

# Projet Réseau

## Plateforme de Cloud Computing



Tuteur : Olivier RICHARD

CALVI Jordan  
MAURICE Alexandre

RICM4  
2013

# Sommaire

- 1 Le Cloud Computing
  - 1.1 Services
  - 1.2 Standard Amazon EC2
  - 1.3 Offres courantes
- 2 Le projet OAR Cloud
  - 2.1 Virtualisation système avec LXC
    - 2.1.1 Présentation
    - 2.1.2 Fonctionnement
    - 2.1.3 Utilisation
    - 2.1.4 Avantages et Inconvénients
  - 2.2 Virtualisation réseau avec Open vSwitch
    - 2.2.1 Présentation
    - 2.2.2 Fonctionnement
    - 2.2.3 Utilisation
    - 2.2.4 Compatibilité
- 3 OAR
- 4 Conception
  - 4.1 Diagramme des cas d'utilisations
  - 4.2 Diagramme de classe
  - 4.3 Diagramme d'objets
  - 4.4 Diagramme de séquence
    - 4.4.1 Utilisateur
    - 4.4.2 Administrateur
- 5 Conclusion
- 6 Liens

# 1 Le Cloud Computing

## 1.1 Services

Le Cloud Computing est l'utilisation de ressources informatiques distantes suivant un niveau de service :

- SaaS (Software as a Service) : Ce model de Cloud Computing consiste à fournir l'accès à des logiciels depuis le web comme par exemple les GRC (Gestion de la Relation Client), outils collaboratifs, messagerie et PGI (Progiciel de Gestion Intégré). Nous pouvons citer les services suivants : GMail pour Google et Office 365 pour Microsoft. Ainsi, on parle plutôt d'une facturation à l'utilisation plutôt que de l'achat de licence.
- PaaS (Platform as a Service) : Mise à disposition des utilisateurs des outils de développement (ex : Google App Engine)
- IaaS (Infrastructure as a Service) : Fournit un accès à des ressources informatiques simple et adaptable aux besoins telles que des serveurs, des moyens de stockage et du réseau. Les utilisateurs peuvent réserver des grappes de serveurs et leur facturation est faite sur la base de la quantité de ressources utiliser (ex : Amazon EC2).

## 1.2 Standard Amazon EC2

Amazon ne vend pas que des livres et propose aussi un service de cloud computing EC2. Cette solution permet d'utiliser son propre OS ou des templates fournis par Amazon et ce sur un ou plusieurs serveurs en choisissant la puissance de calcul utilisée. Amazon facture au temps ou au débit de transfert comme tout ses concurrents. Les moyens de l'entreprise lui permettent d'avoir une quantité énorme d'équipements et donc d'assurer une excellent qualité de service.

Toute la gestion de leurs offres se fait par le web et chaque client dispose d'adresse IP associée à son compte et non à une machine. L'allocation de ressources se fait soit sur réservation soit à la demande, ce qui permet de réduire les coûts pour le client. Le service est donc sujet à de soudaine charges complètement aléatoire. Enfin, le service propose d'ajuster la taille du service (puissance, stockage, trafic) en temps réel.

Cette offre complète et éprouvé permet de se faire une idée des contraintes de mise en place et les évolutions à prévoir pour le projet.

### 1.3 Offres courantes

Le principe du cloud computing est de ne payer que ce qui est utilisé (ressources CPU, mémoire, transfert de données, etc). Amazon propose une tarification par heure d'instance consommée, propre à un type de d'instance fournissant une quantité de ressources prévisible, allant de 0,020 à 3,580 dollars de l'heure.

## 2 Le projet OAR Cloud

Le projet OAR Cloud consiste à étudier les possibilités fournies par le système de virtualisation spécifique à Linux LXC (Linux Containers) ainsi que par le logiciel de réseau virtuel Open vSwitch de manière à concevoir une architecture de Cloud Computing basée sur le gestionnaire de ressources OAR.

### 2.1 Virtualisation système avec LXC

#### 2.1.1 Présentation

LXC est un système de virtualisation au niveau du système d'exploitation Linux. Il permet de créer des environnements virtuels appelés conteneurs, disposant de leurs propres processus et réseau. Ce système de virtualisation a la particularité d'être basé sur l'isolement en s'appuyant sur les Cgroups(Control Groups), récemment intégrés au noyau Linux.

