Retour d'expérience sur l'évolution d'une infrastructure à l'ancienne vers des clouds institutionnels

Damien Ferney, Philippe Depouilly, Laurent Azema, David Delavennat, Romain Theron

Retour d'expérience sur l'évolution d'une infrastructure à l'ancienne vers des clouds institutionnels

Damien Ferney
LMBP
3 Place Vasarely
63 178 Aubière

Philippe Depouilly
IMB
351, cours de la Libération
33 405 Talence

Romain Theron
Institut Denis Poisson
Rue de Chartres
45 067 Orléans

Laurent Azema
GRICAD
700, Avenue Centrale
38 400 Saint-Martin d'Hères

David Delavennat
CMLS
École polytechnique
91 128 Palaiseau

Résumé

Depuis 2004, la PLMteam, une équipe d'ASR du réseau métier MATHRICE, déploie des services numériques pour la communauté mathématique dans le cadre d'infrastructures de laboratoires. À l'occasion de son renouvellement quinquennal, le réseau s'est interrogé sur la refonte de son modèle en anticipant des évolutions métiers. Nous avons opté pour la migration de notre infrastructure sur quatre clouds : OpenStack (Gricad-Nova à Grenoble et Virtualdata à Paris-Saclay) et VMware (GRICAD et RENATER). Nous nous orientons donc vers une consolidation multicloud, générique et redondante des services dans le périmètre des outils pour la recherche.

Le contexte de la pandémie nous a soudainement fait basculer de la réflexion à la concrétisation avec le déploiement de services de visioconférence et de plateformes de recherche sur la Covid19.

Nous avons alors eu à aborder les problématiques suivantes : les libertés et particularités d'implantation des IaaS ; les conceptions différentes de connectivité réseau ; l'état de l'art dans la gestion de services cloud ; la variété des modes d'organisation de chaque site (interlocuteur, tickets, procédures, etc.) ; l'orchestration de conteneurs.

Ces aspects ont bousculé notre organisation. Nous avons dû appréhender de nouveaux concepts tels que l'Infrastructure-as-Code et nous détacher ainsi de l'administration du serveur BareMetal.

Mots-clefs

cloud, DevOps, infrastructure, IaaS, orchestration, OpenStack, VMware

1 Contexte

Le réseau métier MATHRICE[1] (rassemblant les informaticiens des laboratoires de mathématiques) a constitué en 2004 un GdS (Groupement de Service CNRS) afin de mettre en œuvre des services numériques[2] à destination de la communauté mathématique française.

1 https://mathrice.fr

2 https://plm-doc.math.cnrs.fr/doc/spip.php/article5

La PLMteam est une équipe d'une douzaine de Mathriciens qui développent la Plateforme en Ligne pour les Mathématiques (PLM) en parallèle du travail d'ASR dans leurs laboratoires. Elle maintient des services collaboratifs et distants, accessibles par 6500 utilisateurs réguliers. Ces services sont aussi divers que la coédition, l'accès nomade aux revues scientifiques, le déploiement d'applications *web* sur *Platform-as-a-Service* (*PaaS*) ou les *notebooks jupyter*.

1.1 Description de la plateforme historique

La situation de départ de cet article est l'architecture PLM décrite dans un article JRES-2013[1]. Celle-ci, composée de serveurs physiques hébergés dans plusieurs laboratoires de mathématiques (Bordeaux, Lille, Angers, Lyon et Grenoble), a été financée par l'INSMI[3] du CNRS. Ces sites avaient des particularités comme l'autonomie électrique, la politique de filtrage réseau du campus, le nombre de collègues présents sur place ou le nombre d'adresses IP dévolues pour les déploiements. Ces sites proposaient des alternatives mutuelles en cas de problème et la PLM tirait profit du travail réalisé pour l'hébergement des ressources informatiques des laboratoires.

Chaque site hébergeait un serveur de fichiers *NFS* et son secours, des serveurs hôtes de virtualisation *libvirt/KVM* et un serveur d'administration avec l'accès *SSH* et les services de base comme le cache *DNS*, le relai *SMTP*, la référence *NTP*, le serveur de log. Les services étaient découpés en machines virtuelles mono-fonction avec différents systèmes d'exploitation[4] liés au libre choix de l'instigateur du projet.

Pour faciliter les communications internes, chaque *VM* avait une interface dans un réseau privé local en plus de son interface publique sur laquelle exposer son service. Le service *L3VPN* de RENATER permettait d'interconnecter ce réseau privé avec celui des autres sites PLM. En parallèle une zone *DNS* privée facilitait la résolution des noms de services internes. Un exemple d'usage était le service de sauvegarde backuppc qui accédait en *SSH* à chaque *VM* par son nom interne.

La PLMteam avait pris des habitudes de travail. La configuration de la plateforme, construite de manière monolithique et au fur et à mesure, avait abouti à un système complexe, global, difficile à maintenir de manière parcellaire. Il devenait impossible de développer un service sans une connaissance complète de la plateforme. Seules les personnes à la base du projet pouvaient appréhender la complexité globale, compliquant l'intégration de nouveaux collègues. Il était de plus en plus difficile de maintenir l'architecture. L'interdépendance des mises à jour et des modules utilisés avait conduit à dupliquer le système de déploiement Puppet en plusieurs versions parallèles afin de ne pas prendre le risque de casser l'existant.

Le projet de portail pour les Mathématiques en collaboration avec le RNBM[5] et la cellule Mathdoc[6], nous avait guidé vers une démarche DevOps et une meilleure intégration des services[2]. Il avait nécessité du temps de formation et de développement pris sur l'administration système et ce modèle d'intégration a complexifié le déploiement de nouveaux services et apporté des singularités.

