



**IN2P3**

Institut national de **physique nucléaire**  
et de **physique des particules**

# Installation d'OpenStack

Jérôme PANSANEL

Ecole Informatique de l'IN2P3  
1 – 5 juillet 2014

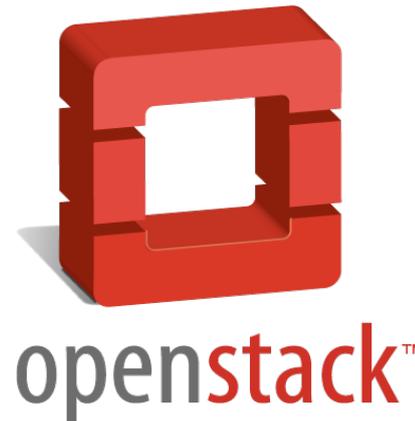
- OpenStack – Introduction et présentation des modules
- Les différentes méthodes d'installation
- Feuille de route
- Préparation de l'installation
- Exemple d'installation
- Documentation

# OpenStack – Introduction et présentation des modules

---

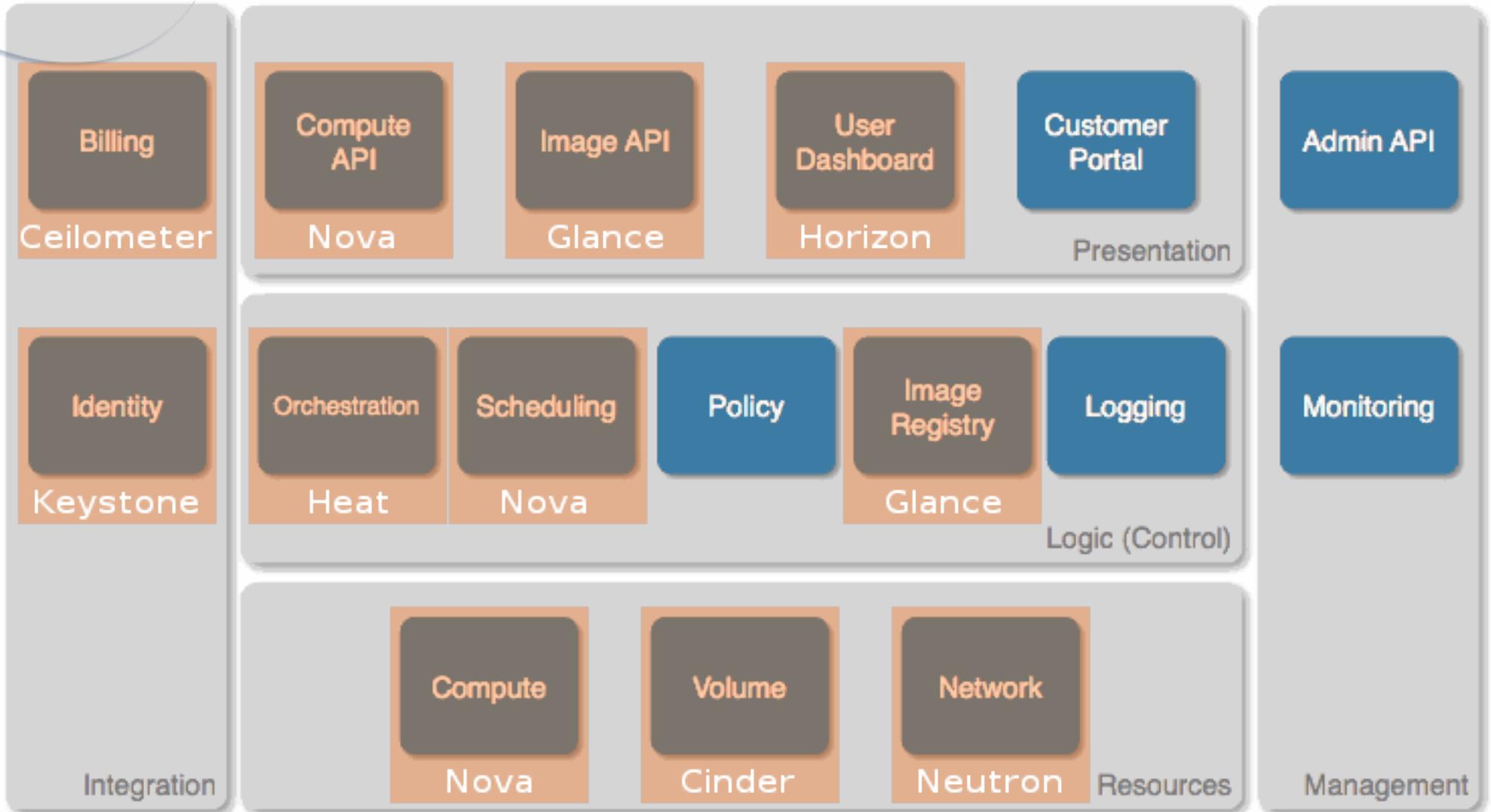
OpenStack en quelques mots :

