



DevOps

Managing disruption

Guillaume Schott
Avril 2016

Agenda

01

DevOps ?

02

Culture & Organisation

03

Design & Architecture

04

Implémentation & outils

05

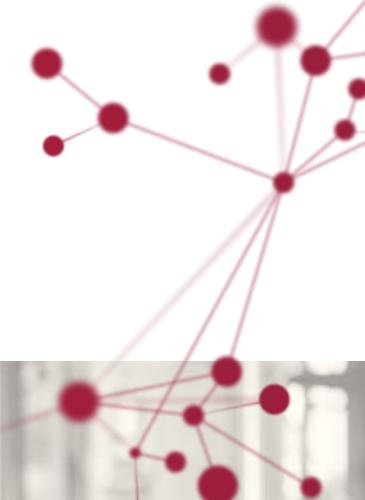
Pilotage & gouvernance

06

Retours d'expérience



DevOps, Une culture, une organisation, des méthodes et des outils



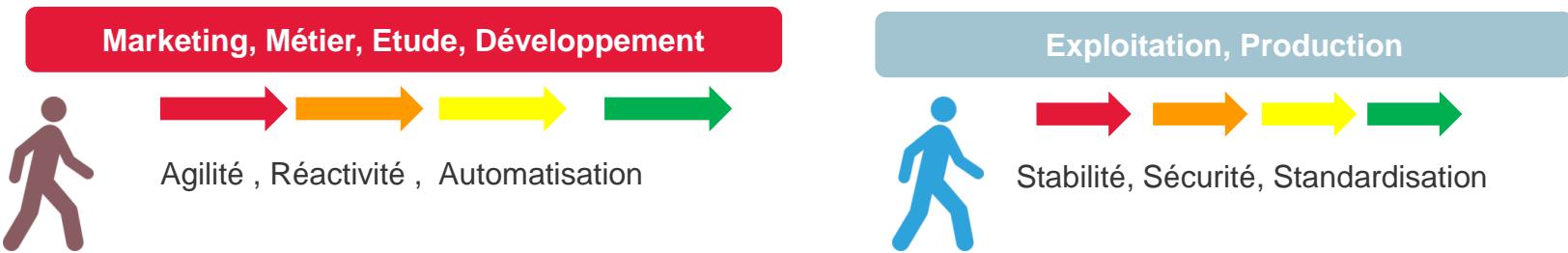
CGI

Experience the commitment®

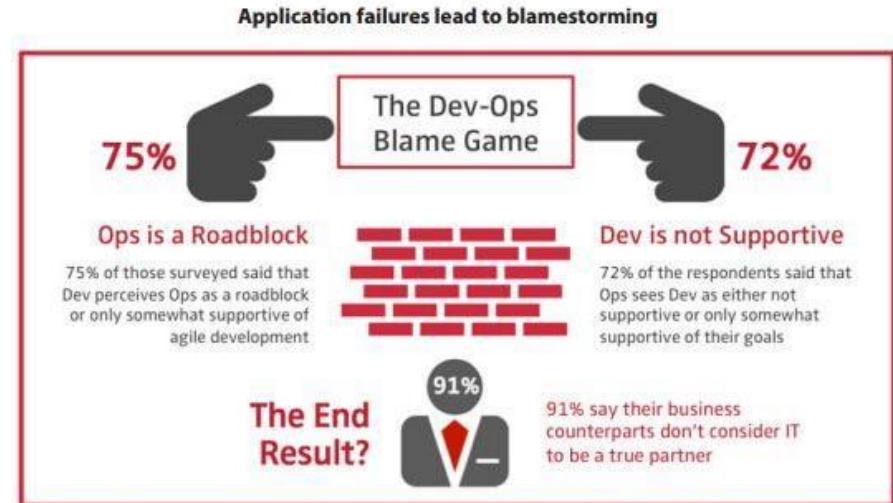
Constats

Des priorités divergentes et des organisations en silos

- Les méthodes agiles se limitent aux cycles de développement et n'adressent pas encore les problématiques des services de la production et de l'exploitation



