

REX DEVOPS – G2S

Innover



Construire



Ensemble



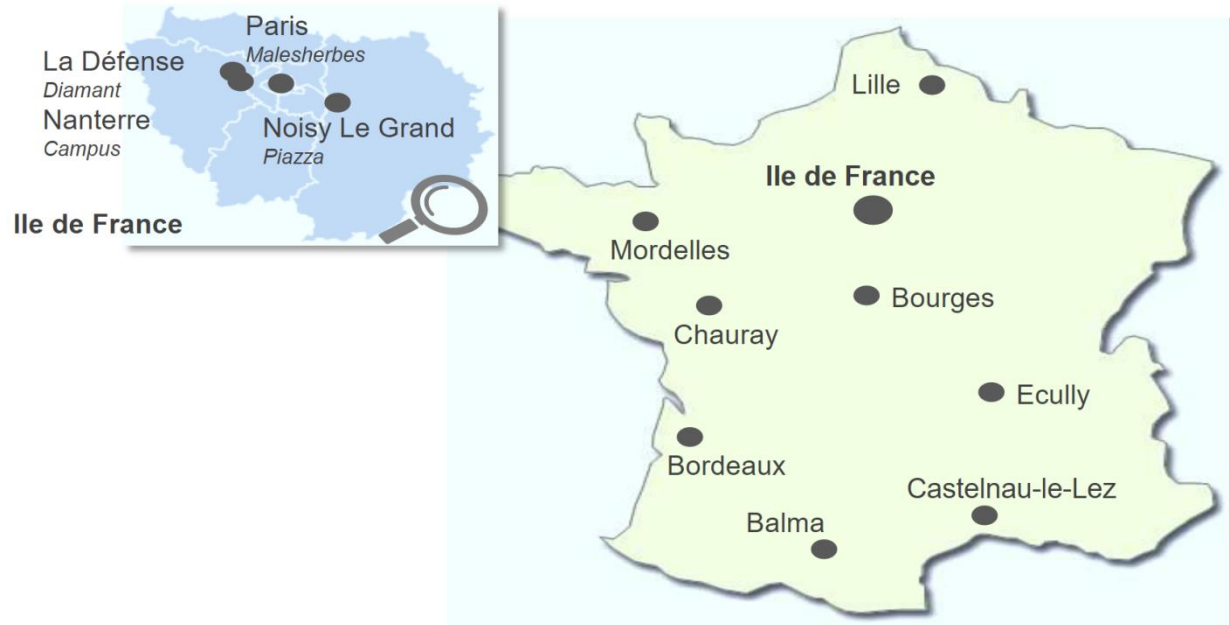
Emmanuel Lieurain – Directeur de Projets DevOps

emmanuel.lieurain@groupama.com



G2S : Groupama Supports et Services

- > GIE Groupama, 1500 collaborateurs
- > Au service des Caisses Régionales et des filiales du groupe



Chez G2S, un panorama applicatif conséquent

Le Système d'Information Communautaire représente l'ensemble des composants du SI qui sont partagés entre les Caisses Régionales, Groupama Assurances Mutuelles, Gan Assurances et GGVié, pour gérer l'activité cœur de métier du Groupe.



Des applications métier

destinées aux gestionnaires, commerciaux, clients ou partenaires pour souscrire un contrat, déclarer un sinistres, consulter un relevé, etc., sur l'ensemble des métiers du Groupe

Des applications support

pour gérer les fonctions comme la comptabilité, la fiscalité, la réassurance ou la gestion des risques,

Des applications de pilotage

pour piloter les activités communautaires du Groupe, s'appuyant sur une Architecture orientée Data avec une plateforme Data servant les besoins du décisionnel, les besoins de la Datascience, les cas d'usage ayant besoin de consolidation de données interdomaines (fraude, vision 360...)

Des applications socle

pour gérer les fonctions transverses au SI comme la Gestion Electronique de Document (GED), la dématérialisation (DEMAT), la gestion des corbeilles (COMPAS), l'édition (EDITO, etc.), la signature électronique (SED), la gestion des habilitations, etc.