#### 2.1.2 Fonctionnement

L'utilisation des Cgroups fourni la possibilité de limiter l'accès aux ressources telle que la mémoire, de prioriser l'accès au CPU, de mesurer la quantité de ressources utilisées, de séparer les espaces de nommage permettant l'isolement (des processus, des connections réseaux et des fichiers) et enfin la gestion. Toutes ces fonctionnalités sont proposées par les sous systèmes de contrôle et d'isolement suivants :

- Cpuset : permet de lier les processus (noeuds) de sous-ensembles du système de CPU et de la mémoire
- Freezer : permet de stopper les processus d'un groupe donné lorsque leur consommation en ressources dépasse un certain seuil, utile pour le HPC (High Performance Computer) tel que Grid5000
- Checkpoint/Restart : permet de sauvegarder l'état d'un container en vu de le restaurer plus tard ou sur une autre machine physique
- Devices : permet de restreindre l'accès à des appareils ou des systèmes de fichiers pour les processus se trouvant dans un groupe (isolation)

- Cpuacct : mesure l'utilisation du CPU
- CPU : partage le temps CPU entre les différents groupes
- Memory : Limite l'utilisation de la mémoire

L'utilisation des Cgroups plutôt que Chroot permet d'avoir en plus une isolation des systèmes de fichiers, une isolation des processus et la capacité de limiter les ressources qu'ils utilisent.

### 2.1.3 Utilisation

LXC permet de créer des conteneurs dans lesquels peuvent s'exécuter des programmes de manière isolés. En effet, cela permet de s'affranchir de la lourdeur d'un deuxième système d'exploitation en utilisant le noyau Linux de la machine hôte. La configuration des conteneurs est simple dans le sens où des fichiers XML directement éditables permettent de spécifier les paramètres de création et aussi d'autres options prises en compte au démarrage du conteneur comme par exemple l'interface réseau et l'adresse IP associée.

### 2.1.4 Avantages et Inconvénients

Les principaux atouts de LXC sont des performances bien meilleurs que les hyperviseurs tels que XEN et KVM grâce à une virtualisation basée sur l'isolation en utilisant directement le noyau de la machine hôte. De cette manière, il n'y a pas le surcoût de l'émulation d'un matériel virtuel nécessaire au fonctionnement des autres solutions de virtualisation. De plus, puisque le système principal a connaissance de tous les processus, cela optimise l'ordonnancement.

Cependant, LXC est instable et est toujours en développement (version 0.7.5 utilisée). Cela pose en effet des problèmes de généricité (fonctionne uniquement sur certaines distributions) et des problèmes de compatibilité (intégration avec libvirt et Open vSwitch capricieuse).

De plus, le partage du noyau de l'hôte avec les conteneur nécessite que le système d'exploitation déployé dedans soient compatibles avec le noyau hôte et les Cgroups (impossibilité par exemple de lancer le système d'exploitation Windows).

## 2.2 Virtualisation réseau avec Open vSwitch

### 2.2.1 Présentation

OpenvSwitch est un logiciel de switch virtuel conçu pour relier entre-elles des machines virtuelles tel que le ferait un switch physique entre plusieurs machine. C'est un logiciel entièrement libre et qui fonctionne sur un grand nombre de plateforme.

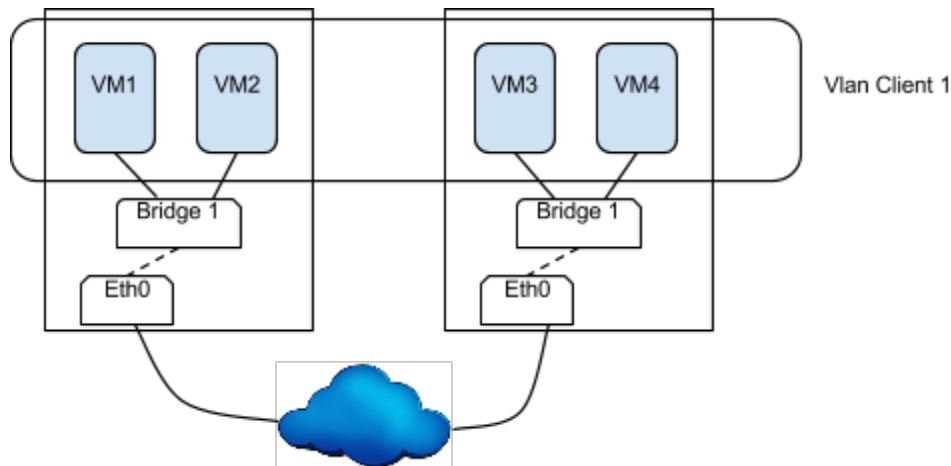
### 2.2.2 Fonctionnement

Openvswitch permet de créer des bridges virtuels entre plusieurs VMs pour les relier ensemble tout en garantissant un isolement complet entre les bridges. Cela permet de créer plusieurs réseaux indépendants sur une même machine physique.

### 2.2.3 Utilisation

Open vSwitch permet de créer autant de bridge que l'on veut et autant de VLAN sur chaque bridge. Cette aspect permet de facilement isoler chaque client en lui attribuant un bridge dédié, on évite ainsi les problèmes de hack ou de fort trafic généré par le rejet de paquets marqués.

