

MATHRICE - Retour d'expérience DevOps

Damien Ferney, David Delavennat

PLM-Team MATHRICE

Jeudi 13 Avril 2017



MATHRICE

Une aventure humaine

Contexte

Évolutions inspirées du mouvement DevOps

des choix

Vers une seule équipe?

Infrastructure

PLM 2.0

Bilans



- ▶ Profil
 - ▶ Développeur & ASR
- ▶ Privé
 - ▶ Intégration : Industriels, Automobile, Aéronautique, Banques, Militaire
- ▶ CNRS
 - ▶ Périmètres : INSMI, INP, INSB
 - ▶ Projets:
 - ▶ Services numériques/Portail des mathématiques (MATHRICE)
 - ▶ Clusteurs HPC et de stockage, virtuels (Mésocentre PHYMATH)



- ▶ Profil
 - ▶ ASR
- ▶ Laboratoire de Mathématiques Blaise Pascal de Clermont-Ferrand (LMBP)
 - ▶ Périmètres : UCA – INSMI
 - ▶ Projets:
 - ▶ PLM – infrastructure MATHRICE.



MATHRICE



"Tout seul on va plus vite, ensemble on va plus loin."

— proverbe africain —

MATHRICE I

- ▶ Réseau de métier des ASR des laboratoires de Mathématiques
- ▶ Chercheurs Maths = [petites équipes, disséminées, mobiles, mouvantes]

=>

- ▶ Groupement de service du CNRS : GDS2754 (INSMI) :
"Faciliter les échanges d'expériences et d'expertises entre informaticiens des laboratoires de mathématiques, contribuer au maintien de leurs connaissances et savoir-faire. Expérimenter et mettre en œuvre des services numériques en appui à la recherche en mathématique, accompagner les chercheurs dans l'utilisation de ces services"



MATHRICE II

- ▶ Besoin et volonté de fournir des ressources numériques mutualisées, collaboratives, accessibles de manière nomade
- ▶ Basé sur l'idée :
 - ▶ ce que je fais dans mon laboratoire peut servir à la communauté
 - ▶ ce que je fais pour la communauté peut servir dans mon labo
- ▶ PLM-TEAM : <http://www.mathrice.fr/spip.php?article204> équipe d'ASR & PLM-DEV
- ▶ $\simeq 30$ personnes qui de près ou de loin participent ou ont participé.
- ▶ un vivier important : tous les admins et dev du réseau de métier.



Avant

- ▶ PLM 1.0 [Depouilly et al., 2005]
 - ▶ Identification LDAP (correspondants PLM)
 - ▶ Annuaire de la communauté (correspondants annuaires)
 - ▶ Accès distant et nomade aux services Numériques :
 - ▶ A travers des développements horde : proxy d'accès à des ressources documentaires, vpn, webconférence, ...
 - ▶ Au travers de comptes pour chaque services : hébergement web, listes de diffusion à la demande, dépôts subversion...
- ▶ PLM 1.5 :
 - ▶ Identification CAS => quelques applications en profitent

MAIS



Un Constat

- ▶ Sortir du carcan de horde et des développements php (difficulté de migration, d'évolution, monobloc).
- ▶ Difficultés d'intégrer et d'homogénéiser les différentes identités numériques d'un utilisateur.
- ▶ Possibilité pour un utilisateur d'interagir hors du domaine des maths.
- ▶ Répondre à une demande de d'interface unique pour regrouper l'ensemble des services, homegenéisation , authentification unique
- ▶ Besoin de rationaliser l'infrastructure, d'urbaniser le système pour répondre à la demande croissante et gagner en disponibilité tout en limitant l'impact sur les ASR.
- ▶ Perte de la maîtrise de la globalité : besoin d'un cadre, pour garder l'autonomie de développement des services.
- ▶ ASR essentiellement mais qui ont besoin de développements et veulent garder la dynamique de déploiement rapide par des équipes restreintes sur env. systèmes de leur choix.



Évolutions inspirées du mouvement DevOps

Des Choix

- ▶ De ce constat :
 - ▶ Choix d'une démarche pragmatique liée à la dynamique du groupe qui associe nos métiers ASR et développement
 - ▶ Services isolés dans une VM => explosion de VM => outil de gestion de configuration (puppet) + historisation (git).
 - ▶ Micro-services évoluent plus facilement & indépendamment.
 - ▶ Web Services associés aux services = interfaces d'interconnexion aux autres services ou aux utilisateurs.
 - ▶ Protocole d'authentification unique, fiable et simple permettant de reconnaître un utilisateur de la PLM au travers de ses multiples identités numériques,



