

ISEP

THÈSE

Solutions ouvertes et Open Source vers le multi-cloud

Auteur :
Mathias KOWALSKI

Tuteur de Thèse :
Louis NAUGES

*Thèse entrant dans le cadre de
l'obtention du Master spécialisé Expert Cloud Computing*

20 septembre 2020

« Liberty means responsibility. That is why most men dread it. »

George Bernard Shaw

ISEP

Résumé

ISEP

Master spécialisé Expert Cloud Computing

Solutions ouvertes et Open Source vers le multi-cloud

par Mathias KOWALSKI

La présente thèse traite des solutions de portabilité non propriétaires qui se profilent en 2020 et qui permettent d'atteindre l'objectif de migrer un SI vers une organisation en multi-cloud. L'objectif multi-cloud est ici considérée comme un but à atteindre pour diminuer les risques associés à l'attachement contractuel à un seul fournisseur cloud. J'insisterai sur les solutions de portabilité non-propriétaires dites « open source » pour ne pas répercuter la responsabilité des risques sur des outils ou des solutions de portabilité dits « propriétaires ».

Le sujet de cette thèse a pour vocation d'accompagner une partie des prédictions mondiales pour 2022 dans le cloud-computing. En effet selon IDC FutureScape, d'ici 2022, 70% des entreprises mondiales auront mis en place des processus et des outils de gestion pour gérer un environnement et une gouvernance multi-cloud robuste à travers des clouds sur site et des clouds public

Pour appuyer l'argument de portabilité en utilisant des logiciels open source ce document lui-même a été écrit en \LaTeX . Ainsi pour vulgariser le concept même si les sources de ce document, donc les données brutes qui lui sont associées sont open source, cela n'empêche pas qu'il soit écrit avec un logiciel SaaS comme Overleaf qui comporte une offre payante et qui tire parti du cloud. L'open source accélère donc la transition numérique et ouvre à ses utilisateurs des possibilités innovantes. Les principaux fournisseurs d'infrastructures cloud l'ont bien compris et en tirent parti. On va donc découvrir dans ce document comment ces fournisseurs tirent leur épingle du jeu en s'ouvrant aux autres dans un contexte multi-cloud.

Ce document est destinée à un public ayant une connaissance modérée des infrastructures informatiques ainsi que des principes de développement d'un logiciel informatique. Un développeur senior connaîtra probablement la plupart des outils cités mais ce document vise essentiellement à cibler une liste non exhaustive de logiciels open source utiles à la mise en place d'une stratégie multi-cloud. Chaque outil cité ne sera pas expliqué dans le détail mais l'effort d'explication et de définition sera orienté principalement par rapport au besoin de multiplicité et d'indépendance vis à vis des fournisseur cloud. Ce document vise aussi à rapprocher les personnes plus proches des opérations sur la vision et la préoccupation des développeurs et des administrateurs afin de comprendre la nécessité d'utiliser des logiciels ou formats dits open-source pour accroître la possibilité de réussite d'une stratégie multi-cloud. L'étude se confrontera aussi aux limites des outils open source et tentera d'expliquer de quelle manière utiliser les solutions propriétaires sans perdre de vue l'objectif du multi-cloud.

Remerciements

Merci à Louis Naugès pour avoir pris le temps d'étudier le sujet de thèse et de m'avoir aiguillé sur les problématiques de cloud souverain et de cloud public. Merci à Gerard Richter pour sa vision critique sur le multi-cloud et sur sa vision sur le cloud en général. Je tiens aussi à remercier plus particulièrement mon épouse qui s'est organisée avec notre fils de six mois comme elle pouvait pendant une période de pandémie afin de me laisser la sérénité nécessaire à l'écriture de cette présente thèse ...

Table des matières

Résumé	iii
Remerciements	v
1 Vers le Multi-Cloud	1
1.1 Contexte cloud actuel	1
1.1.1 Les avantages du cloud en général	1
1.1.2 Autour de la donnée	2
1.1.3 Converger sur le Cloud Hybride et le Multi-Cloud	2
1.1.4 L'offre IaaS, PaaS pour du SaaS	3
1.2 Pourquoi "Open source" et non pas "Logiciel Libre"	3
1.2.1 Quelle licence pour un usage dans le multi-cloud?	5
1.3 Niveaux de maturité dans le cloud	5
1.3.1 Cloud Resilient	6
1.3.2 Cloud Native	6
2 Cas d'usage de l'utilisation du multi-cloud	7
2.1 Une volonté naturelle	7
2.1.1 La combinaison réglementation / prix	7
2.1.2 Éviter le verrouillage fournisseur	9
2.1.3 Être au plus haut niveau technologique, accélérer l'innovation	9
2.1.4 Gestion du risque	9
2.1.5 Un objectif stratégique à atteindre	9
2.2 La problématique	9
3 Contexte initial et phase de préparation	11
3.1 Contexte multi-fournisseur d'infrastructures	11
3.1.1 Le cloud public	11
3.1.2 Orchestration multi-cloud managée	11
3.1.3 Le rôle d'OpenStack	13
3.1.4 La constellation des cloud privés	13
3.1.5 Le projet européen GAIA-X	13
3.2 Les principaux fournisseurs et leur ouverture aux autres	13
3.2.1 AWS Outpost	13
3.2.2 Microsoft Azure Arc	14
3.2.3 Google : Anthos	14
3.2.4 IBM Cloud Pak for Multicloud Management	15
3.2.5 Les autres	15
3.3 Comment se préparer à l'adoption du multi-cloud?	17
3.3.1 Prévoir une stratégie de sortie	17

4	Outils entre usages et infrastructures multi-cloud	19
4.1	De quels usages parles-t-on ?	19
4.2	l'écosystème Kubernetes	19
4.2.1	Une architecture multi-cloud et multi-cluster avec Kubernetes	20
4.2.2	Prometheus	22
4.3	Cloud foundry foundation	22
4.3.1	Plateformes certifiées Cloud Foundry 2020	24
4.4	OpenShift	24
4.5	Usage de surveillance multi-cloud	25
4.6	l'écosystème Européen GAIA-X	25
4.7	Usage entre IaaS et IaC	27
4.7.1	Le cas du FaaS	27
5	Migration des données et des applications	29
5.1	Vers une indépendance sans écosystème	29
5.1.1	Quelle est la probabilité que les applications changent de fournisseur d'infrastructure tout au long de leur durée de vie ?	29
5.2	Portabilité des données, un format d'échange ouvert pour une compatibilité sur les infrastructures	29
5.3	Migration d'une architecture	30
5.3.1	Terraform	30
5.4	Portabilité des instances de calcul	31
5.4.1	Le trio gagnant Packer Terraform Ansible.	31
6	Comment aborder le multi-cloud et les obstacles à son adoption	33
6.1	Procédures à l'adoption	33
6.1.1	Améliorer le niveau de maturité dans le multi-cloud progressivement	33
6.2	Les fragilités du multi-cloud	34
6.2.1	Système de secours	34
6.2.2	Des économies ?	34
6.2.3	La complexité menace le succès à l'adoption	34
6.2.4	Les données éparpillées	35
6.3	Le frein de l'open source	35
6.4	Red Hat OpenShift vs Google Anthos	35
7	Conclusion	37
7.1	Le multi-cloud est une stratégie	37
7.1.1	L'open source fédère ses adeptes autour de plusieurs communautés multi-cloud.	37
7.1.2	service managés pour l'hybride et open source pour le multi-cloud	37
7.2	Effectuer un choix selon le scénario d'utilisation	38
7.2.1	Pour les multi-cloud natifs : Docker + Kubernetes + Spinnaker + Istio	38
7.2.2	Pour ceux qui veulent garder leur architecture historique	38
7.2.3	Pour les adeptes du cloud souverain : OpenStack et Terraform	38
7.3	Kubernetes est la pierre angulaire du multi-cloud	39
7.3.1	Deux grand acteurs dans le panorama multi-cloud open source : Red Hat et Google	39
7.4	Le mot de la fin	39

A	Index outils et licences	41
B	Liste des licences open source principales	43
B.1	GNU AGPLv3	43
B.2	GNU GPLv3	43
B.3	GNU LGPLv3	43
B.4	Mozilla Public License 2.0	43
B.5	Apache License 2.0	44
B.6	MIT License	44
B.7	Boost Software License 1.0	44
B.8	Sans licence	44
	Bibliographie	45

Table des figures

1.1	Multi-Cloud adoption	4
2.1	Besoins multi-cloud par Gartner	8
3.1	quadrant magique pour l'infrastructure cloud et les services de plate- forme	12
3.2	Azure Arc	14
3.3	GKE On-Prem	16
4.1	Pyramide des besoins DevOps Kubernetes IaaS	21
4.2	Multi-cluster Multi-cloud	22
4.3	Architecture Open Shift	24
4.4	Offre Open Shift	25
4.5	Écosystème GAIA-X	26
4.6	Usage OpenFaaS	28

Liste des tableaux

1.1	revenus prévisionnels du cloud public mondial (en milliards de dollars américains), Source : Gartner (Novembre 2019)	1
-----	--	---

Liste des abréviations

AWS	Amazon Web Services
CAPEX	Capital Expenditure (dépenses d'investissement)
CI / CD	Continuous Integrations / Continuous Delivery (Intégration / Livraison continue)
CNCF	Cloud Native Computing Foundation
DNS	Domain Name System (système de noms de domaine)
DSI	Directions des systèmes d'information
FaaS	Function as a Service (Fonction en tant que service)
GCP	Google Cloud Platform
GNU	GNU's Not UNIX
GPL	Global Public License
IaaS	Infrastructure as a Service (Infrastructure en tant que service)
IaC	Infrastructure as Code (Infrastructure en tant que code)
OPEX	Operational Expenditure (dépenses d'exploitation)
PaaS	Platform as a Service (plateforme logicielle en tant que service)
RPO	Recovery Point Objective
RTO	Recovery Time Objective
SaaS	Software as a Service (Logiciel en tant que service)
SI	Système d'Information

Chapitre 1

Vers le Multi-Cloud

1.1 Contexte cloud actuel

Cela fait maintenant près de dix ans que le concept de "cloud computing"¹ pour un système d'information existe. De nombreuses raisons comme la fusion d'entreprise, la rapidité de mise à disposition d'un produit informatique ou l'optimisation des coûts poussent les entreprises à migrer toute ou partie de leur infrastructures dans le cloud. le tableau 1.1 ci dessous prévoit selon les sources Gartner une adoption croissante et massive des technologies du cloud computing. « À ce stade, l'adoption du cloud est la norme », explique dans un communiqué Sid Nag, vice-président de la recherche au cabinet Gartner. « Le paysage des services gérés dans le cloud devient de plus en plus sophistiqué et compétitif. En fait, d'ici 2022, jusqu'à 60% des organisations utiliseront l'offre de services gérés en nuage d'un fournisseur de services externe, ce qui représente le double du pourcentage observé depuis 2018 », commente Sid Nag. « Les fonctionnalités cloud natives, les services applicatifs, les multi-clouds et clouds hybrides constituent un écosystème cloud diversifié qui constituera un élément de différenciation important pour les responsables de produits technologiques. L'intérêt pour les bénéfices stratégiques offerts par les services cloud marque une transformation organisationnelle vers l'entreprise numérique. »(DIRK, 2019)

	2018	2019	2020	2021	2022
Cloud Business Process Services (BPaaS)	41.7	43.7	46.9	50.2	53.8
Cloud Application Infrastructure Services (PaaS)	26.4	32.2	39.7	48.3	58.0
Cloud Application Services (SaaS)	85.7	99.5	116.0	133.0	151.1
Cloud Management and Security Services	10.5	12.0	13.8	15.7	17.6
Cloud System Infrastructure Services (IaaS)	32.4	40.3	50.0	61.3	74.1
Total Market	196.7	227.8	266.4	308.5	354.6

TABLE 1.1 – revenus prévisionnels du cloud public mondial (en milliards de dollars américains), Source : Gartner (Novembre 2019)

1.1.1 Les avantages du cloud en général

En bref, parmi les arguments les plus évoqués qui poussent les entreprises à adopter les technologies du cloud on peut citer :

- Une puissance de calcul flexible
- Des ressources informatiques abondantes
- La capacité à sortir un produit rapidement
- L'ajustement rapide sur la consommation des ressources

1. informatique en nuage (*informatique en nuage / FranceTerme / Ressources / Accueil - Culture.fr p. d.*)

- La capacité à optimiser les coûts des infrastructures informatiques
- La suppression coûts d’investissement liés aux infrastructures informatique
- La suppression de la gestion physiques des infrastructures informatiques

L’idée autour de la facturation dans le cloud c’est : *ne payez que ce que vous utilisez*
 La possibilité en termes d’économies réalisées tourne autour de l’idée : *ne consommez que ce dont vous avez besoin* Pour pourvoir y arriver il va falloir appliquer une règle d’or dans le cloud : *ne devinez pas votre besoin en capacité informatique, automatisez-le*

1.1.2 Autour de la donnée

On tire le meilleur parti de l’information lorsqu’elle est accessible à toute personne ou équipe concernée. Les informations de qualité sont désormais vitales pour les entreprises. Des pratiques industrielles obsolètes et des outils difficiles à utiliser freinent l’exploitation de ces données. Le rôle de l’**open source** et du **cloud computing** influence profondément la technologie, ils donnent la puissance et les outils nécessaires à la transformation de cette donnée. De plus cette combinaison facilite l’exploitation des découvertes scientifiques et technologiques vis à vis des entreprises et des consommateurs. Il est bientôt tellement facile de traiter la données que les outils qui permettent de le faire deviennent communs et finissent par passer en arrière plan. Le cloud computing a permis l’explosion de l’exploitation des données ces dernières décennies grâce à un prix du stockage en baisse constante, une multiplication des sources de données et une analyse accessible de ces données.

Cependant le potentiel des données et de leur analyse se retrouve encore trop souvent piégé dans des centres de calculs sur site ou chez des fournisseurs de cloud privé. L’impact c’est que ce qui peut être collecté est souvent incomplet, lent à mettre à jour et difficile à analyser. De plus ce type de procédé ne bénéficie pas de la baisse du prix du stockage aussi rapidement.