2 Le projet de renouvellement, une opportunité

Tous les 5 ans, MATHRICE, devenu Réseau Thématique depuis 2021, propose un bilan et un projet de renouvellement de son organisation et de ses actions à la direction de l'INSMI. L'année 2020 était dévolue à la finalisation et à la rédaction du projet par un groupe constitué d'une vingtaine de

3 Institut National des Sciences Mathématiques et de leurs Interactions

4 FreeBSD, CentOS, Debian, Ubuntu

5 Réseau National des Bibliothèques de Mathématiques

6 Cellule de coordination documentaire nationale pour les mathématiques

membres du réseau. Le passage vers une « PLM dans les nuages » était un objectif du renouvellement[7].

2.1 Objectif du projet « PLM dans les nuages »

Au vu du bilan et de nos réflexions, les buts recherchés étaient multiples.

- Se décharger de l'administration de l'infrastructure physique pour compenser : la diminution des ressources humaines dans les laboratoires, les délais de recrutement lors de mouvements de personnels, le besoin de temps pour s'occuper des services et du soutien aux mathématiciens.

- Se réorienter résolument vers la démarche DevOps comme évolution métier des ASR avec : la configuration décrite par un code, l'adoption des normes du développeur (spécification des fonctionnalités, utilisation de langages, formalisation, documentation, tests de validation, gestion de version du code), l'industrialisation système avec le provisionnement et le déploiement en nombre.

- Utiliser un hébergement efficient, rationaliser et mutualiser les équipements et les *OS*, adapter à la charge l'allocation des ressources pour se tourner vers une démarche écoresponsable.

- Dissocier le déploiement d'application de la maîtrise de l'architecture PLM pour : intégrer de nouvelles recrues, gagner de la souplesse dans la gestion des droits, faciliter les tâches support et l'utilisation de nouvelles *API*, faciliter l'émergence de nouveaux services.

- Ouvrir les codes et configurations pour : permettre aux utilisateurs de s'en inspirer, présenter le savoir-faire de la PLMteam, susciter des vocations pour venir contribuer et enrichir l'offre.

2.1.1 Notre notion du *cloud*

Le *cloud* est le fait de créer et d'utiliser de manière distante et autonome des ressources d'infrastructures, des ressources d'applications, ou des instances de services.
Nous distinguons les services *clouds IaaS*, *PaaS* et *SaaS*. La PLM propose à ses utilisateurs du *PaaS* voire du *SaaS*. Mais le passage dans le nuage consiste à utiliser un *IaaS* pour gérer l'architecture PLM elle-même.

En effet, un service de virtualisation donne accès à une souscription de ressources virtuelles pour obtenir un système complet avec un accès *root*. C'était insuffisant pour avoir la souplesse nécessaire. Nous étions de plus incapables de provisionner 5 ans à l'avance le nombre et la taille des *VM* de l'infrastructure. Par contre cela répondait à certains de nos besoins où nous avons la nécessité de mettre à disposition une machine virtuelle gérée de manière presque autonome par d'autres personnes.

L'*IaaS* est une forme d'évolution mais qu'il ne faut pas confondre avec un simple système de virtualisation. L'*IaaS* propose des interfaces d'administration des ressources virtuelles pour créer, lancer, interrompre, détruire les machines virtuelles selon le besoin. Ces interfaces peuvent être interactives ou programmables, par interface web (cf. Figure 1) ou ligne de Commande (*CLI*) ou par *API*. L'*IaaS* propose aussi des réseaux internes virtuels et des interconnexions publiques pour construire les topologies souhaitées. L'*IaaS* permet de créer et connecter les volumes utiles sur chaque *VM*.

