

[1] 0.4pt

= = = 1 =1

vers un outils de configuration et deploiement dans le cloud 2017 koudjil mohamed
amine ***** Mme deba Université d'Oran 1 ***** Université d'Oran
1
***** Université d'Oran 1

Ceci contient les remerciements

Je dédie ce travail,
A mes chères parents
A mes frères et sœurs

Table des matières

1	cloud computing	7
1.1	introduction	7
1.2	Concepts et technologies à l'origine du « Cloud Computing » :	7
1.2.1	L'informatique utilitaire de John McCarthy :	8
1.2.2	Le concept de « grille informatique » :	8
1.2.3	Le concept de « ferme de serveurs » :	8
1.2.4	Les fournisseurs de services d'application :	9
1.2.5	La virtualisation :	9
1.3	Définition	10
1.4	L'évolution de Cloud computing :	11
1.5	Les caractéristiques essentielles de cloud computing :	11
1.5.1	Self-service à la demande :	11
1.5.2	Elasticité et rapidité :	12
1.5.3	Accès aux ressources :	12
1.5.4	Allocation des ressources :	12
1.5.5	Mesure du service :	12
1.6	Taxonomie du cloud computing	13
1.6.1	raison de développement :	13
1.6.2	Le modele de services :	14
1.6.3	L'accessibilité :	17
1.7	providers cloud computing	18
1.7.1	Amazon :	18
1.7.2	Salesforce :	19
1.7.3	Google AppEngine :	19
1.7.4	Windows Azure :	19
1.8	Conclusion :	20

2	Déploiement d'applications	21
2.1	Introduction	21
2.2	Définition du déploiement	22
2.2.1	Définition de C. Szyperski	22
2.2.2	Définition de l'Object Management Group	23
2.2.3	Définition de l'Université du Colorado	25
2.3	Travaux connexes	27
2.3.1	Les approches de déploiement	28
2.3.1.1	Les scripts	28
2.3.1.2	Langage de programmation	28
2.3.1.3	Les approches semi-automatiques	28
2.3.1.4	Les Workflow	29
2.3.1.5	Approche basée sur les modèles	29
2.4	Outils et frameworks de déploiement	30
2.5	Conclusion :	32
3	Implémentation	33
3.1	Introduction	33
3.2	Notions générales	33
3.2.1	Aspect syntaxique	33

Liste des figures

Liste des tableaux

Introduction générale

Chapitre 1

cloud computing

1.1 introduction

Nous étudions dans un premier temps le Cloud Computing d'une manière générale (Ces origines et ces définitions), dans un second temps nous décrivons les trois services principaux sur lesquels repose le Cloud Computing : applicatif, plateforme et infrastructure (Saas,Paas,Iaas) ainsi les modèles de déploiement . On termine ce premier chapitre par la présentation des principaux acteurs du Cloud Computing.

1.2 Concepts et technologies à l'origine du « Cloud Computing » :

Le Cloud Computing c'est l'ensemble de différents concepts informatiques, étudiés et développés depuis les années 60, associés à certaines technologies, récemment accessibles financièrement et matures techniquement. Dans la suite nous allons détailler 5 concepts et technologies qui nous semblent être fondamentaux pour définir les fondements du cloud computing.

1.2.1 L'informatique utilitaire de John McCarthy :

Ce concept se base sur la notion de consommation, il est proposé en 1961, lors d'une conférence au MIT (Massachusetts Institute of Technology), par John McCarthy aussi connu comme l'un des pionniers de l'intelligence artificielle (dont il proposa le nom en 1955) et pour avoir inventé du LISP en 1958. Lors de ce discours, John McCarthy suggéra que la technologie informatique partagée « time-sharing » pouvait construire un avenir prospère dans lequel la puissance de calcul et même les applications spécifiques pouvaient être vendues comme un service public. Cette idée, très populaire dans les années 60, disparu au milieu des années 70 : à l'époque, les technologies matérielles, logicielles et réseaux n'étaient tout simplement pas prêtes. Le CC met en oeuvre l'idée d'informatique utilitaire du type service public, proposée par John McCarthy.

1.2.2 Le concept de « grille informatique » :

Ce concept est apparu en 1997 lors d'un séminaire au laboratoire national d'Argonne intitulé « Building a Computational Grid », le Grid Computing est une technique de calcul consiste à partager les ressources de tout élément informatique permettant l'exécution d'une tâche ou le stockage d'une donnée numérique. Les ressources informatiques de cette infrastructure virtuelle, reliées de façon logique, sont délocalisées et autonomisées. Cette technique permet de résoudre des problèmes que l'utilisation d'un super-ordinateur ne pourrait pas les solutionner dans un temps réaliste. Il fournit d'autre service comme le débit, la disponibilité et la sécurité, Il s'inspire profusément du concept d'informatique utilitaire.

1.2.3 Le concept de « ferme de serveurs » :

Les fermes de serveurs sont apparues au début des années 2000, Ces « fermes » hébergent jusqu'à plusieurs centaines de serveurs montés en Cluster. Le « Clustering » est une technique qui consiste à regrouper plusieurs serveurs (ou « noeuds ») indépendants dans le même lieu afin de dépasser les limitations d'une machine unique. Les différents noeuds mis en réseau ensemble vont apparaître comme une

seule machine ayant plus de capacité, de puissance, de mémoire et de stockage.

1.2.4 Les fournisseurs de services d'application :

Les fournisseurs de services d'application ont aussi leur importance dans l'origine du « Cloud Computing ». Une ASP désigne une application fournie comme un service. Plutôt que d'installer le logiciel sur le poste client en ayant assuré les phases d'installation et de maintenance sur chaque poste, les applications ASP sont hébergées et centralisées sur un serveur unique et accessible par les clients à travers des protocoles standards, comme par exemple le cas avec des applications Web qui est accessibles par le protocole http : il n'y a alors plus de déploiement ou de maintenance à effectuer sur le poste utilisateur, celui-ci n'a alors besoin que d'un simple navigateur Internet. Le déploiement, la configuration, la maintenance, la sauvegarde, etc. sont désormais de la responsabilité du fournisseur du service, le client est alors consommateur.

1.2.5 La virtualisation :

La virtualisation est un concept beaucoup plus ancien qui constitue le socle du Cloud Computing. La virtualisation regroupe l'ensemble des techniques matérielles et logicielles permettant de faire fonctionner, sur une seule machine physique, plusieurs configurations informatiques (systèmes d'exploitation, applications, mémoire vive,...) de manière à former plusieurs machines virtuelles qui reproduisent le comportement des machines physiques. Les premiers travaux peuvent être attribués à IBM, qui a travaillé depuis les années 60, sur les mécanismes de virtualisation en développant dans les centres de recherche de Cambridge et de Grenoble, CMS (Conversation Monitor System), le tout premier hyperviseur. C'est donc depuis presque 50 ans que l'idée d'une informatique à la demande est présentée dans les esprits même si les technologies n'étaient jusqu'alors pas au rendez-vous pour pouvoir concrétiser cette idée.