- Le travail des uns s'arrête souvent où celui des autres commence
- Déploiement et mise en production sont des activités très consommatrices en charge et en délai pour les équipes opérationnelles



Enjeux

Des livraisons plus rapides et mieux maîtrisées

Des enjeux complémentaires pour des objectifs communs:

- Time to market !
- Qualité
- Sécurisation des déploiements en production
- Des équipes plus collaboratives



Satisfaction des services métiers

Augmentation de la confiance des autres services internes envers l'IT

Réalisation

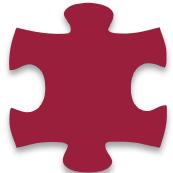
Production



DevOps

Vision et objectifs

Quatre chantiers majeurs à adresser au niveau de la stratégie IT

[Culture & Organisation]	[Design & Architecture]	[Implémentation & outillage]	[Pilotage & gouvernance]
			

Constituer des équipes pluri-disciplinaires du métier à l'exploitation

Découpler les composants du SI afin d'améliorer son agilité

Accélérer les livraisons par la généralisation de l'automatisation

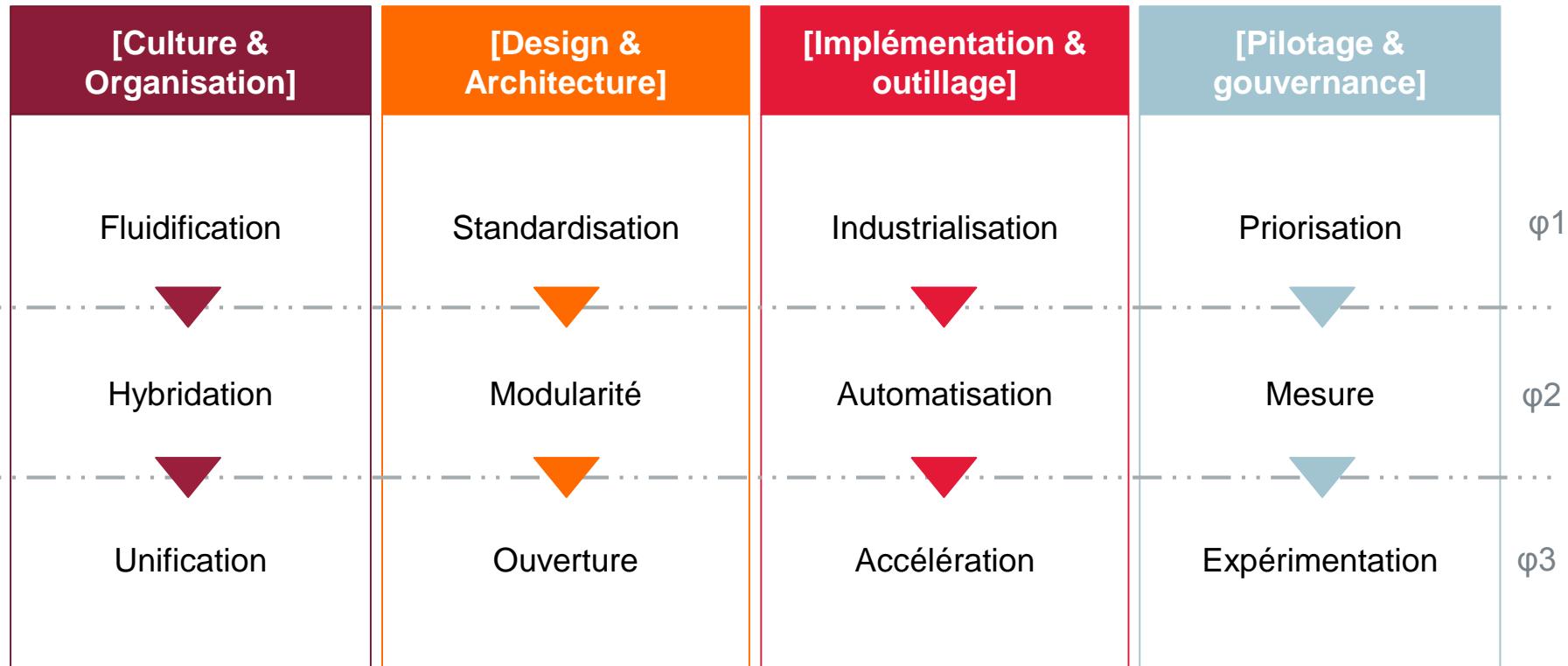
Monitorer et mettre en place un feedback rapide avec le métier

Ces chantiers doivent être contextualisés et peuvent être découplés en fonction des organisations.