7 https://plmbox.math.cnrs.fr/f/ec9406d5aba3409faccf/?dl=1

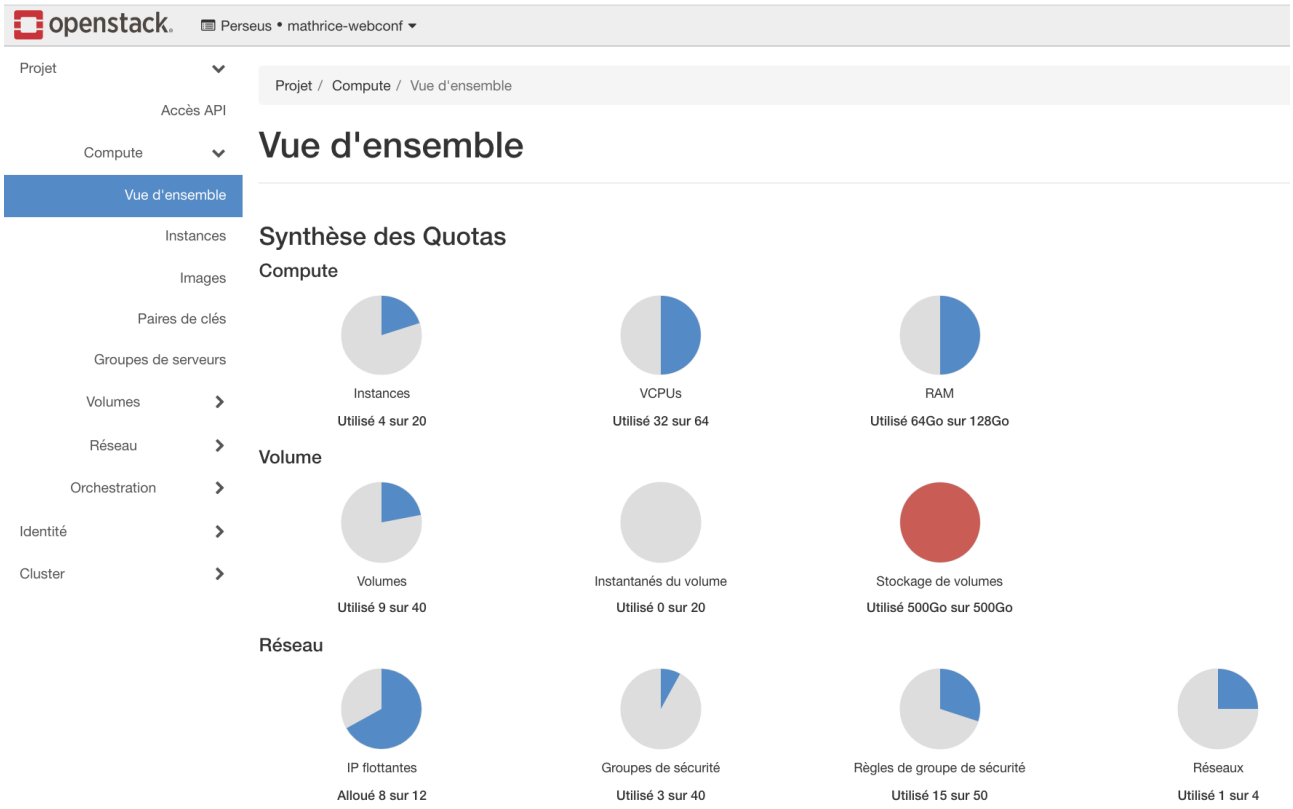

Figure 1 – Exemple d'interface d'administration d'OpenStack

2.1.2 Une transition accélérée

La covid a été une accélératrice. Nous avons développé de nombreux services supplémentaires (*CodiMD*[8], *rocketchat*, *BBB[4]*, …). L'usage d'*OKD*[9] a explosé et nous avons pu produire la plateforme MODCOV19[10] en un temps très court (les outils existaient, la plateforme *PaaS* a permis de nouvelles instances et des configurations facilitées.

Nous avons cherché des « *workers BBB* » en nombre pour tenir la charge des webconférences de nos chercheurs et pour pouvoir accueillir des groupes nombreux. Nous avions des relations privilégiées avec IJClab[11] et avec GRICAD[12]. Nous nous sommes tournés vers eux pour nos besoins urgents de *VM*.

La discussion d'un projet PIA3 autour de l'*openscience* avec MATHRICE et le groupe CALCUL comme soutien technologique n'a pas abouti mais a apporté des idées neuves. De plus, nos chercheurs ne se déplaçant pas, nous avons bénéficié d'un report des frais de mission vers un besoin d'infrastructure.

8 Éditeur de notes collaboratif, temps réel, au format Markdown

9 *Origin Kubernetes Distribution* : La distribution communautaire basée sur kubernetes sur laquelle est fondée RedHat/OpenShift

10 https://modcov19.math.cnrs.fr/

11 Institut Joliot-Curie Laboratory (IN2P3/CNRS)

12 Infrastructure de Calcul Intensif et de Données (CNRS, Université Grenoble Alpes, Grenoble-INP, INRIA)

3 Souscription des ressources

3.1 Choix des sites

Nous tourner vers des *clouds* institutionnels de l'ESR, offrant un accès direct au réseau de la recherche RENATER et disposant d'un niveau de sécurité que nous connaissons sur nos campus universitaires, nous paraissait plus proche de notre type de fonctionnement. De plus, pour envisager le grand saut, des collaborations entre collègues nous semblaient plus appropriées qu'un rapport client-fournisseur.

Différents types de *cloud* sont proposés, les DSI proposent des *clouds* de service, les mésocentres proposent des *clouds* de recherche. Les premiers apportent un niveau de sécurisation important avec de la réplication, de la continuité et de la sauvegarde nécessaires à des services mutualisés devenus stratégiques. Nous avons pensé bénéficier de ce niveau de redondance pour nos services critiques. Les *clouds* de recherche apportent plus de souplesse et de facilité d'accès. Nous avions en tête d'utiliser les ressources qui sont proposées aux chercheurs eux-même. Souscrire à ces services était aussi une manière de les connaître et de les promouvoir dans nos laboratoires.

En 2019, une première étude avait été initiée avec les Mathriciens disposant d'une offre d'hébergement *cloud* de proximité. Les premiers sites contactés étaient Virtualdata (plateforme informatique opérée par IJCLab et l'IAS[13]) à l'Université Paris-Saclay et l'Université de Grenoble Alpes (UGA). D'une part, Virtualdata fournissaient déjà de la ressource à notre réseau MATHRICE à titre expérimental, et d'autre part, le site était relié au réseau *L3VPN* de la PLM. GRICAD est administrativement géré par l'INSMI notre institut de référence et MODCOV19 était déjà issu de notre collaboration.

Le financement exceptionnel qui nous a été attribué à la faveur des reliquats de budget de l'année 2020 nous a contraint à prendre une décision de façon accélérée. Le choix a donc été fait de répartir principalement les investissements sur les deux sites.

3.2 Descriptif des sites retenus

Les sites ont chacun leurs spécificités en termes d'accès aux ressources et d'organisation.

3.2.1 Le site de Grenoble

L'UAR[14] GRICAD[3] est née de l'initiative de mutualisation du site universitaire grenoblois. Elle est chargée des ressources pour la recherche comme les clusters de calcul du mésocentre ou le *cloud* OpenStack et prend part aux différents comités techniques qui pilotent les projets de regroupement des serveurs dans un datacentre, de partage du stockage, ou de virtualisation.

Les discussions entre MATHRICE et GRICAD ont permis l'accès à plusieurs de ces infrastructures. Gricad-Nova est le *cloud* recherche basé sur OpenStack. WINTER est la solution de virtualisation VMWare mettant à disposition des machines virtuelles nues et sécurisées. SUMMER est la solution de stockage, sauvegardé et réparti sur 3 sites, pour disposer d'un espace de données sécurisé.