Le DevOps chez G2S

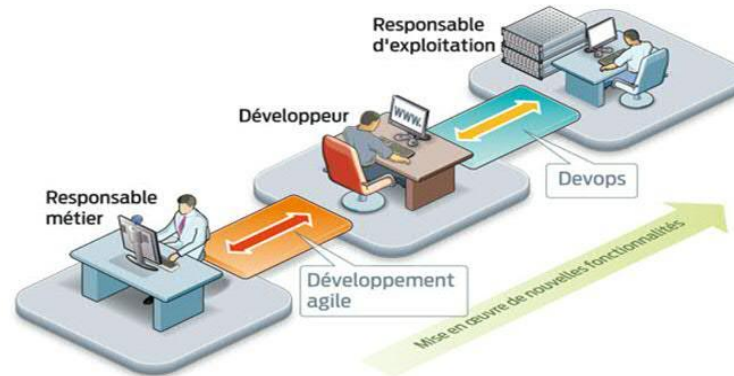
La philosophie DevOps

Le métier souhaite développer de nouvelles offres

- Pour être en avance sur la concurrence
- Pour expérimenter de nouvelles approches
- Pour faire face aux nouveaux acteurs

L'agile et DevOps ont pour objectif de transformer une idée de façon rapide **ET** sécurisée en une réalité client.

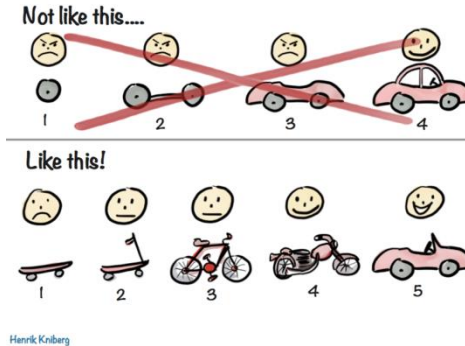
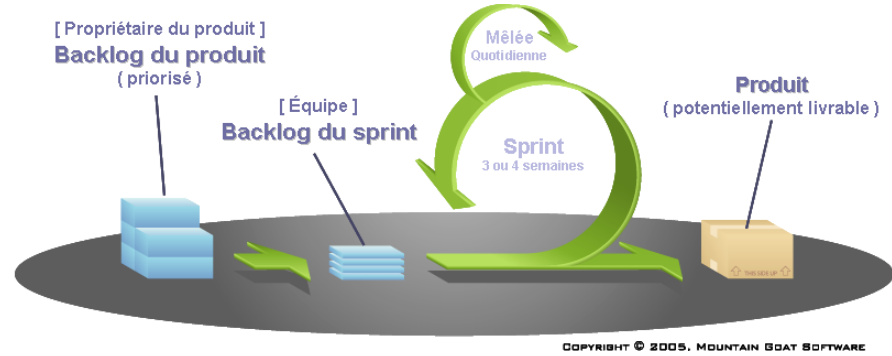
Alors que l'agilité amène au potentiellement livrable, DevOps permet que le potentiellement livrable soit **livré plus rapidement et de façon plus sécurisée.**



La philosophie DevOps

Méthode Agile aux Etudes → s'arrête aux environnements de Recette

Objectif du DEVOPS → être agile jusqu'en Prod

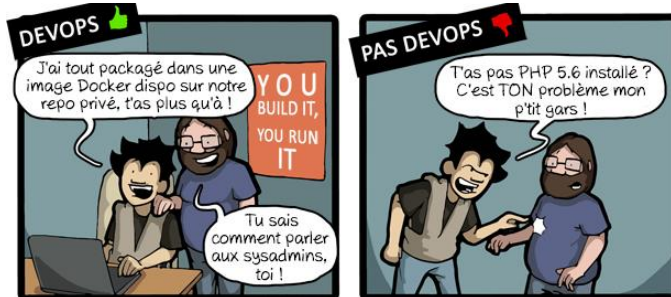


DevOps – naissance en Novembre 2017

Organisation



Infra As Code - DOCKER



Continuous Deployment



Objectifs

- ✓ Délivrer du projet avec la méthode DEVOPS
- ✓ Accompagner le projet du dev à la prod
- ✓ Etablir une relation privilégiée avec les équipes de dev
- ✓ Eviter le mode contributif
- ✓ Construction d'outils pour les équipes de développement : CI/CD Openshift (containers Beluga), CI/CD APIM, CI/CD GEMS.