DevOps

Modèle de maturité



L'avancement des chantiers n'est pas obligatoirement synchronisé.

Culture & Organisation



CGI

La force de l'engagement^{MD}

Une culture...

Une culture de collaboration et de communication

Comprendre les enjeux de chacun

- Se comprendre mutuellement (Dev, Ops, Métier)
 - Expliciter les besoins de chacun

Etablir une communication fréquente entre Dev et Ops

- Meetings réguliers Dev + Ops
 - Capitalisation partagée sur les problèmes de production (livraison, bugs, etc.)
 - Capitalisation partagée sur les évolutions de l'infrastructure

Objectifs

- Climat de confiance
 - Ecoute mutuelle
 - Coopération entre les équipes



Mettre en place une culture commune est un pré-requis indispensable et constitue la première étape.

Une organisation agile en mode intégré

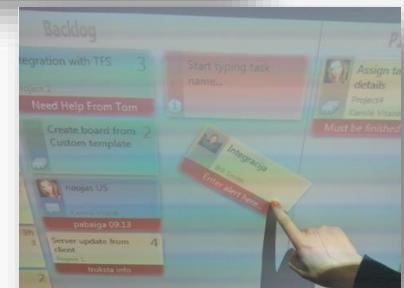
Organisation des équipes

- Equipes projet pluridisciplinaires (Dev, Ops, Métier)
- Nouveaux rôles.
- Suivi visuel des activités et partage d'information (Kanban, Visual Management, ...)
- Equipes distribuées ? Pas de murs ? Nouveaux outils !



Gestion des cycles de vie des projets en mode agile

- Adopter des cycles de mise en Production courts en mode agile
- Méthodologies Scrum, Safe, Scrumban, ...



Bénéfices

- Un statut général clair et connu de tous
- Une équipe en responsabilité sur l'ensemble du cycle de vie
- Une application testable et testée rapidement
- Plus de retours des utilisateurs, et plus rapidement
- Plus de métriques de production
- Des cycles courts centrés sur les priorités métiers
- Augmenter capacité d'innover



... et une organisation

Constituer des équipes pluridisciplinaires du métier à l'exploitation

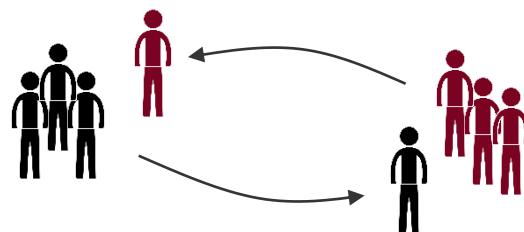
1. Fluidification



Conduite du changement : initialisation de la transformation culturelle et organisationnelle

Mise en place de l'agilité sur l'équipe étendue

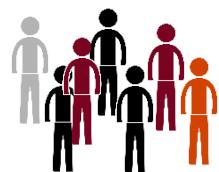
2. Hybridation



Accompagnement du changement
Équipes virtuelles responsable du build et de la gestion des incidents en production

Compétences infra. et dev. délocalisées

3. Unification



Accompagnement du changement
Equipe pluridisciplinaire et responsable « you build it, you run it »

Avec de nouvelles compétences, rôles et métiers

- **DevOps evangelist** : promouvoir les bénéfices de DevOps en identifiant et quantifiant les bénéfices business engendrés par la plus grande agilité des livraisons, accompagner les changements culturels.
- **Change Manager** : anime et pilote le changement culturel et organisationnel. Coach les acteurs clefs du projet dans la mise en œuvre du changement.
- **Release manager**: assure le management et la coordination du produit du développement à la mise en production.
- **Architecte Automatisation**: spécialiste de l'intégration et de l'automatisation, il analyse et implémente les stratégies de déploiement continu et garanti la stabilité des environnements.
- **Software developer/tester**: responsable non seulement de l'implémentation de nouvelles fonctionnalités, mais également de l'implémentation des tests unitaires, des scripts d'automatisation.
- **Ingénieur Sécurité**: intégré au développement le plus tôt possible pour tous les « non functional requirements »
- **Ingénieur DevOps ou FullStack**: mouton à 5 pattes impliqué tout au long du processus maîtrisant les process de développement et l'infrastructure.

