Module 3:

Les serveurs de l'infonuagique

L'infonuagique, qu'est-ce?

- Distributed and Cloud Computing, from Parallel Processing to the Internet of Things, 2011:

  virtualization, scalability, SOA, MOM, Security, Discovery, Databases, Grid, Peer-to-peer, Overlay networks, Fault tolerance, Ubiquitous computing, Internet of things, Wireless sensor networks, Social networks...
- John Gage, 1988, Sun Microsystems: the network is the computer!
- Jacques Gélinas, 2001: avec les vserver, chaque service est dans un serveur séparé de configuration plus générique.
- 2013, Quel modèle de serveur: cat or cow, pet versus cattle.
- Découpler les services des serveurs! Eliminer les serveurs individuels au profit de parcs de serveurs qui offrent une grande économie d'échelle.

Environnements infonuagiques

- VMWare, 1998: consolider plusieurs services sur quelques serveurs redondants grâce à la virtualisation.
- Amazon EC2, 2006: instances d'ordinateurs à louer, "Infrastructure as a service" (IaaS). Virtualisation avec Zen.
- Eucalyptus, 2008: "Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems", clone du logiciel de EC2, initialement ouvert et ensuite à base ouverte.
- OpenStack, 2010: démarré par un large consortium de compagnies de haute technologie, en réponse à Eucalyptus qui n'était plus vraiment ouvert.
- Kubernetes, 2015: les conteneurs sont plus efficaces que les machines virtuelles, pour les infrastructures privées, ou par-dessus des machines virtuelles infonuagiques.

Le marché de l'infonuagique

- Amazon demeure la référence et le leader avec environ 40% du marché. Tous reprennent ses API pour assurer une comptabilité et une transition facile.
- Microsoft a réussi à percer ce marché, avec une part d'environ 10% pour Azure, grâce à des prix agressifs (gratuit à l'essai), sa force de vente, et ses applications infonuagiques (Outlook, Office 365…).
- D'autres joueurs ont une présence importante comme Google, Alibaba et IBM.
- OpenStack (RackSpace) et VMWare (OVH) sont utilisés par plusieurs fournisseurs Internet pour offrir des services infonuagiques. Plusieurs compagnies utilisent VMWare mais sont membres du consortium appuyant OpenStack, attendant qu'il soit plus mature.
- Pour leurs systèmes internes, plusieurs compagnies utilisent Kubernetes ou des solutions équivalentes de conteneurs.

Virtualisation

- Conteneurs: plusieurs espaces de nom dans le système d'exploitation, gérés par le noyau (Linux LXC, Docker).
- Virtualisation logicielle: simulateur d'exécution (Bochs), traducteur dynamique (VMWare, Valgrind).
- Virtualisation matérielle: le processeur peut intercepter ou déléguer certaines instructions privilégiées afin de supporter efficacement la virtualisation (VMWare, KVM).
- Paravirtualisation: le système d'exploitation invité collabore en redirigeant ses requêtes vers le système hôte (Xen, VMWare, KVM).
- Hyperviseur (micro-noyau qui gère les interruptions et protections, e.g. Xen) ou système d'exploitation hôte (e.g. KVM).

Conteneurs

- Un seul noyau avec des espaces de noms séparés par conteneur pour les PID, IPC, usagers, réseau, /proc, hostname, fichiers.
- Chaque conteneur peut rouler une version différente des librairies, applications... mais il n'y a qu'un seul noyau en exécution.
- Aucun coût additionnel en performance, sauf la mémoire non partagée par les versions différentes, si c'est le cas.
- Linux LXC (aussi V-server, OpenVZ, Docker), FreeBSD jails, Solaris containers.

Simulateurs d'exécution

- Programme qui lit et interprète les instructions en simulant le matériel. L'hôte peut être un Intel et l'ordinateur simulé un ARM.
- Certains simulateurs peuvent aussi calculer le nombre de cycles écoulés et s'interfacer à GDB.
- Différentes techniques: interprétation une instruction à la fois, recompilation dynamique par segments, remplacement de certaines instructions et exécution directe des autres.
- BOCHS, QEMU, VMWare et VirtualBox sans support matériel, Valgrind.
- Environ 2 (remplacement de certaines instructions), 5 (recompilation dynamique) ou 50 (interprétation) fois plus lent.