1.1.3 Converger sur le Cloud Hybride et le Multi-Cloud

On retrouve fréquemment la confusion entre cloud hybride et multi-cloud. Le cloud hybride est une combinaison entre le choix d’un fournisseur d’infrastructures privé et celui d’un fournisseur de cloud public. Il y a quelques années le cloud hybride faisait parti du panorama des architecture à explorer et à adopter avec parcimonie. Aujourd’hui les pratiques d’architecture en cloud hybride sont bien encrées et pour gérer des charges de travail efficaces et rentables. Le cloud hybride implique l’intégration et l’orchestration de services suite à un déploiement sur plusieurs infrastructures cloud. Le terme «multi-cloud» quant à lui peut signifier différentes choses pour différents utilisateurs et clients. Il peut entre autre s’agir de :

- l’exploitation de plusieurs technologies de cloud public à la fois
- l’utilisation d’un cloud public en parallèle avec des systèmes non cloud traditionnels (Hybride)
- l’utilisation simultanée de plusieurs clouds publics pour différentes charges de travail

En règle générale on va appeler multi-cloud une combinaison de fournisseurs cloud utilisés par une même entreprise ou un même client. On peut noter que le *Shadow*

IT² peut mettre involontairement une entreprise dans un contexte multi-cloud. En effet la multiplication des services SaaS gratuits peut pousser des collaborateurs peu contentieux à outrepasser les processus trop restrictifs d'une entreprise et finir par héberger des données d'entreprise chez un fournisseur cloud non répertorié par la DSI.

Pour résumer lorsqu'on parle de multi-cloud on y inclut les concepts de cloud hybride. En termes d'adoption du multi-cloud, les résultats du cloud computing pour 2020 montrent que les entreprises continuent d'adopter des stratégies de cloud multi-cloud et utilisent déjà en moyenne plus de deux clouds publics et deux cloud privés (KIM, 2020).

« 93% des entreprises ont une stratégie multi-cloud. »

Rapport sur l'état du cloud Flexera 2020

Les clients adoptent des stratégies multi-cloud dans le but de minimiser le verrouillage fournisseur, de tirer parti des meilleures solutions technologiques, d'améliorer la rentabilité et d'augmenter la flexibilité par le choix. Nous avons un diagramme d'adoption du cloud ces dernières années 1.1. Nous voyons surtout que les entreprises partent de plus en plus sur une stratégie multi-cloud et que le multi-cloud privé-public (hybride) se démarque à l'adoption. Il y a donc un réel enjeu pour faire transiter la donnée et son traitement entre les sites privés et les fournisseurs de cloud publics.

1.1.4 L'offre IaaS, PaaS pour du SaaS.

Dans l'optique où on une entreprise va se servir en logiciels support il est assez facile de dire qu'elle utilise le multi-cloud lorsqu'elle utilise des solutions SaaS dont l'hébergement des données dépend du fournisseur de services. Au cours de ce document je n'évoquerai pas les offres SaaS ni leur interopérabilité. Je me concentrerai plutôt dans l'optique d'une entreprise qui est elle même éditeur de logiciels SaaS et me concentrerai sur l'utilisation de différents fournisseurs Paas ou IaaS afin d'atteindre l'objectif du multi-cloud.

1.2 Pourquoi "Open source" et non pas "Logiciel Libre"

Les désignations "logiciel libre" et "open source" sont deux désignations concurrentes pour un même type de licence de logiciel. En pratique, l'open source défend des critères un peu moins stricts que ceux du logiciel libre (RICHARD M., 2020). On va donc difficilement imaginer un solution entièrement basée sur le logiciel libre dans le cloud. En effet la notion de fournisseur de services d'infrastructure n'est pas compatible avec l'idée du logiciel libre car elle impose un choix à effectuer en termes de responsabilité notamment sur l'hébergement des données. C'est pour cela que la thèse présentera des outils "open source" et non pas forcément "logiciel libre" ainsi que des solutions "ouvertes" sans pour autant être des solutions sous licence "open source". Une liste des type de licence open source et leur explication est présentée

2. Shadow IT (parfois Rogue IT) est un terme fréquemment utilisé pour désigner des systèmes d'information et de communication réalisés et mis en œuvre au sein d'organisations sans approbation de la DSI. Selon la définition du cabinet Gartner, il s'agit aussi des "logiciels, services ou appareils informatiques qui n'appartiennent pas au service informatique d'une organisation ou sont hors de son contrôle"

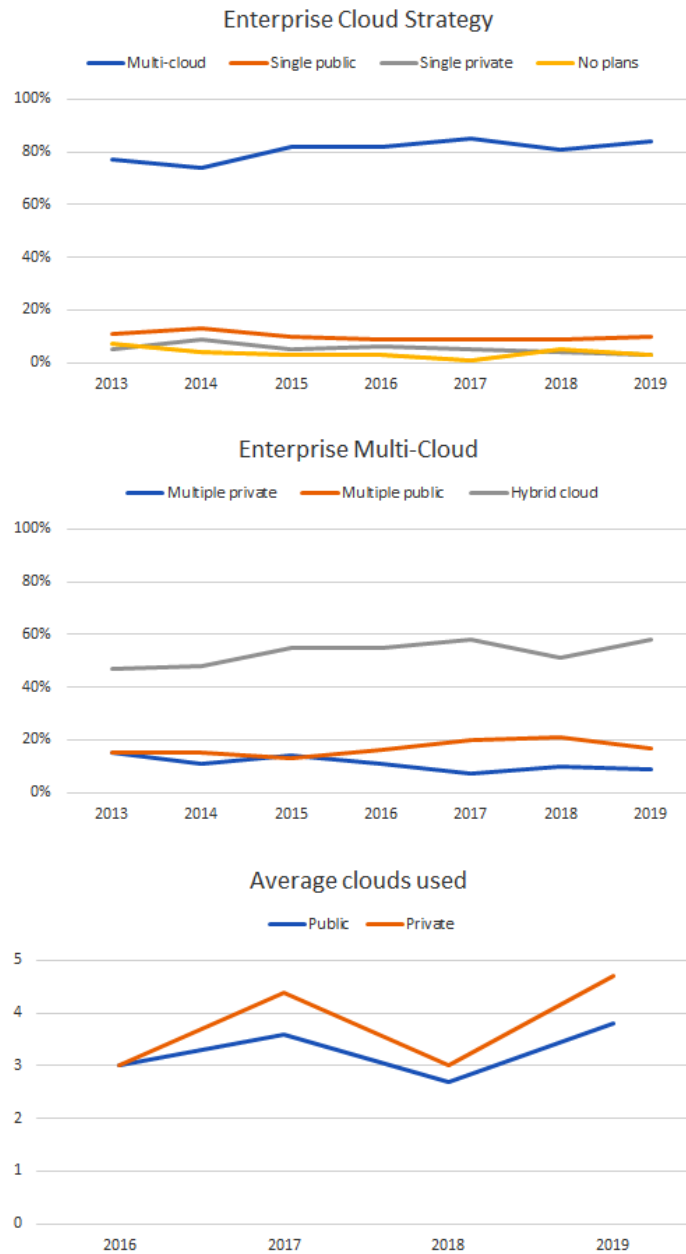


FIGURE 1.1 – Multi-Cloud adoption

en annexe B. On remarquera dans l'annexe A la faible proportion de licence LGPL utilisées qui constituent l'écosystème du logiciel libre.

1.2.1 Quelle licence pour un usage dans le multi-cloud ?

Adopter plusieurs solutions SaaS est une bonne option pour l'adoption de logiciels support et va faire intervenir une multitude de licences open source sans qu'on ne le sache. En effet elles sont gérées par l'éditeur de solutions SaaS. En terme de licence l'utilisation, un logiciel SaaS n'est pas contaminant même si les logiciels open source qui le constituent utilisent des licences fortes comme GNU AGPLv3 B. Il est donc possible d'utiliser des logiciels libres dans une optique multi-cloud mais il faut faire attention que la licence ne soit pas trop contaminante dans le cas d'utilisation au coeur de métier. Un format d'échange peut par exemple très bien adopter une licence open source forte sans que les données ne deviennent libres pour autant. En regardant la liste des outils que j'évoque dans l'annexe A il est indéniable que la licence Apache 2.0 est la plus utilisée. Pourquoi un tel engouement vers cette licence ? Lorsqu'on parcourt les principales licences open source présentées dans l'annexe B nous remarquons que c'est la plus permissive en terme de commercialisation et de valeur ajoutée en rendant possible la modification sans obligation de diffuser le code source de la modification.

1.3 Niveaux de maturité dans le cloud

Vers le multi-cloud, qu'est ce que ça implique ? Sur le sujet, le fait d'*aller* vers le multi-cloud montre une direction à prendre. Ici avant de prendre le large sur les sujets multi-cloud on va s'intéresser à savoir si l'organisme qui veut aller dans la direction du multi-cloud est prêt. Dans le cloud on parle de 4 niveaux de maturité croissants :

1. Cloud ready³
2. Cloud friendly⁴
3. Cloud Resilient⁵
4. Cloud Native⁶

Pour chaque niveau de maturité il y a alors un certain nombre de critères à accomplir pour qu'un projet informatique (applicatif) se dise dans une ou l'autre catégorie. Le *cloud ready* définira un projet qui comporte une certaine autonomie, sur machines virtuelles avec une base de code par application, des sauvegardes, des trace et une gestion des dépendances. En général un projet *cloud ready* ne contient pas encore ses données dans le cloud.

Ensuite le *cloud friendly* est un projet qui comporte un flux de développement continu et d'intégration continue. Il peut suivre le principe du 12-factors (*The twelve-factor app is a methodology for building software-as-a-service apps p. d.*) pour son organisation comporte un couplage faible sur son architecture et implémente une découverte de services.

Le niveau de maturité cloud friendly va être utilisé comme niveau minimal de maturité pour une application dans le reste du document pour aller vers le multi-cloud et les deux niveaux de maturité suivants :

-
3. Prêt pour le cloud
 4. Compatible avec le cloud
 5. Résilient avec le cloud
 6. cloud natif

1.3.1 Cloud Resilient

Pour ce niveau de maturité recherché, il va falloir être tolérant aux pannes et faire partie d'une infrastructure agnostique avec une surveillance du système systématique. On y pratique aussi l'injection de pannes intentionnelle afin de tester la résilience du système. C'est dans cet objectif de maturité qu'une volonté de rejoindre une architecture du multi-cloud apparaît. C'est alors dans cette optique que prend naissance une stratégie multi-cloud au sein d'une entreprise ou d'un organisme. Nous allons voir dans la suite du document au chapitre 5 comment rendre les infrastructures résilientes en mettant en place des outils open source pour effectuer une migration.

1.3.2 Cloud Native

Ce niveau de maturité est le plus haut niveau de maturité qu'on puisse atteindre pour une application dans le cloud. On introduit ici la notion d'architecture en micro-services avec une approche API-first et une authentification et des autorisations bien configurées. On va vouloir atteindre l'élasticité et assurer la continuité de l'application. Implémenter sa propre application *cloud native* implique qu'elle soit :

- Distribuée
- Performante
- Automatisable
- Résiliente
- Élastique
- Facile à mettre à jour
- Auditable

Nous allons voir dans la suite du document au chapitre 4 un ensemble d'outils qui permettent aux application *cloud natives* d'aller dans la direction du multi-cloud en faisant partie d'un écosystème cloud natif. J'aimerais aussi évoquer par rapport à ce niveau de maturité l'existence de la Cloud Native Computing Foundation (CNCF) qui est un projet de la Linux Foundation fondé en 2015 pour aider à faire progresser la technologie des conteneurs et aligner l'industrie technologique autour de son évolution. Il a été annoncé aux côtés de Kubernetes 1.0, un gestionnaire de cluster⁷ de conteneurs open source. Ce projet a été contribué par Google à la Linux Foundation en tant que technologie d'amorçage. Les membres fondateurs incluent Google, CoreOS, Mesosphere, Red Hat, Twitter, Huawei, Intel, Cisco, IBM, Docker, Univa et VMware. Aujourd'hui, la CNCF est soutenue par plus de 450 membres.

7. grappe de serveurs

Chapitre 2

Cas d'usage de l'utilisation du multi-cloud

2.1 Une volonté naturelle

Nous avons vu précédemment que le cloud se présente sous différentes formes. Chaque fournisseur va se positionner sur une politique réglementaire ou une politique de prix différente. On va ainsi trouver des fournisseurs qui se disent "Cloud privé" en prétendant répondre à des besoins réglementaires plus strictes et d'autres fournisseurs dits de "Cloud public" qui vont proposer une même offre à leur clients sous une réglementation plus globale et une mutualisation du matériel. Face à des offres hétérogènes les dirigeants sont face à des choix de fournisseurs à faire. Naturellement ce choix va pouvoir se faire sur plusieurs fournisseurs et comme il est possible de faire ce choix multiple on va alors explorer les cas d'usages qui vont le déclencher.

Selon Gartner les besoins de portabilité et de multi-cloud sont représentés comme sur une pyramide de Maslow avec des besoins plus ou moins importants comme sur la figure 2.1. Ainsi le besoin le plus important serait celui de vouloir survivre aux anomalies liées au fournisseur, puis vient le besoin d'élasticité, de performance, de négociation (réglementations / prix) et pour finir le besoin d'une stratégie multi-cloud. On peut alors affirmer que selon cette configuration le besoin de portabilité des applications et des données est plus important que celui d'avoir une stratégie multi-cloud. Dans le chapitre 5 nous verrons quels sont les moyens de répondre à ces besoins de portabilité avec des formats open-source et dans le chapitre 4 nous verrons comment mettre en place une stratégie multi-cloud avec des outils open-source.