- Project collaboratif de développement d'un gestionnaire de Cloud IaaS initié en 2010
- Utilisé dans plus de 130 pays et par plus de 14.000 personnes
- Chaque mois, plus de deux cents contributeurs différents
- Plus de deux cents sociétés impliquées dans le développement d'OpenStack
- Développé en Python
- Communauté ouverte
- Licence libre (licence Apache 2.0)
- API Java, Node.js, Ruby, .NET, PHP et Python



Plus d'informations sur : <http://www.openstack.org/>

# OpenStack – Introduction et présentation des modules



# OpenStack

## Présentation du module Nova

---

Nova est le cœur d'OpenStack et est par conséquent l'un des composants le plus complexe. Il comporte plusieurs sous-modules ayant chacun une fonction bien précise :

- `nova-api` : ce démon gère les appels API de l'utilisateur. Il supporte l'API native d'OpenStack ainsi que l'API EC2 d'Amazon. Il initie également le démarrage des machines virtuelles et vérifie que certaines règles sont bien respectées (quotas)
- `nova-compute` : ce démon tourne sur les serveurs hôtes. Il gère le cycle de vie des machines virtuelles via l'API de l'hyperviseur (XenAPI pour XenServer/XCP, libvirt pour KVM ou QEMU, VmwareAPI pour VMWare, etc).
- `nova-network` : ce composant de Nova gère les réseaux. Depuis l'arrivée de Quantum et Neutron, ce composant est facultatif et devrait devenir obsolète dans un avenir proche.

# OpenStack

## Présentation du module Nova

---

- `nova-schedule` : ce composant s'occupe de récupérer les demandes de création de machines virtuelles en queue et de déterminer sur quelle machine hôte chaque nouvelle instance doit s'exécuter. Le gestionnaire de queue est un système centralisé pour passer les messages entre les démons. Elle est actuellement implémentée avec RabbitMQ ou Qpid. Il s'agit de l'un des composants le plus simple d'OpenStack OpenStack.
- Nova propose également un service permettant aux utilisateurs d'accéder à la console de leur machine virtuel. Ce service repose sur plusieurs démons (`nova-console`, `nova-vncproxy` et `nova-consoleauth`).

Le module Nova, comme la majorité des modules composant OpenStack, dépend d'une base de données SQL (sqlite3, MySQL et PostgreSQL). Elle permet de stocker les types d'instance disponibles, les instances en cours d'exécution, ...

# OpenStack

## Présentation du module Keystone

---

Le module Keystone gère les autorisations, l'authentification, le catalogue de services et les tokens dans OpenStack. Il est conçu de manière modulaire grâce à l'utilisation de plugins. Ils permettent, par exemple, d'utiliser différents types pour l'authentification (LDAP, VOMS, ...).

# OpenStack

## Présentation du module Glance

---

Glance est le module gérant le catalogue d'images. Il se compose de quatre parties principales :

- `glance-api` : ce démon traite les appels à l'API pour la gestion des images. Il permet notamment de lister les images disponibles, récupérer une image ou en créer une nouvelle.
- `glance-registry` : ce démon stocke, traite et récupère les méta-données associées aux images (taille, type, requis, ...).
- Base de données : la base de données est utilisée pour stocker les méta-données.
- Un répertoire de stockage : c'est dans ce répertoire qu'est stocké les fichiers d'image. Ce répertoire peut être un point de montage d'un système de fichier distribué (GlusterFS, Ceph, ...).