1.3 Définition

Il existe de nombreuses définitions du terme Cloud Computing (informatique dans le nuage) et il y a peu de consensus sur une seule et universelle définition. Cette multitude de définitions reflète sur la diversité et la richesse technologique du Cloud Computing. Nous citons quelque définition :

cloud computing : la définition proposée par CISCO : 'le Cloud Computing est une plateforme de mutualisation informatique fournissant aux entreprises des services à la demande avec l'illusion d'une infinité des ressources'

pour Microsoft : L'ensemble des disciplines, technologies et modèles commerciaux utilisés pour délivrer des capacités informatiques (logiciels, plateformes, matériels), comme un service à la demande. 'Le service à la demande (vous ne payez que ce que vous utilisez) ; 'Le service est accessible n'importe où ; 'Le service est mesuré, ce qui permet de préserver les ressources ; 'La quantité est modulable à la location (élasticité infinie!) ; 'Les ressources sont mises en commun, ce qui réduit les coûts.

Pour EMC, Fournisseur de solutions d'archivage et de stockage : Le Cloud Computing permet aux utilisateurs d'accéder aux pools de ressources quand ils le souhaitent et bénéficient ainsi de l'efficacité partagée et de souplesse.

Cependant, la définition proposée par la National Institute of Standards and Technology (NIST)[6] est devenue la référence acceptée par le public. NIST définit le Cloud Computing comme étant un modèle qui permet l'accès via un réseau d'une façon simple et à la demande à un ensemble des ressources informatiques mutualisées et configurables (ex réseaux, serveurs, stockage, applications et services). Ces ressources informatiques peuvent être allouées et libérées rapidement avec le minimum d'effort de gestion ou d'interaction avec les fournisseurs de services.

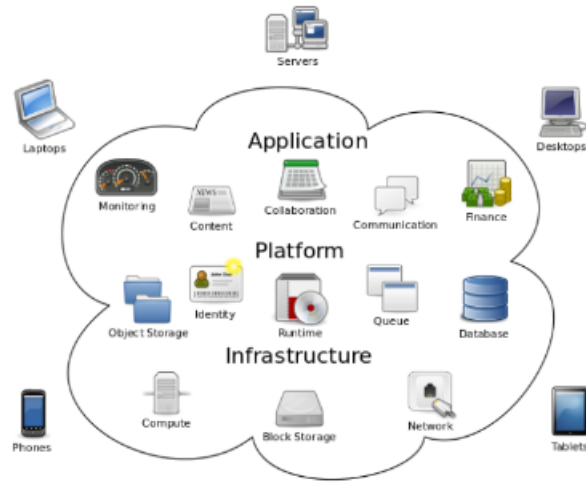


figure : cloud computing

1.4 L'évolution de Cloud computing :

Le Cloud computing est le fruit d'une évolution technologique peuvent être présentée en 5 phases :

- * Elle débute avec les Fournisseurs d'Accès Internet (FAI 1.0). Ils ont pour but de mettre en place des moyens de télécommunication pour assurer le raccordement des personnes ou entreprises au réseau internet.
- * La seconde phase est l'orientation des FAI vers l'hébergement de pages web (FAI 2.0). Cette phase marque un grand bond dans le développement d'internet.
- * La troisième phase (FAI 3.0) est la possibilité qu'offrent les FAI à héberger des applications métiers des entreprises.
- * Une connaissance des besoins applicatifs des entreprises permettent aux FAI de faire évoluer leur domaine d'intervention. Ils mettent en place des plateformes de génération d'applications à la demande. Il s'agit des ASP (Application Service Provider) dont les "Software as a Service" sont des dérivés :c'est le FAI 4.0.
- * La généralisation des pratiques précédentes, la prise en compte de nouvelles pratiques et l'intégration des principes que nous présentons dans les sections

suivantes donnent naissance au cloud computing [5].

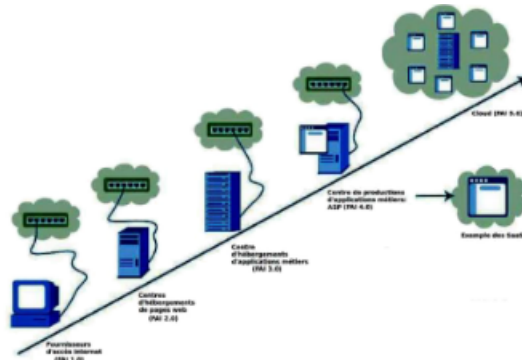


figure : evolution de cloud computing

1.5 Les caractéristiques essentielles de cloud computing :

1.5.1 Self-service à la demande :

L'utilisateur d'un service de « Cloud Computing » a la capacité (« self-provisioning ») d'approvisionner lui-même de nouvelles ressources telles que de l'espace disque, des serveurs virtuels, du temps CPU, mémoire vive RAM.

1.5.2 Elasticité et rapidité :

Les ressources peuvent être allouées ou désallouées rapidement. C'est une des caractéristiques essentielles du Cloud computing , la capacité d'augmenter et de réduire le volume des ressources utilisées en fonction des besoins, de même que libérer les ressources qui ne sont plus nécessaires, au profit d'autres utilisations, Ces opérations peuvent se faire par demande ou de façon automatique, par programmation ou triggers.

1.5.3 Accès aux ressources :

Les ressources sont disponibles via le réseau Internet, ou via l’Intranet dans le cas d’un Cloud privé. Les ressources sont accessibles via des protocoles standards (TCP/IP,SSL, HTTP,...)L’accès aux ressources peut se faire à partir d’un grand nombre de périphériques clients (ordinateurs, portables, téléphones mobiles, smart-phones, ...)et depuis n’importe quelle plate-forme (Windows, Unix, MacOS, Linux, systèmes propriétaires,')

1.5.4 Allocation des ressources :

Les ressources mises à disposition par les providers de « Cloud Computing » sont mutualisées pour répondre aux besoins de plusieurs clients dans une architecture multi-tenant. Une architecture « multi-tenant » ou « multi-locataire » fait qu’une seule instance d’une application est adaptée aux besoins de tous les utilisateurs et offre malgré tout un certain niveau de customisation pour s’adapter de manière individuel aux différents clients.

1.5.5 Mesure du service :

Toutes les ressources allouées peuvent être surveillées et contrôlées de manière automatique ainsi que l’utilisation optimisée de ressources est assurée en s’appuyant sur le modèle “payez uniquement ce que vous consommez”.

1.6 Taxonomie du cloud computing

Avant de présenter les différents types de cloud pouvant être développés, nous établissons dans un premier temps quelques critères de classification :

1. La raison de développement (business model) : c’est la raison qui justifie la mise en place de la plateforme. Elle peut être commerciale, scientifique ou communautaire...ex.
2. Le modele de services :c’est le modele de service que peut être délivré par le cloud aux clients.

3. L'accessibilité :Le cloud peut être accessible par tous ("cloud public") ou restreint à un public particulier ("cloud privé"), C'est la raison qui justifie le déploiement de la plateforme de cloud.