- **Virtualisation matérielle**

  - Support matériel (Intel VT, AMD V) pour intercepter ou rediriger certaines opérations.
  - Assigner un périphérique à une VM (PCI passthrough), démultiplexer par VM les arrivées de paquets dans la carte réseau, déléguer la table de pages...
  - Linux KVM, VMWare et VirtualBox avec support matériel.
  - Entre même vitesse et 2 fois plus lent selon le degré d'E/S et d'interaction avec le système d'exploitation.

- **Paravirtualisation**

  - Le système d'exploitation est modifié pour faire un appel efficace au système d'exploitation hôte. Pas besoin d'intercepter les opérations d'accès au matériel et d'émuler le matériel.
  - Plus d'une centaine d'opérations de bas niveau du noyau Linux (lire CR0, désactiver interruptions, lire bloc, changer table de page...) sont appelées à travers la table paravirt_ops qui pointe vers la fonction native ou virtualisée.
  - Utilisé par Xen mais aussi VMWare, VirtualBox et KVM avec certains pilotes d'interface virtualisés (disque, réseau, affichage).
  - Entre même vitesse et deux fois plus lent, selon le type de charge et les opérations qui sont virtualisées ou non.

- **Hyperviseur**

  - Linux virtuel sous Windows réel ou l'inverse?
  - Hyperviseur, système d'exploitation minimal qui gère les interruptions et les accès aux périphériques, pour les répartir entre les systèmes d'exploitation des machines virtuelles.
  - Xen. Linux domaine 0 qui parle aux périphériques et Linux domaines 1, 2... qui sont les machines virtuelles invitées dont les requêtes sont passées par Xen au domaine 0.
  - Xen peut maintenant accepter des invités Windows grâce au support de virtualisation matériel.
  - Hyperviseur: solution élégante ou un OS de plus inutilement?

- **Bénéfices de la virtualisation**

  - Image logicielle isolée du matériel, utile lorsque les licenses sont attachées au matériel ou pour portabilité.
  - Possibilité de cohabitation entre plusieurs systèmes d'exploitation ou versions, plutôt que double amorçage.
  - Isolation des services à des fins de sécurité ou de gestion. Plusieurs serveurs virtuels de différents groupes peuvent coexister sur le même serveur physique.
  - Modularisation des services: démarrer les serveurs virtuels voulus: base de donnée, courriel, Web...

Coût de la virtualisation

- Certaines instructions causent des interruptions et sont émulées; moins avec le support matériel.
- Accès indirect aux périphériques; moins avec la paravirtualisation ou la virtualisation des I/O (IOMMU).
- Changements de contexte plus nombreux, application, système d'exploitation invité, hyperviseur; moins avec la délégation de tables de pages aux invités.
- Préallocation de la mémoire à chaque machine virtuelle (Xen), n'est pas toujours requis (Xen balloon, KVM).
- Surcoût d'avoir plusieurs copies en mémoire du noyau et des exécutables courants (libc, bash...); moins avec Kernel Samepage Merging.

- **Virtualisation du réseau**

  - Réseaux et commutateurs virtuels à l'intérieur d'un noeud pour connecter les noeuds virtuels; Linux TUN/TAP (network tunnel, network tap).
  - VLAN: réseau local virtuel séparé du reste du réseau local (étiquette ajoutée à chaque paquet Ethernet, gestion des diffusions générales sur le VLAN).
  - VPN/VPLS: connexions multi-point virtuelles privées par-dessus le réseau public.
  - Le résultat est un réseau dédié virtuel (overlay network); latence, bande passante, qualité de service...

- **Migration**

  - Pour équilibrer la charge ou libérer le matériel qui requiert un entretien.
  - Déplacer une image en exécution d'une machine virtuelle à l'autre; revient à migrer une machine virtuelle d'un ordinateur physique à un autre de manière transparente.
  - Contraintes de même réseau local, mêmes fichiers accessibles, pas de 64 vers 32 bits, matériel virtuel identique.
  - Copier toutes les pages de l'image en traçant celles qui sont remodifiées dans l'intervalle. Faire une seconde et possiblement troisième passe. Tout suspendre, copier les pages encore modifiées et poursuivre sur l'autre ordinateur.