# OpenStack

## Présentation du module Cinder

---

Le composant OpenStack Cinder a pour rôle de gérer le stockage permanent. Cette fonction était précédemment incluse dans Nova, via le composant nova-volume. Il permet ainsi de créer, modifier et supprimer les volumes, de gérer les types de volume et les snapshots. Il repose sur plusieurs briques :

- `cinder-api` : ce démon accepte les requêtes API et les transmet à `cinder-volume` pour exécution.
- `cinder-volume` : c'est le cœur de Cinder. Il réceptionne les demandes de `cinder-api` et interagit avec la base de données et les autres process (tel que `cinder-scheduler`). Il fonctionne avec de nombreuses solutions de stockage (linux iSCSI, NetApp, ...).
- `cinder-scheduler` : comme `nova-scheduler`, il sélectionne le stockage le plus adapté pour créer le volume.

Comme pour les autres composants, Cinder repose sur l'utilisation d'une base de données SQL et sur AMQP.

# OpenStack

## Présentation du module Neutron

---

Neutron est le module gérant le réseau « en tant que service ». Ce service permet aux utilisateurs de créer et gérer leur propre réseau et de se connecter à différents types d'architecture grâce aux différents plugins :

- Brocade
- Juniper
- Linux bridging
- NEC OpenFlow
- Open vSwitch
- Switch Cisco
- ...

Le serveur Neutron gère les appels à l'API et transmet les actions à réaliser aux différents agents Neutron.

# OpenStack

## Projets complémentaires

---

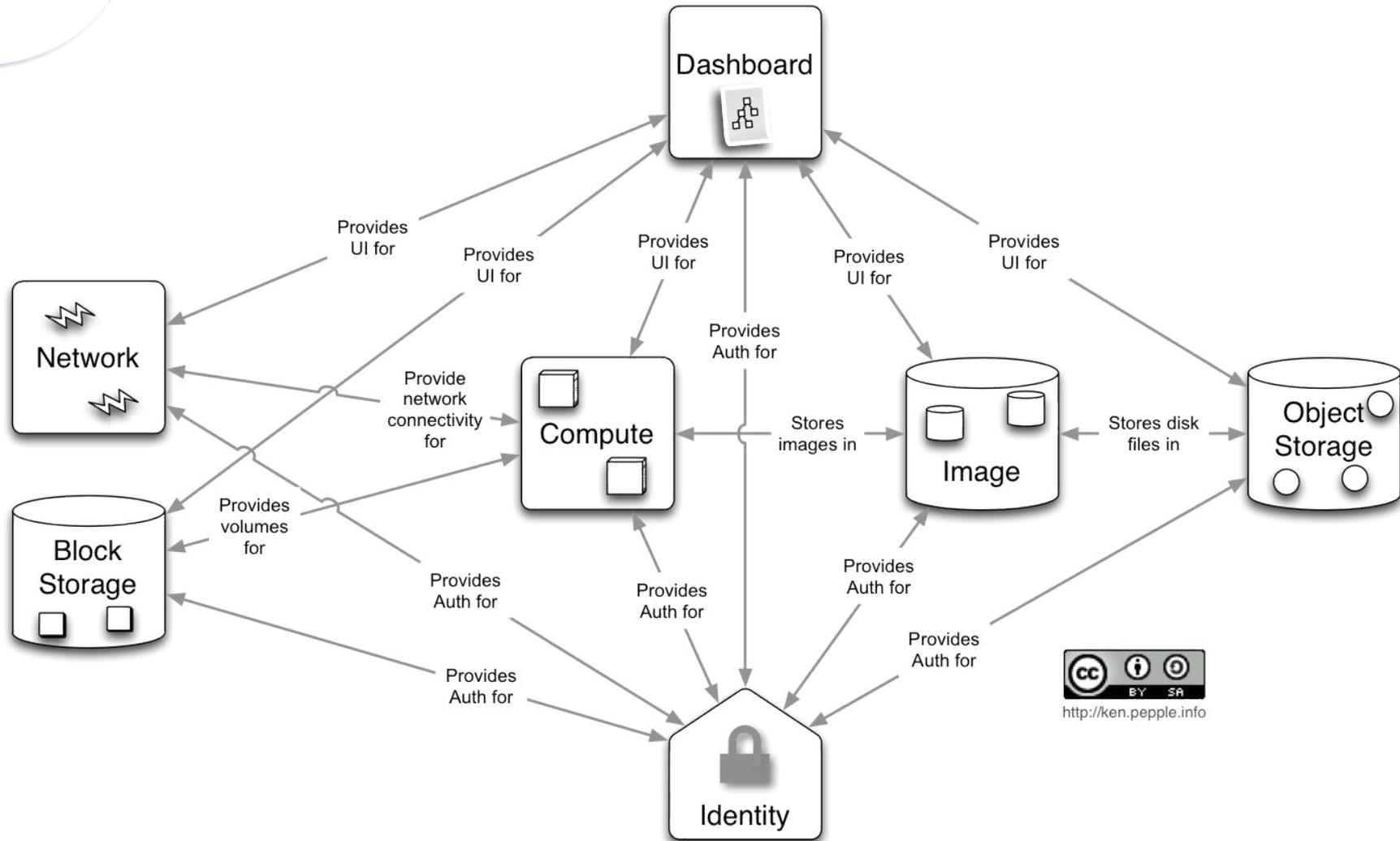
Les principaux projets complémentaires sont :

