

1 juillet 2014 : Ecole IN2P3 — Maîtriser le Cloud

Comment accéder les services cloud ?

Charles Loomis (CNRS/LAL)

Services Distants

- ❖ Un service cloud consist des ressources hébergée dans un site ou plusieurs sites distants
- ❖ Toutes les utilisateurs doivent avoir une moyenne rapide and robuste pour accéder ces services

Besoins différents

- ❖ Il y a plusieurs types des utilisateurs avec les besoins différents :
 - ❖ **Utilisateurs finaux** : accès simple
 - ❖ **Administrateurs** : automation des tâches
 - ❖ **Développeurs** : nouveau services cloud
- ❖ Il y a un moyenne d'accès adapter a chaque type d'utilisation.

Browser Web

- ❖ L'accès aux services cloud vers un browser web est le moyenne plus simple pour les utilisateurs. Beaucoup des logiciels et des service cloud fournissent un portail web. Idéale pour les utilisateurs finaux.
- ❖ Avantages :
 - ❖ Aucune logiciel d'installer
 - ❖ Interface simple avec un browser web
- ❖ Inconvénients :
 - ❖ Parfois des fonctionnalités réduits
 - ❖ Pas bien adapté a l'utilisation intense ou réplétive

Ligne de commande

- ❖ L'accès aux services cloud vers un logiciel léger installé sur un poste de travail. Contrôle des ressources avec la ligne de commande.
- ❖ Avantages :
 - ❖ Compatible avec un utilisation depuis des scripts
- ❖ Inconvénients :
 - ❖ Installation du logiciel et ses dependencies
 - ❖ Parfois une portabilité limitée (p.e. Windows)

Les APIs

- ❖ L'accès depuis un vrai langage de programmation. Faciliter la création des nouveaux services qui utilisent le cloud.
- ❖ Avantages :
 - ❖ Accès adapté aux programmes
 - ❖ Toutes les fonctionnalités des services sont disponibles
- ❖ Inconvénients :
 - ❖ Les bibliothèques pour un nombre de langages limité
 - ❖ Les APIs varient (parfois beaucoup) entre les services

Les types des APIs

- ❖ API native : accès direct aux services
 - ❖ (+) Toutes les fonctionnalités
 - ❖ (-) Beaucoup de variations
 - ❖ (-) Pas de support pour toutes les langues
- ❖ API unifiantes : un accès vers un API “abstract”
 - ❖ (+) API unique pour la programmation dans un langage donné
 - ❖ (-) Abstraction limitée aux fonctionnalités communes entre les clouds
 - ❖ (-) Doit développer un plugin pour les cloud pas encore supporté
- ❖ API standardisées : un API normalisée
 - ❖ Très peu des standards et aucune avec une supporte universelle

Libcloud

- ❖ Une API en Python qui supporte plusieurs services cloud avec les plugins.
- ❖ Sémantique de EC2/S3 applique aux autres clouds
- ❖ Mature et fortement utilisé
- ❖ Assez simple de créer un plugin pour un nouveau service cloud

jClouds

- ❖ Une API en Java qui supporte plusieurs services cloud.
- ❖ Inclure aussi les fonctionnalités “high-level” pour gérer les systèmes avec plusieurs machines virtuelles
- ❖ Mature, mais ça vient plus et plus complex

REST sur HTTP

- ❖ Presque toutes les services cloud aussi présentent un API “REST”. Ces APIs réutilisent HTTP pour avoir une interface universelle:
 - ❖ “Create” → POST
 - ❖ “Read” → GET
 - ❖ “Update” → PUT
 - ❖ “Delete” → DELETE
- ❖ Les bibliothèques pour les langages spécifiques normalement repose sur l’API REST.

Les APIs standardisées

- ❖ Amazon
 - ❖ EC2/S3 sont les standards “de facto”
- ❖ Open Grid Forum (OGF)
 - ❖ OCCI : Open Cloud Computing Interface (OGF)
- ❖ Distributed Management Task Force (DMTF)
 - ❖ CIMI : Cloud Infrastructure Management Interface
 - ❖ CDMI : Cloud Data Management Interface

Conclusions

- ❖ Les moyennes d'accéder des services cloud existe pour toute types d'utilisation
 - ❖ Browser web
 - ❖ Ligne de commande
 - ❖ Les APIs : natives, unifiantes, standardisées

Questions...