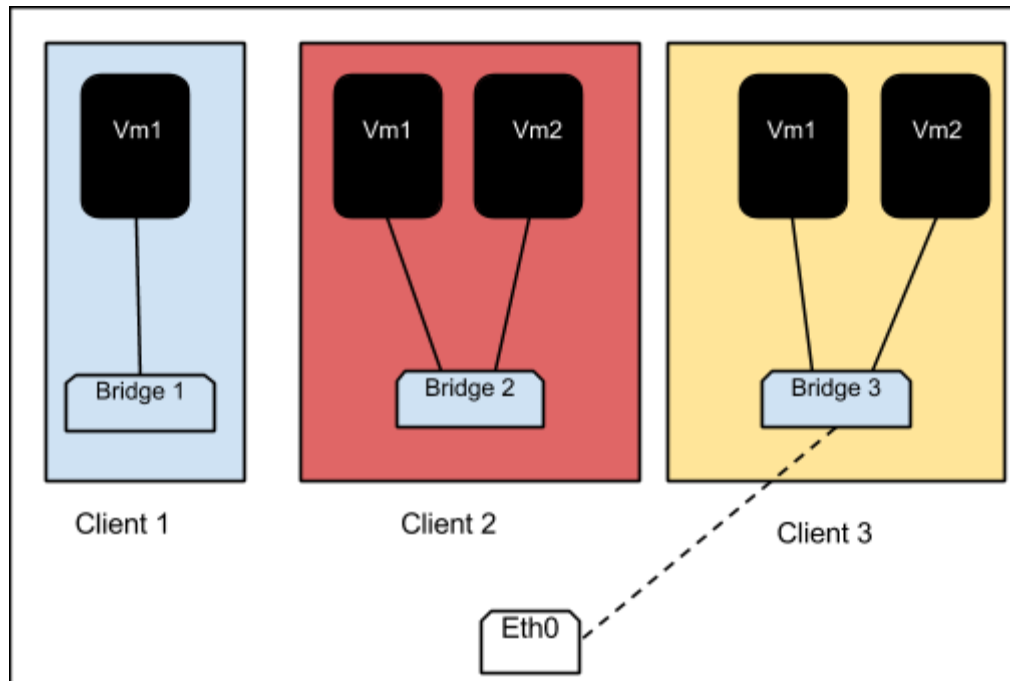
L'usage des VLAN peut cependant permettre de relier les VMs d'un même client qui sont déployées sur plusieurs machines physiques. Dans ce cas les différents bridges sont reliés par un VLAN sur le réseau du serveur. On garanti ainsi que les clients ne voient que leur VMs et sont protégé du "monde extérieur". Les paquets tagués ne circulant qu'entre les bridges d'un même client, l'isolement des machines n'est pas remis en cause. L'exemple ci dessous montre l'usage des VLAN.



L'illustration suivante montre un serveur hébergeant cinq machines appartenant à trois clients différents.

Chaque client possède son propre bridge ce qui permet d'assurer un isolement complet et une administration plus facile, en effet il devient plus facile d'isoler un bridge pour repérer un problème ou simplement le détruire si le client résilie le service. De plus on peut proposer en option une connexion internet ou d'autres services, on relie alors un bridge et non pas un ensemble de port par bridge.

Enfin, on peut connecter un bridge à l'interface réseau du serveur pour communiquer avec les machines d'un même client mais situées sur un autre serveur par l'intermédiaire de paquets tagués pour permettre à Open vSwitch de les rediriger sur le bon bridge distant.



#### 2.2.4 Compatibilité

La compatibilité avec LXC a posé quelque soucis de performance, en effet en le reliant aux VM on obtient des ralentissements très important au démarrage. Le bridge linux ne pose pas autant de problèmes. Il ne propose cependant pas l'isolement ni les services nécessaires à une plateforme de cloud computing.

### 3 OAR

OAR est un gestionnaire de ressource qui permet d'allouer des machines en fonction des ressources demandées. Il se compose de trois applications :

- OAR frontal : le frontal permet d'accéder aux serveurs d'OAR
- OAR serveur : il est responsable de la gestion d'un cluster
- OAR client : il est installé sur chaque noeud

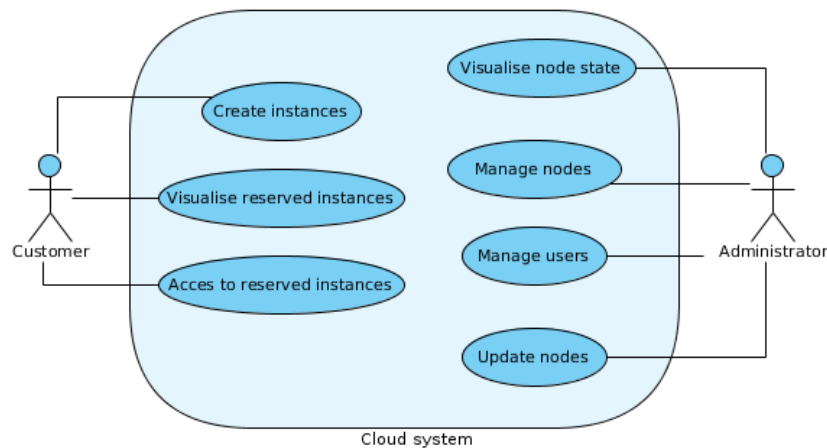
Les réservations de ressources sont appelées des « jobs ». Les principales commandes sont les suivantes :

