle et faire la liaison entre les motifs et leurs valeurs de remplacement.

EOF

```
data "template_file" "userdata_bastion" {
  template = "${file("bastion.tpl")}"
  vars {
    ip_docker = "${var.ip_interne_docker}"
    ip_pio = "${var.ip_interne_pio}"
  }
}
```

Voici un exemple de fichier template 'bastion.tpl' et de fichier résultant :

```
#!/bin/bash
tee -a /etc/hosts << EOF
${ip_docker} docker
${ip_pio} pio reverseproxy
EOF</pre>
```

```
#!/bin/bash
tee -a /etc/hosts << EOF
10.10.10.1 docker
10.12.12.1 pio reverseproxy</pre>
```

La liaison avec l'instance se fait comme ceci :

```
resource "openstack_compute_instance_v2" "instance_bastion" {
    name = "bastion"
    flavor_name = "$(var.gabarit_instance_bastion)"
    user_data = "$(data.template_file.userdata_bastion.rendered)"
    network {
       [.../...]
```

Utilisation directe des API

Jusqu'ici nous avons utilisé les API de OpenStack à travers l'interface en commande de ligne, le Dashboard ou un programme externe comme Terraform

La puissance de OpenStack est l'ouverture des API.

Les API utilisent le protocole HTTPS (ou HTTP pour DevStack), le formalisme JSON en entrée et en sortie.

Les opérations offertes par l'API sont de type CRUD (Create, Read, Update, Delete), les verbes HTTP utilisés sont les suivants :

- Create : POST
- Read : GET
- Update : PUT
- Detele : DELETE

Les paramètres sont passés soit dans l'URL (path) soit dans le corps de la requête (body) au format JSON.

Obtention d'un Token et de la liste des endpoints

Nous l'avons dit, KeyStone est la pierre angulaire du système, il permet l'authentification et contient la base de données des utilisateurs, des projets et les points d'entrée pour les autres API.

La première chose à faire pour utiliser les API est de s'authentifier auprès de KeyStone, en retour on obtient un token, on peut aussi obtenir les points d'entrée pour contacter les autres API. On utilise ensuite ce token pour prouver notre identité aux autres API (attention le token obtenu a une durée de validité).

Le détail de l'API KeyStone pour l'authentification sur un domaine est disponible à l'adresse suivante :

https://developer.openstack.org/api-ref/identity/v3/#token-authentication-with-scoped-authorization

Nous utilisons curl pour dialoguer avec l'API de KeyStone, le protocole est HTTP, nous envoyons l'entête 'Content-Type' pour informer l'API que le format des données envoyées est JSON, les données à envoyer sont passées après le paramètre 'data' (et donc la méthode HTTP sera POST), nous utilisons les variables d'environnement valorisées par le script openro

,

\$05_AUTH_URL/v\$05_IDENTITY_API_VERSION/auth/tokens"; echo

La réponse est en deux parties :

- les entêtes HTTP qui vont contenir le code retour (ici '201 Created' qui signifie qu'un token a été créé), ainsi que le token après l'entête 'X-Subject-Token'
- les données qui contiennent la date d'expiration du token ainsi que les différents points d'entrée des autres API :

```
HTTP/1.1 201 Created
Date: Wed, 14 Mar 2018 21:07:53 GMT
Server: Apache/2.4.18 (Ubuntu)
X-Subject-Token:
gAAAAABarDI1U6c4sJJW8WRYQ4jJ1IYXNom8ee8wrzaVgHlQOChh0ZrpNA0VwIK60bZ3eP0F20vkACAjAFpxur15p6N3XYdDKtkICE5qE20aLFCAHseYtFpwqAQom
QCMTKV73cpckXUz16UfAKKtEmrykqKHBez181FTH2icYkp4dq4_JEj9XDQ
Vary: X-Auth-Token
Content-Type: application/json
Content-Length: 5150
x-openstack-request-id: req-67b012fd-8a25-4a42-a034-d5640f48c209
Connection: close
{"token": {"is_domain": false, "methods": ["password"], "roles": [{"id": "c4f81ba8d77b49d3a3d55712dcf42b11", "name":
"Member"}, {"id": "eld0b86682854ef1b56878f07775b3e4", "name": "anotherrole"}], "expires at": "2018-03-14T22:07:53.000002",
"otalog": [{"endpoints": [{"region_id": "RegionOne", "url":
"http://192.168.1.211/heat-api/v1/3f36cd33d14e468293caa7bf311c00de", "regionOne", "interface": "public", "id":
"secadaf6d374f54bff9faed8fc395d9"), {"region_id": "Regionne", "url": "http://192.168.1.211/heat-
[....]
"nova"]], "user": {"password_expires_at": null, "domain": ("id": "default", "name": "2018-03-
14T21:07:53.000002");
"secadaf76596a72573a8ae66f", "name": "demo"}, "audit_ids": ["tH_IWiaRTh62SWPRB9R8Fw"], "issued_at": "2018-03-
14T21:07:53.000002");
"Nova"}
```