Design & Architecture



CGI

La force de l'engagement^{MD}

Design & architecture

Découpler les composants du SI afin d'améliorer son agilité et sa tolérance aux pannes

1. Standardisation



- Composants et technologies
- Logs applicatifs pertinents et accessibles à tous
- Services mocking

2. Modularité



- Architecture Orientée Service
- **Micro-services autonomes et indépendants**
- Déploiements rapides: Canary release / feature toggle

3. Ouverture



- **API management**
- Event-driven architecture

Des stratégies de design

Des choix architecturaux à aligner avec la stratégie business et IT:

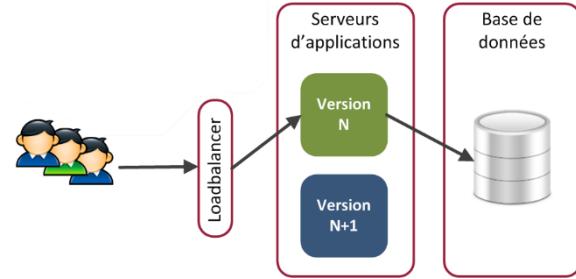
- Utilisation de conteneurs ?
- Utilisation de cloud public/privé ?
- Choix de plateforme ?
- Choix d'architecture, n-tiers, micro-services, API management, event-driven architecture, ...
- Haute disponibilité, clustering applicatif/système ?
- Scalabilité horizontale ?
- Exploitabilité de la solution !



Des stratégies de déploiement

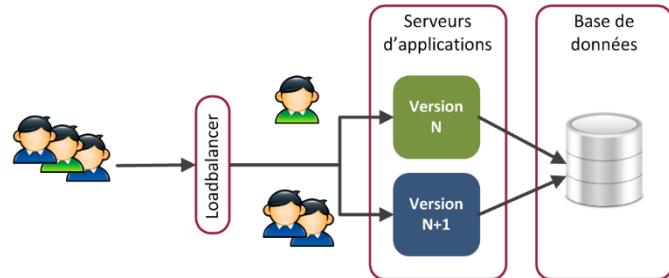
Blue/green deployment

- Permet à minima, d'effectuer des mises en production de manière transparente



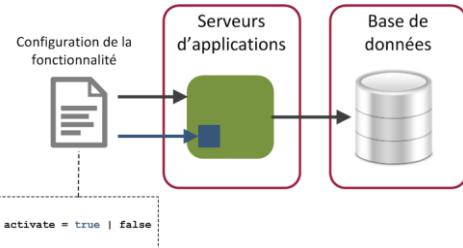
Canary release

- Tester de nouvelles fonctionnalités sur une petite population cible avant ouverture complète



Feature toggle

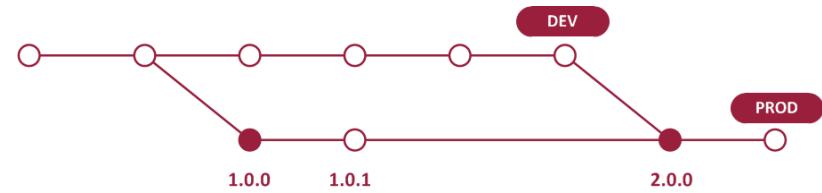
- Activation très rapide de nouvelles fonctionnalités en Production