- **Amazon Elastic Compute Cloud (EC2)**

  - Service offert en 2006.
  - Plusieurs catégories de noeuds.
  - Services de stockage, de base de donnée et de répartition de requêtes réseau.
  - Allocation flexible de noeuds supplémentaires.
  - Règles de filtrage réseau.
  - Différentes zones de disponibilités.
  - Outils de surveillance de la performance.

- **EC2: Virtualisation**

  - Basé sur Xen initialement, en transition vers KVM.
  - Images avec noyau paravirtualisé offertes (Amazon Machine Image, AMI): Ubuntu, Suse, Microsoft Server 2008, Solaris...
  - Possibilité de créer ses propres images Linux, en choisissant les bonnes options de paravirtualisation, et de les importer.

- **EC2: instances**

  - Instances à usage général T2 (nano à 2xlarge), M3 (medium à 2xlarge), M4 (large à 16xlarge), M5 (large à 24xlarge).
  - t2.nano: 1 vCPU, 0.5 GiO, EBS.
  - m3.medium: 1 vCPU, 3.75 GiO, 4 GiO SSD
  - m4.large: 2 vCPU, 8 GiO, EBS.
  - m5.24xlarge: 96 vCPU, 384 GiO, EBS.
  - Instances spécialisées: CPU (C3, C4, C5), mémoire (X1, X1e, R3, R4), GPU (P2, P3, G3, F1), stockage (H1, I3, D2).

- **EC2: stockage**

  - Instance storage: stockage pour la durée d'une instance EC2, avec l'image comme contenu initial de la partition racine.
  - Amazon Elastic Block Store (EBS): partition de disque pour stockage permanent qui peut être attachée à une instance à la fois.
  - Amazon Simple Storage Service (S3): stockage permanent, extensible, accessible de plusieurs instances en lecture et écriture.

- **EC2: utilisation des instances**

  - Interface Web pour commander les instances, ligne de commande ou API.
  - Définition de règles d'accès pour le groupe de sécurité contenant l'instance.
  - Sélection d'une image, d'une taille d'instance, du nombre d'instances, de la zone de disponibilité, et du groupe de sécurité, puis démarrage.
  - Des métadonnées sont disponibles pour chaque instance (paramètres, adresses, numéro d'instance...).
  - Connexion par SSH à l'instance.

- **EC2: enchère de calcul**

  - Il est possible de spécifier un prix de lancement d'instances pour un gros calcul à effectuer à bas prix.
  - Lorsque le prix est sous le seuil spécifié, les instances demandées sont démarrées.
  - Le prix fluctue sous le prix spécifié. Si le prix remonte, les instances peuvent être arrêtées.
  - Requête unique ou persistente.
  - Le prix “spot” est souvent moins de 30% du prix régulier.
  - Modèle pour utiliser les instances qui vont et viennent (nombre variable de noeuds de travail).

- **EC2: adresses**

  - Adresse interne à EC2.
  - Adresse externe visible de l'Internet (NAT), frais pour les transferts avec l'adresse externe.
  - Elastic IP address: 5 adresses statiques associées à chaque compte, peuvent être assignées à n'importe quelle instance.

- **EC2: zones géographiques**
- Choix de nombreuses régions (US-east, US-west, Europe, Asia...).
- Migration d'images ou de données d'une région à l'autre.
- Démarrage d'instances dans une zone spécifiée.
- Meilleure tolérance aux catastrophes naturelles.
- Meilleure performance en raison de la proximité du marché à desservir.
- Contraintes légales (e.g. données de citoyens européens conservées en Europe).

- **EC2: répartiteur de charge**

  - Nom de noeud (e.g. www.macompagnie.com) et numéro de port.
  - Les requêtes arrivant sur ce port sont réparties entre les instances enregistrées pour le servir.
  - Répartition entre les instances dans différentes zones de disponibilité.
  - Maintien de métriques sur le niveau de service dans le répartiteur de charge: latence, nombre de requêtes, nombre d'instances en santé...
  - Envoi de la prochaine requête à l'instance avec la charge la plus petite.
  - Arrêt d'envoi de requête aux instances non fonctionnelles.

- **EC2: le nuage élastique**

  - Maintien d'un niveau de service en surveillant le nombre d'instances valides et le taux d'utilisation du CPU.
  - Redémarrer les instances non valides.
  - Démarrer de nouvelles instances si le taux d'utilisation du CPU dépasse un certain seuil.
  - Arrêter quelques instances si le taux d'utilisation du CPU descend sous un certain seuil.