- Horizon – le tableau de bord
- Swift – le stockage d'objets
- Ceilometer – le service de métrologie
- Heat – API REST pour l'orchestration de plusieurs Cloud
- Trove – service de base de données à la demande

En gestation :

- Ironic : service pour le déploiement *Bare Metal*
- TripleO : service de déploiement de cloud OpenStack grâce à OpenStack (*OpenStack On OpenStack*)
- Marconi : service de messagerie Cloud et de notification
- Sahara : service d'hadoop à la demande

# OpenStack – Introduction et présentation des modules



# Les méthodes d'installation Mirantis / Fuel

---

- Distribution OpenStack avec outil de déploiement Fuel (base puppet)
- Modules testés par Mirantis
- Documentation
- Support Commercial
- Agnostique sur la distribution utilisée (Ubuntu, RedHat)

## Quelques liens :

- <http://docs.mirantis.com/openstack/fuel/fuel-5.0/>
- <https://github.com/Mirantis>
- <https://github.com/stackforge/fuel-library/>

# Les méthodes d'installation Ubuntu OpenStack

---

- Distribution OpenStack testée
- Support commercial
- Documentation
- Système de déploiement automatisé basé sur MAAS et Juju
- Lié au système d'exploitation Ubuntu
- Solution de management basée sur Ubuntu Advantage (commercial)
- Ubuntu 14.04 LTS / IceHouse

## Quelques liens :

- <http://www.ubuntu.com/cloud/ubuntu-openstack>
- <http://www.ubuntu.com/download/cloud>
- <http://www.ubuntu.com/cloud/tools/maas>
- <http://www.ubuntu.com/cloud/tools/juju>

# Les méthodes d'installation PackStack

---

- Système de déploiement automatisé pour OpenStack  
Solution disponible pour différentes versions d'OpenStack
- Basé sur Puppet
- Déploiement en moins de trente minute pour les configurations all-in-one
- Configuration limité
- Pour RedHat et dérivés

## Quelques liens :

- <https://wiki.openstack.org/wiki/Packstack>
- <https://github.com/stackforge/packstack> (branche havana)
- <http://www.blog.sandro-mathys.ch/2013/08/install-rdo-havana-2-on-fedora-19-and.html>

# Les méthodes d'installation RedHat OpenStack

---

- Distribution RedHat et dérivés
- Support IceHouse
- Répertoire logiciel RDO ouvert
- Support commercial
- Permet d'utiliser PackStack pour le déploiement (facultatif)

## Quelques liens :

- [http://openstack.redhat.com/Main\\_Page](http://openstack.redhat.com/Main_Page)
- <http://openstack.redhat.com/Docs>
- <http://openstack.redhat.com/forum/>
- <http://openstack.redhat.com/Quickstart>

# Feuille de route

---

- Sélection des modules à installer
- Définition de l'architecture matérielle
- Déploiement du système d'exploitation sur les OS
- Installation et vérification des pré-requis
- Installation de keystone
- Installation de glance
- Installation de nova
- Installation de cinder
- Installation de neutron
- Tests