2.1.1 La combinaison réglementation / prix

Le cloud privé séduit par le fait que l'entreprise prend ses responsabilités quand à la sécurité et l'emplacement de ses données. Dans ce cas pour pouvoir optimiser les coûts d'une telle décision l'entreprise va faire le choix de payer le prix fort et de se responsabiliser sur les données hébergées sur des infrastructures dites "privées". Elle va en parallèle faire le choix d'un cloud public pour tout les avantages d'un cloud public vu précédemment et notamment en terme de prix. Si l'entreprise préfère investir avec des coûts en CAPEX elle va probablement faire le choix d'avoir son propre centre de données mais si elle est à l'aise avec un centre de coûts en OPEX elle va probablement faire le choix d'un fournisseur de cloud dit "privé" qui va dédier des infrastructures à l'entreprise au sein de son centre de données. Cette manière de faire est fréquente et s'apparente à du cloud dit "Hybride", on peut aussi derrière avoir une combinaison de fournisseurs plus complexe par exemple lorsque

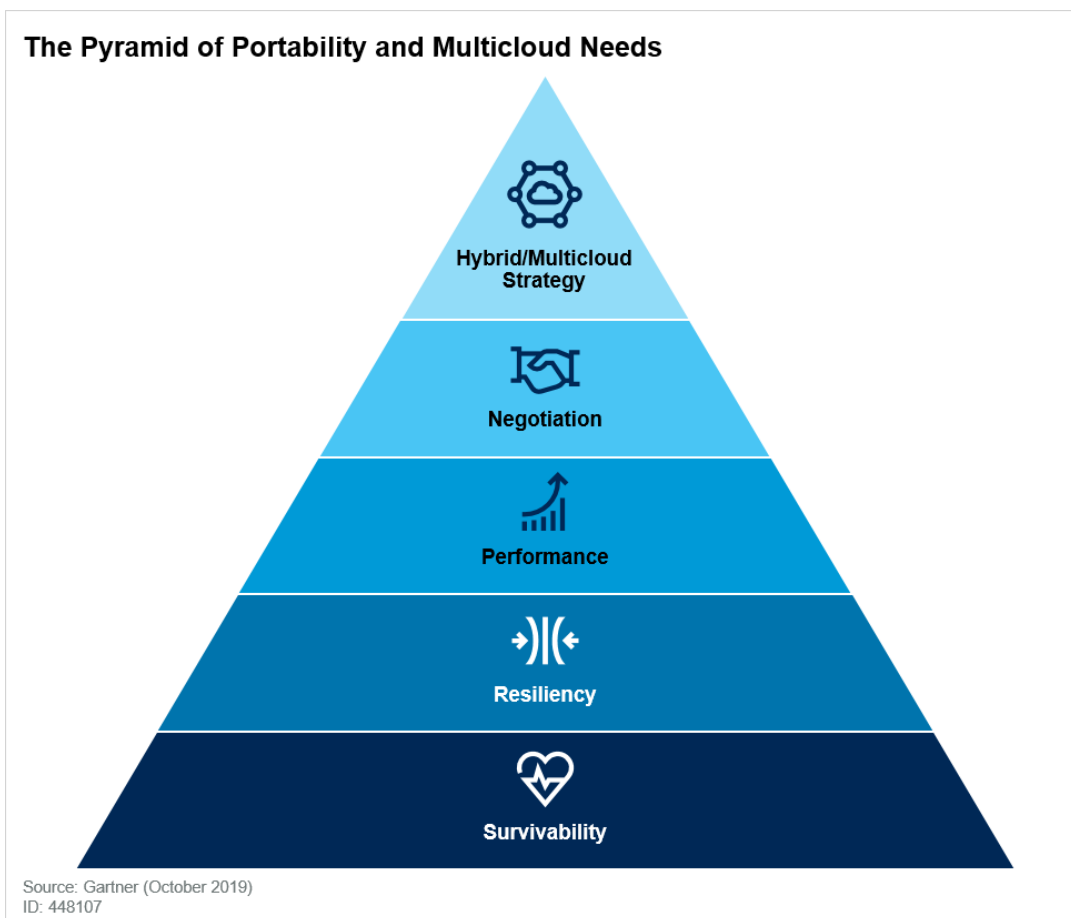


FIGURE 2.1 – Besoins multi-cloud par Gartner

l'entreprise choisit des infrastructures privées dans différents pays chez différents fournisseurs

2.1.2 Éviter le verrouillage fournisseur

Les fournisseurs de cloud public sont désireux de fidéliser leur clients en leur faisant adopter à faible coût leur solutions propriétaires. Ils vont alors proposer des offres d'engagement à long terme et se protéger avec des coûts de transfert pour les données sortantes. Traiter avec plusieurs fournisseurs va donc atténuer les effets de ce verrouillage et va donner au client un meilleur pouvoir de négociation. D'un autre côté grâce aux technologies de mobilité de la charge de travail des applications qu'on va découvrir dans les chapitres suivants, le choix multi-fournisseur cloud est devenu plus simple.

2.1.3 Être au plus haut niveau technologique, accélérer l'innovation

Les entreprises soucieuses de leur produit et de leur image vont tirer parti de la stratégie multi-cloud en adoptant ce qu'il se fait de mieux chez chaque fournisseur. En effet chaque fournisseur va imposer sa différence technologique et proposer ses innovations sur des sujets qui lui sont chers. Même si les grands fournisseurs de cloud public se concurrencent sur les mêmes sujets en terme de technologie ils arrivent à tirer leur épingle du jeu sur des technologies innovantes et marquent la différence par rapport aux autres sur certains sujets. Par exemple sur des produits de l'écosystème Microsoft le cloud Azure va mieux se placer que ses concurrents. Sur la rapidité à lancer une instance de calcul GCP fera la différence... Il y aura donc cette volonté d'adopter ce qu'il se fait de mieux pour un besoin donné. Si les besoins de l'entreprise sont hétérogènes on aura alors cette volonté d'adopter plusieurs fournisseurs pour leur expertise dans chaque domaine.

2.1.4 Gestion du risque

Comme tout produit informatique le cloud d'un fournisseur peut tomber en panne. En général il ne s'agit pas d'une panne généralisée mais cantonnée à certains services ou zones géographiques. Cependant ces pannes sont rares mais répertoriées. Chaque fournisseur propose contractuellement un taux de disponibilité de ses services et donc prévoit l'apparition de ces pannes. Certaines entreprises voient dans le multi-cloud un moyen d'atténuer les risques liées à une panne fournisseur. On imagine un peu plus cette volonté vis à vis de fournisseurs cloud moins connus ou proposant une disponibilité de leur service plus faible ou proposant moins de zones de disponibilité.

2.1.5 Un objectif stratégique à atteindre

En résumé le multi-cloud c'est l'utilisation de plusieurs plate-formes cloud de différents fournisseurs pour plusieurs tâches. On adopte alors le multi-cloud pour atteindre les objectifs dictés par la politique d'une entreprise.

2.2 La problématique

Maintenant qu'on a parcouru les différents besoins relatifs à l'adoption du multi-cloud et que l'on sait qu'il s'agit d'une décision *stratégique* à mettre en place au sein

d'un organisme, il va falloir se trouver les moyens de mettre en oeuvre cette *stratégie*. Quelle clé permet d'atteindre cet objectif? Par quels moyens peut-on assurer cette *transition numérique*? Il va alors falloir trouver la combinaison technique appropriée pour converger en direction du multi-cloud. Il s'agira de **construire un écosystème** capable de supporter les besoin et les contraintes liées au multi-cloud. Ceci implique lors de la phase de conception et de développement d'aborder les problèmes liées à :

- l'interopérabilité
- le déploiement de services et la gestion de la charge de travail associée
- La persistance et la disponibilité des donnée
- Une connexion fiable et performante entre les tâches qui s'exécutent sur différents clouds

Cette transition numérique vers une infrastructure dispersée va remettre en question les procédures et les méthodologies déjà mises en place. Par exemple l'approche DevOps¹ va devoir être mise à jour. Dans tous les cas il faudra garder en tête qu'il ne sera pas forcément possible d'inter-connecter toutes les pièces du puzzle. Il sera alors essentiel de se poser les bonnes questions en termes :

- d'intégration
- de portabilité
- d'interopérabilité

1. le mouvement Devops se caractérise principalement par la promotion de l'automatisation et du suivi (monitoring) de toutes les étapes de la création d'un logiciel, depuis le développement, l'intégration, les tests, la livraison jusqu'au déploiement, l'exploitation et la maintenance des infrastructures

Chapitre 3

Contexte initial et phase de préparation

3.1 Contexte multi-fournisseur d'infrastructures

3.1.1 Le cloud public

Le cloud public est l'ensemble de fournisseur IaaS, PaaS, SaaS. Il est le plus puissant et le plus à jour au niveau mondial. Il apporte toujours en 2020 le plus grand nombre d'innovations informatiques et un nombre pléthorique de services managés. Comme le montre le bureau d'étude Gartner dans son quadrant magique en Août 2020 [3.1](#)¹ Les acteurs principaux dans les infrastructures IaaS et le PaaS sont les suivants :

1. AWS
2. Microsoft
3. Google
4. Alibaba Cloud
5. Oracle
6. IBM
7. Tencent Cloud

Nous verrons alors dans ce chapitre comment les trois premiers clouds public s'ouvrent au autres et au cloud hybride. Le multi-cloud chez Alibaba Cloud implémente essentiellement la mise en place d'outils de synchronisation de données ou de tunnels de données open source. Oracle de son côté propose un outil "Oracle cloud management" qui se dit leader en multi-cloud mais il y a peu d'informations sur ce produit qui semble proposer de visualiser une topologie applicative multi-cloud.

3.1.2 Orchestration multi-cloud managée

à l'image du projet Hybrid Cloud Management X (HCMX) de Micro Focus, il existe une offre de multi-cloud managé sur le marché. Ces solutions se vantent de couvrir l'intégration avec les principaux acteurs de cloud public. Ils se placent alors comme une solution de gouvernance du multi-cloud et proposent de faciliter l'orchestration multi-cloud.

1. Gartner, Magic Quadrant for Cloud Infrastructure & Platform Services, Raj Bala, Bob Gill, Dennis Smith, David Wright, Kevin Ji, 1 September 2020. This graphic was published by Gartner, Inc. as part of a larger research document and should be evaluated in the context of the entire document.

Figure 1. Magic Quadrant for Cloud Infrastructure and Platform Services



FIGURE 3.1 – quadrant magique pour l'infrastructure cloud et les services de plateforme

3.1.3 Le rôle d'OpenStack

OpenStack est un ensemble de logiciels open-source à déployer sur les infrastructures d'un centre de calcul. C'est le résultat, une collaboration entre la NASA et Rackspace Inc. Les utilisateurs d'OpenStack ont en général la volonté de prendre la responsabilité entière de leur centre de calcul et de leur données. C'est alors un ensemble de logiciels qui permettent l'utilisation de l'appellation "Cloud privé" car les outils qui composent OpenStack permettent de gérer les infrastructures de manière similaire à celle utilisée avec les clouds publics. Dans le cadre du multi-cloud l'adoption d'OpenStack est intéressante dans le cas du cloud hybride ou du multi-cloud privé-privé. En effet les outils PaaS multi-fournisseur tels que CloudFoundry, Kubernetes, Ansible ou Terraform que nous allons découvrir plus loin vont pouvoir être déployés sur la base logicielle d'OpenStack. Pour ce qui est du IaaS le fait que les outils soient open source va faciliter les échanges multi-fournisseurs. OpenStack prend alors un rôle de facilitateur à l'adoption du multi-cloud mais ne résout pas les problématiques liées à la centralisation des services multi-fournisseurs. Il y a plein d'autres plateformes telles que Apache CloudStack, ManageIQ, Cloudify, Zenko Salt Cloud mais je ne rentrerai pas dans le détail. Je reste concentré sur la plate-forme OpenStack qui "définit" plus ou moins le terme de "cloud privé".

3.1.4 La constellation des cloud privés

Vu précédemment un certain nombre de besoins pousse les organisations à adopter un cloud sur site ou privé dans une configuration de cloud-hybride. Ainsi parmi différents pays il y a un choix pléthorique de fournisseur d'infrastructures pour différents besoins. Le plus ouverts adoptent OpenStack et les autres se présentent comme fournisseurs IaaS. Ils peuvent être plus difficiles à intégrer comme fournisseur multi-cloud s'ils n'implémentent pas une plateforme cloud open-source. En effet un fournisseur IaaS a tout intérêt à adopter une plateforme cloud open-source pour être présent sur le marché du cloud hybride ou du multi-cloud.

3.1.5 Le projet européen GAIA-X

Le projet européen GAIA-X que je présenterai plus en détail dans le chapitre 4 est un ensemble d'acteurs du cloud qui se regroupent pour former un cloud européen et souverain. C'est alors dans notre contexte initial vers le multi-cloud en ensemble fermé d'acteurs de cloud privé ou de cloud public fédérés. Pour l'instant le liant multi-cloud entre ces acteurs est OpenStack. En effet OVHcloud et T-Mobile qui font parti des acteurs importants du projet intègrent cette plateforme même s'ils ne se considèrent pas comme des cloud privés.

3.2 Les principaux fournisseurs et leur ouverture aux autres

3.2.1 AWS Outpost

AWS Outpost est le produit phare d'AWS pour le cloud hybride sur site. Ici on ne peut pas dire que AWS s'ouvre aux autres clouds hybrides mais il propose plutôt une solution pour déployer ses services sur site comme s'ils faisaient partie intégrante des infrastructures AWS publiques. Cette solution répond aux besoins multi-cloud suivants :

- Performance sur site à faible latence

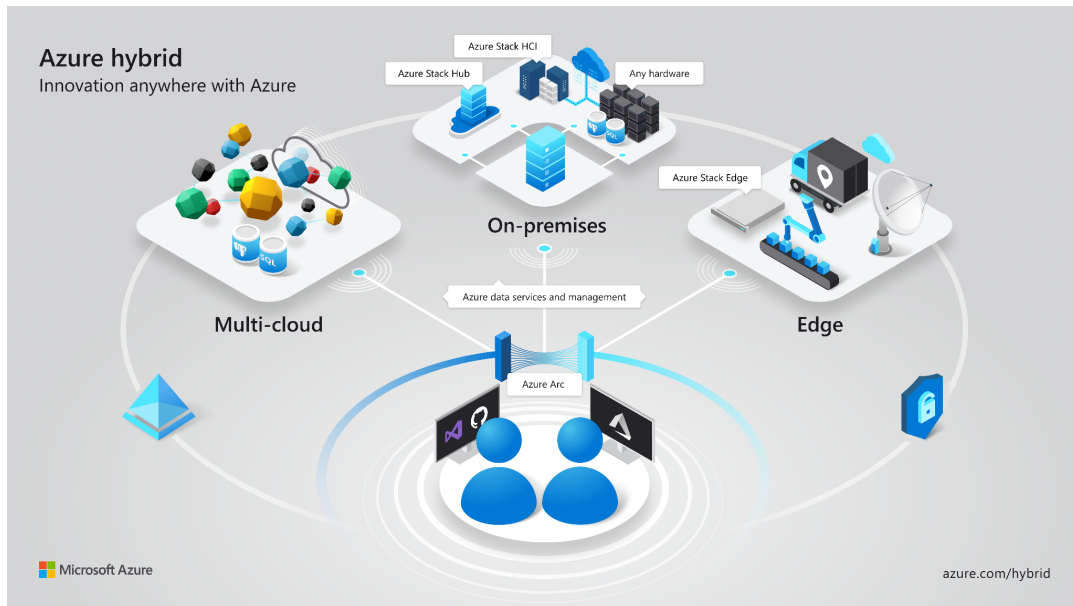


FIGURE 3.2 – Azure Arc

— Conformité de la localisation des données

Outpost est une solution physique qu’AWS installe sur site par ses soins mais en contrepartie la sécurité et la responsabilité physique de l’installation est déléguée chez le client. AWS comporte le plus grand nombre de services managés et c’est l’une des caractéristiques qui le place en haut du podium mondial. Cependant ce fournisseur ne s’ouvre pas aux autres et ne propose pas de plateforme managée et ouverte aux autres fournisseurs sur le multi-cloud. Sa stratégie est plutôt de maximiser ses propres solutions aux problématiques des entreprises.