- **EC2: surveillance avec CloudWatch**

  - Métriques mesurées aux 1 ou 5 minutes et conservées pendant 2 semaines à propos des instances et volumes EBS.
  - Instance: taux d'utilisation du CPU, accès en entrée et en sortie (opérations et octets), nombre d'octets envoyés et reçus par réseau.
  - Volume EBS: accès en entrée et en sortie (opérations et octets), temps de lecture et d'écriture, temps morts, longueur moyenne de la queue.

- **EC2: nuage privé**

  - Réseau virtuel.
  - Adresses IP choisies par l'utillisateur.
  - Tables de routage.
  - Couche de sécurité supplémentaire avec listes de contrôle des accès.
  - Filtrage des sorties en plus des entrées pour chaque instance.
  - Possibilité de limiter l'accès de l'Internet à une passerelle IPSec.

- **EC2: base de donnée (RDS)**

  - Base de donnée MySQL ou Oracle disponible pour les instances.
  - Différentes tailles possibles (small à quadruple extra large) payées à l'heure et au transfert.
  - Sauvegarde et mises à jour intégrées.
  - Simplifie la conception de systèmes avec répartiteur de charge, instances de service élastiques sans données, et base de donnée centrale.

- **EC2: Discussion**

  - Permet de mettre sur pied une grappe ou un service Web de grande envergure très rapidement avec un coût initial presque nul.
  - Evite les coûts de locaux ou d'équipe d'entretien.
  - Attention, les frais d'utilisation et de transferts s'accumulent.
  - Plus cher qu'une solution maison si on possède une très grande grappe, gérée efficacement et pleinement utilisée.
  - Excellente solution pour les petites ou moyennes entreprises, la capacité excédentaire ou ponctuelle, comme plan de contingence, pour un démarrage rapide...

- **OpenStack**

  - Projet démarré en juillet 2010 par Rackspace et la NASA regroupant maintenant plus de 200 compagnies (AT&T, Ubuntu, HP, IBM, Red Hat, SUSE, Cisco, Dell, Ericsson, Hitachi, Huawei, Intel, Juniper, NEC, VMWare ...).
  - Nouvelle version environ deux fois par an, Austin en 2010, Havana en octobre 2013, Icehouse en avril 2014, Mitaka an avril 2016.
  - API compatible avec EC2.
  - Fonctionalité initialement limitée mais progresse très vite.

- **Composantes OpenStack**

  - Nova: infrastructure de calcul.
  - Neutron: infrastructure réseautique.
  - Swift: stockage de fichiers.
  - Cinder: stockage de blocs.
  - Keystone: gestion des identités.
  - Glance: création et partage des images.
  - Horizon: panneau de commande.
  - Ceilometer: collecte de métriques.
  - Heat: configuration par recettes (template).
  - Trove: base de données.
  - Marconi: service de queue et notification.
  - Savannah: service Hadoop.

- **Nova**
  - Grand nombre de noeuds avec un daemon qui les rend disponible comme noeud de calcul.
  - Serveurs Nova qui commandent les noeuds de calcul.
  - Différentes technologies de virtualisation sont supportées: KVM, XenServer, LXC (Linux containers), natif (bare metal), QEMU, VMWare ESX, Microsoft Hyper-v, IBM PowerVM.
  - Architecture répartie, hautement disponible, asynchrone.
  - Démarrage, réinitialisation, redimensionnement, suspension ou arrêt d'instances.
  - Contrôle des accès, quotas, contrôle du débit.
  - Cache locale d'images pour démarrage plus rapide.

- **Ordonnanceur dans Nova**
  - Parmi les noeuds disponibles, noter ceux qui satisfont les critères demandés pour la création d'une nouvelle instance:
    - Architecture
    - Zone géographique
    - Mémoire vive
    - Nombre de coeurs...
  - Parmi les noeuds éligibles, assigner un poids selon différents critères:
    - Charge
    - Nombre d'instances
    - Puissance du noeud...
  - Le noeud éligible avec le plus fort poids est choisi.