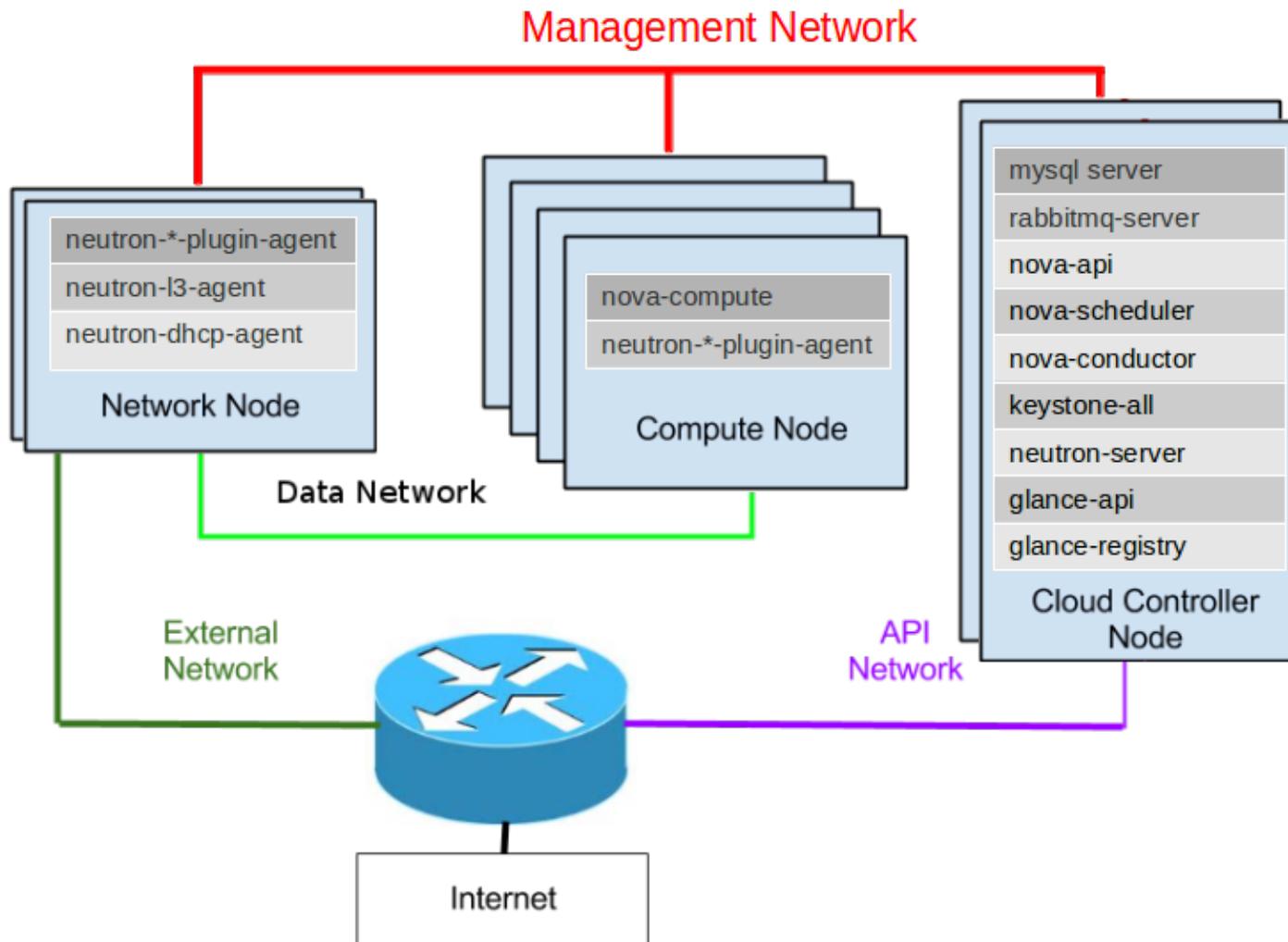
# Préparation de l'installation : Définition des objectifs

---

- Déployer une solution OpenStack
- OpenStack version Havana
- Trois nœuds : Cloud Controller, Network et Compute Nodes
- Services déployés : nova, keystone, glance, cinder
- Service réseau géré avec neutron
- OS : Scientific Linux 6.5 (ou équivalent)
- Utilisation de RDO