3.2.2 Le site Paris Saclay

P2IO[15] est l'un des 100 laboratoires d'excellence (labEx). Il réunit les laboratoires du Campus Paris-Saclay impliqués dans la physique des 2 infinis et des origines. Virtualdata[16] est une plateforme

13 Institut d'Astrophysique Spatiale

14 Unité d'Appui à la Recherche

15 http://www.p2io-labex.fr

16 https://platforms.in2p3.fr/platform/709/details

opérée depuis 2013 par IJClab et l'IAS ouverte aux partenaires de l'Université Paris-Saclay. À côté de la partie IJClab (IPNO & LAL) de la grille GRIF préexistante, un *cloud* OpenStack, devenu la pierre angulaire de la plateforme est complété par une infrastructure de stockage CEPH distribuée.

3.2.3 Le site RENATER

RENATER a toujours fourni à MATHRICE la possibilité d'expérimenter des technologies avancées. Nous sommes utilisateurs du service *L3VPN* qui relie nos sites historiques. Nous avons expérimenté le *L2VPN* pour tenter de réaliser de la redondance de sites pour les services de la PLM. RENATER nous a, cette fois, gracieusement permis d'instancier quelques machines virtuelles nues pour installer notre brique d'authentification, dans leur datacentre Interxion à Aubervilliers.

3.2.4 Le site de Bordeaux

Nous avons choisi de garder le site historique situé à l'Institut de Mathématiques de Bordeaux (IMB) pour plusieurs raisons :

- nous y avons réalisé récemment des investissements matériels importants pour déployer une première plateforme OKD *baremetal* et soutenir certains besoins de stockage ;
- quatre membres de la PLMteam sont hébergés à l'IMB ;
- nous souhaitons, dans un premier temps, garder une part d'infrastructure complètement maîtrisée au service d'une réactivité accrue ;
- nous avons besoin d'un site où installer notamment des serveurs à base de *GPU* en attendant que l'offre apparaisse sur les clouds.

3.3 Les ressources souscrites

Nous sommes dans un contexte où les *clouds* institutionnels ne sont pas organisés comme des fournisseurs de services privés type GAFAM. La refacturation n'est pas le but premier et les coûts mutuels sont souvent financés à la source, parfois sur des financements liés à des projets. Virtualdata prend en compte les dépenses de fluides mais ne facture pas la couche OpenStack. Gricad-Nova a plutôt un modèle de fonctionnement par projets scientifiques via le financement de nœuds de calcul. Sur WINTER la facturation est fonction du nombre de *VM*. SUMMER nous fournit du stockage hautement sécurisé et GRICAD a pris sous sa coupe les ressources de MATHRICE pour une location annuelle auprès de l'UGA. À cela s'ajoutent quelques *VM* sur le datacentre de RENATER. Les ressources ont été souscrites pour 5 ans.

	Total des ressources disponibles				Ressources allouées à MATHRICE			
Sites :	Noeud	Coeurs HT	Mémoire	Stockage	Coeur HT	Mémoire	Stockage	FIP Max
Gricad-Nova	14	1152	4,8To	173To Net (x3)	864 (576 x 1,5 suralloué)	2,3To	53To	512
GRICAD Winter/ Summer	24		1,6To	460To vSan / 6,6 Po NetApp	20	28Go	10To	/
Virtualdata	130	10000	26To	300To Net (x3)	512	1To	64To	256

Figure 2 – ressources souscrites

À travers GRICAD, nous avons ainsi accès à des ressources OpenStack « Ussuri ». Quelques *VM* VMWare (Vsphere version 6.7) sur WINTER sont dédiées à la Webconférence BigBlueButton.

À travers IJClab, nous avons accès à des ressources OpenStack « Queens » de Virtualdata.

À travers RENATER, nous utilisons des machines virtuelles VMWARE.

3.4 Accès aux sites

Les systèmes d'information des *clouds* se basent généralement sur l'annuaire interne de leur personnel. Notre intégration a demandé la création de comptes pour chaque membre de la PLMteam. De notre côté, cela a engendré une multiplicité d'authentifications, chaque site ayant ses propres règles de constitution d'identifiant et de validité de mot de passe.
L'accès aux consoles VMWare ou vSphere sont contrôlées à travers des *VPN* différents (global protect, Cisco-anyconnect) qui nécessitent leur propre identification. Certains *VPN* routent l'ensemble du trafic et sont donc mutuellement exclusifs y compris avec notre OpenVPN. Cela rend leur usage simultané impossible sauf à utiliser des machines virtuelles d'administration dans lesquelles fonctionnerait le client *VPN* spécifique au site auquel on souhaite accéder.

3.5 Des acteurs locaux connus

La PLMteam a des liens particuliers avec ces deux sites à travers deux de ses membres. Laurent Azema[17] travaille maintenant à GRICAD et David Delavennat[18] travaille de longue date sur Virtualdata. Ces 2 personnes œuvrent grandement dans la confiance mutuelle. Ils connaissent le fonctionnement et les besoins de la PLM. Ils nous aident à comprendre comment sont perçues nos demandes. Ils facilitent la compréhension des besoins. Ils contactent les bons interlocuteurs sur des demandes bloquantes ou urgentes. Ils appuient nos suggestions d'évolution auprès des administrateurs locaux. Celles-ci, partagées avec d'autres usagers, motivent ou accélèrent la mise en place de nouvelles fonctionnalités et font progresser le service au bénéfice de tous.

17 Directeur de Mathrice (2016-2020)

18 Ingénieur au Centre de Mathématiques Laurent Schwartz - Ecole polytechnique

4 La démarche

Les sites étant choisis et disposant de plus que les ressources nécessaires, le travail semblait simple pour migrer les *VM* des sites historiques à l'identique et seulement changer le nommage. Mais le travail commençait !