2 équipes dont l'intersection grandit

- ▶ De la superposition des personnes agissant aussi bien au niveau développement (PLM-Dev) qu'au niveau opérationnel (PLM-Team) a émergé une approche DevOps avec un profil par individu variant au cours du temps et une porosité importante entre les 2 mondes.
- ▶ Transfert de compétence développeurs vers les ASR, en retour les ASR sont plus à l'écoute des besoins dev et même certains ont migré
- ▶ L'infrastructure évolue pour permettre les développements et le développement améliore l'infrastructure. Un va et vient continu qui permet de progresser.
- ▶ Une organisation stigmergique : les outils et méthodes utilisés contiennent une notion de trace qui permet de pérenniser le savoir

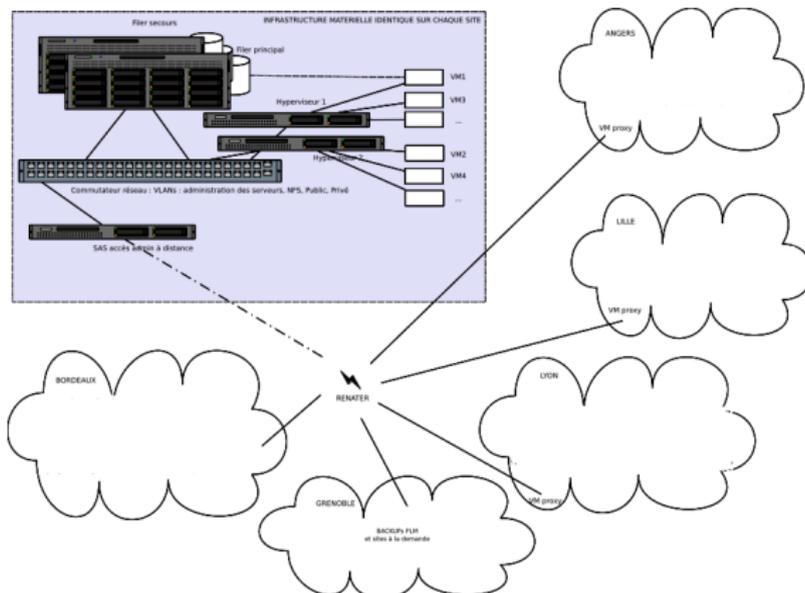


Infrastructure I

- ▶ Uniformisation et rationalisation de l'infrastructure (4 sites identiques)[Charbonnel, 2012]
 - ▶ sas ssh
 - ▶ Filer et Pra (ZFS + NFS)
 - ▶ Hyperviseurs en équilibrage de charge et interchangeables (KVM)
 - ▶ Un filtrage réseau à plusieurs niveaux.(campus, site(vlan ou bridge), service)
- ▶ Automatisation de la création de VM (filer et hyperviseurs)



Infrastructure II



Infrastructure III

- ▶ configuration depuis un serveur puppet (filers et VM). Une VM est décrite complètement
 - ▶ En fonction de son nom (IP)
 - ▶ Selon son rôle (/profils), cela permet d'ajouter une couche d'abstraction entre un noeud et les modules qu'on lui affecte. Chaque noeud inclut un seul rôle et chaque rôle appelle la notion de profils.
 - ▶ (en cours) Configuration dirigée par la demande : les Facts, afin de gérer des configurations dans une infrastructure Cloud (IAAS : OpenStack ou OpenNebula)
- ▶ Vers une démarche de développement continu (PDCA): VM
 - ▶ xx-dev
 - ▶ xx-pre
 - ▶ xx(-prod)
- ▶ Interconnexion des sites (L3VPN - RENATER)
- ▶ Redondance de services entre sites (L2VPN - RENATER)



fonctionnalités nouvelles

- ▶ Les éléments constitutifs de la PLM sont inter-opérables (via leur Web Service). Ils deviennent pilotables par d'autres services ou par des demandes d'utilisateur.
- ▶ Un module d'authentification par la fédération d'identité et autorisation après convergence d'identité (Shibboleth, CAS, LDAP) a été développé. L'accès à la Plateforme est facilité :
 - ▶ Activation autonome de comptes PLM
 - ▶ Permet aux membres de la communauté mathématiques d'associer leurs multiples identités numériques.
 - ▶ Permet l'invitation de chercheurs hors de notre périmètre
- ▶ L'utilisation du protocole Oauth2 permet aux applications de profiter de l'authentification unique et facilite l'intégration de nouveaux services.