La partie JSON n'est pas très lisible par un être humain, il est possible de le rendre lisible humainement, nous devons supprimer l'affichage des entêtes pour ceci (et donc le token ne sera pas visible).

Nous voyons au passage le endpoint du service 'cinderv3' ('volumev3') qui gère les volumes, celui de 'neutron' qui gère le réseau ('network') et du service 'nova' ('compute') qui gère les instances:

```
stack@microwave:~$ curl --silent \
      --request POST \
     --header "Content-Type: application/json" \
      --data
  "auth": {
    methods : [ password ],
"password": { "user": { "domain": { "id": "'$OS_PROJECT_DOMAIN_ID'" },
"name": "'$OS_USERNAME'",
"password": "'$OS_PASSWORD'" } }
     "scope": {
        "project": { "name": "'$0S_PROJECT_NAME'", "domain": { "id": "'$0S_PROJECT_DOMAIN_ID'" } }
    1
     "$05_AUTH_URL/v$05_IDENTITY_API_VERSION/auth/tokens" | python -m json.tool | pygmentize -1 json
    "token": {
         "audit ids": [
              "lg6y1QFKT0Czcy2601wjKA"
         "catalog": [
   [.../...)
             ł
                  "endpoints": [
                             'id": "17069516fedb40cc8e44f15ea4b02dba",
                            "interface": "public",
"region": "RegionOne",
                            "region id": "RegionOne"
```



L'URL du endpoint de Cinder renvoyée par KeyStone est :

http://192.168.1.211/volume/v3/3f36cd33d14e468293caa7bf311c00de

La fin de l'URL du endpoint de Cinder est l'identifiant du projet :

stack@microwave	e:~\$ openstack project show demo	,
Field	Value	
description	/ I	+
domain_id	default	
enabled	True	1
id	3f36cd33d14e468293caa7bf311c00de	e I
is_domain	False	1
name	demo	1
parent_id	default	1
tags	[]	1
+	+	+

Utilisation de l'API Cinder : Obtenir la liste des volumes

Nous allons utiliser l'API Cinder pour obtenir la liste des volumes de notre projet, à titre de comparaison la liste des volumes obtenue en utilisant le CLI est la suivante :

<pre>stack@microwave:~\$ openstack volume list</pre>	±			,	,
ID	Name	Status	Size	Attached to	
	donnees ubuntu-16.04_vol	available in-use 	5	Attached to ubuntu-16.04_inst on /dev/vda	-+ -+

Il s'agit d'un des cas les plus simples d'utilisation d'une API, la documentation de cette API est disponible à cette adresse :

<u>https://developer.openstack.org/api-ref/block-storage/v3/index.html#list-accessible-volumes-with-details</u>