1.6.1 raison de développement :

L'utilisation du CC ne se limite pas uniquement aux entreprises à caractère commercial. En fonction des raisons de sa mise en place, nous distinguons quatre catégories de plateformes de CC à savoir :

Cloud d'Entreprises :Dans cette catégorie, nous retrouvons des entreprises de petites et de moyennes tailles disposant chacune de peu de ressources et de moyens de maintenance de leurs infrastructures. Elles se regroupent donc autour d'un projet de cloud afin de mutualiser leurs capacités.La plateforme qui en découle est privée, c'est-à-dire accessible uniquement par les entités des différentes entreprises. Cette plateforme à l'avantage d'être de petite taille et d'accès restreint à des utilisateurs connus.

Cloud Gouvernemental et Recherche Scientifique :des instituts de recherche mettent leur pieds sur des environnements de cloud pour des raison de recherche scientifique de developement ,L'accès est réservé aux personnes appartenant aux instituts de recherche associés.

Cloud pour Réseaux Sociaux et Jeux :Le développement des réseaux sociaux et des jeux en ligne nécessite de plus en plus de grandes quantités de ressources.ca implique la développement d'une plateforme similaire au cloud devient une évidence pour optimiser l'utilisation des ressources et faciliter le partage de données. En effet, elles sont considérées comme un cloud .

Cloud pour Fournisseurs de Services :C'est le modèle le plus répandu. Une entreprise, appelée fournisseur, met à la disposition d'autres (appelées clients) une plateforme d'exécution d'applications et assure le service informatique inhérent.

1.6.2 Le modele de services :

Le but principal du Cloud est d'offrir des services à des utilisateurs ,suivant différents modèles. NIST précise que le Cloud Computing a trois modèles de services principaux Iaas,Paas,Saas

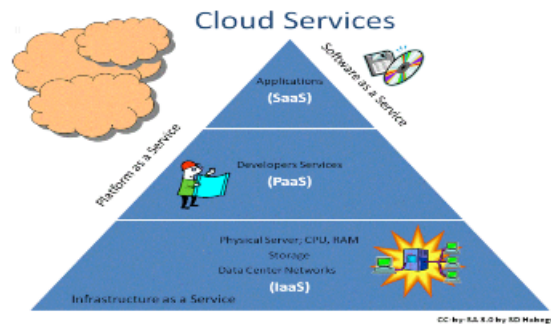


figure : modele de srvice

Infrastructure en tant que service (infrastructure as a Service « Iaas ») : Dans ce modèle, l'infrastructure physique (le matériel réseau, le matériel serveur, la plate-forme de virtualisation, les moyens et capacités de stockage) est « dématérialisée » et hébergée. Le fournisseur procure donc une couche matérielle (serveurs, réseau, stockage, hyperviseur, solution de supervision, solution de management) sur laquelle les clients vont pouvoir déposer leurs environnements système et leurs applications. Mais ce service va encore plus loin grâce à la virtualisation. L'utilisation de cette technologie permet aux clients de créer leur propre infrastructure personnalisée (serveurs virtuels, réseau virtuel, stockage) en quelques clicks. Cette infrastructure est par ailleurs extrêmement flexible, accessible sans restriction et configurable en temps réel. Les clients n'ont pas à se soucier de la scalabilité de leur infrastructure, cette tâche étant gérée par le fournisseur. Celui-ci gère également tous les coûts de gestion liés au fonctionnement du matériel (électricité, climatisation, etc) ainsi que le contrôle de la consommation s'il y a une facturation à l'usage (au Go, au temps d'utilisation, etc). Il y a un avantage principal dans ce niveau de service. Le client n'a plus à construire son propre Datacenter, ni à gérer l'infrastructure physique et les coûts qui lui sont inhérents. L'ingénieur ou

l'administrateur système peut se reconcentrer sur l'optimisation de son environnement système, et le développeur sur ses applications. En fait, le client a un contrôle total de son Datacenter virtuel sans se soucier de son élasticité, ni de l'infrastructure physique qu'il y a derrière. Mais ce modèle a également des inconvénients. Déjà, comme pour toute infrastructure informatique classique, il est indispensable d'avoir un administrateur ou un ingénieur systèmes dans son entreprise. Et enfin, malgré la facilité technique de création d'une infrastructure personnalisée grâce à la virtualisation, un important travail de réflexion et d'expertise préalable reste à fournir pour sa mise en oeuvre. Plusieurs offres existent dans cette catégorie comme Amazon Elastic Compute Cloud (EC2) ainsi que d'autres ElasticHosts, GoGrid, Rackspace Cloud, RightScale, Skytap et Orange Business. Plateforme en tant que service (Platform as a Service « PaaS ») : Le PaaS fournit un niveau d'abstraction supplémentaire par rapport à l'IaaS. Dans cette catégorie, non seulement l'infrastructure est dématérialisée, mais aussi le système d'exploitation, et la plateforme d'exécution, de déploiement et de développement d'application. Le fournisseur procure donc aux clients développeurs l'infrastructure, le système d'exploitation, les bases de données, la couche middleware, et une plateforme de développement complète, fonctionnelle et performante. Ces plateformes sont équipées d'outils de développement, de modules, d'un langage de programmation, d'un type de base de données. Le client développeur peut utiliser cette plateforme pour héberger, développer et/ou exécuter des applications. L'avantage pour le développeur est qu'il ne se soucie pas du matériel. Il peut développer, déployer puis exécuter son application sans avoir à gérer, ni les technologies sous-jacentes nécessaires, ni les configurations matérielles. Le cycle de développement est fortement réduit et le client peut se concentrer sur son application.

Plateforme en tant que service (Platform as a Service « PaaS ») : Le PaaS fournit un niveau d'abstraction supplémentaire par rapport à l'IaaS. Dans cette catégorie, non seulement l'infrastructure est dématérialisée, mais aussi le système d'exploitation, et la plateforme d'exécution, de déploiement et de développement d'application. Le fournisseur procure donc aux clients développeurs l'infrastructure, le système d'exploitation, les bases de données,

la couche middleware, et une plateforme de développement complète, fonctionnelle et performante. Ces plateformes sont équipées d'outils de développement, de modules, d'un langage de programmation, d'un type de base de données. Le client développeur peut utiliser cette plate-forme pour héberger, développer et/ou exécuter des applications. L'avantage pour le développeur est qu'il ne se soucie pas du matériel. Il peut développer, déployer puis exécuter son application sans avoir à gérer, ni les technologies sous-jacentes nécessaires, ni les configurations matérielles. Le cycle de développement est fortement réduit et le client peut se concentrer sur son application. L'inconvénient apparaît lorsque l'on veut déplacer une application d'une plateforme à une autre. La compatibilité n'est pas avérée car en fonction des solutions les langages sont différents. Il faut choisir sa plateforme en fonction de son langage, et ensuite y rester.