→ <http://docs.openstack.org/havana/install-guide/install/yum/content/>

# Préparation de l'installation : Définition de l'infrastructure



# Préparation de l'installation : Vérification des pré-requis

---

Pour tous les nœuds :

- Scientific 6.5
- Dépôt EPEL activé
- Python 2.6 ou 2.7
- ntp
- Service network-manager désactivé
- Au moins deux interfaces réseaux (trois pour le network node)

Pour le Cloud Controller Node :

- Serveur MySQL
- Serveur RabbitMQ (ou Qpid)
- Avoir un volume LVM dénommé `cinder-volumes`
- Paquet `scsi-target-utils` et service `tgtd` configuré

# Préparation de l'installation : Installation de RDO

---

## Installation du dépôt RDO :

```
# wget http://repos.fedorapeople.org/repos/openstack/openstack-havana/rdo-  
release-havana-8.noarch.rpm  
# yum localinstall rdo-release-havana-8.noarch.rpm  
# yum update -y
```

# Exemple d'installation

## Installation de keystone

---

- Installation des paquets `openstack-keystone` et `python-keystoneclient`
- Configuration de l'accès à la base de données
- Initialisation de la base de données *keystone*
- Configuration des tokens PKI
- Démarrage du démon `openstack-keystone`
- Configurer l'exécution de ce service au démarrage du serveur
- Configuration des tenants, utilisateurs (administrateur, services, utilisateur standard) et rôles

```
# keystone service-create --name=keystone --type=identity \  
--description="Keystone Identity Service"
```

- Définition des points d'accès

```
# keystone endpoint-create \  
--service-id=the_service_id_above \  
--publicurl=http://controller:5000/v2.0 \  
--internalurl=http://controller:5000/v2.0 \  
--adminurl=http://controller:35357/v2.0
```

- Vérification de l'installation

# Exemple d'installation

## Installation de glance

---

- Installation du paquet `openstack-glance`
- Configuration de l'accès à la base de données
- Initialisation de la base de données *glance*
- Configuration des paramètres dans `/etc/glance/glance.conf`
- Configuration de l'accès à keystone et des points d'accès
- Démarrage des démons :  
`openstack-glance-api`  
`openstack-glance-registry`
- Configurer l'exécution de ces services au démarrage du serveur
- Vérification de l'installation

# Exemple d'installation

## Installation de nova (controller)

---

- Installation des paquets `openstack-nova` et `python-novaclient`
- Configuration de l'accès à la base de données
- Configuration des paramètres dans le fichier `/etc/nova/nova.conf`
- Initialisation de la base de données *nova*
- Configuration de l'accès à RabbitMQ
- Configuration de l'accès à keystone et des points d'accès
- Démarrage des démons :
  - `openstack-nova-api`
  - `openstack-nova-cert`
  - `openstack-nova-consoleauth`
  - `openstack-nova-scheduler`
  - `openstack-nova-conductor`
  - `openstack-nova-novncproxy`
- Configurer l'exécution de ces services au démarrage du serveur
- Vérification de l'installation

# Exemple d'installation

## Installation de nova (compute)

---

- Installation du paquet `openstack-nova-compute`
- Configuration de l'accès à la base de données
- Configuration des paramètres dans le fichier `/etc/nova/nova.conf`
- Configuration de l'accès à keystone et des points d'accès
- Démarrage du démon :  
`openstack-nova-compute`
- Configurer l'exécution de ces services au démarrage du serveur
- Vérification de l'installation

# Exemple d'installation

## Installation de cinder

---

- **Installation des paquets** `openstack-cinder`, `openstack-utils` et `openstack-selinux`
- **Configuration de l'accès à la base de données**
- **Initialisation de la base de données *cinder***
- **Configuration des paramètres dans le fichier**  
`/etc/cinder/cinder.conf`
- **Configuration de l'accès à keystone et des points d'accès**
- **Configuration de** `/etc/tgt/targets.conf`
- **Démarrage des démons :**  
`openstack-cinder-api`  
`openstack-cinder-scheduler`  
`openstack-cinder-volumes`
- **Configurer l'exécution de ces services au démarrage du serveur**
- **Vérification de l'installation**