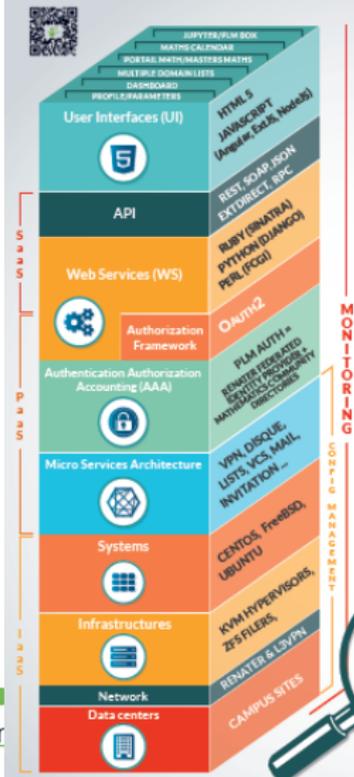
PLM 2.0

- ▶ PLM 2.0 [Azema et al., 2013]: Explosion des services
 - ▶ > 3000 Utilisateurs
 - ▶ > 200 Sites Web
 - ▶ > 600 Listes de diffusions multi-domaines
 - ▶ > 1000 dépôts (git – svn) avec une vision homogène
 - ▶ > 4To Bibliothèques Plmbox + espaces disques personnels (& pages web)
 - ▶ Indico : Sites Conférences + Sites séminaires (agenda des maths)
- ▶ Intégration fluide dans le portail des mathématiques (Bibliothèque numérique, Annuaires, Services numériques) autour d'un protocole d'authentification unique.

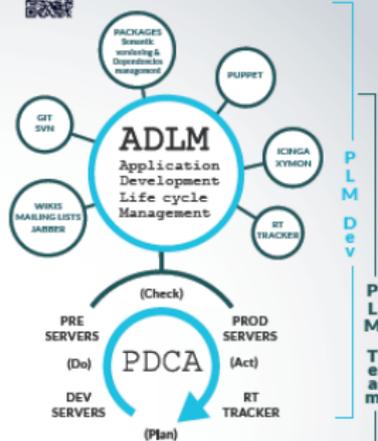


Synthèse

The MATHRICE Services Stack



DevOps Life Cycle & Project Management



DevOps stigmergic collaboration



Bilans

Bilan : technologies, produits

- ▶ Sencha (MVVM)
- ▶ Ruby Passenger
- ▶ Couchbase / N1QL
- ▶ RabbitMQ
- ▶ Shibboleth / CAS
- ▶ OpenLDAP
- ▶ Libvirt / KVM
- ▶ Puppet
- ▶ OAuth / OpenID Connect
- ▶ Sympa SOAP
- ▶ ShareLatex
- ▶ Seafile
- ▶ OpenNebula
- ▶ OpenStack
- ▶ Indico
- ▶ GIT / SVN
- ▶ ExtDirect
- ▶ ...



Bilan : technologies, produits

Il faut des protocoles de communications

- ▶ adaptés :
 - ▶ SOAP (puissant mais complexe)
 - ▶ REST (nomenclature d'appel non uniforme)
 - ▶ GET = / = POST, PUT, DELETE
 - ▶ ExtDirect (méconnu : JSON-RPC)
 - ▶ Bi-Directionnel sans nécessiter de WebSocket, Exceptions, Pooling
- ▶ et versionnés
 - ▶ minimiser les *barrières* lors du développement



Bilan : ASR

- ▶ Il faut maîtriser les concepts Objets (le scripting shell ne suffit pas)
 - ▶ Classe, Instance, Héritage simple, Héritage multiple (mixin)
- ▶ Il faut moderniser les infrastructures pour les rendre agiles (CLOUD)
 - ▶ Software-Defined-Infrastructure / Infrastructure pilotée par le besoin
 - ▶ Les outils tels que Puppet doivent être adaptés
 - ▶ traiter des classes de machines
 - ▶ permettre de déployer des instances de machines dont ils n'ont pas de pré-connaissance (modèle de sécurité inadapté)

Bilan : développeurs

- ▶ Il faut comprendre que
 - ▶ ce qui est faisable en mode développement
 - ▶ n'est pas forcément souhaitable en mode production
 - ▶ besoin de sécurité des infrastructures
 - ▶ environnement multi-utilisateur



Bilan : global

- ▶ Processus (peut être) très long à mettre en place
 - ▶ Le travail d'équipe est primordial
 - ▶ La maîtrise des technologies doit être partagée (formation interne)
 - ▶ Difficulté de concilier
 - ▶ la production
 - ▶ l'ingénierie nécessaire à l'évolution de l'infrastructure
- ▶ Très valorisant pour les personnels
 - ▶ Montée en compétence
 - ▶ fun ;)



Références

[Azema et al., 2013] Azema, L., Charbonnel, J., Delavennat, D., Facq, L., Ferney, D., Layrisse, S., Shih, A., and Théron, R. (2013).

Mathrice, une communauté, une organisation, un réseau, des projets pour les mathématiques.

[Charbonnel, 2012] Charbonnel, J. (2012).

Architecture de serveurs virtualisés pour la communauté mathématique.

[Depouilly et al., 2005] Depouilly, P., Grancher, G., and Marchand, J. (2005).

Mathrice, un réseau métier pour les mathématiques.



Questions ?