Logiciel en tant que service (Software as a Service « SaaS ») : logiciel en tant que service (SaaS) offre des applications complètes fournies à la demande. Ce type de service fournit différents types d'applications telle que webmail, suite de bureautique en ligne ainsi que les réseaux sociaux et les jeux. Ces applications s'exécutent sur les infrastructures du provider. Elles sont hébergées sur le cloud et accessibles via un navigateur Internet. L'utilisation des services SaaS est plus simple que les autres services où la facturation s'adapte dynamiquement avec la consommation et la paramétrisation des services offerts sont limités. L'utilisateur n'a aucun souci sur l'installation des logiciels ou de leur mise à jour et il n'a aucun contrôle sur l'infrastructure du Cloud tel que serveurs virtuel, les composants réseaux, l'emplacement de stockage, la version de l'application et les fonctions d'application disponibles [14]. On peut distinguer deux types principaux de logiciels ou applications en tant que services :

- Des applications sont disponibles au public général totalement gratuit, par exemple Services Gmail et Facebook, où les e-mails, pièces jointes des email, photos, vidéos et musiques. Généralement ce type d'application est indirectement financé par la publicité, ou par des produits dérivés de l'analyse statistique à grande échelle.

- des applications sont disponibles pour répondre aux attentes des entreprises payantes selon la consommation. Ces applications sont conçues pour fournir des logiciels faciles à utiliser

1.6.3 L'accessibilité :

En plus des modèles de livraison qui permettent de concrétiser les services Cloud, on trouve un ensemble de modèles de déploiement de services basés Cloud computing. Ces modèles permettent de définir le degré d'accès de l'utilisateur final aux fournisseurs de services Cloud. Ces modèles sont divisés en quatre grandes catégories d'après NIST[1].

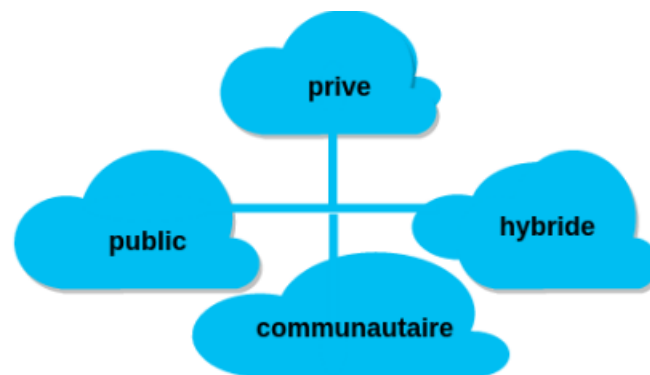


figure : modele de deployment

Le Cloud privé : Un cloud privé est une infrastructure de cloud n'est utilisé que par une seule organisation. Le cloud privé peut être gérée par l'organisation elle-même ou par un prestataire de service qui peut être située dans les locaux de l'organisation ou à l'externe chez le prestataire de services. L'avantage supplémentaire du cloud privé c'est le contrôle total des données par l'organisation, d'autre façon il permet la confidentialité et la sécurité des données grâce aux ressources qui ne peuvent pas être partagées par d'autres organisations [3].

Le Cloud communautaire : L'architecture est dédiée à une communauté professionnelle spécifique, pour permettre de travailler de manière collaborative sur un même projet.

Le Cloud public : Le cloud public est accessible publiquement à tous les particuliers et les groupes industriels. Son propriétaire est un fournisseur de service IaaS, PaaS ou SaaS. Les consommateurs ne sont facturés que pour les applications, les services ou les données qu'ils utilisent. Les ressources sont illimitées et il n'y a donc aucun investissement initial. Les « Clouds » publics sont généralement exploités par des fournisseurs de cloud commerciaux comme Amazon, Google, Microsoft, GoGrid.

Le Cloud hybride : Le cloud hybride est une composition de deux ou plusieurs infrastructures de Cloud publics et privés. Généralement les données sensibles sont gérées au niveau interne par l'organisation ou chez le prestataire de cloud privé et l'autre type de données « non-sensibles » sont gérées par le cloud publique

1.7 providers cloud computing

L'existence du « cloud computing » et sa valeur pour les consommateurs sont aujourd'hui plus évidentes pour le grand public, les différents acteurs du monde IT comme Google et Microsoft proposent des services de Cloud Computing.



figure : cloud provider

1.7.1 Amazon :

« Amazon Web Services » (AWS) met à disposition un cloud public depuis 2006. Au départ le but d'Amazon était de rentabiliser ses énormes infrastructures requises pour assumer les montées en charge pendant la période de Noël sur leur boutique en ligne. Aujourd'hui Amazon propose de nombreux services en ligne, à commencer par l'IaaS probablement le plus connu : Elastic Compute Cloud (EC2).

1.7.2 Salesforce :

Salesforce.com est une société qui a lancé dès 2003 des offres de Cloud public. C'est officiellement le plus ancien prestataire dans ce domaine. Aujourd'hui encore, leurs offres sont uniquement composées de Cloud Public, et adressées aux entreprises (surtout les grands comptes). Les outils proposés sont tournés vers le travail collaboratif, la gestion des ventes et le marketing relationnel.

1.7.3 Google AppEngine :

Google mis et propose beaucoup de services PaaS et SaaS sur le cloud computing . A grande échelle, les solutions de Google dans le cloud sont surtout connues des consommateurs privés au travers des ses Google Apps telles que Google Docs, Calendar ou encore Gmail. Toutes ces « web apps » sont dans le domaine du SaaS et gratuites pour une utilisation privée. Google App Engine dont la première version beta est sortie en avril 2008 est le service PaaS de Google. Au départ le service ne supportait que le développement d'applications en Python. Depuis, le support de Java Virtual Machines (JVMs) a été ajouté et permet de développer des applications non seulement en Java mais aussi au moyen de JRuby, JPython, Scala ou Clojure. Le SDK (software development kit) inclut un environnement complet de développement qui simule App Engine en local sur le bureau du développeur (Rhoton, 2010). La plateforme inclut également des services sous formes d'API permettant de manipuler des images, d'envoyer des mails ou encore d'utiliser les comptes Google pour les identifications au sein de l'application.

1.7.4 Windows Azure :

Microsoft s'avère être actuellement l'un des seuls acteurs au niveau mondial à proposer des offres sur les trois modèles SaaS, PaaS et IaaS. Les principales solutions SaaS sont Office 365 et Microsoft Online Services. Windows Azure est le service PaaS phare de Microsoft. Avec la plateforme il est possible de migrer une application existante ou de développer dans différents langages de programmation. La plateforme met également à disposition le service SQL Azure, un service de base de données basée sur Microsoft SQL Server. Au niveau IaaS, Microsoft propose une solution de cloud privé qui repose sur Windows server.

1.8 Conclusion :

Le Cloud Computing se positionne actuellement en tête de liste des nouvelles technologies. Il se caractérise par son extensibilité et élasticité et son exploitabilité par un grand nombre d'utilisateurs dans le monde entier. Il offre une grande puissance de calcul et espace de stockage, comme toute innovation technologique qui se respecte, le nuage informatique fait économiser de l'argent, le coût, le temps d'utilisations des logiciels, leur développement et leur installation.

Chapitre 2

Déploiement d'applications

2.1 Introduction

Le Cloud Computing se positionne actuellement en tête de liste des nouvelles technologies. Il se caractérise par son extensibilité et élasticité et son exploitabilité par un grand nombre d'utilisateurs dans le monde entier.