Nova

```
$ euca-run-instances ami-00000003 -k mykey -t m1.tiny
INSTANCE i-00000005 ami-00000003
2011-10-19T12:48:04Z unknown zone ami-00000000 ami-00000000

$ euca-describe-instances
INSTANCE i-00000005 ami-00000003 192.168.4.3 192.168.4.3 running mykey
    (proj,Openstackvalidationserver1) 0 m1.tiny 2011-10-19T12:48:04Z
    nova ami-00000000 ami-00000000

$ euca-reboot-instances i-00000005
$ euca-terminate-instances i-00000005

$ euca-get-console-output i-00000005
i-00000005
2011-10-07T07:22:40.795Z
[0.000000] Initializing cgroup subsys cpuset
[0.000000] Initializing cgroup subsys cpu
[0.000000] Linux version 2.6.32-21-server (buildd@yellow)
4.3 (Ubuntu 4.4.3-4ubuntu5) ) #32-Ubuntu SMP Fri Oct 07 09:17:34 UTC
.............
```

- **Neutron**
  - Peut commander un réseau local, des réseaux vlan, un réseau défini par logiciel (SDN) avec OpenFlow.
  - Adresses privées et publiques, statiques ou flottantes, DHCP...
  - Services additionnels comme détection d'intrusion, pare-feu, réseau privé virtuel, équilibrage de charge...

```
$ nova-manage network create private 192.168.4.0/24 1 256

$ nova-manage floating create --ip_range=10.10.10.224/27

$ euca-allocate-address 10.10.2.225

$ euca-associate-address -i i-0000008 10.10.2.225
```

- **Contrôle des accès sur le réseau**
  - Définition de groupes de sécurité avec des règles pour chacun.
  - Un groupe de sécurité est associé à une instance.

```
$ euca-add-group -d "My Servers" myservers

$ euca-authorize -P tcp -s 192.168.1.1 -p 22 myservers

$ euca-authorize -P icmp -s 192.168.1.1 -t -1:-1 myservers

$ euca-authorize -P tcp -s 192.168.1.1 -p 3389 myservers

$ euca-describe-groups
GROUP proj myservers my servers
PERMISSION proj myservers ALLOWS tcp 22 22 FROM CIDR 192.168.1.1
PERMISSION proj myservers ALLOWS icmp -1 -1 FROM CIDR 192.168.1.1
PERMISSION proj myservers ALLOWS tcp 3389 3389 FROM CIDR 192.168.1.1

$ euca-revoke -P tcp -s 192.168.1.1 -p 22 myservers

$ euca-delete-group myservers

$ euca-run-instances ami-XXXXXXXX -k mykey -g myservers
```

- **Swift**
  - Stockage permanent d'objets (fichiers).

```
$ curl -X PUT -H 'X-Auth...' http://10.10.10.2:8080/v1/AUTH_admin/mycontainer
$ curl -H 'X-Auth-Token: AUTH_tk3bb59...' http://10.10.10.2:8080/v1/AUTH_admin/
$ curl -X PUT -T file1 -H '...' http://10.10.10.2:8080/v1/AUTH_admin/mycontainer/
$ curl -H 'X-Auth-Token:...' http://10.10.10.2:8080/v1/AUTH_admin/mycontainer
$ curl -H '...' http://10.10.10.2:8080/v1/AUTH_admin/mycontainer/?prefix=fi
$ curl -H '...' http://10.10. 10.2:8080/v1/AUTH_admin/mycontainer/file1
$ curl -o localfile1 -H... http://10.10.10.2:8080/v1/AUTH_admin/mycontainer/file1
$ curl -X DELETE -H '...' http://10.10.10.2:8080/v1/AUTH_admin/mycontainer/file1
$ curl -v -X HEAD -H '...' http://10.10.10.2:8080/v1/AUTH_admin/
```

- **Cinder**
  - Les fichiers locaux d'une instance sont volatils.
  - Possibilité de créer des volumes (disque virtuel sous forme de séquence de blocs) et les associer à une instance.

```
$ euca-create-volume -s 10 -z nova
VOLUME  vol-00000002  1  creating (proj, None, None, None)
  2011-04-21T07:19:52Z

$ euca-describe-volumes
VOLUME  vol-00000001  1  nova  available (proj, server1, None, None)
  2011-04-21T05:11:22Z

VOLUME  vol-00000002  2  nova  available (proj, server1, None, None)
  2011-04-21T07:19:52Z

$ euca-attach-volume -i i-00000009 -d /dev/vdb vol-00000002

$ euca-detach-volume vol-00000002
```

- **Keystone**
  - Répertoire des usagers qui peut s'intégrer à LDAP.
  - Authentification par mot de passe ou par jetons.
  - Permissions et politiques d'accès.
  - Liste des services disponibles.