Des stratégies de développement

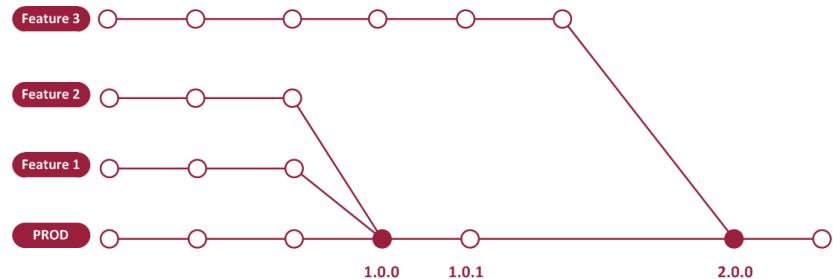
Simple branching

- Livrer rapidement un correctif en Production
- Faciliter des retours arrières



Feature branching

- Ne pas attendre la complétude de la release pour déployer une fonctionnalité



Implémentation & outillage



CGI

La force de l'engagement^{MD}

Implémentation & outillage

Accélérer les livraisons par la généralisation de l'automatisation

1. Industrialisation



- Généraliser l'intégration continue
- Automatiser les déploiements
- Automatiser les tests

2. Automatisation



- Infrastructure as code
- Provisioning automatique
- Tests d'acceptation automatisés

3. Accélération



- Cadencement des releases
- Déploiement continu

Les 5 concepts clés DevOps

Continuous Integration

- Plateforme d'intégration continue avec compilation des sources, tests unitaires, qualité du code, packaging, dépôt du package dans un référentiel

Continuous Testing

- Exécution automatique des tests d'intégration et fonctionnel via des déploiements à la volée sur différents environnements

Continuous Delivery

- Processus permettant de s'assurer qu'après chaque modification l'application est dans un état « **prêt à déployer en production** »
- CI + CT + provisionnement/gestion de configuration des environnements

Continuous Monitoring

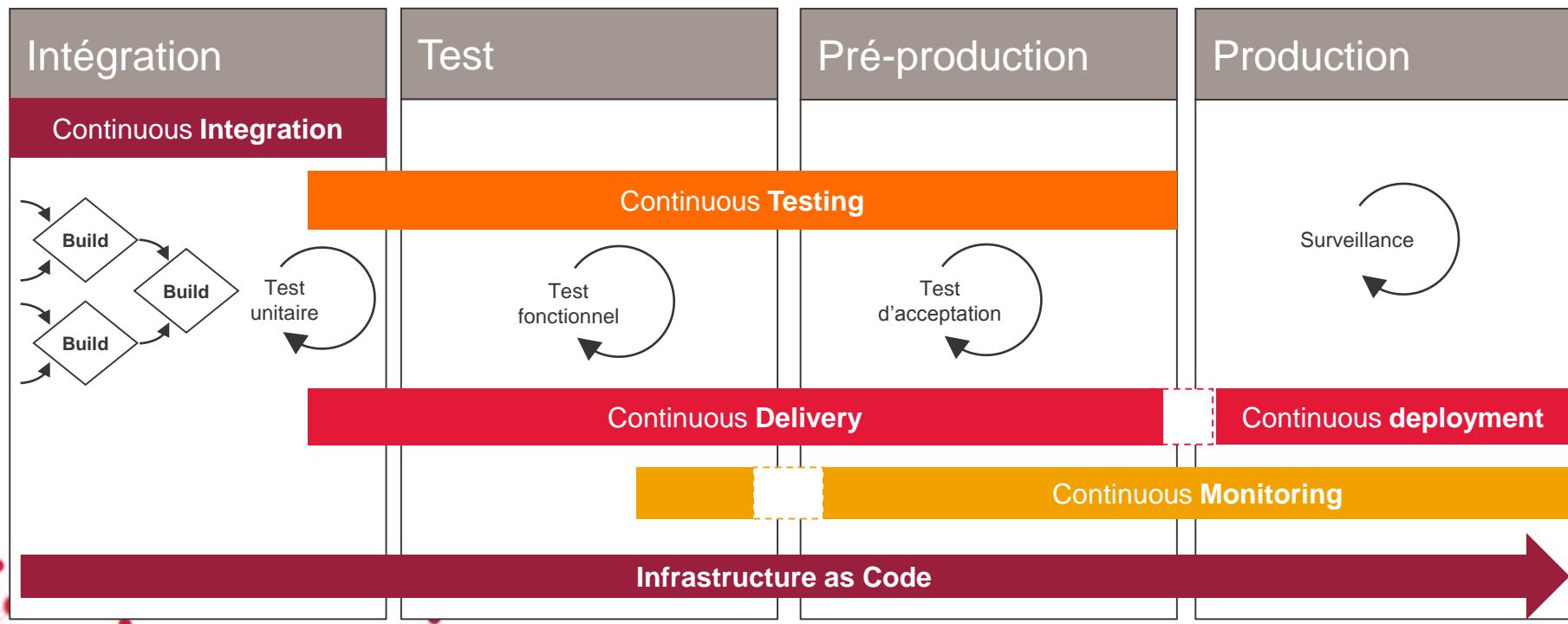
- Surveillance et supervision des environnements, remontée d'alertes, prise de mesures, QA