3.2.2 Microsoft Azure Arc

Microsoft a de son côté présente un certain nombre de solutions aux entreprises souhaitant gérer l’hybridation sur site avec les services Azure et le Multi-Cloud. **Azure Arc** est le nom d’un ensemble de technologies visant à faire cohabiter l’écosystème Azure comme entité qui centralise le contrôle des ressources venant de l’extérieur (sur site ou d’un autre fournisseur cloud). L’idée d’azure Arc est de s’interconnecter avec des services comme **kubernetes** et **Azure SQL Analytics**. **Azure Arc** se place comme la succession à une hybridation cloud initiée par **Azure Stack** en 2017.

Par rapport à l’ouverture aux autres ici Microsoft propose un produit qui permet d’appliquer la gestion avec les produits Azure sur du cloud natif hébergé chez un autre fournisseur avec Kubernetes ou sur du cloud hybride sur site. Comme le montre la figure 3.2 propose avant tout de un outil de gestion propriétaire à destination du multi-cloud mais pas vraiment dans le sens inverse.

3.2.3 Google : Anthos

Google Anthos une plate-forme d’applications moderne pour les environnements hybrides et multi-cloud. Il s’agit principalement de gérer de manière centralisée les clusters kubernetes dans le Multi-Cloud. Ses principales fonctionnalités sont :

- Service d’orchestration et de gestion des conteneurs pensé pour les entreprises sur Kubernetes
- Automatisation des règles et de la sécurité à grande échelle
- Maillage de services entièrement géré avec visibilité intégrée
- Sécurité modernisée pour les déploiements hybrides et multi-cloud

Anthos permet la cohérence sur plusieurs Clouds : il permet d’avoir une vue globale des services s’exécutant sur une architecture multi-cloud. Anthos Service Mesh gère et sécurise le trafic en cours d’exécution dans un centre de données, sur Google Cloud et sur d’autres clouds tels qu’AWS. Anthos Config Management permet de distribuer et d’appliquer des politiques hiérarchiques (telles que l’autorisation, les quotas de ressources et les limites) à grande échelle sur plusieurs applications exécutées dans plusieurs clouds. En bref, Anthos donne les outils pour exécuter des applications n’importe où. Anthos s’intègre principalement avec AWS et compte aussi le faire bientôt avec Microsoft Azure. La prise en charge d’Anthos pour AWS est généralement disponible. Cette version inclut plusieurs d’avantages en termes de Fiabilité élevée, de mise à l’échelle automatique, d’intégration avec un environnement AWS existant, de cohérence opérationnelle et d’intégration avec la pile Anthos complète. Anthos vient aussi supporter l’offre hybride de google cloud avec sa version kubernetes sur site **GKE On-Prem**. En effet avec GKE On-Prem il est possible de créer, gérer et mettre à niveau des clusters Kubernetes dans un environnement sur site. GKE On-Prem s’exécute sur site ou sur un cloud privé avec comme prérequis un environnement vSphere. vSphere est le logiciel de virtualisation du serveur VMware. GKE On-Prem utilise le serveur vCenter de VMware pour gérer les clusters comme présenté sur la figure 3.3.

3.2.4 IBM Cloud Pak for Multicloud Management

IBM arrive avec sa propre plateforme de gestion multi-cloud. Sa plateforme s’exécute sur Red Hat® OpenShift® que l’on verra dans le chapitre suivant 4 et qui est un écosystème multi-cloud pour les containers. LE produit tourne autour des idées de :

- gestion multicluster
- gestion des événements
- gestion des applications
- gestion des infrastructures

Comme la plupart des solutions propriétaires la plateforme propose un support de gouvernance intégré pour la gestion de la conformité

3.2.5 Les autres

Les autres acteurs se reposent pour l’instant sur les plateformes et écosystèmes open source tels qu’OpenStack pour rester ouverts aux autres. Ils font souvent partie d’un écosystème multi-cloud qu’on verra dans le chapitre 4 pour faire partie d’une communauté de clouds inter-connectés par ce dernier. On aura alors accès à des fournisseurs d’infrastructures différents qu’ils soient sur OpenShift, CloudFoundry ou GAIA-X par exemple. Le multi-cloud sera alors limité par l’écosystème souvent open-source qu’il y a derrière.

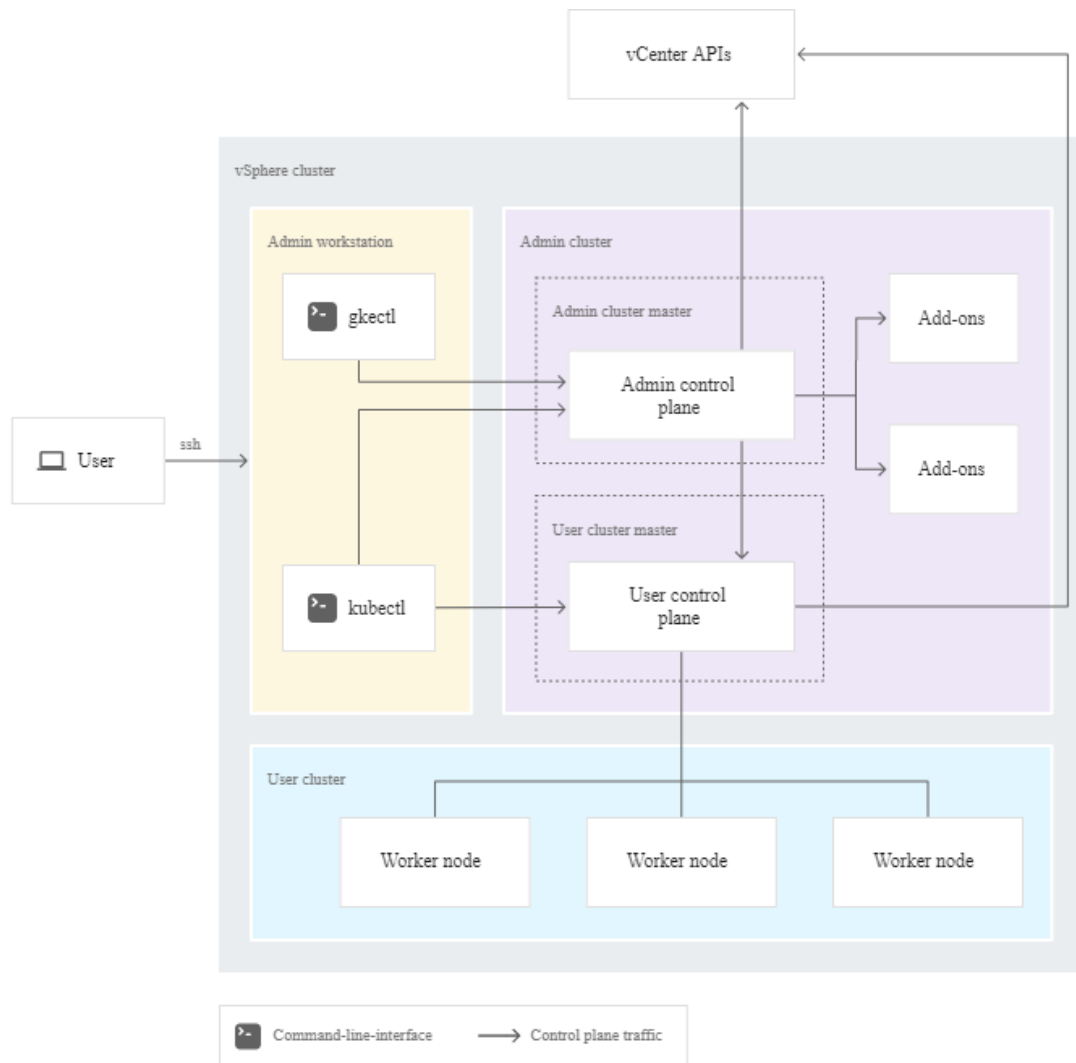


FIGURE 3.3 – GKE On-Prem

3.3 Comment se préparer à l'adoption du multi-cloud ?

Avant de se lancer dans le vif du sujet c'est à dire sur le choix de solutions qui pourraient nous mener sur la bonne voie du multi-cloud, attardons nous sur la stratégie à mettre en place. On a vu précédemment qu'on ne pouvait pas forcément intégrer toutes les briques d'un SI dans un environnement multi-cloud. Il va donc falloir identifier les briques qui peuvent interagir ensemble sans opposer trop de difficultés. La granularité de cette difficulté et de cette nouvelle complexité doit être mesurée vis à vis des équipes de conception et vis à vis des clients. **Quel est l'impact de la transition d'un composant du SI vers le multi-cloud en adoptant tel ou tel écosystème ?** Que ce soit la conception ou le déploiement il y aura un certain nombre de points à considérer (TITTEL et al., 2019) :

- **Orchestration** : vos outils d'orchestration sont-ils en lien avec les nouveaux clouds à adopter
- **DevOps** : Les outils d'automatisation peuvent-ils fusionner ? Le processus d'automatisation peut-ils être facilité ?
- **Stratégie de stockage** : Sur un environnement conteneurisé le stockage est-il cohérent au niveau de la portabilité de la charge de travail ?
- **Produits support, self-service et portails** Est-il possible de réutiliser les API ? Comment est-il possible d'étendre leur utilisation

Il est important de pouvoir tester toutes ces interaction et d'adopter un écosystème multi-cloud permettant d'effectuer ces tests. Nous verrons dans le chapitre 4 suivant les écosystèmes open sources qui permettent d'atteindre cette phase de préparation. La bonne pratique serait alors de choisir un fournisseur sur lequel on déploie cet écosystème pour ensuite y déployer la constellation de sous services qui peuvent pour certains être hébergés chez un autre fournisseur.

3.3.1 Prévoir une stratégie de sortie

La préparation au passage dans le multi-cloud de la même manière qu'avec le passage dans le cloud nécessite de prévoir une stratégie de sortie. Il faut alors comprendre ce qu'implique le passage d'un fournisseur à un autre. Lors de la signature d'un contrat avec plusieurs fournisseurs il faut selon la stratégie multi-cloud négocier les modalités de sortie des données. Il n'est en général pas nécessaire de récupérer l'architecture des ressources sachant qu'en termes de bonnes pratiques on est censé automatiser le déploiement de cette architecture. Nous verrons dans le chapitre 4 comment stocker les plans d'une architecture IaaS sans pour autant devoir migrer toutes les ressources en cas de migration chez un autre fournisseur. Il est alors important de prendre en compte l'automatisation de la création des ressources lorsqu'on prévoit une stratégie de sortie orientée multi-cloud. En terme de modalité de sortie des données il faut s'assurer que le format des données à faire transiter soit Open Source et il faut pouvoir effectuer cette transition par un moyen de transport sécurisé. Via le réseau la sortie de ces données va devoir être chiffrée mais il faudra aussi faire attention au débit et effectuer le calcul sur le temps nécessaire à la sortie de ces données. Si le temps nécessaire à la sortie de ces données est trop long, certains fournisseurs de cloud public comme AWS proposent d'exporter les données de manière logistique et physique comme avec AWS Snowball².

2. AWS Snowball est un service destiné aux clients qui souhaitent transporter des téraoctets ou des pétaoctets de données vers et depuis AWS, ou qui souhaitent accéder à la puissance de stockage et de calcul du cloud AWS au niveau local et de manière rentable, à des endroits où une connexion à Internet pourrait ne pas être envisageable.

Chapitre 4

Outils entre usages et infrastructures multi-cloud

4.1 De quels usages parles-t-on ?

La séparation entre usage et infrastructure est la pierre angulaire d'une bonne architecture multi-cloud. En effet d'un point de vue d'éditeur de logiciels nous allons devoir séparer usages et infrastructures. Louis Naugès en parle dans son billet "B I S D" (NAUGÈS, p. d.) de l'importance de séparer usages et infrastructures. Comme le démontre Ici en termes d'usage on va s'attarder aux activités *DevOps* et plus particulièrement à la partie "Business" et celles qui concernent le coeur de métier d'un éditeur de logiciels et le cycle de vie d'un logiciel. Dans les usages il va donc falloir pouvoir effectuer les activités suivantes :

- écrire un logiciel avec du code
- construire le logiciel pour pouvoir le tester
- le tester
- le livrer en lui assignant une version si il passe les tests
- le déployer pour le faire tourner en production
- le faire tourner en production
- surveiller l'état du logiciel en production
- planifier des correctifs ou gérer son cycle de vie

Les principaux fournisseurs de cloud proposent les outils le plus souvent propriétaires pour effectuer ces tâches et les incluent dans leur pilier de bonnes pratiques à adopter. Cependant pour effectuer une réelle séparation entre usage et infrastructures on va regarder quels sont les outils non propriétaires qui permettent réellement d'appliquer ce principe au multi-cloud. Nous allons alors dans ce chapitre évoquer les moyens qui peuvent faire passer une organisation ou une entreprise au niveau de maturité *cloud native* et en gardant en tête la stratégie multi-cloud. Dans un deuxième temps nous allons voir quelles sont les possibilités de migration pour chaque type d'infrastructure sans pour autant appartenir à un écosystème. Comme vu dans le chapitre 3 précédent Il s'agira de cibler les besoins pour les organisations ou entreprises d'atteindre niveau de maturité plutôt *cloud resilient*.