Pour obtenir la liste des volumes de notre projet il nous suffit de faire un 'GET' en HTTP et d'utiliser comme URL celle du endpoint retournée par KeyStone (qui contient l'id de notre projet) et de rajouter '/volumes/detail' à la fin :

http://192.168.1.211/volume/v3/3f36cd33d14e468293caa7bf311c00de/volumes/detail

Nous récupérons le token KeyStone en appelant l'API KeyStone et nous le plaçons dans la variable MON_TOKEN, puis nous appelons l'API de Cinder en utilisant le endpoint renvoyé par KeyStone en même temps que le token, nous ajoutons un entête HTTP 'X-Auth-Token' :

```
stack@microwave:~$ export MON_TOKEN=gAAAAABarDI1U6c4sJJW8WRYQ4jJ1 [.../..] qKHBezi8iFTHZicYkp4dq4_JZj9XDQ
stack@microwave:~$ curl --silent \
                 --request GET \
--header "X-Auth-Token: $MON_TOKEN" \
                 "http://192.168.1.211/volume/v3/3f36cd33d14e468293caa7bf311c00de/volumes/detail" | python -m json.tool
      "volumes":

→ PREMIER VOLUME

                 "attachments": [],
                 "availability_zone": "nova",
"bootable": "false",
                 "consistencygroup_id": null,
"created_at": "2018-03-03T16:07:40.000000",
"description": "",
                  "encrypted": false,
                            '609fd47f-0c07-46b6-925a-46e8abadeb89",
                  <mark>"id": "609:</mark>
"links": [
                             "href": "http://192.168.1.211/volume/v3/3f36cd33d14e468293caa7bf311c00de/volumes/609fd47f-0c07-46b6-925a-
46e8abadeb89",
                             "rel": "self"
                       },
                             "href": "http://192.168.1.211/volume/3f36cd33d14e468293caa7bf311c00de/volumes/609fd47f-0c07-46b6-925a-
46e8abadeb89",
                             "rel": "bookmark"
                      }
                  "metadata": {
                        "readonly": "False"
                  "multiattach": false,
                  "name": "donnees",
"os-vol-tenant-attr:tenant_id": "3f36cd33d14e468293caa7bf311c00de",
"replication_status": null,
                 "size": 5,
"snapshot_id": null,
"source_volid": null,
"status": "available",
"updated_at": "2018-03-04T21:15:53.000000",
"user_id": "cf025806abdb476f936a72573a8ae66f",
"volume_type": "lvmdriver-1"
                  "size": 5,
            },
    SECO
                VOLUME
                  "attachments": [
                             "attached_at": "2018-02-18T16:46:57.000000",
                             "attached_at": "2018-02-18716:46:37.000000",

"attachemen_id": "75c0ad36-7aa9-475d-b830-ec2d6a93123b",

"device": "/dev/vda",

"host_name": null,

"id": "417394d7-dc15-48ec-9f22-09b034019459",

"server_id": "39ea09c2-659a-4b2a-9552-71fcee89ef4a", ⇒ ID du serveur ubuntu-16.04_inst

"volume_id": "417394d7-dc15-48ec-9f22-09b034019459"
                       ł
                  "availability_zone": "nova",
"bootable": "true",
                  "consistencygroup_id": null,
"created_at": "2018-02-18T16:08:57.000000",
                  "description": null,
                 "encrypted": false,
"id": "417394d7-dc15-48ec-9f22-09b034019459",
                 "links": [
                             "href": "http://192.168.1.211/volume/v3/3f36cd33d14e468293caa7bf311c00de/volumes/417394d7-dc15-48ec-9f22-
096034019459".
                             "rel": "self"
                       },
```

"href": "http://192.168.1.211/volume/3f36cd33d14e468293caa7bf311c00de/volumes/417394d7-dc15-48ec-9f22-
095034019459",
"rel": "bookmark"
}
],
"metadata": {
"attached_mode": "rw",
"readonly": "False"
1,
"multiattach": false,
"name": "ubuntu-16.04_vol",
"os-vol-tenant-attr:tenant_id": "3f36cd33d14e468293caa7bf311c00de",
"replication_status": null,
<mark>"size": 10</mark> ,
"snapshot_id": null,
"source_volid": null,
"status": "in-use",
"updated_at": "2018-02-18T16:46:57.000000",
"user_id": "cf025806abdb476f936a72573a8ae66f",
"volume_image_metadata": {
"checksum": "9cb8ed487ad8fbc8b7d082968915c4fd",
"container_format": "bare",
"disk_format": "qcow2",
"image_id": "0885d2ea-9445-4a6a-ab45-61ecd42e01d8",
"image_name": "ubuntu-16.04",
"min_disk": "0",
"min_ram": "0",
"size": "289603584"
},
"volume_type": "lvmdriver-1"
J.