4.1 Répartition des services

Nous avons imaginé une répartition avec redondance en fonction des capacités des différents sites. WINTER et SUMMER offrent de la sécurité en terme de stockage et de taux de disponibilité. Le paramétrage réseau (*openvswitch* et *arp-responder*) de Gricad-Nova permet d'utiliser l'installeur openstack d'OKD. Sur Virtualdata, nous avions déjà déployé une infrastructure BBB. Enfin, il semblait naturel de profiter du site de RENATER pour déployer la brique d'authentification des utilisateurs, pour des raisons de disponibilité mais aussi de proximité des composants et notamment avec la fédération d'identité.

Sur Gricad-Nova :
- reproduire l'infrastructure des services, OpenStack servant ici simplement à remplacer KVM et l'architecture réseau d'un site de la PLM ;
- une *PaaS* à l'aide d'OKD pour l'hébergement web ;
- des *VM* vierges pour les projets utilisateurs qui le nécessitent ;
- une redondance de l'architecture de webconférence BigBlueButton (BBB).

Sur WINTER :
- les services scalelite, greenlight, coturn, supervision pour disposer d'un niveau de fonctionnement sécurisé du service BBB.

Sur SUMMER :
- le stockage sécurisé des données, en particulier pour les enregistrements des webconférences.

Sur Virtualdata :
- une réplication de la PLM pour une éventuelle défaillance de Gricad-Nova ;
- l'architecture de webconférence BBB ;
- un bac à sable pour tester des projets ;
- un potentiel de croissance des ressources de calcul pour *jupyterhub*.

Sur RENATER :
- les éléments critiques pour l'authentification sur la PLM : le fournisseur de services de la fédération d'identité, notre brique d'authentification oauth2 et la convergence d'identité.

4.2 Recherche de la topologie réseau adéquate.

Nous avions pensé reproduire la topologie des anciens sites PLM avec un réseau interne interconnecté aux autres sites par le *L3VPN* de RENATER. Ce raccordement *L3VPN* est apparu aux différents *clouds* comme une demande singulière. Il est même incompatible avec le réseau du datacentre grenoblois, basé sur le SdN Cisco ACI. En appliquant une configuration globale sur les flux réseaux par projet, la Cisco ACI a besoin de partager un même plan d'adressage pour tous les flux. Faire transiter le réseau d'interconnexion de la PLM impose une compatibilité de son plan d'adressage. Une telle contrainte généralisée sur les différents sites, deviendrait ingérable avec un adressage privé.

Comme alternative au *L3VPN*, nous avons envisagé d'utiliser Wireguard. Basé sur *UDP*, il n'est pas communément ouvert en entrée voire en sortie des différents campus, nous avons donc dû demander la modification du filtrage. Par mesure de sécurité *anti-spoofing*, OpenStack limite tout

trafic sortant d'une *VM* avec une adresse IP source non attribuée à la *VM*. Nous avons réussi à créer une passerelle connectée aux autres sites PLM avec une interface dans un réseau *interco* local d'un projet Gricad-Nova. Nous étions alors obligés d'autoriser l'ensemble des réseaux privés distants.

Par ailleurs, le partage de ce réseau d'interconnexion entre projets OpenStack a fonctionné mais la spécification d'une adresse fixe ou de groupes de sécurité particuliers à une interface est réservée au projet définissant le réseau et ne peut être réalisée dans le projet définissant la *VM*. Une modification de la sécurité le permettrait mais finalement nous ne l'avons pas demandé.

Les choix d'implémentation réseau diffèrent entre les deux *cloud* : OpenVSwitch pour Gricad-Nova et LinuxBridge pour Virtualdata. À Grenoble, les paquets qui transitent entre deux serveurs physiques sont encapsulés dans un VxLAN. Le plan d'adressage de chaque projet est indépendant de celui du datacentre. À Paris-Saclay, il est possible de faire correspondre VLAN du site et réseau interne au projet. Cela aurait pu servir à l'interconnexion *L3VPN* si l'idée n'avait pas été abandonnée.

En souhaitant déléguer la gestion de *VM* sans contrôle direct de la PLMteam, le niveau de confiance du réseau privé est somme toute équivalent à celui du réseau public. Or, connecter les services internes PLM (sauvegardes, configurations, authentifications, nommages, journaux) via un réseau interne n'est pas une obligation. Surtout si la connexion se fait à l'initiative de clients disposant d'une adresse publique. Les flux sont sécurisés indépendamment de la topologie réseau. Ainsi, chaque projet devient autonome avec un bastion pour fournir aux *VM* locales les services internes comme le regroupement des journaux d'événements, le relai éventuel de messages système, la métrologie ou la supervision mais aussi l'accès *ssh* administrateur depuis l'extérieur.

Pour gérer la plage d'adresses publiques, Gricad-Nova utilise le mécanisme des IP flottantes (FIP). Elles sont allouées puis associées aux instances. Chaque instance a une interface dans le réseau du projet doté d'un cache DNS qui associe dynamiquement le nom d'instance et l'adresse privée. Chaque création de FIP produit la déclaration d'un nom DNS dans le domaine technique u-ga.fr. Ce nom est utilisé pour les demandes d'ouverture de flux en entrée de campus. Un changement d'adresse IP, lors de la recréation d'une FIP par exemple, sera pris en compte automatiquement sur les pare-feux en quelques minutes. Le nom public d'un service est déclaré dans le DNS de la PLM comme un CNAME du nom technique.