Organisation

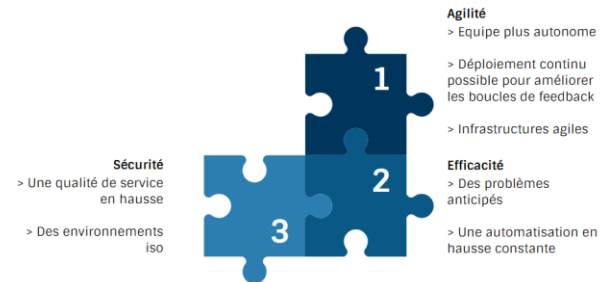
- ✓ Facilitateur de mise en œuvre des environnements
- ✓ Interface avec le reste de la DTPI
- ✓ Support aux équipes MOE (objectif autonomie)

Eligibilité des projets

- ✓ Projets en méthode agile (*dans la mesure du possible*)
- ✓ Techniquement éligibles
- ✓ Besoin de CI/CD ou de rapidité de mise en œuvre
- ✓ Equipes DEV curieuses des technologies utilisées

Impacts équipes projets

- ✓ De "nouveaux membres" dans l'équipe projet
- ✓ De nouveaux outils pour gagner en :

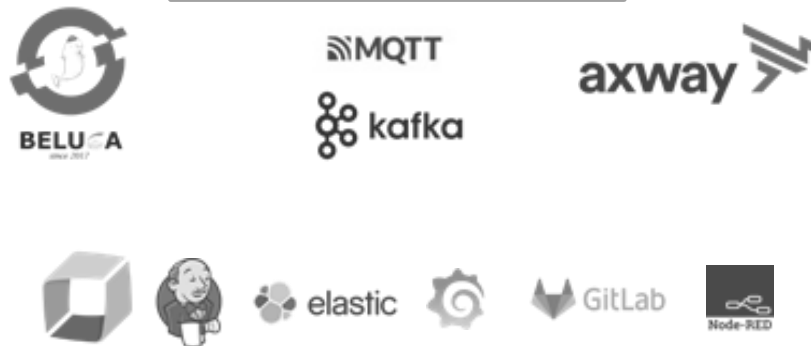


de l'équipe DevOps à Platforms Team

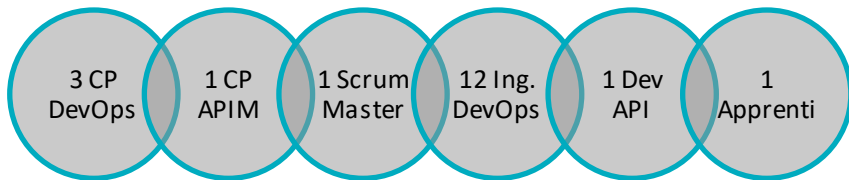
Une nouvelle équipe



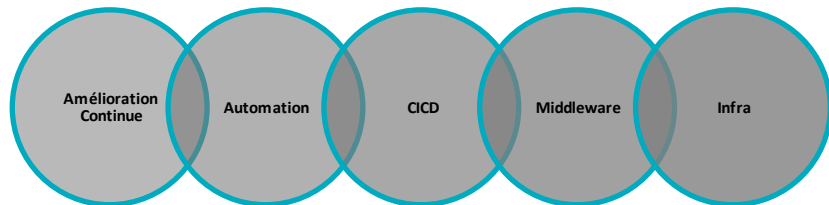
Des Technologies



Des Compétences



Un savoir faire



Les Changements avec le DevOps

1. Changement Technologique : Les containers

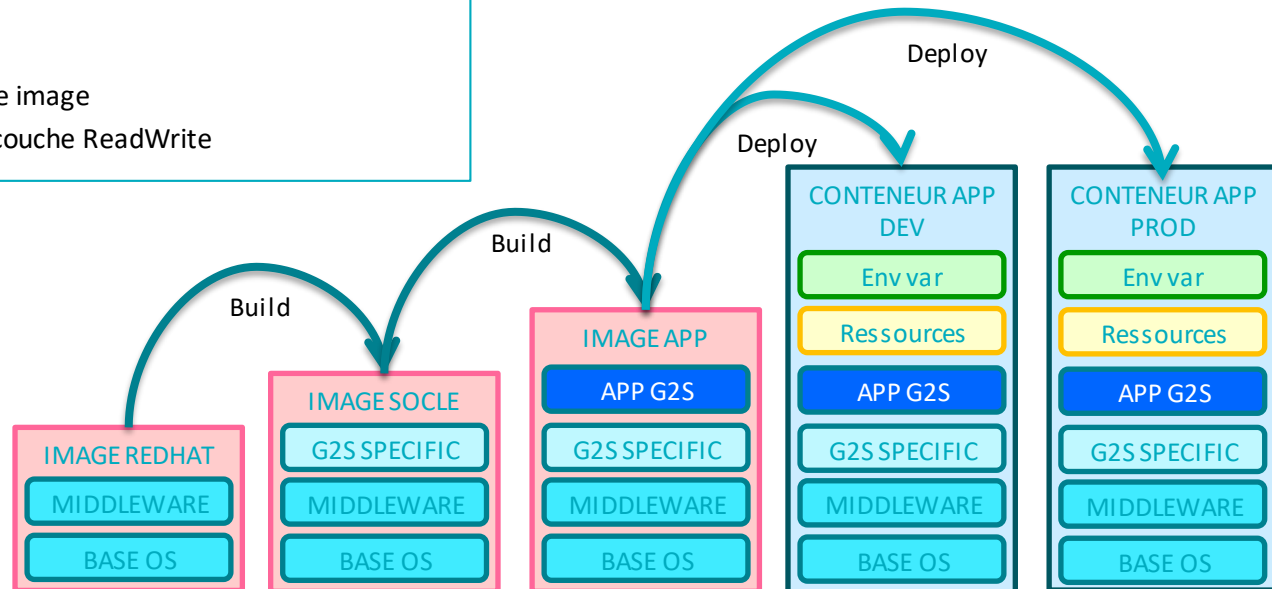
- > Projet technique de mise en œuvre d'une **plateforme d'orchestration de containers**
 - ✓ Code SA : « BELUGA »
 - ✓ Basé sur le produit RedHat Openshift
 - ✓ Fondé sur la technologie Docker et Kubernetes



1. Changement Technologique : La plateforme Beluga

Les images et les conteneurs

- ✓ Image
 - Assemblage de couche en lecture seul (Read)
 - Création à partir d'une image existante
- ✓ Conteneur
 - Instancie une image
 - Ajoute une couche ReadWrite



1. Changement Technologique: les gains

- > Changement de paradigme: nouvelle approche « de zéro »
- > Normalisation
- > Templates
- > Catalogue

- > Concentration des applications : toutes les applications sur les mêmes serveurs

- > Pipelines techniques
 - « Build » : Source, image docker, Registry
 - « Deploy » : Zero Downtime Deployment
 - Trigger depuis le pipeline CI des devs (Gitlab-CI)