Des nombreux individus et organisations ont choisi de déplacer leurs serveurs ou applications dans un environnement en nuage afin d'optimiser l'utilisation de leur infrastructure informatique ,passage à l'échelle, haute disponibilité, etc.., L'exploitation des ressources d'infrastructure dans un environnement partagé, ou mis en commun, permet de générer des économies de coûts,le déplacement d'une application correspond au processus de redéploiement de celle-ci, généralement sur de nouvelles plates-formes et au sein d'une nouvelle infrastructure.

D'une manière générale, il existe deux modèles de service cible potentiels pour la migration d'une application Infrastructure en tant que service (IaaS) et Platform as a Service (PaaS). pour deployer un application chez les cloud provider il faut la bien configurées et dimensionner l'environnement d'exécution,les dépendances de l'application pour prendre en compte l'hétérogénéité et l'élasticité inhérentes au paradigme du Cloud Computing .

Notamment, il existe un grand nombre de provider ayant des niveaux de fonctionnalité et des possibilités de dimensionnement hétérogènes parmi les différentes

solutions de cloud disponibles (Amazon EC2, Rackspace cloud , WindowsAzure). Lors d'un déploiement d'application sur un PaaS, il faut choisir un serveur d'application, la fréquence CPU nécessaire à l'exécution, la base de données à utiliser, sa capacité en terme espace mémoire, etc. Cette variabilité en termes de configuration et de dimensionnement offre une multitude de possibilités de configurations qui sont généralement effectuées de manière ad hoc et sont sources d'erreurs lorsqu'elles sont configurées à la main[10].

2.2 Définition du déploiement

Après avoir développé un application , vous pouvez le déployer, nous présentons trois grandes sur le déploiement : Nous commencerons par la définition de Clemens Szyperski [13], puis nous poursuivrons par celle proposée par l'Object Management Group (OMG) [8] et enfin, nous finirons par la définition de l'université du Colorado [2]. [HHC+98].

2.2.1 Définition de C. Szyperski

La définition du déploiement proposée par C. Szyperski c'est une définition générale du déploiement pour le domaine des composants logiciels, sans pour autant préciser si elle se limite au déploiement sur le cloud computing ou non. Le déploiement est défini comme étant l'étape de préparation d'un composant en vue de l'installer dans un environnement spécifique. C. Szyperski précise explicitement que le déploiement (c'est-à-dire, la préparation) revient à renseigner les paramètres d'un descripteur de déploiement [13].

Nous présentons dans cette section les étapes qui précèdent et qui suivent cette phase de déploiement. Cela va nous permettre de préciser où cette phase de déploiement est située dans le cycle de vie logiciel et quelle est sa portée. L'étape qui précède le déploiement est l'acquisition. Cette étape permet d'obtenir un composant logiciel. C. Szyperski énonce que tout composant acquis est déployable. Ce qui signifie qu'un composant proposé à l'acquisition est produit exécutable par une machine physique ou virtuelle. Plus précisément, dans [13], l'auteur met en avant le fait que

tout composant doit être une unité de déploiement (page 686). Il définit ensuite une unité de déploiement comme un livrable exécutable dans un environnement d'exécution, sans besoin d'intervention humaine pour modifier le composant afin de le rendre effectivement installable et prêt à être exécuté. L'installation est l'étape qui suit immédiatement le déploiement. Elle rend un composant disponible sur un site (host) particulier, dans un environnement particulier. Il est précisé que cette étape d'installation est souvent automatisée. Enfin, l'étape d'installation est suivie par le chargement (Loading) qui consiste à lancer l'exécution d'un composant dans un contexte d'exécution particulier.

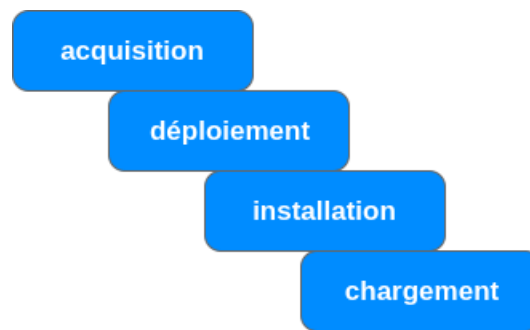


figure :la phase de déploiement et son context selon Szyperski

2.2.2 Définition de l'Object Management Group

La phase de déploiement est constituée de cinq étapes, à savoir, l'Installation, la Configuration, le Planning, la préparation et le Launch. Ces cinq étapes sont structurées les unes après les autres de manière linéaire et séquentielle.

La première étape est l'installation. Elle est définie comme la récupération, l'acquisition d'un package logiciel (software package) publié et son rapatriement dans un dépôt logiciel sous le contrôle du déployeur. Un dépôt peut se situer, ou non, sur la même machine ou sur le même système de fichier que celui dans lequel le logiciel va être déployé. C'est une zone qui permet au déployeur d'appliquer des politiques au logiciel, comme l'authentification ou la certification (du logiciel), avant de lancer toute étape concernant l'exécution du logiciel à proprement dit. Un dépôt ne doit pas nécessairement être persistant et ne doit pas, non plus, nécessairement être stocker ou avoir une copie du logiciel ou des métadonnées du logiciel. L'OMG propose, aussi,

sa propre définition de l'activité d'installation. Ainsi, l'installation n'est pas définie comme le déplacement d'un logiciel vers l'environnement d'exécution où il pourra ensuite être activé, mais simplement comme le déplacement du logiciel acquis vers le dépôt du déployeur.

La seconde étape du processus de déploiement est la configuration. Pour réaliser une configuration, un package logiciel doit être installé dans le dépôt du déployeur. Le déployeur est le seul à pouvoir effectivement le configurer. Les possibilités de configuration offertes dépendent de chaque package logiciel. Un package logiciel peut, par exemple, offrir des options de configuration concernant la langue à utiliser, le système de mesure, le délai d'attente entre chaque mesure ou encore la fréquence et le formatage des rapports de mesures générés. Plusieurs configurations peuvent être associées à un même package logiciel. Enfin, cette seconde étape ne concerne que la configuration des fonctions du package logiciel et n'est en aucun cas destinée à la prise de décisions concernant le déploiement comme, par exemple, le choix de l'implémentation à utiliser ou, le cas échéant, le choix de la distribution des différents éléments du package logiciel.

La troisième étape s'articule sur la planification (Planning). Son but est de définir comment et où une configuration va être exécutée dans l'environnement cible. Pour rappel, la spécification définit un environnement comme étant une infrastructure composée d'ordinateurs et de réseaux. L'activité de planification prend en compte les requis de la configuration à déployer, ainsi que les ressources offertes par l'environnement cible et sélectionne les implémentations à utiliser. Elle décide également comment et où la configuration donnée sera exécutée dans l'environnement cible. Le résultat de l'étape de planification est un plan de déploiement spécifique (à la configuration en cours de déploiement, ainsi qu'à l'environnement cible). Le déploiement ainsi mis en $\frac{1}{2}$ uvre est un déploiement sensible au contexte.