# Exemple d'installation

## Installation de neutron

---

*This chapter is a bit more adventurous than we would like. We are working on cleanup and improvements to it. Like for the rest of the Installation Guide, feedback through bug reports and patches to improve it are welcome.*

# Exemple d'installation

## Installation de neutron (network)

---

- Configuration des plugins Neutron (Open vSwitch)
- Configuration de l'accès à la base de données
- Configuration de l'accès à keystone et des points d'accès
- Installation du paquet `openstack-neutron`
- Démarrage des démons :  
`neutron-dhcp-agent`  
`neutron-metadata-agent`  
`neutron-l3-agent`
- Configurer l'exécution de ces services au démarrage du serveur
- Autoriser le transfert de paquet (`/etc/sysctl.conf`) :  
`net.ipv4.ip_forward=1`  
`net.ipv4.conf.all.rp_filter=0`  
`net.ipv4.conf.default.rp_filter=0`
- Configuration du plugin OpenVSwitch

# Exemple d'installation

## Installation de neutron (compute)

---

- Configuration des plugins Neutron (Open vSwitch)
- Installation du paquet `openstack-neutron-openvswitch`
- Autoriser le transfert de paquet (`/etc/sysctl.conf`) :  
`net.ipv4.conf.all.rp_filter=0`  
`net.ipv4.conf.default.rp_filter=0`
- Configurer nova pour prendre en compte la configuration de Neutron (`/etc/nova/nova.conf`)
- Bug avec de nombreuses cartes :  
`ovs-vsctl set interface eth1 other-config:enable-vlan-splinters=true`
- Redémarrer le service :  
`openstack-nova-compute`
- Redémarrer le service :  
`neutron-openvswitch-agent`

# Exemple d'installation

## Installation de neutron (controller)

---

- **Installation du paquet** `openstack-neutron`, `python-neutron` **et** `python-neutronclient`
- **Configuration des plugins Neutron (Open vSwitch)**
- **Configuration de Nova pour l'utilisation de Neutron et redémarrage des services :**  
`openstack-nova-api`  
`openstack-nova-scheduler`  
`openstack-nova-conductor`
- **Configuration de Neutron via le fichier**  
`/etc/neutron/neutron.conf`
- **Redémarrage du service**  
`neutron-server`
- **Vérification de l'installation**

# Ouverture de ports

---

Ports à ouvrir :

- 5000/tcp vers Cloud Controller Node (keystone)
- 8773/tcp vers Cloud Controller Node (nova-api)
- 8774/tcp vers Cloud Controller Node (nova-api)
- 8775/tcp vers Cloud Controller Node (nova-api)
- 8776/tcp vers Cloud Controller Node (cinder-api)
- 9292/tcp vers Cloud Controller Node (glance-api)
- 9696/tcp vers Cloud Controller Node (neutron-api)

## Documentation en ligne

- [https://wiki.openstack.org/wiki/Main\\_Page](https://wiki.openstack.org/wiki/Main_Page)
- <http://docs.openstack.org/user-guide/content/>
- <https://ask.openstack.org/en/questions/>

## Listes de diffusion

- [https://wiki.openstack.org/wiki/Mailing\\_Lists](https://wiki.openstack.org/wiki/Mailing_Lists)
- <http://lists.openstack.org/cgi-bin/mailman/listinfo/openstack-fr>

## Canal IRC

- IRC : Freenode / #openstack-fr

## Guide d'installation :

- **Ubuntu 12.04 :**  
<http://docs.openstack.org/havana/install-guide/install/apt/content/>
- **RedHat 6.5 et dérivés :**  
<http://docs.openstack.org/havana/install-guide/install/yum/content/>
- **Debian 7.0 :**  
<http://docs.openstack.org/havana/install-guide/install/apt-debian/content/>

## Exemple de configuration pour les différents modules OpenStack :

- <http://docs.openstack.org/havana/config-reference/content/>

## Guide pour les administrateurs :

- <http://docs.openstack.org/admin-guide-cloud/content/>
- <http://docs.openstack.org/security-guide/content/>

## Blogs :

- <http://planet.openstack.org/>
- <http://www.mirantis.com/blog/>
- <http://openstack-in-production.blogspot.fr/>
- <https://grid.ifca.es/archives.html>