



# OpenStack : Solution de Cloud et Virtualisation sur mesure

Josy

Objectif *libre*

# Christophe Sauthier

- Ancien Président de Ubuntu-fr
- Développeur Ubuntu Officiel
- Développeur OpenStack
- Fondateur de Objectif Libre



Objectif *libre*

# Objectif Libre

SSLL spécialisée dans l'infrastructure linux:

- Mise en place / Audit
- Gestion
- Formations

Gros focus sur des outils "modernes" et innovants :

- Gestion de parc (Puppet et GLPI)
- Virtualisation (KVM, LXC, OpenStack..)

La contribution dans les gènes : Top 20 mondial pour OpenStack



Objectif *libre*

# OpenStack ?

- Projet "récent" (Juillet 2010)
  - Regroupement Plateforme Nebula (Nasa) et RackSpace Cloud File (Rackspace)
  - Licence Apache 2.0
  - Vocation de permettre à tous de créer des clouds (privés et/ou public) sur du matériel "standard"
- Plusieurs API d'appel (dont EC2 et S3)
- Leader sur le marché de l'OpenSource

Objectif *libre*

# OpenStack ?

- Articulé autour d'une fondation (Septembre 2012)
  - Membres individuels (plus de 15000)
  - Membres sociétés (différents niveaux de sponsoring)
    - Regroupe plus 360 sociétés (aujourd'hui)
      - Rackspace
      - Nasa
      - Canonical
      - Cisco
      - HP
      - ...
      - Objectif Libre
  - Différents comités (user, technical) et différents Boards (nommés / élus)

# OpenStack ?

- IaaS
- Divisé en sous-projets
  - **"Incorporés"** ou **Core** pour les projets principaux qui ont une certaine stabilité
  - **"En incubation"** ou **Incubated** pour les projets en cours d'intégration (pendant au moins 1 cycle)
  - **"Non officiel"** pour le lancement du projet (et pendant les premiers mois)

Objectif *libre*

# OpenStack ?

- Une nouvelle version tous les 6 mois
  - IceHouse (stable, avril 2014)
  - Juno (future dev, octobre 2014)
- Ubuntu comme choix de fait (et majoritaire) de plateforme de déploiement
  - plusieurs "sources" pour l'installation
  - Autres Linux possibles : gros efforts de Red Hat, mais aussi OpenSuse, Fedora, CentOS...

Objectif *libre*

# OpenStack ?

- Écrit en Python
- Communication entre les composants
  - API Rest et RPC
  - AMQP (RabbitMQ ou QPid)
- KVM comme premier choix d'hyperviseur
  - LXC pour des tests rapides
  - Xen/XenServer (dont XCP)
  - VMware/ESX
  - Docker
  - Baremetal

... Objectif *libre*





**Les principaux composants**

Objectif *libre*

# Les principaux composants

## Nova : Gestion des traitements

- Responsable de la gestion des instances (lien fort avec la virtualisation)
- La gestion des ressources (par des quotas)
- Possède la notion de zones de disponibilité
  - Personnalisables
  - Pour une réduction des risques
- Nova regroupe de nombreux "projets internes" pour assurer ces tâches comme nova-compute

Objectif *libre*

# Les principaux composants

## Swift : le stockage dans le cloud

- Permet de gérer un stockage de plusieurs pétaoctets en 1 seul système
- Assure la réplication et l'intégrité des données
- Stockage de blobs bruts accompagnés d'attributs (nécessité de xattr)
- Maintenance d'un mapping entre nom et localisation (the ring)
- Notion de zones (similaires à nova)
  - Personnalisables
  - Réplication (par défaut) dans 3 zones différentes
- Subit la concurrence de Ceph (projet non OpenStack mais très utilisé)

Objectif *libre*

# Les principaux composants

## Glance : La gestion des images

- Stockage et manipulation d'un ensemble d'images (personnalisées) utilisables
  - Lancement rapide
  - Déjà configurées
  - Intégrité vérifiée
- Nova supporte plusieurs systèmes de virtualisation => plusieurs formats d'images supportés par Glance
- Stockage multiple
  - Disques locaux
  - NAS/iSCSI
  - Swift
  - Ceph

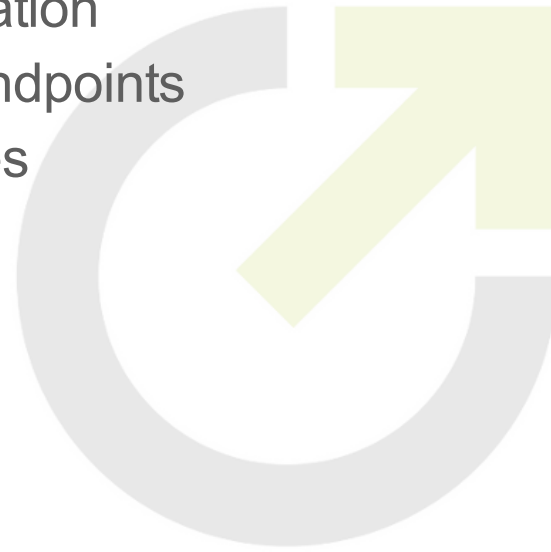
objectif libre

# Les principaux composants

## Keystone : La gestion des identités

Keystone est le point d'entrée dans son cloud

- centralise l'authentification
- liste des services et endpoints
- gère les différents rôles



Objectif *libre*

# Les principaux composants

## Neutron : La gestion avancée du réseau

Neutron constitue la nouvelle génération de gestion du réseau sur OpenStack

- Propose une API unifiée qui est capable de piloter plusieurs architectures réseaux
  - Linux Bridge
  - Open vSwitch
  - Cisco (matériel Nexus)
  - NSX (VMWare) ...
- Permet la création de réseaux complets et autonomes par tenant
- Base de fonctionnalités avancées
  - LBaaS
  - DNSaaS
  - FWaaS

