Virtualisation basique

Alves Lobo Michael

- https://www.linkedin.com/in/michael-alves-lobo
- https://github.com/kairel/learning
-
- Devops, dev, réseau et administration système
- En poste chez Vade/Hornet
- Automatisation
  - ansible
  - puppet
  - terraform
  - docker

SOMMAIRE

Introduction et Concepts Fondamentaux

- Principe de base
- Pourquoi virtualiser ?
- Histoire de la virtualisation
- Concepts clés à retenir

Types de Virtualisation et Hyperviseurs

- Virtualisation de ressources
- Architecture d'un environnement virtualisé
- Un exemple Vagrant

SOMMAIRE

Haute disponibilité et continuité d’activité

- Concept
- Architecture
- Migration a chaud
- Sauvegarde et restauration
- Bonnes pratiques
-

Introduction générale

- Définition de la virtualisation

La virtualisation est une **technologie qui permet de créer des versions virtuelles** (simulées) de ressources physiques, comme :

- Des **serveurs** (machines virtuelles).
- Des **postes de travail** (PC virtuels).
- Des **réseaux** (switchs, routeurs virtuels).
- Des **stockages** (disques virtuels).
- Des **applications** (conteneurs).

Objectif : Découpler le matériel physique du logiciel, pour une utilisation plus flexible et optimisée.

Introduction générale

- Pourquoi Virtualiser

- Réduction des coûts matériels (moins de serveurs physiques). Economies
- Création/suppression rapide de machines virtuelles (VM). Flexibilité
- Chaque VM est indépendante (sécurité, tests sans risque). isolation
- Ajout de ressources (CPU, RAM) sans changer le matériel. Scalabilité
- Meilleure utilisation des ressources (un serveur physique peut héberger plusieurs VM).
- Sauvegarde et restauration facile des VM.

Introduction générale

- Histoire de la virtualisation

- **Années 1960** : Premiers concepts sur les mainframes IBM
- **Années 1990** : VMware révolutionne la virtualisation x86
- **Années 2000** : Démocratisation avec VirtualBox, Hyper-V
- **Années 2010** : Émergence de la containerisation (Docker)

Introduction générale

- Les différents type de virtualisation

A. Virtualisation de serveur

- **Principe** : Un serveur physique héberge plusieurs serveurs virtuels.
- **Exemples** : VMware ESXi, Microsoft Hyper-V, KVM (Linux).
- **Cas d’usage** : Hébergement web, bases de données, cloud privé.

Introduction générale

- Les différents type de virtualisation

B. Virtualisation de poste de travail

- **Principe** : Exécuter un OS complet dans une fenêtre sur un PC.
- **Exemples** : VirtualBox, VMware Workstation, Parallels (Mac).
- **Cas d’usage** : Tests de logiciels, formation, compatibilité (ex : Windows sur Mac).

Introduction générale

- Les différents type de virtualisation

C. Virtualisation de réseau

- **Principe** : Simuler des équipements réseau (routeurs, switchs).
- **Exemples** : SDN (Software-Defined Networking), NFV (Network Functions Virtualization).
- **Cas d’usage** : Laboratoires de réseau, cloud networking.

Introduction générale

- Les différents type de virtualisation

D. Virtualisation de stockage

- **Principe** : Regrouper plusieurs disques physiques en un seul espace virtuel.
- **Exemples** : SAN (Storage Area Network), NAS virtuels.
- **Cas d’usage** : Stockage partagé, sauvegardes centralisées.

Introduction générale

- Les différents type de virtualisation

E. Virtualisation d’applications (conteneurs)

- **Principe** : Isoler une application et ses dépendances dans un conteneur léger.
- **Exemples** : Docker, LXC.
- **Cas d’usage** : Développement, déploiement rapide (microservices).

Introduction générale

- Comment ca marche ?

- **Grace a un Hyperviseur**

Un **hyperviseur** est une plate-forme de virtualisation qui permet à plusieurs systèmes d'exploitation de fonctionner en parallèle dans une seule machine physique.

Introduction générale

- Comment ca marche ?

Rôle de l’hyperviseur

- **Hyperviseur** : Logiciel qui gère les machines virtuelles.
  - **Type 1 (bare-metal)** : S’installe directement sur le matériel (ex : VMware ESXi).
  - **Type 2 (hosted)** : S’installe sur un OS existant (ex : VirtualBox).
- **Fonctionnement** :
  - L’hyperviseur alloue dynamiquement les ressources (CPU, RAM, disque) aux VM.
  - Chaque VM pense avoir son propre matériel dédié.

