IVC

Infrastructures Virtuelles et Conteneurs)

Legond-Aubry Fabrice

fabrice.legond-aubry@parisnanterre.fr

Plan du Cours

Infrastructures Virtuelles et Conteneurs

Introduction

Principes Théoriques

Applications pratiques (bases)

Outils de déploiements

Sécurité

Plan du Cours

Infrastructures Virtuelles et Conteneurs

Introduction

Introduction

Utilité des IVC. Pourquoi ?

- Depuis les débuts de l'informatique, on veut
  - ✓ Faciliter la gestion des applications
  - ✓ le déploiement et la configuration
  - ✓ la migration (physique)

- Utilisation de scripts de déploiement
- Configuration (métrologie) à la main du système pour l'application
  - ✓ Problèmes de cohabitation des applications

Introduction
Utilité des IVC. Pourquoi ?

- Utilisation des composants logiciels
  - ✓ Gestion partielle de la configuration
  - ✓ Morceaux d'applications gros grains adaptés aux applications BackEnd
  - ✓ Permet l'isolation de chaque morceaux d'applications
  - ✓ Migration possible entre les serveurs d'applications

- Composants restent insuffisants
  - ✓ Isolation extrêmement limitante sur la conception et l'architecture des applications
  - ✓ Nécessite un environnement applicatif lourd (le serveur d'application)
  - ✓ Manque de nombreux services systèmes

Introduction

Utilité des IVC. Pourquoi ?

- En parallèle, virtualisation de certains services systèmes
  - ✓ Ex: les supports persistants
  - ✓ Disque accessibles par le réseau
    - ➢ Gros disque accessibles par le réseau
  - ✓ Disques virtuels (réseaux de disque)
    - ➢ Création de réseaux spécifiques pour agréger des disques

- Insuffisant :
  - ✓ Il manque les services applicatifs
  - ✓ Le déploiement

Introduction

Utilité des IVC. Pourquoi ?

- IVC : Tentent de fusionner les deux approchent
  - ✓ l'isolation
  - ✓ La gestion des services systèmes
  - ✓ La gestion des services applicatifs
- Le seul moyen :
  - ✓ Créer des systèmes, à la volée, spécialisés pour chaque applications
  - ✓ Une Application DEVIENT une nouvelle entité :
    Une application + SON système dédié
- Objectifs de dématérialisation
  - ✓ Terriblement accentués par le COVID
  - ✓ Gestion à distance
  - ✓ Indépendance vis-à-vis des infrastructures

Introduction

Utilité des IVC. Pourquoi ?

- Un peu d'histoire :
  - ✓ Idée développée au centre IBM de Cambridge et de Grenoble en 1972 (VM/CMS) (pseudo-machine.)
  - ✓ UNIX chroot pour unix 7 et bsd (1979-1982)
  - ✓ Mi-90's émulateurs d'Atari, Amiga, NES, SNES,…
  - ✓ BSD Jail (1998/2000)

Introduction

Utilité des IVC. Pourquoi ?

- Début des années 2000 :
  - ✓ Propriétaires : Vmware, Parallels Virtuozzo (2001)
  - ✓ Logiciels libre : Xen, Qemu (2003)
  - ✓ Propriétaire (mais gratuits) : VirtualBox
  - ✓ Solaris Containers – zones (2005)
  - ✓ Intel VTx / AMD V (assistance physique pour la virtualisation)
  - ✓ KVM (2007)
  - ✓ Linux LXC (2008)
  - ✓ Microsoft Hyper-V (2009)
  - ✓ Docker (2013)

Introduction

Utilité des IVC. Pourquoi ?

- Matériel physique :
  - ✓ Surchauffe
  - ✓ Plante, Casse, Brûle
  - ✓ Se périme (voir les CPU/Carte Calcul/Disque dur)
  - ✓ La distribution nécessite des protocoles spécifiques
  - ✓ On atteint des limites des fréquences CPU (>3GHz) au delà duquel il y a des problèmes de désynchronisation des horloges, perturbation.
    On multiplie les "core" dans les CPUs
    - ➢ Parfois les machines sont inadaptées à l'application (problèmes de métrologie)
    - ➢ En sur ou sous capacité !

Introduction
Utilité des IVC. Pourquoi ?