1. Changement Technologique: les « bonus »

Equipe de Dev Autonome	Scalabilité horizontale automatique	Auto-remédiation automatique	Fiabilité des livraisons	Déploiement sans interruption de service
<ul style="list-style-type: none"> • Consulter logs, quel que soit les environnements • Déployer dev to < préprod sans demande • Consulter fichiers config sans demande 	<ul style="list-style-type: none"> • Pic de charge : nouveaux conteneurs démarrés • Creux : suppression des conteneurs 	<ul style="list-style-type: none"> • Relance automatique en cas de plantage • En cas de crash d'un connexe, le conteneur redémarre tant qu'il n'est pas disposans intervention humaine 	<ul style="list-style-type: none"> • Obligation de déploiement avant la production • Plus de fiabilité des livraisons (tests/qualité) • Image docker immuable : pas de modification « à la main » possible 	<ul style="list-style-type: none"> • Zero Downtime Deployment (ZDD) • EasyChange

2. Changement de méthodes

- > Agile : Scrum puis Kanban
- > Au plus proches des devs pour qu'ils deviennent autonomes:

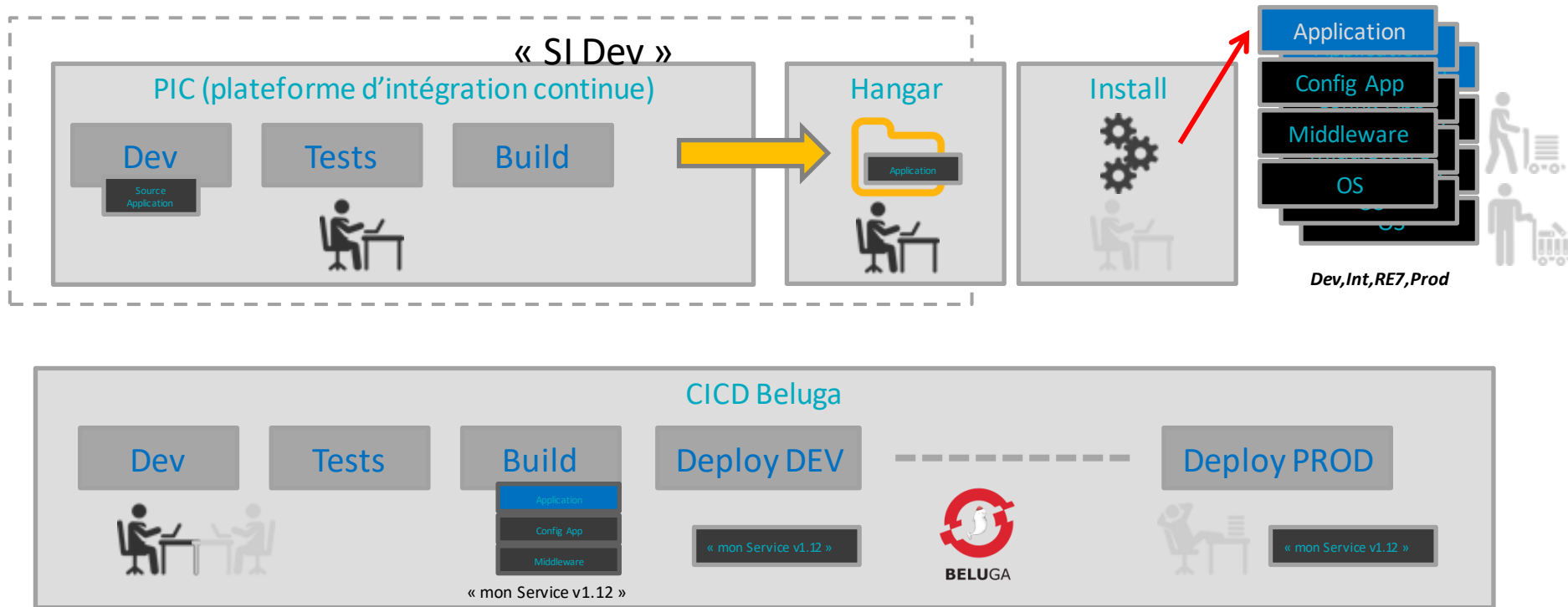
Dans leur repositories projet

Observabilité partagée : Traces (Dynatrace), Logs (Elastic), Metrics (Prometheus)