Virtualdata propose directement un adressage public aux instances avec une plage d'adresses dédiée à MATHRICE. Pour éviter de bloquer la création d'une instance par manque d'adresse *IP*, il est possible d'utiliser les *FIP* pour découpler la création d'instance et l'allocation d'adresse. Les noms des services sont déclarés dans les zones de la PLM, sans déclaration *DNS* inverse.

Lors de son installation, OKD utilise des *VIP*s pour basculer les IP de service kubernetes (API) et d'applications (APPS) entre les nœuds. Un *bug* sur la configuration du *arp-responder* OpenStack empêche le fonctionnement du protocole *VRRP*. L'installation d'*OKD* se termine en déplaçant les *IP*s à la main entre les nœuds, mais nous ne pouvons l'utiliser en production sans activer cette option.

Sur le *IaaS* hautement disponible de RENATER, nous souhaitions utiliser Contour[19] comme *reverse-proxy* et *load-balancer* vers des services tournant sur les *clouds* Virtualdata et Gricad-Nova.

Contour est un *ingress*[20] Kubernetes.
Envoy est un *reverse-proxy* cloud-natif, programmable et haute performance.
Contour monitore des « *Custom Resources Definitions* »[21] pour les traduire en configuration Envoy.

19 https://github.com/projectcontour/contour

20 https://kubernetes.io/docs/concepts/services-networking/ingress

21 https://kubernetes.io/docs/tasks/extend-kubernetes/custom-resources/custom-resource-definitions

Un *bug* sur la génération des clusters Envoy par Contour rend inopérant les *healthchecks* HTTP vers des services de type ExternalName[22].
Il est donc pour l'heure impossible d'utiliser Contour pour ce cas d'usage.

D'autres control-plane existent pour Envoy, par exemple, Enroute[23] qui est même utilisable en dehors de Kubernetes.

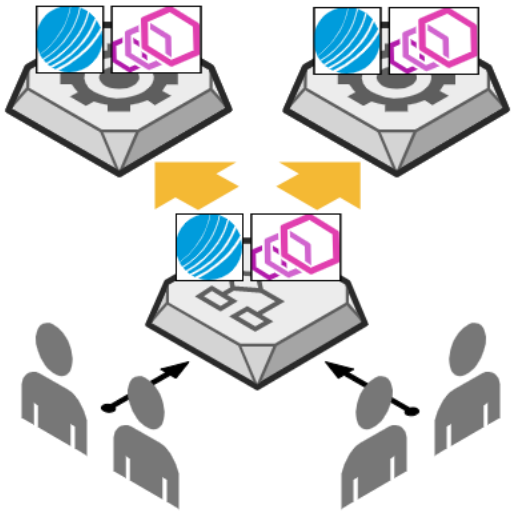

Figure 3 – Utilisation de Contour au centre pour adresser les services sur les 2 clouds

4.3 Évolutions de l'administration de la PLM

4.3.1 Consolider la démarche DevOps

Cette nouvelle architecture est l'occasion de consolider la démarche DevOps présentée sous la forme d'un poster aux JRES-2015 [2] Pour notre architecture historique, nous avions écrit des scripts de création de *VM* fondés sur virt-install à destination de nos hyperviseurs libvirt. La configuration et la maintenance des *VM* se faisait à l'aide de Puppet menant à une forme monolithique de l'ensemble de l'infrastructure. Celle-ci est antinomique du grain nécessaire à l'intégration de nouveaux administrateurs et au découpage en micro-services. Nous pouvons bénéficier de l'API d'OpenStack pour reconsidérer le mode d'administration de la PLM et avoir une réelle démarche DevOps. Chaque service est géré comme un conteneur avec sa création, sa configuration et sa destruction, défini de façon autonome du reste de la PLM. L'objet est ici d'intégrer des gestionnaires extérieurs de *VM* sans avoir de droits sur l'infrastructure elle-même.

4.3.2 Infrastructure as Code (IaC)

Nous avons choisi de découper en projets Openstack les différentes sphères de gestion : la PLMteam gère l'ensemble de l'infrastructure et les machines de service propres à la gestion de la PLM sont dans un projet mathrice-plm. Les workers BBB sont dans un projet mathrice-webconf. Un projet mathrice-vm-hosting accueillera les *VM* gérées par des mathriciens, ...

Les *VM* et leurs services sont instanciées, dans un mode d'Infrastructure as Code représenté par la Figure 4. Un fichier Openstack RC permet l'accès à Openstack et terraform va permettre de créer les objets nécessaires.