- **Glance**

  - Supporte différents types d'images: KVM (Raw, qcow2), VirtualBox (VDI), VMWare (VMDK), Hyper-V (VHD)...
  - Banque d'images déjà construites (Ubuntu, Rest Hat, SUSE, Windows...).
  - Les images sont souvent configurées avec la paravirtualisation et les pilotes associés, et avec un accès réseau (Linux avec ssh, ou Windows avec pilotes VIRTIO installés et pare-feu ouvert pour ICMP et RDP).
  - Les images peuvent être éditées en attachant les fichiers contenus (mount) ou en utilisant un interprète de commande comme guestfish.
  - Les images peuvent ensuite être ajoutées ou enlevées à OpenStack avec les commandes offertes par Glance.

- **Horizon**

  - Application Web de panneau de commande pour les utilisateurs.
  - Alternative à l'accès par programme (API) ou par ligne de commande.
  - Permet configurer les services et objets comme les instances, le réseau, le stockage, les images, les usagers...
  - Peut afficher l'état et des statistiques sur les différents services et objets.

- **Ceilometer**

  - Système modulaire et flexible pour collecter et agréger des données sur l'opération d'OpenStack.
  - Utile pour la facturation, la caractérisation pour l'application de politiques (équilibrage de charge) ou le monitoring et déboguage.
  - Un très grand nombre de métriques ont été programmées et peuvent être activées et consultées (CPU, accès lecture/écriture, accès réseau...).
  - Il est ensuite possible de consulter ou s'abonner aux métriques voulues.
  - Service d'alarme lorsque certaines métriques dépassent des seuils définis.

- **Heat**

  - Recettes (template) en format natif ou Amazon Web Services (Cloud Formation).
  - Format déclaratif qui peut facilement être mis dans un système de gestion du code source avec version.
  - Outils pour activer la configuration définie par la recette.
  - Définition des paramètres à spécifier.
  - Déclaration de la configuration en utilisant les paramètres qui devront être fournis par l'usager.

```
parameters:
  KeyName:
    type: string
  InstanceType:
    type: string
    description: Instance type for the instance to be created
    default: m1.small
    constraints:
      - allowed_values: [m1.tiny, m1.small, m1.large]
  ImageId:
    type: string
  db_port:
    type: number
    default: 50000

resources:
  my_instance:
    type: AWS::EC2::Instance
    properties:
      KeyName: { get_param: KeyName }
      ImageId: { get_param: ImageId }
      InstanceType: { get_param: InstanceType }
outputs:
  instance_ip:
    description: The IP address of the deployed instance
    value: { get_attr: [my_instance, PublicIp] }
```

- **Trove**

  - Service de base de données.
  - Accès à une base de donnée par appel de procédure à distance (RPC).
  - Création et gestion de base de données.
  - Accès à la base de donnée.
  - Un agent reçoit les appels à distance et fait faire le travail par la base de donnée.
  - Supporte différentes bases de données relationnelles ou non (MySQL, PostgreSQL...).

- **Zaqar (était Marconi)**

  - Queues de messages, publication abonnement, notification.
  - Pour par exemple: distribution de tâches à des noeuds, collecte de données, envoi de commandes à plusieurs destinataires...
  - Pour un très grand nombre de queues, messages, destinataires…
  - Utilise un API RESTful ou basé sur les Websocket.

- **Savanna**

  - Service Hadoop.
  - Semblable à Amazon Elastic Map-Reduce.
  - Réutilise les autres services de OpenStack
    - Commandé par Horizon
    - Images avec Hadoop dans Glance
    - Noeuds de calculs activés à la demande avec Nova
    - Stockage sur Swift
    - Authentification avec Keystone

- **OpenStack: Discussion**

  - Convergence d'un grand nombre de joueurs autour de quelques technologies clé (Linux, KVM, API de EC2, Réseaux virtuels).
  - Infrastructure infonuagique entièrement libre qui offre des fonctionnalités semblables à ce qui a été mis de l'avant par Amazon.
  - Progrès très rapide. Grandes différences d'une année à l'autre.
  - L'essentiel de la fonctionnalité requise est maintenant disponible.