L'étape avant dernière étape est nommée préparation. Elle est définie comme la mise en $\frac{1}{2}$ uvre effective des décisions prises lors de la planification. Ces décisions sont spécifiées dans le plan de déploiement. Le but de la préparation est de rendre le logiciel prêt à être exécuté. Cette mise en œuvre consiste, entre autre, à déplacer des binaires exécutables dans les ordinateurs qui leur ont été attribués. Comme pour la planification deux modes de préparation sont distingués. Le premier est le mode

juste à temps, le second est le mode par avance. Le mode juste à temps implique que l'étape de préparation doit immédiatement être suivie par l'étape de lancement. Le mode par avance n'impose pas cette contrainte. La dernière étape est le lancement (Launch). Elle consiste à démarrer l'exécution du logiciel, en collectant, toutes les ressources requises par le ou les packages logiciels dans l'environnement d'exécution. Afin de lancer une application à base de composants ,il faut instancier des composants dans les ordinateurs de l'environnement cible, la configuration de ces instances de composants et la mise en place des interactions entre les différentes instances et enfin, le démarrage de l'exécution de l'application en elle-même. Dès que l'application s'exécute alors, soit l'application s'exécute malgré que son exécution n'est pas complète, soit elle est arrêtée (c'est-à-dire désactivée) en utilisant la même infrastructure de déploiement que celle par laquelle son lancement a été effectué [lien vers le mémoire].

Les phases que nous venons de présenter peuvent tout à fait être déroulées une par une séparément, ou comme une seule grande étape de déploiement entièrement automatisée.

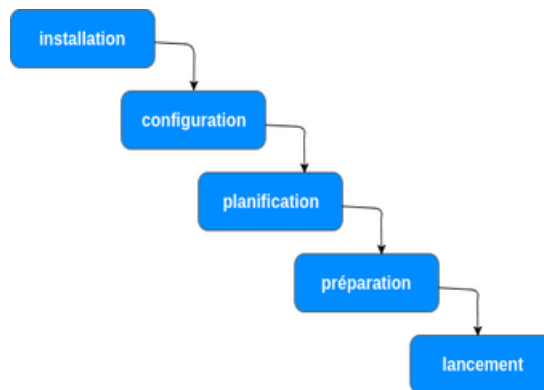


figure :la phase de déploiement et son context selon OMG]

2.2.3 Définition de l'Université du Colorado

Nous allons présenter maintenant la définition de la phase de déploiement proposée dans [2].la phase de déploiement est un processus, qui contrairement aux propositions de C.Szyperski et de l'OMG n'est ni linéaire, ni séquentiel. Il est formé de huit

activités distinctes et de plusieurs transitions, le tout formant une phase de déploiement minimale qui peut être considérée comme générique. Nous allons maintenant nous focaliser sur chacune des huit activités qui sont proposées. Ces huit activités sont la mise à disposition (release), l'installation (install), l'activation(activate) , la désactivation , la mise à jour , l'adaptation , la désinstallation . Il est intéressant de remarquer que cette phase de déploiement est plutôt symétrique, ainsi, derelease, deInstall et desactivate sont, respectivement, l'inverse de release, Install et activate. La première activité est la mise à disposition. Cette activité fait l'interface entre le processus de développement du logiciel et le processus de déploiement du logiciel. Elle englobe toutes les actions nécessaires pour préparer un système logiciel afin qu'il puisse être correctement assemblé au niveau du site du consommateur. Le résultat de cette activité est un package qui contient les composants du système, ses dépendances et ses contraintes, ainsi que les informations nécessaires aux autres activités de déploiement. Enfin, l'activité de mise à disposition est aussi chargée d'informer les consommateurs de la mise à disposition du système. La seconde activité est l'installation. Cette activité permet d'ajouter le système requis sur le site d'un consommateur. C'est l'activité la plus complexe du processus de déploiement puisqu'elle doit trouver et assembler toutes les ressources nécessaires au système. Concrètement, l'installation d'un système commence par deux étapes. La première étape est l'interprétation des informations fournies par le package du système. La seconde étape est l'inspection du site cible afin de configurer correctement le système logiciel.

La troisième activité est l'activation. Cette activité consiste à démarrer les éléments d'un système afin de rendre le système utilisable (c'est-à-dire afin de lancer l'exécution du système). Pour un système simple, l'activation peut se résumer à l'appel d'une méthode de l'environnement cible. par-contre, pour un système complexe, il peut être nécessaire d'activer un certain nombre d'éléments de l'environnement, tels des serveurs ,avant de pouvoir activer le système en lui-même.

La quatrième activité est la désactivation. Cette activité est applicable pendant l'activation. Le but de la désactivation est d'arrêter l'exécution d'un système, cet arrêt se faisant via l'arrêt des différents éléments du système. L'activité de désactivation est souvent requise avant que d'autres activités comme la mise à jour statique puissent commencer.

La cinquième activité est la mise à jour. Les auteurs présentent cette activité comme étant un cas particulier d'installation. Cependant, cette activité est moins complexe que l'installation, puisque la plupart des ressources nécessaires au système ont déjà été obtenues durant l'installation du système. Deux situations possibles se présentent pour la mise à jour. La première situation nécessite la désactivation du système, puis sa mise à jour et enfin sa réactivation. La seconde situation ne nécessite pas la désactivation du système. Certains systèmes peuvent, en effet, être mis à jour tout en restant actifs. Enfin, tout comme l'installation, la mise à jour inclut aussi les sous-activités de transfert de packages et de configuration des éléments.

La sixième activité est l'adaptation. Comme la mise à jour, elle permet de modifier un système logiciel précédemment installé. Cependant, l'activité d'adaptation diffère de l'activité de mise à jour puisque, contrairement à cette dernière qui est initiée par des événements distants (par exemple, par une commande du consommateur), l'adaptation est, quant à elle, initiée par des événements locaux (par exemple, par un changement dans l'environnement dans lequel le système est déployé). Une adaptation peut, par exemple, être initiée afin de maintenir la correction d'un système logiciel. L'avant dernière activité de la phase de déploiement est la désinstallation. Elle consiste à supprimer un système d'un environnement donné. Pour qu'un système puisse être désinstallé, il faut qu'il ait été préalablement désactivé. L'activité de désinstallation n'est pas nécessairement triviale. En effet, lors de la désinstallation d'un système, il convient de porter une attention toute particulière aux ressources que le système en cours de désinstallation partage avec d'autres systèmes. Lors de la désinstallation du système, les dépendances d'autres systèmes vers le système ciblé par la désinstallation c'est un paramètre qui doit être pris en compte,

Enfin, un système peut être marqué comme obsolète, ce faisant, le producteur arrête de le maintenir. L'activité de fin de support n'est pas similaire à l'activité de désinstallation puisque, dans le cas d'une fin de support, le système devient indisponible pour de futures installations, mais n'est pas désinstallé des sites sur lesquels il est déjà déployé. De plus, l'activité de fin de support est aussi en charge d'informer tous les consommateurs du système de la fin du support.