Collaboratif : Canaux Teams communs

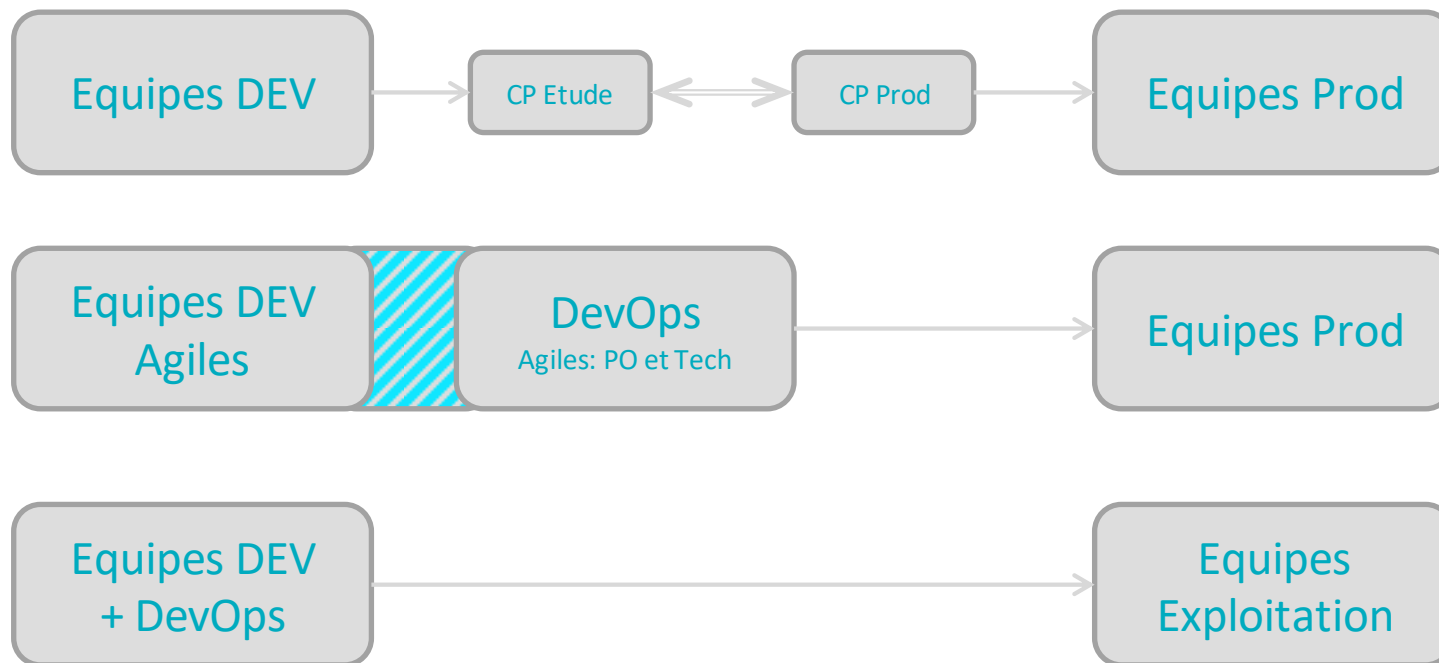
« *You build it, you ~~run~~ deploy it* »

2 bis. Changement de méthode d'installation



3. Changement de pilotage

- > **Chefs de projets DevOps:** à mi-chemin entre le CP Technique (Prod), et le CP Etude