Infrastructure as code

- Ecrire sous forme de code la gestion de configuration de l'infrastructure
- Bénéficier des pratiques issues du développement (versioning des sources, tests unitaires)

Une industrialisation de l'ensemble des phases

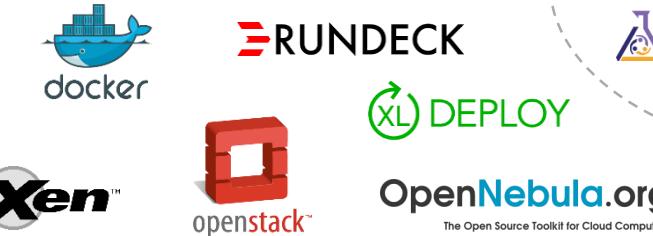
- Développer et tester sur des systèmes similaires à la Production
- Déployer au travers de processus répétables et stables
- Surveiller, contrôler et valider la qualité
- Accélérer les feedbacks utilisateurs



Des processus automatisés

Quelques exemples d'outils...

Continuous Integration



Infrastructure as Code

Continuous Delivery



Continuous Testing



Continuous Monitoring

CGI

Des processus automatisés

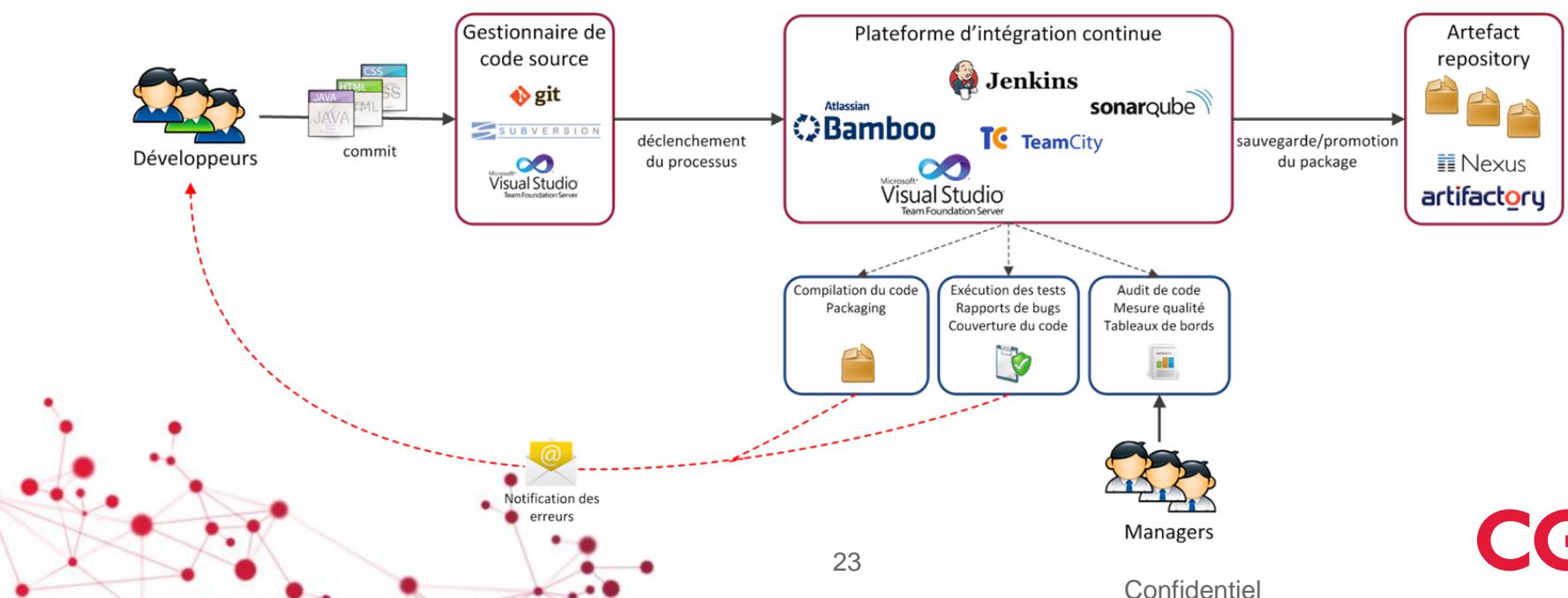
Continuous Integration

Processus essentiel à tout projet de développement

- Permet d'industrialiser les développements
- Repose sur une usine logicielle composée d'outils permettant d'automatiser la phase d'intégration des développements

Objectifs

- Mesurer et améliorer la qualité du code
- Contrôler le respect des standards de développement
- Automatisation des tests unitaires
- Déetecter les anomalies au plus tôt
- Traçabilité du package déployé dans les # env.
- Visibilité sur l'avancement du projet



Des processus automatisés

Continuous Integration - exemple de rapports

Hudson Hudson > Suivi2-Quality

Projet Suivi2-Quality Application web Suivi2

Historique des builds (tendances)

- #173 9 juil. 2012 22:17:02 105MB
- #172 6 juil. 2012 22:14:00 35MB