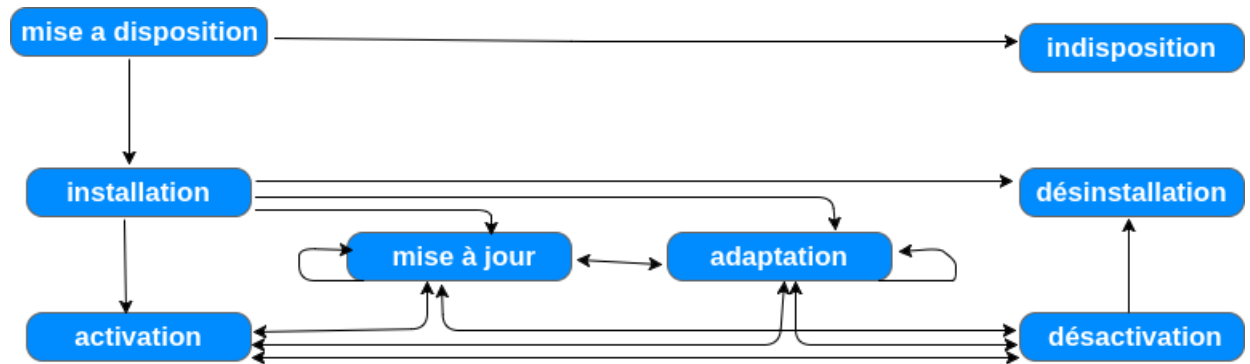


figure :La phase de déploiement définie par l'Université du Colorado.

2.3 Travaux connexes

Nous présentons dans cette section les principaux approches de déploiement automatique d'applications sur le cloud existent dans la littérature , ainsi nous citons les solutions les plus réponsus actuellement

2.3.1 Les approches de déploiement

Il existe plusieurs solution « ?outils et frameworks » de déploiement automatique des logiciel dans le cloud computing s'implique l'existence de plusieurs approche de base derrière ces solutions.

2.3.1.1 Les scripts

Pour faire face au déploiement automatique, différents outils et solutions utilisent la codification manuelle par le biais de scripts, ce qui nécessite plus de temps pour déployer le logiciel. Ce type d'approche réduit le risque d'erreur humaine lors du processus de déploiement manuelle, et pour le cas de développeur, il s'occupe de développement d'application au lieu la configuration du cloud. Le seul inconvénient de ces approches de déploiement, c'est qu'il augmente les coûts associés à la codification, le temps et les efforts humains [RJdRSM16].

2.3.1.2 Langage de programmation

Nous avons également noté d'autres solutions utilisant le déploiement automatisé de services dans le cloud à l'aide de langage de programmation. En comparaison avec les approches présentées à l'aide des scripts, ces approches sont plus avancées en fonction de l'effort humain et le temps appréciée dans le processus de déploiement.

2.3.1.3 Les approches semi-automatiques

En outre, les modèles virtuels proposés par Disnix [Disnix] ont présenté des mécanismes semi-automatisés pour le déploiement. En d'autres termes, ces approches nécessitent encore que certaines étapes sont effectuées manuellement pendant le processus de déploiement. La solution proposée par Ardagna [Ardagna] a présenté une approche partiellement orientée et semi-automatisée pour le déploiement de logiciel. Elle exige un certain niveau de la compréhension de l'utilisateur final sur les détails de la structure des clouds et une charge lourde de l'information dans les modèles de déploiement demandé à l'utilisateur.

2.3.1.4 Les Workflow

Cette solution propose que les développeurs aient les services à déployer pour créer les modèles UML du déploiement de logiciels. Ces modèles définissent tous les informations requises pour le déploiement (machines virtuelles, services, applications, systèmes d'exploitation des machines virtuelles, des bases de données, fournisseur de services et les clés d'accès) en tant que paramètres sans la nécessité du codage de la configuration d'une machine virtuelle. Ces modèles sont transformés pour générer le code de déploiement à l'aide d'outils spécialisés. le code de déploiement à l'aide d'outils spécialisés.

2.3.1.5 Approche basée sur les modèles

L'objectif est de déployer le logiciel à un niveau plus élevé d'abstraction pour réduire les efforts humains et le temps passé à effectuer la tâche de déploiement, car l'approche basée sur un modèle est une meilleure façon d'augmenter la productivité

de développeur. Cette approche est la proposition de l'OMG qui est en cours de standardisation. La spécification de l'OMG, elle a pour objectif de fournir un modèle de données et d'exécution permettant de gérer le développement, le packaging, le déploiement et la configuration d'applications à base de composants. La spécification est décrite à travers une entité appelée "Platform-Independent Model" (PIM), composée d'un ensemble de modèles UML et de règles sémantiques associées. Le PIM est indépendant de tout modèle de composant particulier. Pour utiliser cette spécification avec un modèle particulier de composant, il faut créer une entité appelée "Platform-Specific Mapping" (PSM). Le PSM est un ensemble de règles qui transforme les modèles UML du PIM en données et modèles d'exécution, dans un format approprié pour le déploiement du modèle de composant cible. La spécification n'a pour l'instant standardisé que le PSM pour le modèle de composant Corba, dans lequel les modèles de données et d'exécution sont transformés en deux formats : XML schéma pour le stockage sur disque et l'échange entre outils, et IDL (Interface Definition Language) pour la représentation du modèle d'exécution et des communications entre les entités du déploiement.

2.4 Outils et frameworks de déploiement

D'après une étude bibliographique sur les travaux qui s'articulent sur le déploiement automatique des logiciels sur le cloud computing, nous avons sélectionné les principaux outils et frameworks de déploiement afin de les décrire :

1. L'outil SALOON est un outil de configuration et de déploiement. Basé sur des ontologies, SALOON permet de spécifier une configuration technique pour l'application à déployer et de dimensionner cette configuration
2. La solution de Franklin est basée sur les modèles pour le déploiement automatique des logiciels sur le cloud computing. Cette solution ne nécessite que le développeur d'avoir des connaissances à propos de la clé d'accès et le nom du fournisseur de services, étant donné que les détails spécifiques sont distraits. L'objectif est de déployer le logiciel à un niveau plus élevé d'abstraction et réduire les efforts humains et le temps passé à effectuer pour le déploie-

ment des tâches, car l'approche basée sur les modèles est une meilleure façon d'augmenter la productivité du développeur[1] .