- oarsub : permet lancer une réservation de ressources selon les critères fournis
- oarsh : permet d'accéder à un « job »

## 4 Conception

### 4.1 Diagramme des cas d'utilisations

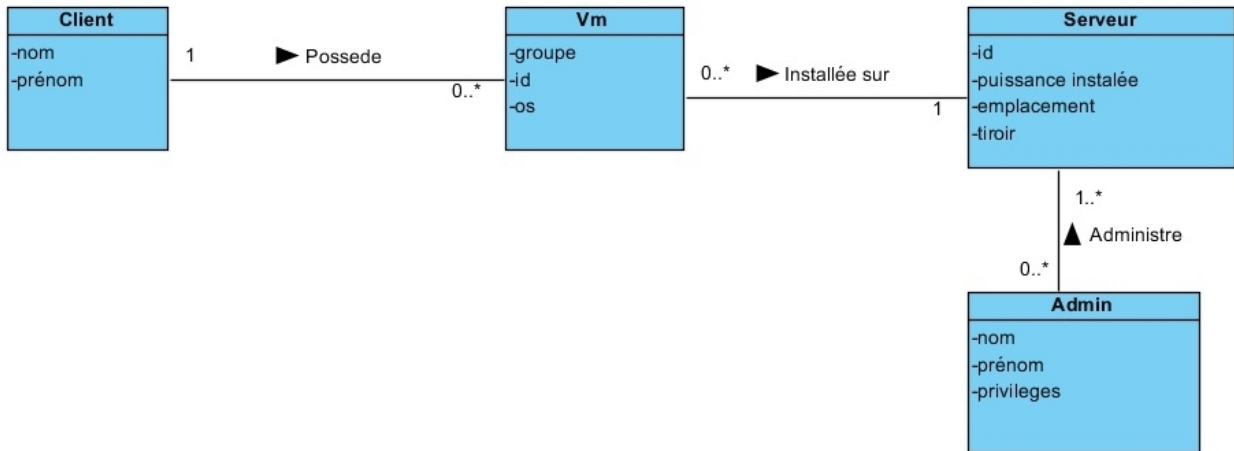
Deux acteurs peuvent interagir avec le système de cloud computing : les utilisateurs et les administrateurs. Un utilisateur peut créer des instances en lançant des « jobs ». Il peut aussi visualiser l'état des instances qu'il a réservées auparavant et y accéder. L'administrateur peut quant à lui gérer les noeuds en procédant à des mises à jours, gérer les utilisateurs et avoir une vue globale de l'état du système.



### 4.2 Diagramme de classe

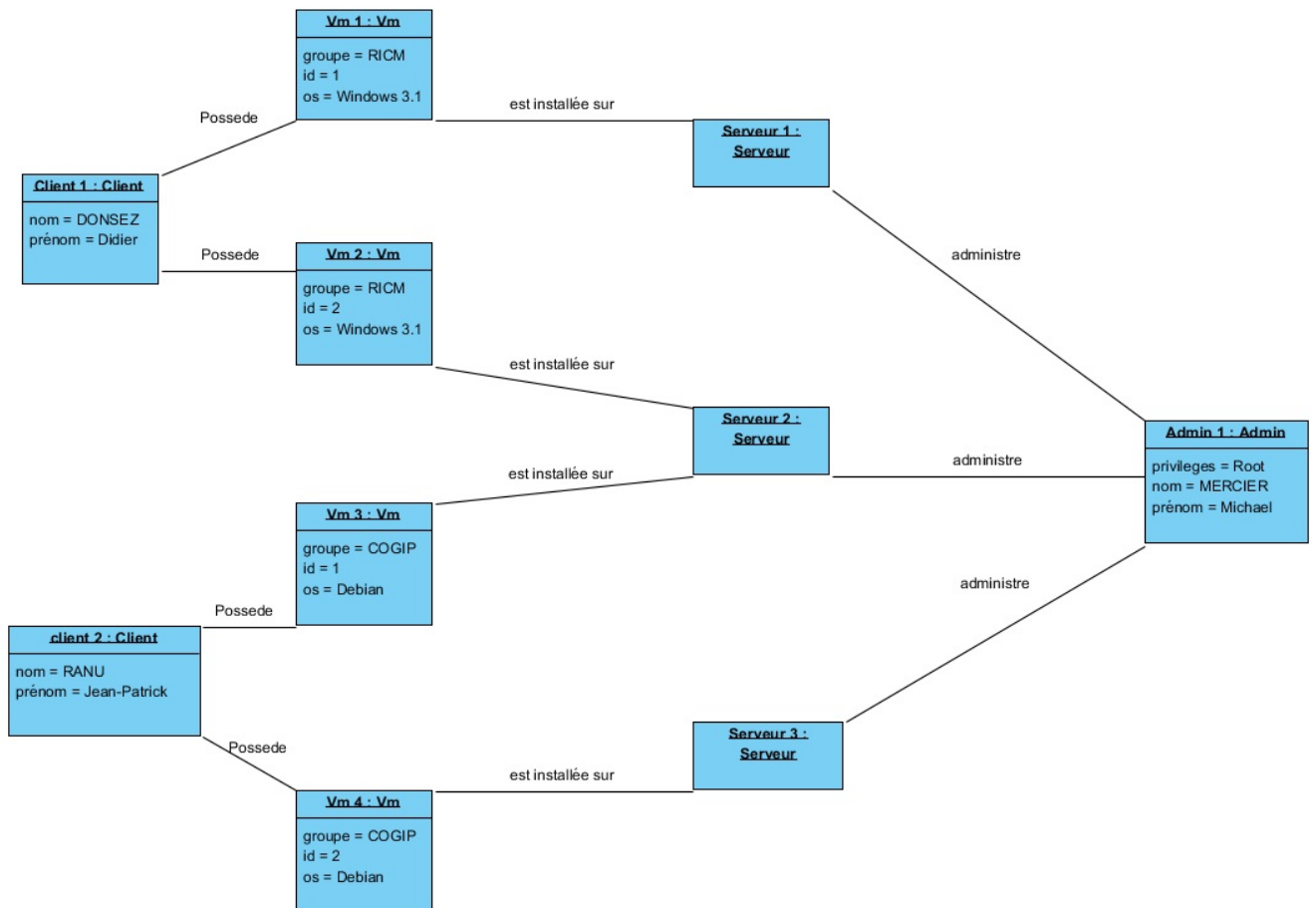
Un client possède des conteneurs dans lesquels des « jobs » sont exécutés. Une VM est déployée sur un serveur mais les VMs d'un utilisateur ne se trouvent pas nécessairement sur un même serveur. Des administrateurs sont responsables d'au moins un serveur.