22 https://github.com/projectcontour/contour/blob/main/design/external-names.md

23 https://getenroute.io

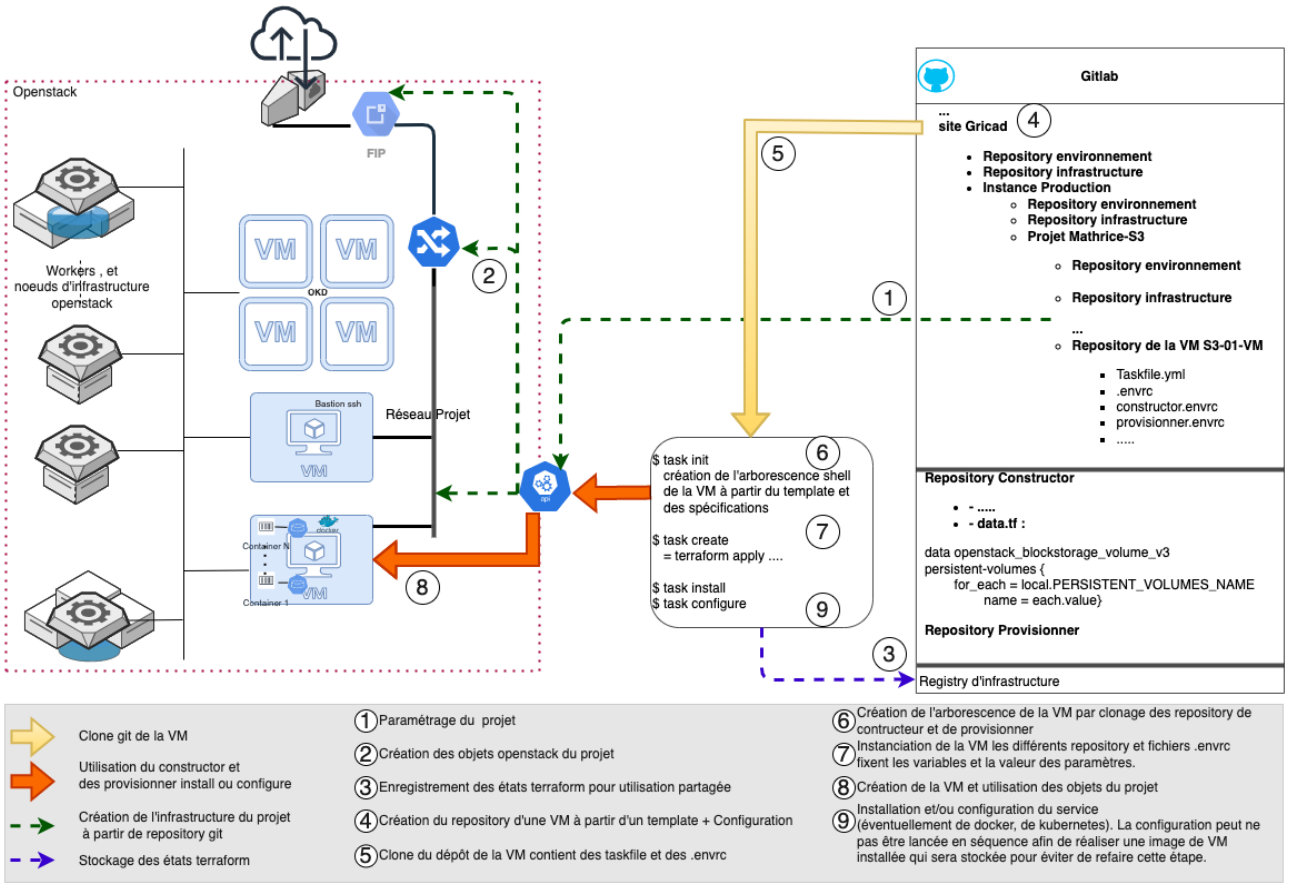

Figure 4 – gestion d'un projet (IaC)

Le constructeur : il contient le code mutualisé permettant de fabriquer un objet. Nous en avons défini pour deux types d'objet : image et *VM*.

Pour une image, nous utilisons Packer[24] comme outil central. Nous avons un constructeur particulier pour les images de base. Packer utilise Qemu[25] pour créer l'image d'une *VM* locale à partir du disque ISO de l'installeur de la distribution souhaitée. Qemu permet l'envoi indispensable de codes de touche au cours de l'installation.
Le constructeur d'image avec le pilote Openstack permet de spécialiser des images à partir d'une image parente. Ces images sont stockées sur le registre du *cloud* ou sur un stockage S3 indépendant.

Pour une *VM*, nous utilisons Terraform[26] pour gérer les ressources du *cloud*. Le premier constructeur OpenStack/Systemd va spécifiquement réaliser l'installation d'une image, utilisant l'init systemd pour permettre le montage[27] des volumes persistants du SAN CEPH et son formatage avec un système de fichiers particulier comme ZFS. Le but de ce constructeur est d'obtenir un système fonctionnel avec la distribution souhaitée et les services de base comme le client NTP, le relais SMTP ou l'envoi des logs et des métriques.

Le provisionneur : il contient le code mutualisé permettant l'installation et la configuration de services. Découpés en modules, seuls les éléments sélectionnés par paramètres seront utilisés.

24 https://github.com/hashicorp/packer

25 https://github.com/qemu/qemu

26 https://github.com/hashicorp/terraform

27 https://www.freedesktop.org/software/systemd/man/systemd.mount.html

Les provisionneurs que nous avons développés pour l'instant font essentiellement de l'instanciation de fichiers de configuration ou de l'exécution d'installeurs. Ainsi nous utilisons un binaire go « Sup[28] » qui, à partir de fichiers Yaml, permet de déployer des fichiers et éventuellement de les exécuter. Ansible fait cela aussi. Nous pourrions capitaliser sur notre expérience en utilisant Puppet (l'interdépendance entre systèmes étant moins importante et plus facilement gérable ici dans la mesure où Puppet s'occuperait seulement de la configuration des services). L'utilisation d'un des trois outils est laissé à la discrétion du développeur.
Les 2 étapes peuvent être jouées séparément ou enchaînées pour créer une *VM* entièrement configurée depuis sa création jusqu'à sa mise en service.

L'environnement : il s'agit d'un ensemble de variables d'environnement qui peuvent être partagées à différents niveaux : global, site *cloud*, instance cloud, projet ou fonctionnalité. Un mécanisme d'héritage de valeur permet de surcharger ces variables entre les niveaux.