3. OSGi fournit un environnement d'exécution, basé sur la technologie Java. Le processus de déploiement inclut les activités suivantes : l'installation, la mise à jour, la désinstallation. Il fournit un cadre qui permet le déploiement d'applications Java, extensibles et téléchargeables (appelées "bundle"). "Un bundle" est constitué de classes Java et d'autres ressources (bibliothèques, fichiers, etc.), l'ensemble pouvant fournir un ou plusieurs services aux utilisateurs. Ils sont déployés sous la forme d'archives JAR (Java ARchive). Les "bundles" sont les seules entités utilisées pour le déploiement d'applications. Les appareils OSGi compatibles peuvent télécharger, installer, supprimer les "bundles". L'installation et la mise à jour se font de manière dynamique, en gérant les dépendances entre les "bundles" et les services. Les limitations de ce modèle sont liées au fait qu'il est spécifique à l'environnement Java et à des applications non distribuées [4].
4. moSaIC Le projet européen de recherche mOSAIC [7] propose une API et une plateformePaaS libre permettant de développer et déployer des applications « ca nécessite pas l'intervention directe du développeur d'application » pour un environnement multi-clouds. Ils se concentrent sur l'abstraction pour les développeurs d'applications et l'état peut facilement activer les utilisateurs pour "obtenir les caractéristiques d'application souhaitées (comme l'évolutivité, la tolérance aux pannes,QoS ..etc.) "[9]. Il existe deux couches d'abstraction, une pour l'approvisionnement en cloud et une pour la logique d'application. La solution mOSAIC sélectionnera un cloud approprié sur la façon dont les développeurs décrivent leur application, plusieurs providers cloud peuvent être sélectionnés en fonction de leurs propriétés.La construction d'une application déployée avec mOSAIC exige que cette dernière soit constituée de composants avec des dépendances explicites pour communiquer et échanger des données entre eux. En outre, l'architecture de l'application doit être de type SOA et doit utiliser uniquement l'API mOSAIC pour la communication. mOSAIC utilise uniquement la communication asynchrone « technologies de files d'attente basées sur le cloud

5. Le projet européen de recherche RESERVOIR [11] vise à développer une infrastructure de fédération cloud entre les clouds privé et hybride afin de permettre l'interopérabilité dynamique entre les fournisseurs de cloud. Avec cela, une application déployée peut répartir la charge de travail de manière transparente entre les clouds privés et publics en fonction des exigences des applications. La transparence est réalisée en créant des sites de réservoir, un pour chaque fournisseur. Chaque site est indépendant et exécute un environnement d'exécution virtuelle (VEE) qui est géré par un VirtualExecution Environment Manager (VEEM). L'VEEM communique avec d'autres VEEM et sont en mesure de faire une fédération entre les clouds. Chaque site doit avoir des composants logiciels réservoirs installés, ce qui le rend autonome et autocontrôle .
6. Vega framework est un framework de déploiement destiné à des déploiements cloud complets sur des topologies multi-niveaux, elles suivent également une approche basée sur un modèle. La description d'une topologie donnée se fait en utilisant des fichiers XML (eXtensible Markup Language) avec ces fichiers, les développeurs peuvent répliquer une pile. Le XML contient des informations sur les instances, telles que l'os-type pour le système d'exploitation et la description de l'image pour décrire les propriétés d'une instance telle que la quantité de mémoire (*req_memory*). Ils permettent également que les petits scripts soient écrits directement dans le XML via

2.5 Conclusion :

Dans ce chapitre, nous avons décrit trois définitions les plus utilisées de déploiement automatique des applications sur le cloud computing , Afin de faciliter le choix de la technique de déploiement automatique appropriée dans les prochains chapitres, nous avons montré les approches d'adoption du Cloud computing par les développeurs de logiciels implémentées par certains outils et Frameworks.

Un des aspects essentiels de manipulation des approches de déploiement consiste à automatiser le déploiement grâce à l'utilisation des workflows en implémentant ces méthodes et ces outils, Ceci a donné l'idée de chapitre suivant.

Chapitre 3

Implémentation

3.1 Introduction

3.2 Notions générales

3.2.1 Aspect syntaxique

La syntaxe d'un langage est définie par un ensemble d'expressions (des mots, des phrases, des instructions, ou des diagrammes). Ces expressions peuvent être de deux types : les expressions simples et les expressions construites à partir d'autres expressions. Ainsi, pour exprimer une syntaxe d'un langage, on trouve plusieurs approches où les deux les plus utilisées sont l'approche par métamodélisation et l'approche grammaticale. La syntaxe d'un langage peut être l'un des deux types suivants :

Conclusion générale

Bibliographie

- [1] A model-driven solution for automatic software deployment in the cloud.
- [2] Software deployment : Extending configuration management support into the field. *The Journal of Defense Software Engineering*, 1998.
- [3] Gerard Conway and Edward Curry. Managing cloud computing - a life cycle approach. In Frank Leymann, Ivan I. Ivanov, Marten van Sinderen, and Tony Shan, editors, *CLOSER*, pages 198–207. SciTePress, 2012.
- [4] Maurice-Djibril Faye. *Self-adaptive deployment for middleware on elastic platform*. Theses, Ecole normale supérieure de lyon - ENS LYON, November 2015.
- [5] I. Foster and C. Kesselman. *The Grid. Blueprint for a new computing infrastructure*. Morgan Kaufmann Publishers, San Francisco, USA, 1999.
- [6] Peter Mell and Timothy Grance. The nist definition of cloud computing. Technical Report 800-145, National Institute of Standards and Technology (NIST), Gaithersburg, MD, September 2011.
- [7] Francesco Moscato, Rocco Aversa, Beniamino Di Martino, Teodor-Florin Fortis, and Victor Ion Munteanu. An analysis of mosaic ontology for cloud resources annotation. In Maria Ganzha, Leszek A. Maciaszek, and Marcin Paprzycki, editors, *FedCSIS*, pages 973–980, 2011.
- [8] Object Management Group. <http://www.omg.org>, seen at April 2008.
- [9] Dana Petcu, Ciprian Craciun, Marian Neagul, Silviu Panica, Beniamino Di Martino, Salvatore Venticinque, Massimiliano Rak, and Rocco Aversa. Architecturing a sky computing platform. In Michel Cezon and Yaron Wolfsthal, editors, *ServiceWave Workshops*, volume 6569 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 2010.
- [10] Clément Quinton and Laurence Duchien. Vers un Outil de Configuration et de Déploiement pour les Nuages. In *JLdP - Journée Lignes de Produits*, pages 83–94, Lille, France, November 2012.

- [11] Benny Rochwerger, David Breitgand, Eliezer Levy, Alex Galis, Kenneth Nagin, Ignacio Mart  n Llorente, Rub  n S. Montero, Yaron Wolfsthal, Erik Elmroth, Juan A. C  ceres, Muli Ben-Yehuda, Wolfgang Emmerich, and Ferm  n Gal  n. The reservoir model and architecture for open federated cloud computing. *IBM Journal of Research and Development*, 53(4) :4, 2009.
- [12] Benny Rochwerger, David Breitgand, Eliezer Levy, Alex Galis, Kenneth Nagin, Ignacio Mart  n Llorente, Rub  n S. Montero, Yaron Wolfsthal, Erik Elmroth, Juan A. C  ceres, Muli Ben-Yehuda, Wolfgang Emmerich, and Ferm  n Gal  n. The reservoir model and architecture for open federated cloud computing. *IBM Journal of Research and Development*, 53(4) :4, 2009.
- [13] Clemens A. Szyperski. Component technology - what, where, and how?. In Lori A. Clarke, Laurie Dillon, and Walter F. Tichy, editors, *ICSE*, pages 684–693. IEEE Computer Society, 2003.
- [14] Mark Turner, David Budgen, and Pearl Brereton. Turning software into a service. *IEEE Computer*, 36(10) :38–44, 2003.