### 4.3 Diagramme d'objets

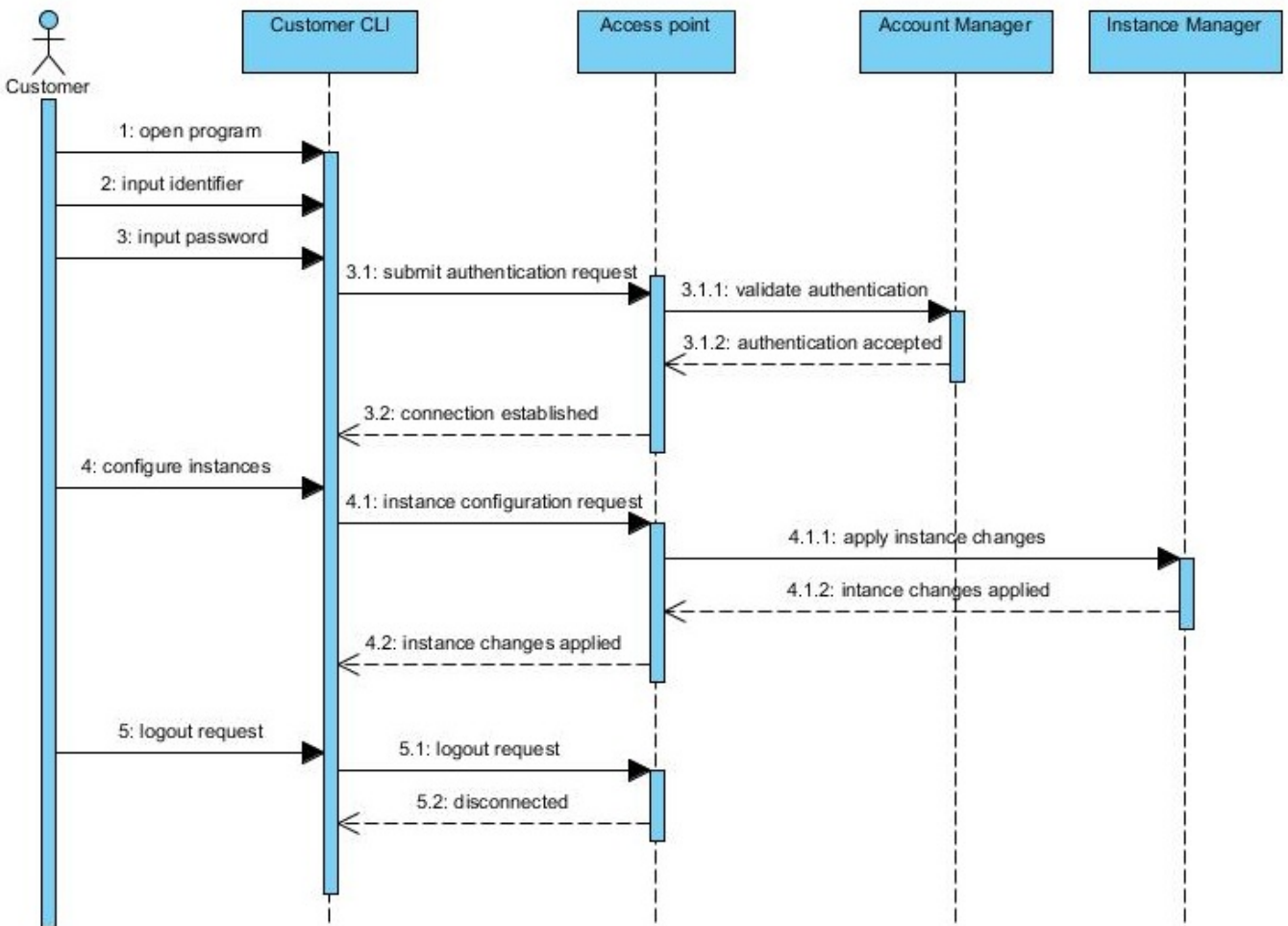
Le diagramme d'objets suivant montre un exemple d'état du système de cloud computing :