4.2 L'écosystème Kubernetes

Kubernetes est une plateforme open source pour automatiser le déploiement et la gestion des applications dans un environnement conteneurisé et en cluster. Nous avons parlé précédemment de la CNCF et de l'outil Kubernetes cédé à la fondation Linux. En août 2018, Google a annoncé qu'il passait le contrôle opérationnel de Kubernetes à la communauté CNCF (*Google takes a step back from running the Kubernetes*

development infrastructure p. d.). 9 millions de dollars de crédits ont alors été fourni à la CNCF par google cloud. Ces crédits seront répartis sur trois ans et sont destinés à couvrir les coûts d'infrastructure de construction, de test et de distribution de Kubernetes.

Pourquoi est-ce important? Jusqu'à présent, Google hébergeait pratiquement toutes les ressources cloud qui soutenaient le projet, comme son infrastructure de test CI / CD, les téléchargements de conteneurs et les services DNS sur son cloud. Mais Google prend maintenant du recul. La communauté Kubernetes ayant atteint un stade de maturité, Google transfère tout cela à la communauté. «*L'important don financier de Google à la communauté Kubernetes contribuera à garantir que le rythme constant d'innovation et d'adoption à grande échelle du projet se poursuive sans relâche*», a déclaré Dan Kohn, directeur exécutif de la CNCF. «*Nous sommes ravis de voir Google Cloud transférer la gestion des projets de test et d'infrastructure Kubernetes entre les mains des contributeurs - rendant le projet non seulement open source, mais ouvertement géré par une communauté ouverte.*»

Il convient de noter que la CNCF comprend désormais un large éventail de membres généralement en concurrence les uns avec les autres. Nous parlons par exemple d'Alibaba Cloud, AWS, Microsoft Azure, Google Cloud, IBM Cloud, Oracle, SAP et VMware. Tous bénéficient du travail de la CNCF et de la communauté Kubernetes.

Comment l'écosystème Kubernetes nous permet de séparer usages et infrastructures dans une approche DevOps? Kubernetes est avant tout un outil utilisé par les DevOps pour déployer des applications cloud natives dans un environnement *open source* et adopté par la grande majorité des fournisseurs cloud. Comme le montre la figure 4.1, Kubernetes répond à la pyramide des besoins DevOps et se repose sur le socle des infrastructures. Kubernetes lui-même se déploie sur des machines physiques ou virtuelles en grappes (clusters). Les avantages à l'adoption de l'écosystème kubernetes sont les suivants :

- éviter le verrouillage chez un fournisseur d'infrastructures. Les application sont agnostiques des infrastructures physiques
- Aide à la modernisation des application (architecture monolithique vers une architectures en micro-services ou architecture en micro-services vers une architecture sans serveur)
- Indépendance des applications, tant au niveau du cycle de vie qu'à leur cloisonnement.

Il y a cependant quelques spécificités à connaître vis à vis de cet écosystème :

- La configuration des cluster est spécifique aux infrastructures de l'hôte. Chaque fournisseur de cloud connu propose son implémentation (GKE chez GCP, EKS chez AWS, AKS chez azure, ACK chez alibaba)
- Avoir sa propre implémentation de contrôle de configuration de cluster kubernetes dans un centre de calcul privé est complexe et demande une certaine expertise sur le sujet. Cependant il existe des facilitateurs tels que GKE On-prem.

4.2.1 Une architecture multi-cloud et multi-cluster avec Kubernetes

Nous allons voir ici qu'il existe des outils plus spécifiques à déployer sur un cluster kubernetes : Istio et Spinnaker Comme nous le montre parfaitement Ameer Abbas dans sa présentation sur la gestion du multicloud avec Kubernetes, Istio et Spinnaker (*Using Kubernetes, Spinnaker and Istio to Manage a Multi-cloud Environment*



FIGURE 4.1 – Pyramide des besoins DevOps Kubernetes IaaS
(IBRYAM, 2017)

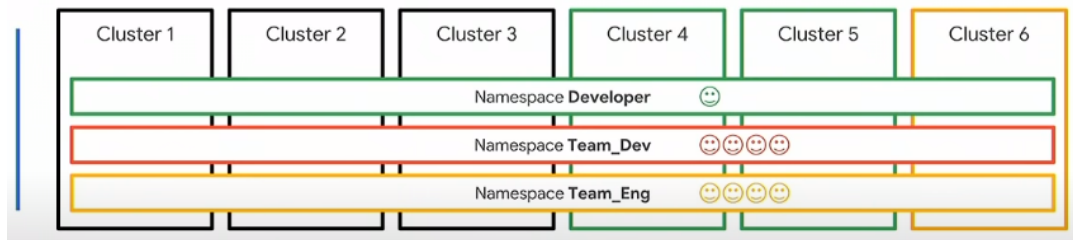


FIGURE 4.2 – Multi-cluster Multi-cloud

(*Cloud Next '18* - *YouTube p. d.*), la combinaison de quatre logiciels open source permettent de répondre aux problématiques d'une fabrique logicielle :

- **Docker** va emballer les applications
- **Kubernetes** va faire tourner les applications
- **Spinnaker** va gérer les applications
- **Istio** va connecter et sécuriser les applications

Dans cette optique on va déployer des cluster Kubernetes sur site ou sur de plateformes cloud compatibles avec kubernetes. Les cluster Kubernetes peuvent ainsi être inter connectés via un *namespace*¹. Une architecture multi-clusters est représenté sur la figure 4.2. Les trois premiers cluster en noir font par exemple partie d'une zone de disponibilité sur trois cluster hébergés sur un cloud public. Les deux autres clusters en vert tournent par exemple sur site et le cluster en orange tourne sur un cluster kubernetes d'un autre fournisseur cloud. Ici le *namespace* qui fait partie intégrante des spécifications Kubernetes va facilement regrouper les applications par type d'équipe. Ensuite on va pouvoir multiplier les cluster Kubernetes et les gérer tout au long du cycle de vie d'une application. On va avoir des applications qui tournent dans une phase de développement, dans une phase de pré-production et une phase en production. C'est là qu'intervient **Spinnaker** comme outil open source, multi-cloud de livraison continue. Le cycle de vie de l'application est maintenant assuré. Maintenant qu'on a tout ça, il faut savoir qu'il est difficile de s'y retrouver sur plusieurs cluster Kubernetes juste avec les outils fournis avec Kubernetes. Il nous faut alors un outil open source comme **Istio** qui propose une vue globale du système qui nous permette d'identifier rapidement les ressources problématiques. Je ne rentrerais pas plus loin dans le détail mais je vous invite à voir la présentation d'Ameer Abbas pour plus d'informations à ce sujet.

4.2.2 Prometheus

Projet membre de la CNCF, Prometheus est un outil de surveillance du cloud sponsorisé par SoundCloud dans les premières itérations. L'outil est actuellement utilisé par Digital Ocean, Ericsson, CoreOS, Docker, Red Hat et Google. C'est l'outil open source principal pour récupérer les métriques sur un cluster Kubernetes.

4.3 Cloud foundry foundation

Les développeurs doivent pouvoir exécuter leurs applications sur n'importe quelle instance Cloud Foundry dans la langue et le cadre de leur choix. Les organisations

1. Espace de nom utilisé pour labelliser des ressources Kubernetes

ont besoin de la portabilité entre les plates-formes d'applications cloud dans un environnement multi-fournisseurs et multi-cloud. Cloud foundry foundation se place dans le multi-cloud comme facilitateur du 12-factor apps (*The twelve-factor app is a methodology for building software-as-a-service apps p. d.*). Il s'oriente aussi vers les besoins DevOps mais plutôt dirigés par les 12 facteurs qui sont les suivants :

1. **Base de code** Une base de code suivie avec un système de contrôle de version, plusieurs déploiements
2. **Dépendances** Déclarez explicitement et isolez les dépendances
3. **Configuration** Stockez la configuration dans l'environnement
4. **Services externes** Traitez les services externes comme des ressources attachées
5. **Assemblez, publiez, exécutez** Séparez strictement les étapes d'assemblage et d'exécution
6. **Processus** Exécutez l'application comme un ou plusieurs processus sans état
7. **Associations de ports** Exportez les services via des associations de ports
8. **Concurrence** Grossissez à l'aide du modèle de processus
9. **Jetable** Maximisez la robustesse avec des démarrages rapides et des arrêts gracieux
10. **Parité dev/prod** Gardez le développement, la validation et la production aussi proches que possible
11. **Logs** Traitez les logs comme des flux d'évènements
12. **Processus d'administration** Lancez les processus d'administration et de maintenance comme des one-off-processes

Cloud foundry permet entre autre d'embarquer une distribution native de Kubernetes. Cette plateforme est représentée un PaaS et la qualité d'open source la distingue des autres plateformes. Il est ainsi possible de la déployer sur l'infrastructure d'un fournisseur cloud comme AWS, HP Helion, VMware vSphere ou OpenStack. Toutefois, il est aussi possible de la déployer sur sa propre infrastructure interne. En tant que PaaS open source, Cloud Foundry est entièrement personnalisable, ce qui permet aux développeurs de coder dans un grand nombre de langages différents et d'utiliser divers frameworks. Ceci évite de se retrouver enchaîné à un vendeur comme c'est souvent le cas avec les PaaS. Licenciée sous Apache 2.0, la plateforme est compatible Java, Node.js, Go, PHP, Python, Ruby, .NET Core et Staticfile. Notons tout de même que Cloud Foundry n'est pas la seule PaaS Open Source. C'est également le cas de OpenShift, Google App Engine et Heroku.

la plateforme est composée de sept groupes de composants :

1. Le routing
2. L'authentification
3. Le cycle de vie des applications
4. La mise en route des applications et leur stockage
5. Les brokers de service
6. La communication
7. La surveillance par les métriques et la journalisation

De son côté CloudFoundry se repose en grande partie sur sa communauté. C'est grâce aux développeurs et l'entraide possible que les entreprises adoptent cet outil.

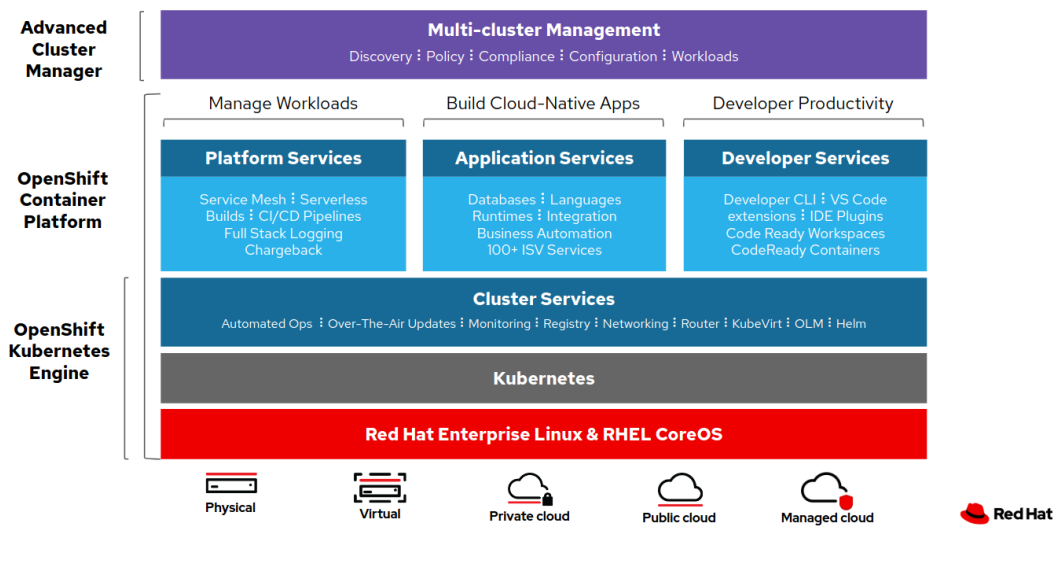


FIGURE 4.3 – Architecture Open Shift

4.3.1 Plateformes certifiées Cloud Foundry 2020

Pour les entreprises qui ont besoin d'une plateforme certifiée Cloud Foundry. Toutes les offres certifiées utilisent le même logiciel de base Cloud Foundry et garantissent la portabilité des applications et des compétences entre les fournisseurs. La liste exhaustive 2020² des plateformes certifiées CloudFoundry est la suivante :

- Atos Cloud Foundry
- Cloud.gov
- IBM Cloud Foundry
- SAP Cloud Platform
- SUSE Cloud Application Platform
- Swisscom Application Cloud
- VMware Tanzu

Chacune de ses plateformes se différencie par rapport à son offre de service ou son offre réglementaire en plus de tirer les avantages du PaaS multi-cloud CloudFoundry.

4.4 OpenShift

OpenShift est un ensemble d'outils pour développer et maintenir des applications cloud. En tant que tel, son travail principal est d'orchestrer les conteneurs d'applications - des unités autonomes standardisées qui contiennent tout le code et les logiciels dont une application a besoin pour s'exécuter dans le cloud. c'est une offre PaaS comme Cloud Foundry. OpenShift est comme «une sorte de deuxième génération» qui a vu le jour pour mieux tirer parti des logiciels cloud comme OpenStack (HALL, 2020). Comme le montre la figure 4.3 le produit OpenShift qui est un produit de Red Hat enterprise est une solution de gestion de cluster Kubernetes qui tourne sur son propre système d'exploitation et qui se présente comme une plateforme PaaS pour la gestion de cluster multiples.