Si vous utilisez un token incorrect ou expiré vous allez avoir une erreur d'authentification
(HTTP 401):
{
 "error": {
 "code": 401,
 "message": "The request you have made requires authentication.",
 "title": "Unauthorized"
 }
}

Utilisation de l'API Cinder : Création d'un volume

Nous allons créer un volume à partir d'une image, comme nous l'avions fait en utilisant le CLI OpenStack.

La documentation de l'API correspondante est disponible à cette adresse :

• https://developer.openstack.org/api-ref/block-storage/v3/index.html#create-a-volume

Cette opération est réalisée par un ordre POST, le nom du projet est passé dans l'URL (path) de la même façon que nous avions obtenu la liste des volumes, les autres paramètres sont passés dans le corps de la requête au format JSON.

Nous commençons par obtenir l'id de l'image Ubuntu à partir de laquelle nous allons créer le volume, nous pouvons utiliser le CLI OpenStack pour bien sûr utiliser l'API pour obtenir cette information :



"min_ram": 0, "name": "ubuntu-16.04", "owner": "f5863f4d4266400cbb2562001ea22eba", [.../...]

Nous allons créer un volume de 10Go, qui aura pour nom 'ubuntu_volapi', tout comme lorsque nous avions créé le volume en utilisant le CLI openstack nous voyons que le statut du volume est 'creating' et qu'il n'est pas bootable.



Nous interrogeons l'API d'OpenStack pour vérifier quand notre volume est disponible, on note au passage qu'il devient bootable, ce qui nous permettra de lancer une instance à partir de ce volume.

• https://developer.openstack.org/api-ref/block-storage/v3/index.html#show-a-volume-s-details

stack@microwave:~\$ curlsilent \request GET \header "X-Auth-Token: \$MON_TOKEN" \
"http://192.168.1.211/volume/v3/3f36cd33d14e468293caa7bf311c00de/volumes/ <mark>160ce17d-df05-43b5-b262-0327f3d0d994</mark> " \
python -mjson.tool
<pre>{ "volume": { </pre>
"attachments": [],
"availability_zone": "nova",
"bootable": "true",
"consistencygroup id": null,
"created at": "2018-03-17117:24:55.000000",
"description": null,
"encrypted": false,
"id": "160ce17d-df05-43b5-b262-0327f3d0d994",
"links": [
ℓ
"href": "http://192.168.1.211/volume/v3/3f36cd33d14e468293caa7bf311c00de/volumes/160ce17d-df05-43b5-b262-
0327f3d0d994",
"rel": "self"



Utilisation de l'API Compute: Création d'une instance

Nous allons utiliser le volume que nous venons de créer pour créer une instance, nous avons déjà l'id du volume, il nous faut l'id d'un gabarit (flavor), l'id de notre réseau privé ainsi que le nom d'une clé publique.

L'API pour obtenir la liste des gabarits est la suivante :

• <u>https://developer.openstack.org/api-ref/compute/#list-flavors</u>



Il vous est laissé à titre d'exercice d'utiliser l'API compute pour récupérer l'id du gabarit 'm1.small'

Indice: Il vous faut le endpoint de l'API 'compute'.