- Machine virtuelle (avantages) :
  - ✓ les API logiciels périment moins vite que le matériel
    - ➢ Plusieurs générations de matériel peut utilisé la même API
  - ✓ Peut être facilement distribué sur plusieurs sites
    - ➢ Facilité de déploiement, de configuration, allocation dynamique
  - ✓ Peut être migrer sur un autre support physique
    - ➢ Souvent même pendant l'exécution avec un délai (freeze) minime.
  - ✓ Le "matériel" virtuel émulé reste plus homogène que les matériels physiques
    - ➢ Cela peut être un problème car limite la puissance
    - ➢ Mais économie sur le matériel → création de ferme
  - ✓ Usage plus optimal des ressources même sur une seule machine
    - ➢ Mutualisation, contrôle des ressources
  - ✓ Sécurité importante
    - ➢ Isolation, création d'image, mise en suspend
    - ➢ surveillance / audit (log) / diagnostiques post-mortem (forensic)

Introduction

Utilité des IVC. Pourquoi ?

- Ex Matériel :
  - ✓ Créer une carte réseau 10Gb
    - ➢ QQS milliers d'ingénieurs
    - ➢ Création de chipset qui nécessite une fonderie

- Ex Virtualisation :
  - ✓ Emuler une carte réseau 10Gb dans une machine virtuelle
    - ➢ 3 mois de travail, 2 ingénieurs
- Possibilité de déployer une carte 10Gb sur des cartes 1Gb ou sur d'autres types de cartes.
- Attention: Certaines parties sont cachés ou non spécifiées
  - ✓ On appelle cela des points de variations sémantiques ou d'incertitudes

Introduction

Utilité des IVC. Pourquoi ?

- Différents domaines / types de virtualisation
  - ✓ Virtualisation d'applications (du contexte d'exécution).
  - ✓ Virtualisation de serveur
  - ✓ Virtualisation du réseau (VLAN)*
  - ✓ Virtualisation du stockage*

* = hors de ce cours

Introduction

Modèles génériques de virtualisation

- **Software-as-a-service (SaaS)**
  - ✓ Accès à un logiciel au travers d'une connexion `a un serveur distant
  - ✓ Accès libre ou sur abonnement
  - ✓ Accès au travers d'un client (ex: navigateur web)
  - ✓ Exemples: Service mail, Editeur en ligne, Github

Introduction

Modèles génériques de virtualisation

- **Platform-as-a-service (PaaS)**
  - ✓ Le client déploie son application sur l'infrastructure cloud
  - ✓ Le fournisseur fournit les briques logicielles de base (OS, base de donnée, etc)
    - ➢ Le client peut configurer ces briques logicielles selon ses besoins
  - ✓ Le fournisseur maintient la plateforme d'exécution
    - ➢ Serveur, Stockage, Réseau
  - ✓ Le fournisseur peut fournir des services en plus
    - ➢ Persistance, Haute disponibilité, Sécurité

Introduction

Modèles génériques de virtualisation

- **Infrastructure-as-a-service (IaaS)**
  - ✓ Le fournisseur donne accès à des ressources informatiques
    - ➢ Calcul, stockage, réseau
  - ✓ Accès sous forme de machines virtuelles
    - ➢ Possibilité de réserver des ressources à grain très fin (1 cœur de calcul)
  - ✓ Le client installe sa propre pile logicielle sur les ressources obtenues
  - ✓ Possibilité de déployer son propre système d'exploitation

Introduction
Quelques termes en vrac

- **CD =** Continuous Delivery
- **CT =** Continuous Testing
- **CI =** Container Integration
- **Système hôte (host) :**
  - ✓ système hébergeant les les conteneurs
- **Système invité (guest) :**
  - ✓ machine virtuelle exécuté sur le système hôte
  - ✓ "Abusivement" : conteneur exécuté sur le système hôte

Introduction
Quelques termes en vrac

- Conteneur (container), VM → voir les définitions
- **Registre (Registry) :**
  - ✓ Dépôts d'images de conteneur / VMs
- **OCI** = Open Container Initiative
  - ✓ définition d'une structure standard de conteneur et de registre
- **Daemonless/Daemonful(l) :**
  - ✓ Nécessite ou non un service en fond pour fonctionner
- **rootless / rootful(l)** :
  - ✓ Nécessite des droits root (ou non) sur le système hôte