Matériel physique → Hyperviseur → Machines virtuelles (VM1, VM2, VM3)

Introduction générale

- Concrètement c'est quoi ?

A. Pour les entreprises

- **Consolidation de serveurs** : Réduire le nombre de serveurs physiques.
- **Cloud computing** : Fournir des VM à la demande (AWS, Azure).
- **Tests et développement** : Créer des environnements isolés.

B. Pour les particuliers

- **Compatibilité** : Exécuter des logiciels Windows sur Mac/Linux.
- **Sécurité** : Naviguer dans une VM pour éviter les malwares.
- **Formation** : Apprendre Linux ou d'autres OS sans risque.

Introduction générale

- Haute disponibilité & continuité d'activité

Concepts de clustering

Un **cluster** est un groupe de serveurs physiques qui travaillent ensemble pour fournir des services hautement disponibles.

Types de clusters :

1. **Active/Active** : Tous les nœuds traitent des charges
2. **Active/Passive** : Un nœud principal, les autres en standby
3. **N+1** : N nœuds actifs + 1 nœud de secours

Introduction générale

- Haute disponibilité & continuité d'activité

Architecture de cluster VMware vSphere HA

Composants principaux :

- **Cluster** : Groupe d'hôtes ESXi
- **Shared Storage** : Stockage partagé (SAN, NAS)
- **vCenter Server** : Gestion centralisée
- **HA Agent** : Service de monitoring sur chaque hôte

Introduction générale

- Haute disponibilité & continuité d'activité

Configuration vSphere HA

Étape 1 : Création du cluster

- Ajouter les hôtes ESXi au cluster
- Activer vSphere HA
- Configurer le stockage partagé

Étape 2 : Politiques de failover

- **Nombre d'échecs tolérés** : 1, 2, 3...
- **Slot policy** : Calcul des ressources disponibles
- **Percentage policy** : Pourcentage de ressources réservées

Introduction générale

- Haute disponibilité & continuité d'activité

Configuration vSphere HA

Étape 3 : Priorités de redémarrage

- **High** : Applications critiques (redémarrage immédiat)
- **Medium** : Applications importantes (après High)
- **Low** : Applications non critiques (en dernier)
- **Disabled** : Pas de redémarrage automatique

- Haute disponibilité & continuité d'activité
- Migration à Chaud

VMware vMotion

Prérequis techniques :

- Processeurs compatibles (même famille)
- Stockage partagé (SAN, NAS, vSAN)
- Réseau de gestion configuré
- vCenter Server opérationnel

Types de vMotion :

1. **vMotion standard** : Migration de la VM en cours d'exécution
2. **Storage vMotion** : Migration du stockage uniquement
3. **Enhanced vMotion Compatibility (EVC)** : Compatibilité CPU étendue
4. **Cross vCenter vMotion** : Migration entre vCenter différents

- Haute disponibilité & continuité d'activité
- Stratégies de sauvegardes et restauration

Différence Snapshots vs Sauvegardes

Snapshots :

- Capture d'état instantané
- Stockage sur le même datastore
- Rapidité de création/restauration
- **Limitation** : Pas une vraie sauvegarde

Sauvegardes :

- Copie complète sur support externe
- Protection contre les sinistres
- Rétention long terme
- **Inconvénient** : Plus lent à restaurer

- Haute disponibilité & continuité d'activité
- Bonnes pratiques

Bonnes Pratiques et Recommandations

Conception de l'architecture

1. **Éviter les points de défaillance unique (SPOF)**
   - Redondance de tous les composants critiques
   - Chemins réseau multiples
   - Alimentation redondante
2. **Dimensionnement approprié**
   - Règle N+1 minimum pour les clusters
   - Marge de performance (20-30%)
   - Croissance anticipée
3. **Séparation des environnements**
   - Production/Test/Développement séparés
   - VLANs dédiés par environnement
   - Policies de sécurité différenciées

Introduction générale

- Un exemple concret VAGRANT

Architecture de base

Vagrant est un **outil en ligne de commande** qui s'appuie sur un **provider** (ex : VirtualBox) pour créer et gérer des VM. Il utilise un fichier de configuration appelé `Vagrantfile` (écrit en Ruby) pour définir :

- Le système d'exploitation de la VM (ex : Ubuntu, CentOS).
- Les ressources allouées (CPU, RAM, disque).
- Les scripts de provisionnement (ex : installation de logiciels).

Introduction générale

- Un exemple concret VAGRANT

La suite ici:

https://blog.stephane-robert.info/docs/infra-as-code/provisionnement/vagrant/