La description de l'API pour obtenir les détails d'un gabarit est la suivante :

<u>https://developer.openstack.org/api-ref/compute/#show-flavor-details</u>



Il est également nécessaire de spécifier à quel réseau connecter le serveur, nous allons le connecter au réseau privé ('private'), nous devons obtenir son id grâce à l'API Neutron :

• <u>https://developer.openstack.org/api-ref/network/v2/index.html#list-networks</u>

[/]		
	"id": "8e7ebaa8-a44e-4a94-9060-9d5fc6143ddf",	
	<pre>"name": "private",</pre>	
[/]		
}		

La dernière information importante est le nom de la clé publique à utiliser :

<u>https://developer.openstack.org/api-ref/compute/#list-keypairs</u>

stack@microwave:~\$ curlsilent \
request GET \
header "X-Auth-Token: \$MON TOKEN" \
"http://192.168.1.211/compute/v2.1/os-keypairs" python -mjson.tool
"keypairs": [
"keypair": {
"fingerprint": "8c:56:76:56:d9:de:50:81:8b:4d:6c:b1:ee:55:b2:ae",
"name": "Foucher Laurent RSA",
"public key": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAACAQDS jn EGxaX1FsDqp6x2TrYACUWCQqHiq5YGtfdZ501z7F5ic9z57Cf5zYqRme40WUpB+6JW8Qv6Jbw8mswyTLjqYGxh97Hp3
5vq3SBIrVfEoGw7jYwRmo2Xf5zMx1orxNcTUnvSGRusoY6CdR28129sfzsIxV7Zh9EMisDt61Xra+CdescSOG3UNiQ0KcIoS24N/
zyImdkXkxrDeeqHPiw9FFNpAbFnwNQx7VCwiwJqjII2wVSYYYBcnpTRzqU8PL0HSU9xcDKn5n9iuD8pFAieF1nSP6k2e6E7hOuPEkIkSkemUedUS6t2BS+pUZqq4b
bMDcSCz70L0acjvvdHmlkcdjNJzPsEFdqZ7eKBTwkOFS3ot7brpRtG8AMjDKx0KQHUawm8S2vPW3RjhD4QB2osPpvmokKLyvP4mZv0II3RN9qa69nwntKEBGUbc1A
wBEQWUE010MhknlymVbABtbm11054X8R940LTyVfXC92RXL/F3j5iJB91WtUwf50DLB96rp09I4IJI31oEkW7r0cV2NbvusYSZBaD/
PVZoV9h5i2oz8UospaGHjoce4cJG+ZOLKB/1UM778rXVL/
oFFm2TRMEOH4x2yBq0EyJo4o+cOquWhj29FFCB2yy1pWhpi42Nf80qXM0d7jKxwaKkS7R7ZdM1eAJ44B/qV57Msqiiw== Foucher, Laurent\n"

Nous avons maintenant toutes les informations nécessaires pour créer une instance,

La description de l'API qui permet de créer une instance est disponible à cette adresse :

https://developer.openstack.org/api-ref/compute/#create-server

```
stack@microwave:~$ curl --silent \
       --request POST \
      --header "X-Auth-Token: $MON_TOKEN" \
      --header "Content-Type: application/json" \
--data '
{"server": {
     "name": "ubuntu_instapi",
"flavorRef": "2",
     "networks": [{
"uuid": "8e7ebaa8-a44e-4a94-9060-9d5fc6143ddf"
     }],
     "key_name": "Foucher Laurent RSA",
     "block device_mapping_v2": [{
       "source_type": "volume",
       "destination_type": "volume",
"boot_index": "0",
       "uuid": "160ce17d-df05-43b5-b262-0327f3d0d994"
    }]
}
}'
   ١
 "http://192.168.1.211/compute/v2.1/servers" | python -mjson.tool
     "server": {
         "OS-DCF:diskConfig": "MANUAL",
"adminPass": "4d8L5QtyWvAV",
         "id": "564754fa-3bf7-44fb-96ce-41a2cf9f6ee2",
"links": [
             ł
                  "href": "http://192.168.1.211/compute/v2.1/servers/564754fa-3bf7-44fb-96ce-41a2cf9f6ee2",
                  "rel": "self"
              },
              {
                   "href": "http://192.168.1.211/compute/servers/564754fa-3bf7-44fb-96ce-41a2cf9f6ee2",
                   "rel": "bookmark"
              ł
          "security_groups": [
             {
                  "name": "default"
              ł
         ]
     }
```

Nous utilisons l'API compute pour vérifier si notre serveur est disponible :

https://developer.openstack.org/api-ref/compute/#show-server-details



Il vous est laissé à titre d'exercice d'utiliser les API pour associer une adresse IP flottante au serveur, le groupe de sécurité qui nous avions créé et qui autorise les connexions en SSH

Vous devriez ensuite être capable de vous connecter à votre serveur en SSH.