2. <https://www.cloudfoundry.org/certified-platforms/>

Red Hat OpenShift Dedicated	RED HAT QUAY	Red Hat IBM	Red Hat OpenShift Container Platform
<ul style="list-style-type: none"> Clusters Red Hat OpenShift privés, haute disponibilité, hébergés sur Amazon Web Services et Google Cloud Fournis en tant que service hébergé et pris en charge par Red Hat 	<ul style="list-style-type: none"> Clusters Red Hat OpenShift à haute disponibilité, hébergés sur Microsoft Azure Fournis en tant que services hébergés, conçus, exécutés et pris en charge conjointement par Red Hat et Microsoft 	<ul style="list-style-type: none"> Un service de Red Hat OpenShift flexible et entièrement géré, dans le cloud public d'IBM Fourni en tant que service hébergé et pris en charge par IBM 	<ul style="list-style-type: none"> Une plateforme Kubernetes sur votre propre infrastructure, conçue avec la sécurité comme maître mot Créez, déployez et gérez vos applications basées sur les conteneurs, de manière cohérente sur les infrastructures cloud et sur site
✓ Hébergé par Red Hat	✓ Hébergé par Red Hat et Microsoft	✓ Hébergé par Red Hat et IBM	✓ Hébergé par l'entreprise
En savoir plus	En savoir plus	En savoir plus	En savoir plus

FIGURE 4.4 – Offre Open Shift

C'est une plateforme qui se veut indépendante d'un fournisseur d'infrastructures de cloud public tel que AWS, Azure ou GCP ou d'une infrastructure sur site. Les différentes possibilités sont présentées autour d'une offre commerciale détaillée sur la figure 4.4 c'est alors un produit que vous pouvez installer sur votre infrastructure avec un support payant inclus et livré avec un abonnement. Parmi les PaaS appréciées par les entreprises, OpenShift tient une bonne place avec Cloud Foundry. Il est intéressant de les différencier afin de faciliter le choix des professionnels. Premier point, Cloud Foundry donne accès à des outils Open Source et gratuits. OpenShift, comme son nom l'indique repose également sur des technologies libres de droit, mais son éditeur facture l'accès à ses services. OpenShift offre davantage de services à ses clients, tout du moins sur le papier. Cette dernière met en avant son programme de partenaires, ses cours et les possibilités de certifications.

4.5 Usage de surveillance multi-cloud

4.6 L'écosystème Européen GAIA-X

Tout d'abord j'aimerais préciser au moment où j'écris que le projet GAIA-X est toujours un projet sur le point de sortir. A l'initiative des gouvernements français et allemand, le projet Gaia-X vise à la mise en place d'une infrastructure de données en forme de réseau performante, compétitive, sûre et fiable pour l'Europe. L'écosystème GAIA-X est divisé en noeuds ou services. Un fournisseur GAIA-X fait parti de ce écosystème en lorsqu'il fournit un noeud ou d'un service de l'architecture GAIA-X. Comme le montre la 4.5 Il y a en fait deux écosystèmes, celui des infrastructures et celui des données. Pour ce qui est du thème multi-cloud on va s'intéresser à l'écosystème des infrastructures. GAIA-X pour l'usage a donc fait le choix de séparer les infrastructure et les données comme dans le modèle B I S D en version 2(NAUGÈS,

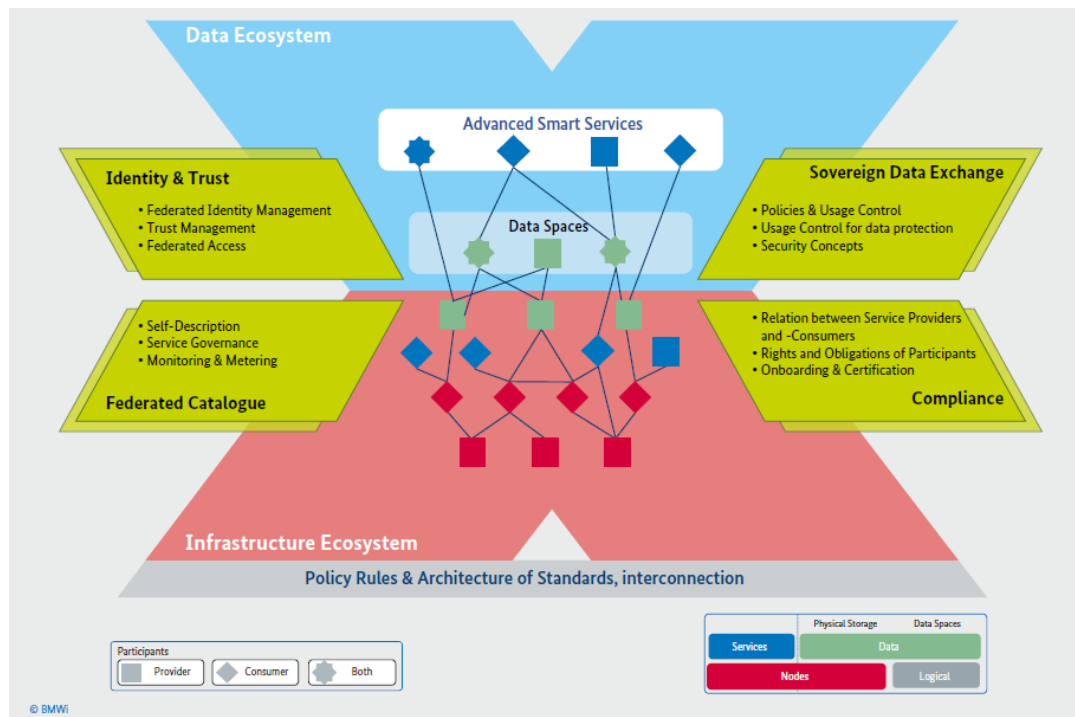


FIGURE 4.5 – Écosystème GAIA-X

p. d.). On ne peut pas dire directement que le projet est multi-cloud mais il est en tout cas multi-fournisseur et les principaux fournisseurs d'infrastructure sont pour l'instant OVHCloud et T-Mobile.

Par rapport au sujet de l'open-source GAIA-X se veut porté par ce type de produit technologique et communautaire pour appuyer son argument d'indépendance vis-à-vis d'un fournisseur de service. En effet dans le document d'architecture du projet³ On trouve l'évocation de l'Open Source Business Alliance⁴ qui propose un modèle d'évaluation technologique de manière indépendante du fournisseur.

« A strong open infrastructure ecosystem is the foundation of digital sovereignty.⁵ »

Source : "GAIA-X : Architecture technique (sortie - juin 2020)"

Sur ce sujet une communauté se détache et crée le **Sovereign Cloud Stack**⁶ en tant que l'une des plus de 60 entreprises et organisations GAIA-X. C'est alors l'un des premiers projets soutenus par l'Agence fédérale pour les innovations de rupture allemande SPRIND⁷. Ils sont actifs dans le groupe de travail qui développe des logiciels indépendants pour l'exploitation d'une infrastructure cloud. Pour eux, il est clair que le code de GAIA-X doit être open source, car c'est le seul moyen de garantir une véritable souveraineté numérique. « Ce dont nous avons besoin aujourd'hui, ce n'est

3. <https://www.data-infrastructure.eu/GAIA/Redaktion/EN/Publications/gaia-x-technical-architecture.pdf>

4. L' OSB Alliance - Association fédérale pour la souveraineté numérique eV. L'OSB Alliance considère l'open source et les normes ouvertes comme des fondements essentiels de la souveraineté numérique, de la flexibilité et de la sécurité dans le changement numérique et donc comme la réponse à l'un des plus grands défis de notre temps. <https://osb-alliance.de/>

5. Un solide écosystème d'infrastructures ouvertes est le fondement de la souveraineté numérique.

6. <https://scs.community/>

7. <https://www.sprind.org/en/projects/sovereign-cloud-stack/>

pas seulement une nouvelle prise de conscience de la souveraineté des données, mais aussi l'infrastructure informatique pour l'accompagner. nous avons besoin d'une toute nouvelle infrastructure open source. » Pour répondre à ce besoin au sein du groupe Sovereign Cloud Stack le produit OSISM⁸ permet d'avoir un aperçu des outils open-source basés sur **Ansible** et **Docker** pour gérer **OpenStack** et **Ceph**. D'une manière générale OpenStack semble être la base OpenSource retenue par exemple par les fournisseurs principaux GAIA-X c'est à dire OVHCloud et T-Mobile. On remarquera que l'écosystème open source GAIA-X n'est pas figé et peut être mené à évoluer notamment par le fait que la pile d'outils open-source dépend du choix de l'entreprise ou de l'organisation qui existe derrière. Il faut alors être conscient que l'adoption de l'écosystème GAIA-X est tellement ouvert que le risque augmente en adoptant un service ou un noeud dont la pérennité n'est ni prouvée ni assurée.

4.7 Usage entre IaaS et IaC

Les infrastructures IaaS définissent notre fournisseur cloud et on a vu précédemment que le multi-cloud rajoute une notion de multiplicité de ces offre IaaS sans pour autant toucher à l'activité principale d'une entreprise ou d'une organisation. Ici je vais vous parler des outils open source qui nous permettent d'abstraire certains services managés par les fournisseurs et ainsi ouvrir la porte à une homogénéité entre fournisseurs.

4.7.1 Le cas du FaaS

De nombreux acteurs du cloud public proposent en 2020 des fonctions lambda et la notion de *Serverless*⁹ pour simplifier au maximum la gestion des fonctions de calcul. On peut ainsi ne plus se préoccuper des "instances de calcul" et de leur élasticité quel que soit la charge de travail. Ici on va parler de **OpenFaaS**. Cet outil open source se situe en amont de la couche IaC dans une architecture cloud comme le montre la figure 4.6.

Pour faire fonctionner OpenFaaS il est alors essentiel que l'infrastructure de cloud public ou de cloud privé propose l'écosystème Kubernetes ou Docker Swarm. Ainsi OpenFaaS permet aux développeurs de déployer facilement des fonctions et des micro-services événementiels sur Kubernetes sans codage répétitif et passe-partout.

8. Open Source Infrastructure & Service Manager <https://www.osism.de/>

9. **Informatique sans serveur** : Le terme "sans serveur" ne signifie pas qu'il n'y a pas de serveurs impliqués. Cela signifie qu'ils sont gérés par les fournisseurs et non par les consommateurs. Sans trop penser à leur maintenance, les ressources informatiques sont utilisées comme des services.

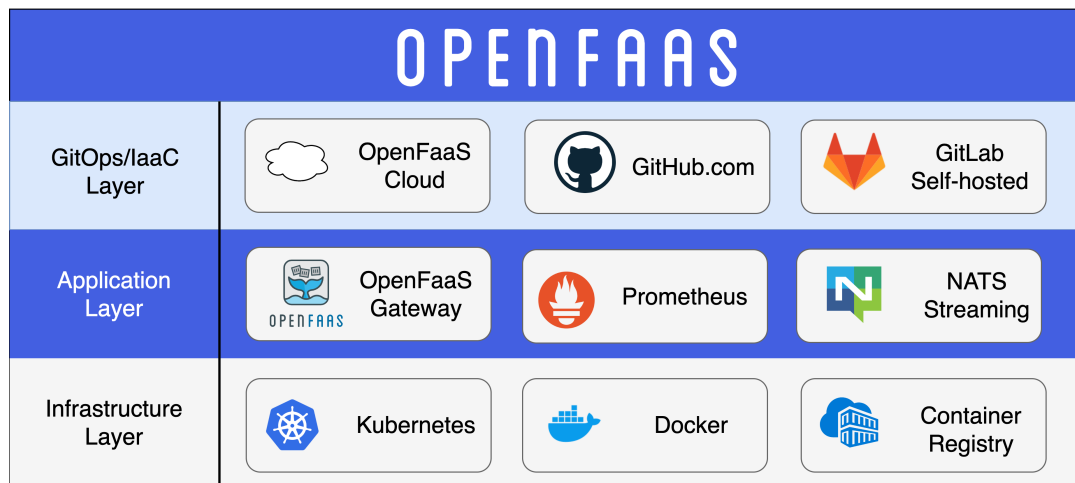


FIGURE 4.6 – Usage OpenFaaS

Chapitre 5

Migration des données et des applications

5.1 Vers une indépendance sans écosystème

Dans ce chapitre nous allons voir de quelle manière il est possible d'être indépendant d'un fournisseur cloud et plus précisément de ses infrastructures IaaS sans pour autant adopter un écosystème indépendant comme vu au chapitre 4. Il s'agira alors plutôt de se concentrer sur les pré-requis et les procédures à suivre pour changer de fournisseur en un temps raisonnable sur un ensemble d'infrastructures à migrer. Vu au chapitre 3 la stratégie est ici d'atteindre un niveau de maturité visé le *cloud resilient*. Cette situation se présente en général lors d'un besoin de migration.

5.1.1 Quelle est la probabilité que les applications changent de fournisseur d'infrastructure tout au long de leur durée de vie ?

Les enquêtes chez Gartner (MARCO, 2020) montrent que les applications déployées chez un fournisseur ont tendance à y rester. Plus une instance de calcul est spécifique à un fournisseur, moins elle a de chance d'être portable de quelque manière que ce soit. Le scénario de migration se présente alors le plus souvent lorsque le risque réglementaire est trop grand ou lorsque la fiabilité et le coût d'un fournisseur de convient plus. Dans notre contexte actuel on peut quand même parler de la volonté de souveraineté numérique en Europe qui pousse vers le haut cette volonté de migration.

5.2 Portabilité des données, un format d'échange ouvert pour une compatibilité sur les infrastructures

En pré-requis nous allons voir les formats d'échange ouvert donc exportables ou importables sur l'ensemble des infrastructures. Il est important de s'assurer que ce format d'échange soit adopté par les principaux fournisseurs de cloud public. Je ne vais pas ici énumérer ou lister les différents formats qui existent en 2020 mais sensibiliser sur le fait qu'ils soient open source et voir même libres. Ils faut aussi que le format d'échange soit assez connu et compatible pour pouvoir au moins être intégré comme un format d'échange en entrée d'un fournisseur cloud.

Prenons l'exemple de MongoDB qui est passé de la licence GNU AGPL v3.0 à la licence SSPL en 2018. L'objectif de MongoDB en sélectionnant la licence publique côté serveur (SSPL) v1.0, une licence introduite par MongoDB, exige que les améliorations de MongoDB soient publiées dans la communauté. De plus MongoDB met également les pilotes de la base de données à disposition sous la licence Apache v2.0.

Dans cet exemple on peut raisonnablement se poser la question si MongoDB continuera-t-il à fournir des logiciels open source ? La réponse de MongoDB est : Oui, les pilotes et connecteurs pris en charge par MongoDB tels que le connecteur MongoDB pour Apache Spark sont sous licence Apache License v2.0. Toutes les versions de MongoDB Community Server publiées avant le 16 octobre 2018 continueront d'être concédées sous la licence GNU AGPL v3.0 de la Free Software Foundation.