4.3.3 L'arborescence git traduit l'organisation

Un dépôt template contient la définition des tâches Taskfile pour construire l'arborescence de fichiers qui regroupe environnement, constructeur et provisionneur. Les tâches ainsi regroupées permettent de construire la *VM* et de provisionner les services à fournir par celle-ci. Ce dépôt est cloné pour chaque *VM* en le paramétrant pour choisir et configurer constructeur et provisionneur selon la fonction de la *VM*. L'arborescence git calque l'organisation de l'infrastructure :

- clouds (cloud.math.cnrs.fr,...)
  - sites (Gricad-Nova, Virtualdata, …)
    - instances (production, test, ...)
      - projets (mathrice-plm, mathrice-webconf, mathrice-s3, ...)
        - webconf01.mathrice-webconf.gricad.cloud.math.cnrs.fr
        - bastion.mathrice-plm.gricad.cloud.math.cnrs.fr

Un dépôt git nommé 'environnement' fournit les variables d'environnement à chaque niveau. Par exemple au niveau site, nous aurons des variables représentant les serveurs NTP, DNS, etc. Au niveau instances, nous aurons une variable contenant l'adresse de l'instance d'Openstack dans lequel sont déployées les *VM*.

Dans chaque projet un dépôt nommé « infrastructure » permet de définir par Terraform l'ensemble des ressources du projet utiles pour les *VM* : FIP, volume persistant, groupe de sécurité ...

L'état terraform d'un projet regroupe les informations sur les objets d'une configuration. Il est partagé entre les membres de la PLMteam via la *registry* d'infrastructure de GITLAB[29]. Ainsi il est possible d'intervenir sur les configurations Terraform des autres. Nous prévoyons dans l'avenir d'utiliser la cli *OpenStack-Inventory*[30] pour gérer les ressources non créées par Terraform. Pour cela nous devrons gérer la création de métadonnées de déploiement.

Ainsi, gitlab devient le point central de configuration de l'infrastructure et des services. Le gain en granularité va permettre à des Mathriciens de contribuer plus facilement à l'administration de services sur la PLM sans une connaissance exhaustive de la configuration et tout en ayant une sécurité et une délégation de droits adéquates.

5 Bilan

Lors du passage au *cloud*, nous avons perdu la spécialisation du matériel.

28 https://pressly.github.io/sup/

29 https://docs.gitlab.com/ee/user/packages/infrastructure_registry/

30 https://pypi.org/project/openstacksdk/

C’est la contrepartie de la banalisation sur des infrastructures mutualisées. Par exemple, nous avons équipé de *GPU* le site *baremetal* de Bordeaux car les nœuds *cloud* n’en proposaient pas. La multiplicité des authentifications et des gestionnaires de tickets ne simplifie pas la gestion quotidienne. Le changement d’architecture a nécessité un apprentissage long et la compréhension des mécanismes internes. Nous avons dû revoir notre manière de fonctionner.

Nous restons dépendant des arrêts, des maintenances et des problèmes des hébergeurs mais cela ne change pas foncièrement de la situation initiale car c’est inhérent au fait de travailler sur une architecture distante et multi-site. Nous nous reposons maintenant sur les équipes de sites.

Nous avons beaucoup appris et évolué dans notre métier d’ASR en ajoutant la dimension DevOps. Nous ne gérons plus des achats matériels, de la climatisation ou de l’électricité. Nous avons augmenté l’efficacité, l’autonomie et l’indépendance des administrateurs. Nous pouvons facilement mettre à disposition des machines virtuelles sans interférer avec la sécurité de notre architecture.

Nous avons obtenu de la généricité et de l’élasticité. La création de nouvelles machines est immédiate ou presque par clonage d’un dépôt git. Nous pouvons instantanément mettre des moyens importants sur une maquette ou face à un besoin. Nous pouvons déployer une infrastructure de développement ou de test en parallèle de la production. Le déploiement de *PaaS* OKD[5] sur cette infrastructure nous a montré combien nous avions gagné en capacité et en facilité d’installation. Nous avons rendu les codes et documentations disponibles afin de partager notre production.

Nous avons rencontré principalement deux écueils : reproduire notre ancienne architecture réseau sur l’*IaaS* existante, ce qui nous a contraint à revoir notre infrastructure ainsi que les différences d’implémentations de ces *IaaS* auxquelles nous devons nous adapter.

L’écoresponsabilité de l’infrastructure matérielle est désormais l’affaire des sites mais notre modèle de déploiement prend en compte l’adaptation de l’usage des ressources aux besoins.

Il nous reste à éprouver l’accueil de nouveaux administrateurs et les solutions pour organiser et profiter de l’interchangeabilité entre les 2 sites. Nous pensons avoir atteint un niveau d’abstraction et de généricité qui facilitera l’expansion et la maintenabilité de la plateforme.

Bibliographie

[1] L. Azema, J. Charbonnel, D. Delavennat, L. Facq, D. Ferney, S. Layrisse, A. Shih, R. Théron. Mathrice, une communauté, une organisation, un réseau, des projets pour les mathématiques. Dans actes du congrès JRES2013, Montpellier, décembre 2013 : https://2013.jres.org/archives/32/paper32_article.pdf

[2] L. Azema, D. Delavennat, P. Depouilly, L. Facq, S. Layrisse. Architecture DevOps de PLM MATHRICE. Dans actes du congrès JRES2015, Montpellier, décembre 2015 : https://conf-ng.jres.org/2015/planning.html#article_63

[3] C. Lenne, G. Feltin. L’UMS GRICAD : un modèle organisationnel original à l’échelle d’un site. Dans actes du congrès JRES2017, Nantes, novembre 2017 : https://2017.jres.org/fr/presentation?id=27

[4] H. Massias, S. Layrisse, M. Khabzaoui, D. Delavennat, A. Shih. Big Blue Button une réponse à l’explosion des besoins en visioconférence. Dans actes du congrès JRES2022, Marseille, Mai 2022 : https://conf-ng.jres.org/2021/planning.html#article_127

[5] P. Depouilly, D. Ferney. Déploiement d’une solution complète de type PaaS dans un environnement de Cloud ou de laboratoire, destinée à une large communauté. Dans actes du congrès JRES2022, Marseille, Mai 2022 : https://conf-ng.jres.org/2021/planning.html#article_122