- **Docker**

  - C’est une image, c’est un conteneur, c’est une compagnie?
  - Format d’image pour exécuter un conteneur sur Linux.
  - Environnement d’exécution pour rouler une image sur Linux… ou sur d’autres systèmes comme Windows.
  - Compagnie qui offre des outils de haut niveau pour gérer les conteneurs, au-delà des fonctions de base de Docker.
  - Le format d’image et l’environnement d’exécution ont été transférés au Open Container Initiative de la Linux Foundation qui regroupe de nombreuses entreprises.

- **Dockerfile**

  – Déclaration et recette pour créer et rouler une image.

```
FROM ubuntu:latest
RUN apt-get update
RUN apt-get install -y wget
RUN apt-get install -y build-essential tcl8.5
RUN wget http://download.redis.io/releases/redis.tgz
RUN tar xzf redis.tgz
RUN cd redis-stable && make && make install
RUN ./redis-stable/utils/install_server.sh
EXPOSE 6379
ENTRYPOINT  ["redis-server"]
```

- **Kubernetes**

  - Contribué par Google au Cloud Native Computing Foundation de la Linux Foundation en 2015.
  - Orchestration de conteneurs typiquement avec Docker.
  - Rapidement supporté par les fournisseurs d'infonuagique et très populaire pour les nuages internes aussi.
  - Peut être déployé par-dessus des noeuds natifs ou des machines virtuelles.
  - Très bonne mise à l'échelle, si c'est assez bon pour Google...

- **Kubernetes nodes (minions)**

  - Pod: groupe de conteneurs qui s'exécutent sur un même noeud pour offrir un service.
  - Noeud: machine virtuelle ou noeud physique disponible pour rouler des conteneurs, qui exécute:
    - Docker runtime: engin pour exécuter les conteneurs;
    - Kubelet daemon: processus pour gérer, arrêter ou démarrer, les conteneurs sur le noeud;
    - Kube-proxy: processus pour gérer les communications, qui redirige les requêtes au bon conteneur;
    - Cadvisor: agent qui collecte diverses métriques qui peuvent être utilisées pour le monitoring ou pour gérer les pannes et la mise à l'échelle.

- **Kubernetes master**
  - Control plane: orchestration des conteneurs, typiquement avec redondance, sur les noeuds à l'aide de:
    - Etcd: base de donnée clé-valeur, répartie et persistente, avec notification de changement, représentant la configuration désirée;
    - API server: reçoit les requêtes REST et accède etcd;
    - Ordonnanceur: choisit quel "pod" (groupe de conteneur) roule sur quel noeud.
    - Controller manager: collection de modules de commande qui gèrent l'orchestration des conteneurs pour la réplication, la mise à l'échelle...

- **Kubernetes service répliqué**

```
$ kubectl create -f svc.yaml

# Répartiteur pour le service
apiVersion: v1
kind: Service
metadata:
  name: simpleservice
spec:
  ports:
    - port: 80
      targetPort: 9876
  selector:
    app: sise
```

```
$ kubectl create -f rc.yaml
# Conteneurs répliqués
apiVersion: v1
kind: ReplicationController
metadata:
  name: rcsise
spec:
  replicas: 2
  selector:
    app: sise
  template:
    metadata:
      name: somename
      labels:
        app: sise
    spec:
      containers:
      - name: sise
        image: img/serv:0.5.0
        ports:
        - containerPort: 9876
```

Kubernetes : Discussion

- Essor très rapide car technologie mature qui répond à un besoin très présent.
- Approche typique de Google avec mécanismes simples mais puissants qui se mettent bien à l'échelle.
- Pourquoi avoir la migration de conteneur lorsqu'on a déjà la tolérance aux pannes.
- Bon pour les déploiements où on contrôle bien l'ensemble de l'application car plus efficace mais moins transparent que les VM.

Conclusion

- Les ordinateurs, comme les voitures, deviennent interchangeables; un signe de maturité.
- Le temps où un technicien s'occupait de 1 à 10 ordinateurs est révolu.
- Le gouvernement américain veut réduire le nombre de ses centres de données de plus de 1200 (sur environ 3000).
- Les ventes de serveurs sont en mutation.