## 4.4 Diagramme de séquence

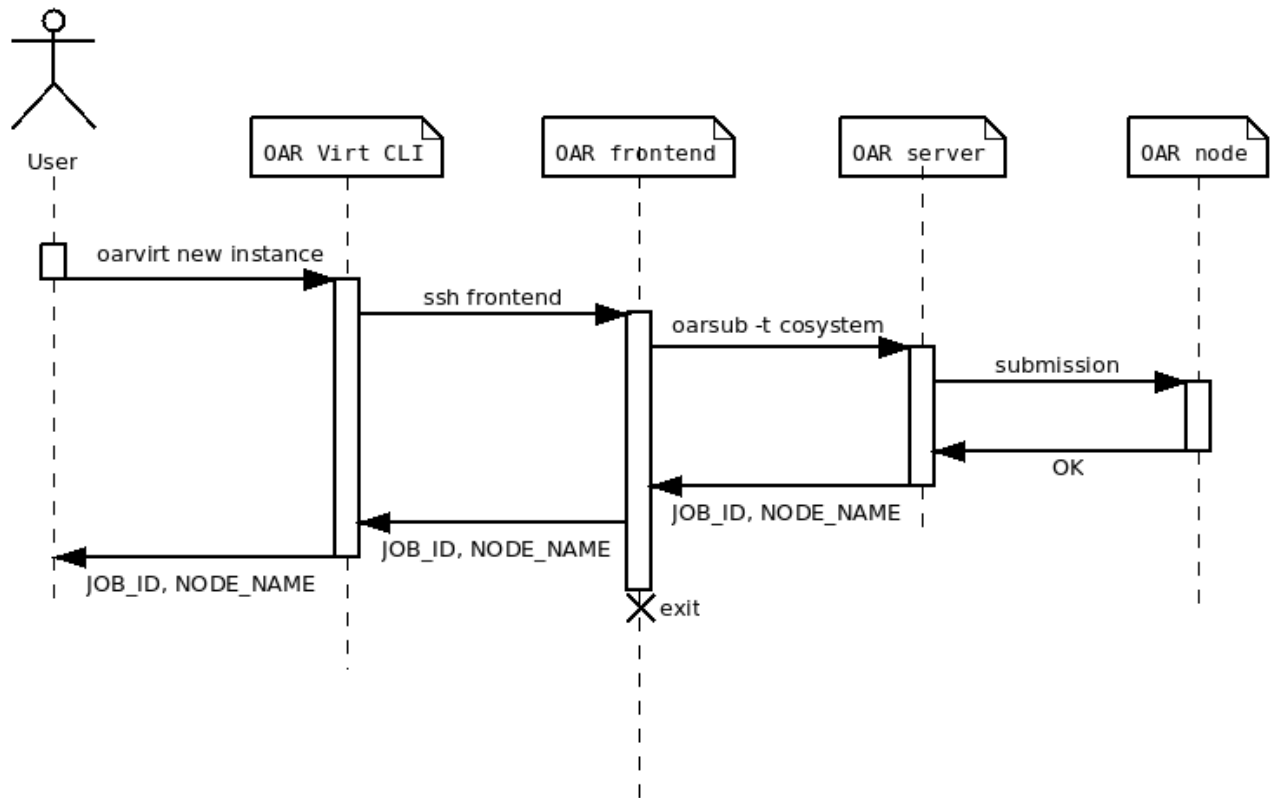
### 4.4.1 Utilisateur

Voici une vue globale de la manière dont un utilisateur interagit avec le système en séquences. L'utilisateur entre des commandes via une CLI (Command Line Interface) qui lui permet de se connecter à distance sur le frontal. Il peut alors manipuler des conteneurs faisant des requêtes au serveur OAR (account manager) qui se charge de faire les modifications adéquates. Enfin, l'utilisateur est libre de se connecter aux conteneurs via SSH.