2. Changement de livrables

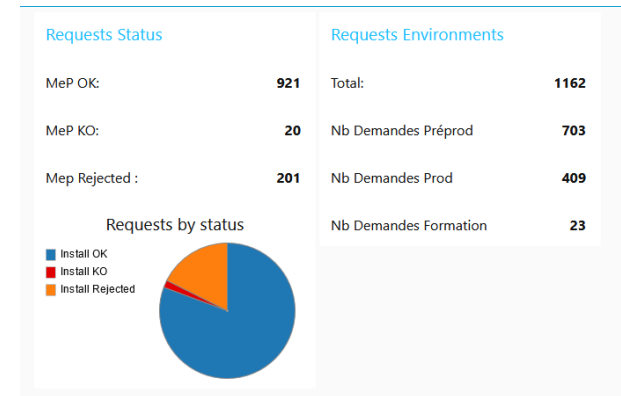
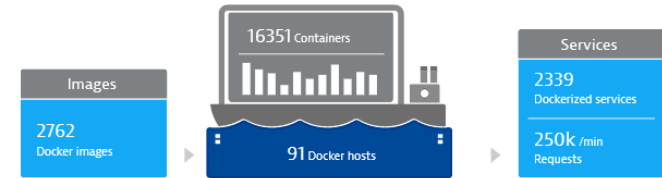
- > De l'accompagnement vers la Production (Exploitabilité)
- > Le livrable majeur de l'équipe: les pipelines (1 par composant)
- > Expertise (Paramétrage, Batch, Middleware)

- > De l'amélioration continue:
 - Vérifications automatiques (Conf Management, Process)
 - Livraison en autonomie
 - Notifications
 - Publication des mises en production (Documentation automatique)

Les Résultats

Les chiffres

- > 180 systèmes applicatifs conteneurisés
- > 2000 composants, entre 2 et 4 exemplaires en moyenne
- > Plus de 100 applications typées « critiques »
 - Souscription
 - Sinistre
 - Sites internet (www.groupama.fr, .com, www.gan.fr, espace client, ...)
- > 400+ installations en journée, directement demandées par les dev, depuis Janvier 2022, sans interruption de service, conforme à ITIL



Les limitations

- > Les Back-Offices majeurs sont livrés 2 fois par an (Avril et Octobre)
- > Certains projets, même agiles doivent suivre ce rythme
- > Nous devons faire cohabiter les deux mondes, sans créer un SI dans le SI
- > Nous sommes encore trop « équipe DevOps », après 5 ans d'évangélisation, nous y travaillons !

- > Certaines habitudes perdurent (choix des dates, dev presque-agiles)

- > On nous oppose les campagnes de Recette faites par les clients
« On ne va pas tester toutes les deux minutes ! »

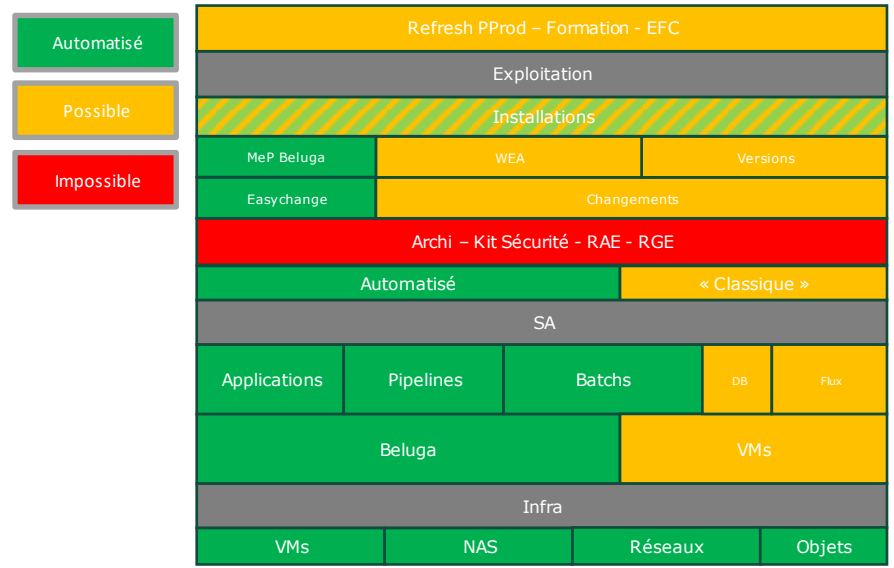
Ce n'est plus une problématique technique, mais bien culturelle !

La suite: Le pipeline est la clé pour faire changer la culture

Travail sur la fiabilisation de la livraison



Continuer l'automatisation du process



Merci !