Bien que la SSPL ne soit pas approuvée par l'OSI¹, elle conserve toutes les mêmes libertés que la communauté a toujours eues avec MongoDB sous AGPL. Les utilisateurs sont libres de réviser, modifier et distribuer le logiciel ou de redistribuer les modifications apportées au logiciel. Cependant, l'Open Source Initiative (OSI) a son propre processus d'approbation de ce qu'elle considère comme une licence open source, et la SSPL n'a pas reçu l'approbation OSI. Le logiciel MongoDB sous licence SSPL n'est pas considéré comme open source par l'OSI.

Ainsi il est extrêmement difficile d'effectuer une liste exhaustive de format d'échange libre mais il n'empêche pas le fait qu'il faut toujours avoir une longueur d'avance et prévoir la migration des données au fur et à mesure du temps, de l'évolution des licences et de l'évolution des technologies.

5.3 Migration d'une architecture

Il s'agit ici de se concentrer sur le besoin de migration d'un ensemble de ressources constituant une architecture. On a vu dans le chapitre 3 qu'il n'était pas judicieux d'effectuer une migration sans au préalable avoir automatisé le déploiement des ressources associées. On va alors seulement avoir besoin d'un **plan** d'architecture. Ce plan constitue à ce moment là la seule donnée critique à garder précieusement. On va voir ici les outils open source qui nous permettent de déployer une architecture de manière quasi identique. On va alors parler de IaC qui permet de décrire une architecture virtuelle à base de fichiers et de scripts. Ensuite lors d'une migration on aura aussi besoin de garder les outils qui nous permettent de gérer le surveillance de cette architecture une fois mise en place.

5.3.1 Terraform

Terraform (*Terraform by HashiCorp p. d.*) est un logiciel open source à déployer chez un fournisseur de cloud public ou adoptant une architecture OpenStack pour les autres. Il est donc possible de déployer Terraform chez les acteurs suivants : AWS, Azure, GCP, Oracle, Alibaba, VMware vSphere, ou même chez de nouveaux acteurs sur le cloud public comme OVH avec un compte utilisateur OpenStack. Cette grande compatibilité en fait un outil multi-cloud déterminant.

Parmi la multitude de fonctionnalités que présente Terraform nous allons nous attarder sur celle qui nous intéresse, c'est à dire **créer une infrastructure reproductible**. On a avec cet outil la possibilité :

- d'avoir un environnements de production, de préparation et de développement reproductibles
- d'avoir des modules partagés pour les modèles d'infrastructure courants
- de combiner plusieurs fournisseurs de manière cohérente

Terraform facilite la réutilisation des configurations pour une infrastructure similaire, ce qui permet d'éviter les erreurs et de gagner du temps. Notons que Terraform est compatible avec l'écosystème Kubernetes en tant que fournisseur ce qui le place

1. L'Open Source Initiative est une organisation dédiée à la promotion des logiciels open source

encore un cran plus loin dans les outils intéressants faisant partie de votre trousse à outils multi-cloud. En terme de ressources supportées, Terraform supporte la mise en place de ressources IaaS, PaaS et FaaS.

5.4 Portabilité des instances de calcul

5.4.1 Le trio gagnant Packer Terraform Ansible.

Une des parties les plus importantes à savoir migrer dans une architecture c'est les instances de calcul. En effet c'est elles qui vont porter la charge des application qui tournent et qui constituent au final le moteur du coeur de métier. Le cloud impose déjà un passage vers la virtualisation des des serveurs. Ces serveur si on veut pouvoir un jour les migrer vont devoir porter en eux un certain nombres de caractéristiques tant au niveau de la configuration réseau, des restrictions de droit, de mises à jour de système d'exploitation et de configuration système. Dans cette optique de migration nous voir comment nous pouvons déployer une machine virtuelle d'un environnement virtualisé à un autre avec la bonne combinaison de logiciels open source. La recette de base la plus utilisée est la suivante :

1. Tout d'abord il va falloir créer des images de machine virtuelles à migrer pour différentes plateformes. **Packer A** est l'outil idéal pour cela, il est compatible avec les acteurs principaux de cloud public mass aussi avec VMWare, Docker, Linux KVM...
2. Ensuite une fois que nos images de machine ont été créées avec Packer et rendues accessibles sur l'environnement de destination, nous allons avoir besoin de créer les machines virtuelles sur l'environnement de destination. La tâche se complique surtout s'il y a des millier de machines virtuelles à déployer, intervient alors ici l'outil **Terraform** que va pouvoir directement utiliser les images créées à partir de Packer comme ressource.
3. Maintenant qu'on a redéployé les machines virtuelles de l'environnement source à l'environnement de destination il ne reste plus qu'à configurer comme ils se doit l'ensemble des machines virtuelles déployées. Cette opération post-migratoire essentielle pour redonner vie aux applications va être opérée avec l'outil **Ansible**, qui lui va exécuter les scripts préparés pour les machines virtuelles à configurer.

Nous venons de parler de migration de machines virtuelles qui est l'approche la plus primitive dans le cloud pour une instance de calcul mais qu'en est-il des conte-neurs ou des fonctions sans serveur FaaS? Nous avons vu dans le chapitre 4 précédent qu'OpenFaaS nous permettait de gérer les fonctions entre les infrastructures et la couche IaaS. C'est alors que se pose la question essentielle de la portabilité des images de conteneurs d'un cluster kubernetes pour la portabilité des instances de calcul cloud natives qui se différencient des instances de machine virtuelle.

Chapitre 6

Comment aborder le multi-cloud et les obstacles à son adoption

6.1 Procédures à l'adoption

On a vu précédemment qu'on pouvait construire des solutions en tant qu'éditeur de logiciels autour d'un écosystème indépendant des fournisseurs d'infrastructures. Il s'agit surtout de tirer parti du IaaS multi-fournisseur avec un centre de contrôle indépendant qui peut être apparenté à du PaaS. Si seule la portabilité des infrastructures IaaS nous intéresse alors il va falloir écrire des procédures de portabilité et les tester régulièrement. Dans le cas où on a plusieurs services qui tournent en production et en parallèle chez différents fournisseurs, les procédures seront plus complexes. Premièrement nous aurons besoin de créer une "**Zone d'atterrissage**" du multi-cloud en utilisant des outils d'automatisation du commissionnement des ressources tel que **Terraform** et **Ansible**. Nous n'avons pas encore évoqué le sujet du multi-cloud comme centre de coûts. En effet avant de basculer dans le multi-cloud il va falloir pouvoir gérer différents coûts liés à chaque fournisseur et pourvoir étiqueter et inventorier les ressources utilisées par projet. Ce travail doit être fait en amont pour ne pas perdre le contrôle sur le budget.

6.1.1 Améliorer le niveau de maturité dans le multi-cloud progressivement

Nous avons vu au Chapitre 2 les différents niveaux de maturité d'une application dans le cloud. Il est important de ne pas brûler les étapes pour que la stratégie multi-cloud fonctionne. Il n'est donc pas nécessaire de faire basculer tout un SI directement dans le bain du multi-cloud. Il faut déjà pouvoir selon le découpage du SI par applications identifier le niveau de maturité pour chacune des applications. Ensuite il est toujours plus facile parmi ces applications d'en choisir une avec le bon niveau de maturité et de criticité et essayer de l'amener vers un environnement multi-cloud. Si cette première expérience fonctionne et remplit les objectifs en temps et en budget et correspond à la stratégie multi-cloud choisie alors on peut considérer que l'application a atteint son objectif multi-cloud. En cas de succès il va falloir selon la stratégie de l'entreprise faire évoluer une à une les briques suivantes du SI. En cas d'échec il faut probablement revoir à la baisse le niveau de maturité des applications ou la stratégie multi-cloud mise en place.

6.2 Les fragilités du multi-cloud

6.2.1 Système de secours

Le système de secours doit-il être hébergé chez un autre fournisseur que le système en production. D'un point de vue pratique non car les fournisseurs de cloud public proposent de leur côté des "zones de disponibilité" dans différentes régions. D'un point de vue politique ou stratégique cela peut se discuter. On peut par exemple pour une raison de politique des coûts héberger des sauvegardes chez un fournisseur moins onéreux. Mais à ce moment là on peu se poser la question selon s'il est plus judicieux d'héberger un système de secours sur un cloud privé voir même s'il s'agit de sauvegardes de les récupérer sous forme physique. Si le RTO/RPO¹ n'est pas une priorité en effet le multi-cloud peut avoir du sens. Dans le cas où on a besoin d'un RTO/RPO réactif avec réplication du système actif/actif il est plus judicieux de faire appel à un seul fournisseur pour profiter d'une automatisation rapide

6.2.2 Des économies ?

Il est judicieux de se poser la question du prix du multi-cloud. Est-ce que multiplier le nombre de fournisseurs en choisissant pour chacun la politique de prix la plus arrangeante pour une type d'usage fait la différence ? Premièrement pour effectuer ce type d'opérations il faut pouvoir à tout moment basculer d'un fournisseur à l'autre selon la fluctuation des prix. Deuxièmement les fournisseurs de cloud public proposent des réductions selon la quantité utilisée. Dans tous les cas, il faut un bon FinOps² pour pouvoir répondre à la question. Le multi-cloud semble plutôt s'orienter vers une augmentations des coûts. En effet en plus de devoir surveiller minutieusement les infrastructures et leur coût associé sur plusieurs sites il va falloir avoir son propre support FinOps pour éviter d'exploser les coûts d'exploitation. Les fournisseurs public eux de leur côté proposent des services "managés" d'optimisation des coûts et fidélisent ainsi les clients.

6.2.3 La complexité menace le succès à l'adoption

Comme vu au chapitre 2 lorsque j'ai évoqué la problématique, la complexité menace le succès à l'adoption du multi-cloud. En effet en règle générale il ne faut pas imaginer que toutes les pièces du puzzle multi-cloud peuvent être connectées. Il y a des choix techniques qui doivent être fait en avance de phase et il faut être pleinement conscient des pièces du puzzle technique qui ne peuvent pas être portables ou qui ne peuvent être déployées chez un autre fournisseur. La volonté de vouloir interconnecter absolument tout un système entre différents clouds peut être d'une complexité déconcertante surtout si il faut intégrer une dette technique à un niveau de maturité dans le cloud assez bas. L'adoption du multi-cloud sans prendre en compte cette complexité ainsi que la redéfinition des processus de gestion mène en général à l'échec à l'adoption.

1. Durée maximale d'interruption admissible (DMIA) / Perte de données maximale admissible PDMA

2. Le Financier des Opérations

6.2.4 Les données éparpillées

Pour les mêmes raisons que celles qui régissent le choix d'un fournisseur par rapport à la conformité réglementaire voulue par rapport à une donnée, il est peu probable de vouloir qu'une donnée se retrouve chez différents fournisseurs à différents endroits géographiques. Le stockage des données est en effet souvent incompatible avec le multi-cloud lorsqu'il s'agit de données critiques. Déjà que le choix d'un cloud pour héberger des données critiques est un choix difficile à faire, il est probablement plus important de s'assurer de la portabilité des données en cas d'avarie plutôt que d'avoir ces données critiques chez différents fournisseurs à tout moment. J'aimerais aussi rajouter que le prix de migration des données d'un fournisseur à un autre peut vite devenir prohibitif. La pratique courante est d'offrir la possibilité d'importer la donnée gratuitement mais de rendre la sortie de cette donnée payante. Il est alors assez illusoire de vouloir faire voyager des données de manière sécurisée entre différents fournisseurs à moindre frais. De plus à chaque fois que la donnée voyage, elle est vulnérable aux attaques et cela est un argument supplémentaire pour appuyer la sédentarité des données critiques.

Pour terminer on peut dire que la criticité de la donnée va orienter la stratégie multi-cloud à mettre en place.

6.3 Le frein de l'open source

Les solutions open source répondent surtout au besoin d'indépendance vis à vis du multi-cloud. Cependant vu au Chapitre 2 le besoin de profiter du meilleur de la technologie est aussi une bonne raison d'adopter le multi-cloud. La plupart des technologies innovantes proposées par les fournisseurs cloud ne sont pas open source et les outils qui permettent de créer un écosystème indépendant et open source peuvent freiner plus qu'autre chose l'adoption de ces innovations. Cela est le cas car les fournisseurs de cloud veulent fidéliser leur client en proposant un ensemble de services managés. Cette position de leader sur des services innovants et propriétaires ne peut qu'être tenue par un leader mondial. De part sa position de leader AWS propose alors pléthore de services managés sans vraiment se soucier des alternatives open source qui pourraient le concurrencer. En effet, à quoi bon opter pour du multi-cloud si on trouve tout ce que l'on veut et moins cher chez un seul fournisseur ? Cet argument prend encore plus de sens depuis qu'AWS propose son service AWS Outpost évoqué dans le chapitre 3.

6.4 Red Hat OpenShift vs Google Anthos

Une chose est sûre les deux principaux poids lourds sur le marché multi-cloud open source sont Red Hat et GCP. Red Hat a commercialisé sa propre version d'OpenStack et des produits de cloud computing open source ultérieurement et continue à être un acteur dans ce domaine. En effet Red Hat a historiquement proposé un ensemble de services support payants sur des produits open source. Il a débuté par un support d'un système d'exploitation Linux basé sur Fedora et aujourd'hui il propose les solutions Open Source évoquées dans ce document tels que OpenShift qui se repose sur OpenStack mais aussi Ansible. GCP appuie sa domination dans ce domaine en fournissant l'outil Kubernetes et avec l'appui de la CNCF va pousser les entreprises à d'abord monter en maturité dans le cloud. On peut ainsi dire que GCP

redéfinit le PaaS avec des produits comme Anthos et vise ainsi être le facilitateur du multi-cloud pour les entreprises dites "cloud natives".

Chapitre 7

Conclusion

7.1 Le multi-cloud est une stratégie

Par rapport aux usage que j'ai couvert dans cette thèse, le multi-cloud ressort comme une stratégie d'entreprise pour atteindre des objectifs relatifs à des besoins. Il ne suffit pas d'avoir une stratégie pour y arriver mais il faut aussi se créer son propre écosystème pour pouvoir mettre en place cette stratégie. Il y a alors un ensemble de choix technique et de compromis architecturaux à faire avant que cela ne soit possible. C'est aussi pour cela qu'un certain nombre d'entreprises abandonne. Si on n'a pas le niveau de maturité technologique adéquat il est clair que cette stratégie peut coûter cher. Il est alors important que la stratégie multi-cloud soit claire, elle doit bien déterminer les procédures à mettre en place à l'image d'une transition numérique vers un système plus complexe. Il est cependant illusoire de vouloir rendre tout un SI interopérable. En pré requis il est conseillé d'identifier les parties qu'il est possible de désolidariser par rapport à un cloud en particulier. Il est notamment déconseillé de vouloir porter dans le cloud public des applications temps réel ou de *trading* financier.