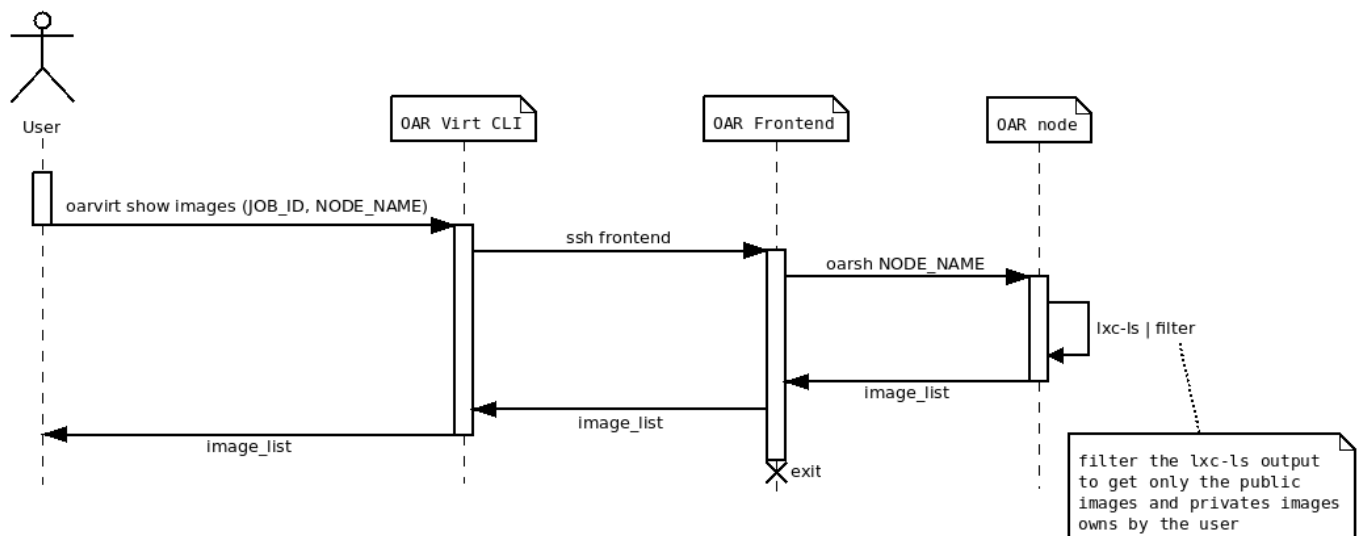


Voici maintenant une vue plus détaillées des séquences d'interactions possibles de l'utilisateur avec le système :

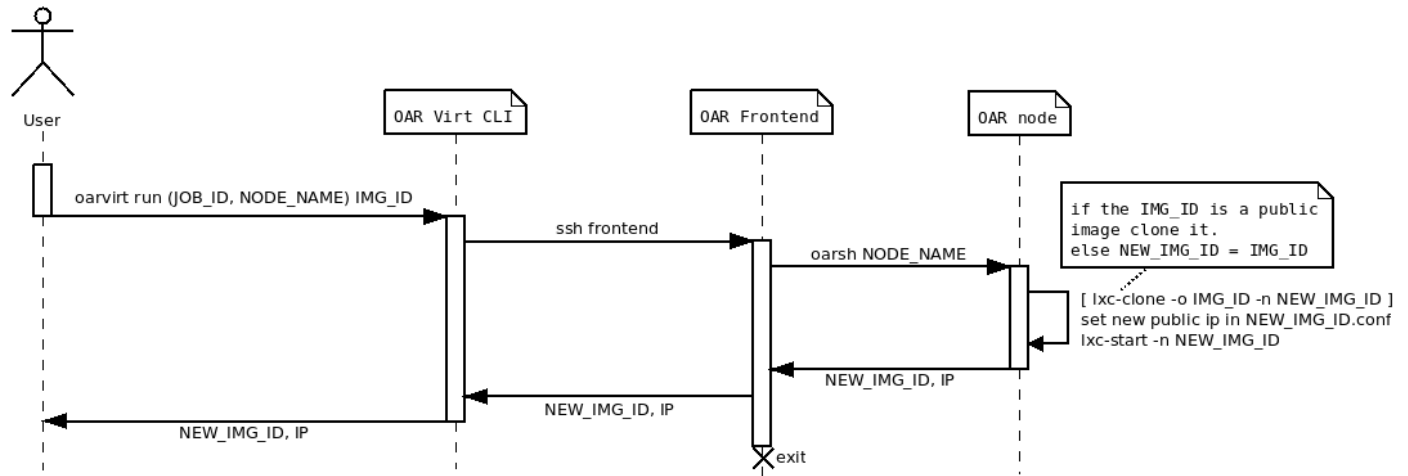
- L'utilisateur réserve un noeud :



- Il récupère la liste des conteneurs disponibles sur ce noeud :