# Les principaux composants

## Cinder : La gestion des volumes

- Nouveau paradigme pour les données :
  - **Ephémère** si non précisé
  - **Persistant** à la demande, on parle de volume  
C'est le rôle de Cinder
- Cinder est l'équivalent de EBS pour AWS
- Permet de stocker sur
  - Systèmes de fichiers (sur plusieurs nodes)
  - SAN / NAS
  - Ceph

Objectif *libre*

# Les principaux composants

## Heat : L'orchestration

- Equivalent de Amazon Cloud Formation
- Permet la description d'un déploiement (une **stack**) d'une infrastructure
  - Lancement et provisionning d'instances
  - Gestion des stockages associés
  - Gestion des IP
  - Haute disponibilité
  - Autoscalling (couplé à Ceilometer)
- 2 Approches
  - Compatibilité Cloud Formation
  - Langage de templates HOT



# Les principaux composants

## Ceilometer : La gestion des mesures

- Gère les mesures d'utilisation  
=> Le socle pour le billing (non inclu)
- De nombreux agents capable de donner des données ou des cumuls
- La base de l'alariming / autoscalling



Objectif *libre*

# Les principaux composants

## Et les autres (à différents stades)

- Horizon (Dashboard web)
- Trove (Base de données as a Service)
- Marconi (Messaging as a Service)
- Oslo (Librairies de code communes)
- Manilla (Systèmes de fichiers partagés)
- Devstack (Déploiement mono-node pour prise en main rapide)
- Ironic (Driver Bare Metal)
- TripleO (Déploiement)
- Sahara (Couplage avec Hadoop)

...

Objectif *libre*



**Pourquoi choisir OpenStack ?**

Objectif *libre*

# Pourquoi choisir OpenStack ?

## Création de cloud public

Le cas d'utilisation le plus courant...



Objectif *libre*

# Pourquoi choisir OpenStack ?

## Création de cloud privé

Proche d'une solution de virtualisation sur mesure (avec tous les composants et possibilités offertes)

- Multi hyperviseurs dans un seul environnement
- Gestion fine des lieux d'exécution (ou pas)
- Haute disponibilité
- Scalable par conception
- Load balancing natif
- Souplesse/facilité des migrations pour permettre une bonne gestion des hotes
- Ajout/intégration simplifié de nouveaux hosts
- Décomposition native en projets simplifiant les permissions
- Possibilité d'avoir une gestion avancée et fine du réseau

# Pourquoi choisir OpenStack ?

## Le futur avec le cloud hybride

Passage simple vers le cloud public (API communes)



Objectif *libre*



**L'Offre Objectif Libre pour OpenStack**

Objectif *libre*

# L'Offre Objectif Libre pour OpenStack

## Formations

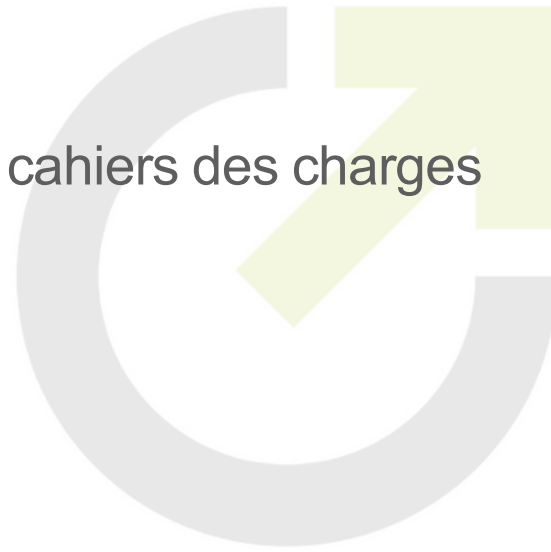
- Formation Mise en Oeuvre (3j)
  - Découvrir les concepts du cloud computing et les différents composants de OpenStack
  - Installer et gérer son propre cloud privé sur OpenStack
- Formation Mise en Oeuvre Avancée (3j)
  - Découvrir les composants avancés de OpenStack
  - Savoir créer une infrastructure Cloud robuste
- Formation Utilisation (2j)
  - Découvrir les concepts du cloud computing et les différents composants de OpenStack
  - Savoir utiliser son cloud privé
  - Savoir programmer son cloud privé en Python



# L'Offre Objectif Libre pour OpenStack

## Accompagnement et maintenance

- Audit existant / Conseil personnalisés
- Installation
- Support
- Aide à la rédaction de cahiers des charges
- Veille technologique



Objectif *libre*

# L'Offre Objectif Libre pour OpenStack

## Packs prototypage

- Pack **starter** (5j)
  - Audit des besoins et attentes
  - Présentation/Formation aux concepts et composants principaux de OpenStack
  - Mise en place d'un PoC (sur matériel non fourni, max 3 serveurs)
    - Nova et 2 computes KVM
    - Neutron
    - Glance
    - Keystone
    - Horizon
  - Transfert de compétence sur l'utilisation (CLI/Python) et automatisation via Puppet

# L'Offre Objectif Libre pour OpenStack

## Packs prototypage

- Pack **advanced** (10j)
  - le pack starter
  - Intégration dans le PoC des composants :
    - Swift ou Ceph et couplage aux autres outils
    - Heat et formation d'utilisation au déploiement de stack
    - Migration pour tests sur des éléments actuel du client (KVM ou ESX)

Objectif *libre*



# Questions ?

Sinon plus tard... [christophe.sauthier@objectif-libre.com](mailto:christophe.sauthier@objectif-libre.com)

Twitter: @chrissauthier

Objectif *libre*