Utilisation du client CLI openstack en mode debug

Vous pouvez utiliser le client CLI openstack en mode debug pour avoir la liste des appels à l'API avec leurs paramètres et les données renvoyées par l'API.

Ceci peut vous permettre de récupérer les appels API nécessaires à vos projets (notez au passage que le token n'est jamais affiché en clair en mode debug, seulement son hash SHA-1) :

stack@microwawa:a\$ openstackdobug co	mor list				
State with optimer (ul-adobud ulored ulort)					
In///// RRO: curl -g -i -X GRT http://192.168.1.211/compute/v2.1/servers/detail -H "User-Agent: python-novaclient" -H "Accent:					
annlication/ison"-H_"X-Auth-Taken: (SHA1)2hc2ff4fr2075Ch5d796fahe72c2d5885haPece"					
http://192.168.1.211:80 "GRT /compute/v2.1/servers/detail HTTP/1.1" 200 3196					
RESP: [200] Date: Sun. 18 Mar 2018 13:25:20 GMT Server: Apache/2.4.18 (Ubuntu) Content-Length: 3196 Content-Type:					
application/ison OpenStack-API-Version: compute 2.1 X-OpenStack-Nova-API-Version: 2.1 Varv: OpenStack-API-Version.X-					
OpenStack-Nova-API-Version x-openstack-request-id: req-3f13e308-890e-4681-ba83-d66e2e188800 x-compute-request-id: req-					
3f13e308-890e-4681-ba83-d66e2e188800 Connection: close					
RESP BODY: {"servers": [{"OS-EXT-STS:task state": null, "addresses": {"private": [{"OS-EXT-IPS-MAC:mac addr":					
"fa:16:3e:9d:db:6c", "version": 4, "addr": "10.0.0.7", "OS-EXT-IPS:type": "fixed"), ("OS-EXT-IPS-MAC:mac addr":					
"fa:16:3e:9d:db:6c", "version": 6, "addr": "fdb0:453a:3cd:0:f816:3eff:fe9d:db6c", "OS-EXT-IPS:tvpe": "fixed"}, {"OS-EXT-IPS-					
MAC:mac addr": "fa:16:3e:9d:db:6c", "version": 4, "addr": "172.24.4.13", "OS-EXT-IPS:type": "floating")]}, "links": [{"href":					
"http://192.168.1.211/compute/v2.1/servers/564754fa-3bf7-44fb-96ce-41a2cf9f6ee2", "rel": "self"}, {"href":					
"http://192.168.1.211/compute/servers/564754fa-3bf7-44fb-96ce-41a2cf9f6ee2", "rel": "bookmark")], "image": "", "OS-EXT-					
STS:vm state": "active", "OS-SRV-USG:launched at": "2018-03-18711:28:35.000000", "flavor": {"id": "2", "links": [{"href":					
"http://192.168.1.211/compute/flavors/2", "rel": "bookmark"}]}, "id": "564754fa-3bf7-44fb-96ce-41a2cf9f6ee2",					
[/]					
+	+	+	+	-+	+
ID	Name	Status	Networks	Flavor	1
+	+	+	+	-+	+
564754fa-3bf7-44fb-96ce-41a2cf9f6ee2	ubuntu_instapi	ACTIVE	private=10	m1.small	1
39ea09c2-659a-4b2a-9552-71fcee89ef4a	ubuntu-16.04_inst	ACTIVE	private=10	m1.small	1
+	+	+	+	-+	+
clean_up ListServer:					
END return value: 0					

Conclusion

J'espère que cette introduction à l'utilisation de OpenStack vous a plu.

N'oubliez pas de visiter régulièrement mon blog : <u>https://blog.u03.fr/</u>

N'hésitez pas à me faire part de vos remarques, soit en commentant sur mon blog ou directement par email : <u>u03@u03.fr</u>