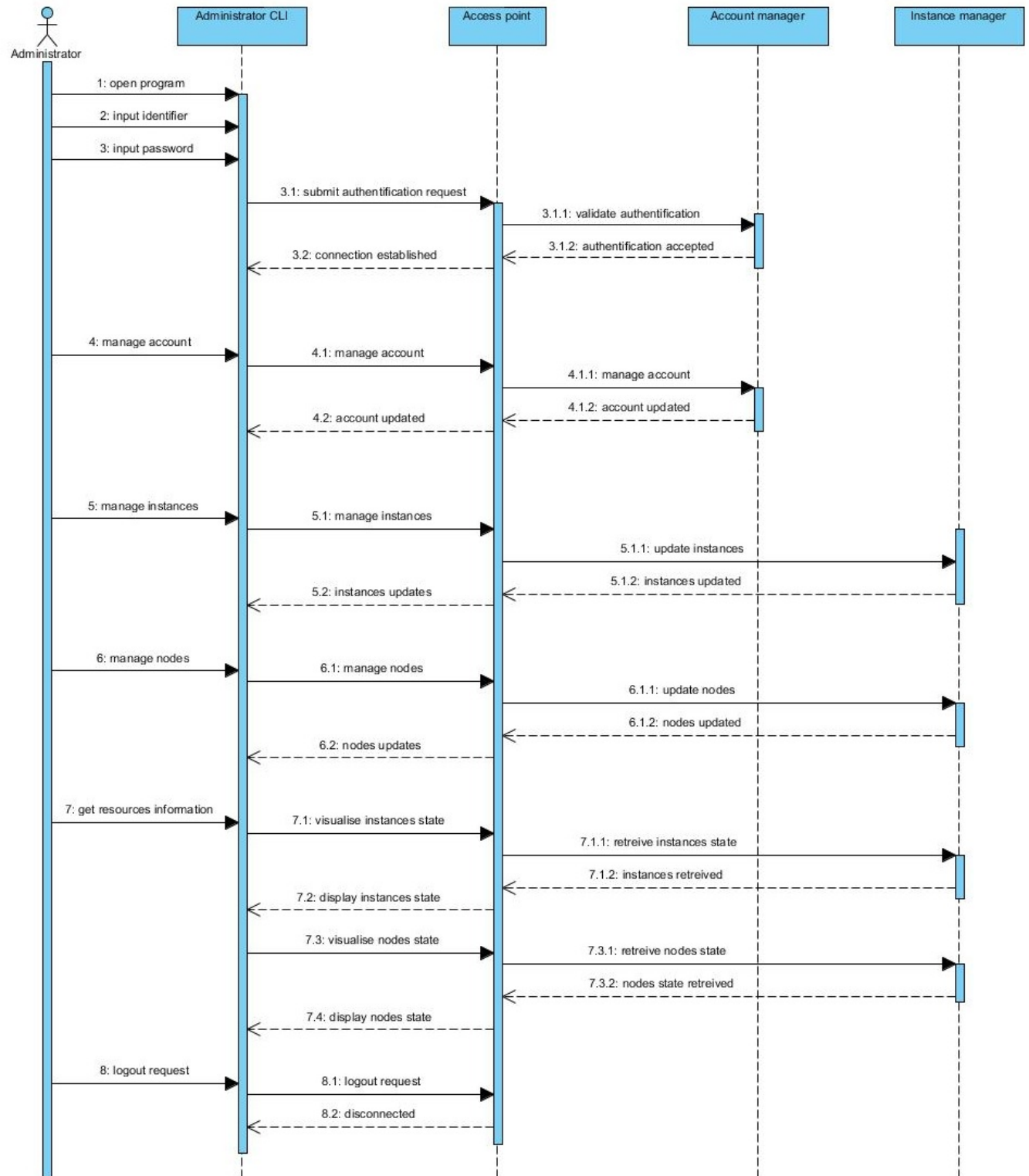


- L'utilisateur lance le conteneur choisi :



#### 4.4.2 Administrateur

L'administrateur peut lui aussi interagir avec le système de manière similaire mais avec des fonctionnalités différentes :



## 5 Conclusion

Ce projet a permis de découvrir les problématiques du cloud computing et en particulier l'interaction entre un grand nombre de machines virtuelles. Que ce soit du point de vue de la virtualisation en elle même avec le partage de ressource et leur déploiement ainsi que l'infrastructure réseau permettant de les relier.

Ce document présente la solution sous son aspect théorique et non en version fonctionnel. L'étape suivante étant de pouvoir déployer une version test sur une plateforme grandeur nature comme Grid5000.

Ce projet nous a permis de découvrir les contraintes liées au développement d'une nouvelle solution en s'inspirant des solutions concurrentes (ex : Amazon) mais aussi de l'imprévu lorsqu'il s'agit d'utiliser ensemble deux technologies distinctes (Open vSwitch et LXC).

## 6 Liens

Fiche du projet : <http://air.imag.fr/mediawiki/index.php/Proj-2012-2013-OAR-Cloud>

Dépôt GIT : <https://github.com/mickours/oar-cloud>