7.1.1 L'open source fédère ses adeptes autour de plusieurs communautés multi-cloud.

On le voit bien OpenStack fédère les fournisseurs de cloud qui peine à trouver leur place. En regardant les fédérations souveraines telles que GAIA-X on voit que l'open source est le fer de lance du projet. Pour un adhérent à un produit open source la communauté va donner la force nécessaire pour se démarquer des leader par le fait d'appartenir à un écosystème résilient à base de multi-cloud. Il y a un ensemble de communautés open-source et multi-cloud à rejoindre selon le la stratégie multi-cloud. La communauté Kubernetes se démarque par son grand nombre d'adeptes, ensuite il est possible de faire parti de la communauté CloudFoundry, OpenShift, GAIA-X ou tout autre communauté non citée. Ceci est un choix d'importance en terme de pérennité et de support à prendre en compte au moment de basculer sur une stratégie multi-cloud.

7.1.2 service managés pour l'hybride et open source pour le multi-cloud

Lorsqu'on va mettre en place la stratégie vers le multi-cloud, le meilleur moyen de rendre une partie du SI compatible "multi-cloud" la première approche est d'utiliser un écosystème PaaS orienté multi-cloud pour les services développés comme coeur de métier. Si certains services utilisés dans le SI peuvent être remplacés par une

solution SaaS c'est probablement le moyen le plus facile d'y arriver sur les applications support. Nous avons vu au chapitre 4 que Kubernetes a été cédé par Google en 2018 et cela place maintenant cet écosystème comme la solution universelle open source pour le multi-cloud natif. On a ainsi d'autres outils qui prennent le relais comme OpenShift pour y rajouter une couche d'abstraction et de gestion. Istio et Spinnaker font aussi partie de ce panorama d'outils dans le multi-cloud. D'un autre côté on remarquera que les solutions managées telles qu'AWS Outpost, Azure ARC ou google Anthos sont plutôt orientées vers le cloud hybride. Même si certains de ces outils managés par les principaux fournisseurs de cloud public se disent multi-cloud, ils n'arrivent pas à s'intégrer aussi bien sur différentes plateformes que les produits open-source.

7.2 Effectuer un choix selon le scénario d'utilisation

7.2.1 Pour les multi-cloud natifs : Docker + Kubernetes + Spinnaker + Istio

En 2020 la meilleure option possible pour la gestion multi-cloud d'un éditeur de logiciels et celui de la gestion multi-cluster Kubernetes comme vu au chapitre 4. Cette configuration n'est disponible que pour les application qui ont veulent atteindre un niveau de maturité cloud natif. la pile de technologies Docker + Kubernetes + Spinnaker + Istio est d'une efficacité redoutable et elle peut même être couplée avec un écosystème open source PaaS OpenShift ou Cloud Foundry ou même avec une solution propriétaire comme Google Anthos.

7.2.2 Pour ceux qui veulent garder leur architecture historique

Pour les entreprises qui cumulent de la dette technique au fur et à mesure du temps, l'adoption du multi-cloud n'est probablement pas une bonne idée. On a vu au chapitre 2 la nécessité de rentrer dans un cadre minimal de degré de maturité dans le cloud avant de pouvoir aborder les besoins liés au multi-cloud.

7.2.3 Pour les adeptes du cloud souverain : OpenStack et Terraform

L'important pour les adeptes du cloud souverain sera la confidentialité des données et la conformité réglementaire. On va parler ici plus spécifiquement des adeptes qui ne veulent pas se lier à un fournisseur de cloud public américain pour ne pas s'exposer au cloud ACT. Au vu du manque de cohérence des solution alternative et même si le projet Gaïa-X commence à voir le jour il va falloir prendre certaines mesures en avance de phase.

Une stratégie multi-cloud est cruciale pour ce type de demande. En effet les échecs passés avec certains projet de cloud souverain comme avec CloudWatt ou Numergy rappellent qu'il est critique d'avoir une stratégie de sortie comme vue au chapitre 3. Il est alors important de se préparer techniquement avec les bons outils et le bon formats d'échange vus au chapitre 5 pour une procédure de migration en cas de besoin. Le prérequis minimal serait alors pour un fournisseur de cloud souverain d'adopter **OpenStack**. Pourquoi OpenStack? Et bien même si OpenStack n'a pas rencontré pas un franc succès dans son passé, il permet en cas de stratégie de migration de pouvoir récupérer le plan architectural des ressources avec l'outil **Terraform** et l'outil **Ansible**. En effet comme Terraform et Ansible sont compatibles avec OpenStack, la combinaison de ces deux outils permettra en cas migration d'avoir la possibilité de

récupérer une grande partie d'un SI. Même si les fournisseurs de cloud souverain sont encore loin de ce qu'il se fait sur les clouds publics en terme d'innovation technologique il est important de prendre en compte la compatibilité avec un écosystème **Kubernetes** lors du choix d'un fournisseur de cloud souverain. En effet nous avons vu précédemment qu'il était possible avec **Kubernetes** de partager des charges de travail entre différents fournisseurs de cloud sans pour autant devoir partager les données hébergées sur un **cluster kubernetes** souverain par exemple.

7.3 Kubernetes est la pierre angulaire du multi-cloud

Pour conclure sur les différentes thématiques Open-Source et multi-cloud, j'aimerais porter une attention particulière sur l'outil Kubernetes. Vu précédemment, Kubernetes s'inscrit systématiquement dans la pile de logiciels open source requise pour vers une stratégie multi-cloud. Que ce soit OpenShift, GAIA-X ou Anthos, un écosystème se voulant indépendant d'un fournisseur d'infrastructures de calcul réserve une place importante à Kubernetes. Il existe un certain nombre de contrepoints quant à l'utilisation de Kubernetes¹ et son usage pour la portabilité des applications. Certes il peut être coûteux de mettre en place tout un système de portabilité basé sur Kubernetes fait maison mais il faut garder à l'esprit qu'il existe au jour d'aujourd'hui des solutions propriétaires comme Anthos qui répondent à ces contrepoints.

7.3.1 Deux grands acteurs dans le panorama multi-cloud open source : Red Hat et Google

Red Hat et son produit OpenShift ainsi que Google qui a initié Kubernetes et qui revient aujourd'hui avec son produit Anthos sont les acteurs principaux dans le panorama multi-cloud. Ils tirent leur épingle du jeu autour d'une même communauté, la CNCF. Il reste maintenant à voir comment le projet européen GAIA-X qui prône les vertus de l'open source va se rapprocher de cette communauté et comment il va se placer par rapport aux produits déjà existants.

7.4 Le mot de la fin

J'espère que la lecture de cette thèse vous a permis de démystifier le passage au multi-cloud. J'espère aussi avoir réussi à expliquer en quoi une combinaison de logiciels open-source est essentielle pour répondre à cette stratégie dans un contexte où les fournisseurs de IaaS et de PaaS essaient tous de tirer leur épingle du jeu. Pour tous les logiciels open source que je n'aurai pas cités mais qui sont certainement intéressants j'espère au moins avoir sensibilisé la curiosité du lecteur qui se pourra les découvrir ou les soutenir au grès de ses besoins. Je vous souhaite alors une réussite lors de l'application de votre stratégie multi-cloud.

1. Un article du blog Gartner sur "Pourquoi adopter Kubernetes pour la portabilité des applications n'est pas une bonne idée"(MARCO, 2020)

Annexe A

Index outils et licences

- **Terraform** – Mozilla Public License 2.0
- **Kubernetes** – Apache License 2.0
- **Docker** – Apache License 2.0
- **Spinnaker** – Apache License 2.0
- **Istio** – Apache License 2.0
- **Git** – GNU GPLv3 2.0
- **Cloud Foundry** – Apache License 2.0
- **OpenStack** – Apache License 2.0
- **OpenShift** – Apache License 2.0
- **HBase** – Apache License 2.0
- **Hadoop** – Apache License 2.0 et LGPL
- **Ceph** – Apache License 2.0
- **OSISM** – Apache License 2.0
- **Packer** | HashiCorp – Mozilla Public License 2.0
- **OpenFaaS** – MIT License

Annexe B

Liste des licences open source principales

B.1 GNU AGPLv3

Les autorisations de cette licence copyleft¹ la plus puissante sont conditionnées à la mise à disposition du code source complet des œuvres sous licence et des modifications, qui incluent des œuvres plus volumineuses utilisant une œuvre sous licence, sous la même licence. Les avis de droit d'auteur et de licence doivent être préservés. Les contributeurs accordent expressément les droits de brevet. Lorsqu'une version modifiée est utilisée pour fournir un service sur un réseau, le code source complet de la version modifiée doit être rendu disponible.

B.2 GNU GPLv3

Les autorisations de cette forte licence copyleft sont conditionnées à la mise à disposition du code source complet des œuvres sous licence et des modifications, qui incluent des œuvres plus grandes utilisant une œuvre sous licence, sous la même licence. Les avis de droit d'auteur et de licence doivent être préservés. Les contributeurs accordent expressément les droits de brevet.

B.3 GNU LGPLv3

Les autorisations de cette licence copyleft sont conditionnées à la mise à disposition du code source complet des œuvres sous licence et des modifications sous la même licence ou la GNU GPLv3. Les avis de droit d'auteur et de licence doivent être préservés. Les contributeurs accordent expressément les droits de brevet. Cependant, une œuvre plus importante utilisant l'œuvre sous licence via des interfaces fournies par l'œuvre sous licence peut être distribuée sous des termes différents et sans code source pour l'œuvre plus importante.

B.4 Mozilla Public License 2.0

Les autorisations de cette licence copyleft faible sont conditionnées à la mise à disposition du code source des fichiers sous licence et des modifications de ces fichiers sous la même licence (ou dans certains cas, l'une des licences GNU). Les avis

1. L'autorisation donnée par l'auteur d'un travail soumis au droit d'auteur (œuvre d'art, texte, programme informatique ou autre) d'utiliser, d'étudier, de modifier et de diffuser son œuvre, dans la mesure où cette même autorisation reste préservée.

de droit d'auteur et de licence doivent être préservés. Les contributeurs accordent expressément les droits de brevet. Cependant, une œuvre plus importante utilisant l'œuvre sous licence peut être distribuée sous des termes différents et sans code source pour les fichiers ajoutés dans l'œuvre plus importante.

B.5 Apache License 2.0

Une licence permissive dont les principales conditions exigent la préservation des mentions de droits d'auteur et de licence. Les contributeurs accordent expressément les droits de brevet. Les œuvres sous licence, les modifications et les œuvres plus importantes peuvent être distribuées sous des conditions différentes et sans code source.

B.6 MIT License

Une licence permissive courte et simple avec des conditions exigeant uniquement la préservation des avis de droit d'auteur et de licence. Les œuvres sous licence, les modifications et les œuvres plus importantes peuvent être distribuées sous des conditions différentes et sans code source.

B.7 Boost Software License 1.0

Une licence permissive simple ne nécessitant que la préservation des avis de droits d'auteur et de licence pour la distribution source (et non binaire). Les œuvres sous licence, les modifications et les œuvres plus importantes peuvent être distribuées sous des conditions différentes et sans code source.

B.8 Sans licence

Une licence sans aucune condition qui consacre des œuvres au domaine public. Les œuvres sans licence, les modifications et les œuvres plus importantes peuvent être distribuées sous des conditions différentes et sans code source.

Bibliographie

- DIRK, Basyn (nov. 2019). *Croissance de 17% du Cloud mondial en 2020, selon Gartner*. fr-FR. URL : <https://www.channelnews.fr/croissance-de-17-du-cloud-mondial-en-2020-selon-gartner-92906>.
- Google takes a step back from running the Kubernetes development infrastructure (p. d.). en-US. URL : <https://social.techcrunch.com/2018/08/29/google-steps-back-from-running-the-kubernetes-infrastructure/>.
- HALL, Laura (juin 2020). *In Cloud Computing, Open Source Becomes Big Business*. und. Text. URL : <http://www.nasa.gov/directorates/spacetech/spinoff/in-cloud-computing-open-source-becomes-big-business>.
- IBRYAM, Bilgin (fév. 2017). *Kubernetes and the Microservices Hierarchy of Needs*. en-US. URL : <https://thenewstack.io/introducing-microservices-hierarchy-needs/>.
- informatique en nuage / FranceTerme / Ressources / Accueil - Culture.fr* (p. d.). URL : <http://www.culture.fr/franceterme/terme/COGE746>.
- KIM, Weins (mai 2020). *Cloud Computing Trends : 2020 State of the Cloud Report*. en-US. URL : <https://www.flexera.com/blog/industry-trends/trend-of-cloud-computing-2020/>.
- MARCO, Meinardi (sept. 2020). *Why Adopting Kubernetes for Application Portability Is Not a Good Idea*. en. URL : <https://blogs.gartner.com/marco-meinardi/2020/09/04/adopting-kubernetes-application-portability-not-good-idea/>.
- NAUGÈS, Louis (p. d.). *B I S D : la version 2 du modèle B I S*. URL : https://nauges.typepad.com/my_weblog/2019/08/-b-i-s-d-la-version-2-du-mod%C3%A8le-b-i-s.html.
- RICHARD M., Stallman (jan. 2020). *En quoi l'open source perd de vue l'éthique du logiciel libre*. fr. URL : <http://www.gnu.org/philosophy/open-source-misses-the-point.html>.
- Terraform by HashiCorp* (p. d.). en. URL : <https://www.terraform.io/index.html>.
- The twelve-factor app is a methodology for building software-as-a-service apps* (p. d.). *The Twelve-Factor App*. URL : <https://12factor.net/>.
- TITTEL, Ed et al. (2019). *Multicloud Portability For Dummies, Red Hat Special Edition*. John Wiley & Sons, Inc. ISBN : 978-1-119-54761-7.
- Using Kubernetes, Spinnaker and Istio to Manage a Multi-cloud Environment (Cloud Next '18) - YouTube* (p. d.). URL : <https://www.youtube.com/watch?v=kKC-VgAptII>.