- #170 5 juil. 2012 13:51:53 35MB
- #169 4 juil. 2012 22:25:16 35MB
- #168 3 juil. 2012 22:13:20 35MB
- #167 2 juil. 2012 22:11:21 35MB
- #166 29 juin 2012 22:23:47 35MB
- #165 28 juin 2012 22:10:46 35MB
- #164 28 juin 2012 10:37:15 35MB
- #162 26 juin 2012 22:09:43 35MB
- tous les builds** **tous les échecs**

Liens permanents

- Dernier build (#173), il y a 11:35 h
- Dernier build ayant succès (#172), il y a 11:12 h
- Last unstable build (#173), il y a 11:12 h
- Last unsuccessful build (#173), il y a 11:12 h

Projets en amont

- Suivi2-QDEV-Littorables

Disk Usage: Workspace 962MB, Builds 417MB

Open Tasks Trend

Coverage Report

Sonar

Espace de travail

Changements récents

Derniers résultats des tests (130 échecs / #0)

Checkstyle Trend

Duplicate Code Trend

Project Coverage summary

Name	Classes	Conditionals	Files	Lines	Methods	Packages	
Cobertura Coverage Report	71%	517/724	100% 0/0	74% 456/617	25% 18604/5733	31% 4785/15400	72% 41/57

Coverage Breakdown by Package

Name	Classes	Conditionals	Files	Lines	Methods	
eu.curia.suivi2.dao	100%	1/1	N/A	100% 1/1	51% 19/373	56% 137/236
eu.curia.suivi2.service.impl	75%	3/4	N/A	75% 3/4	33% 109/319	64% 27/42
eu.curia.suivi2.session.authentication	0%	0/1	N/A	0% 0/1	0% 0/10	0% 0/7
eu.curia.suivi2.view.fxml	100%	1/1	N/A	100% 1/1	73% 307/418	87% 34/39
eu.curia.suivi2.dao.suivi2.impl	100%	3/3	N/A	100% 3/3	58% 718/1342	60% 154/257
eu.curia.suivi2.constant	70%	43/61	N/A	100% 2/2	58% 144/749	37% 57/181
eu.curia.suivi2.bean.impl	83%	10/12	N/A	83% 10/12	19% 61/327	21% 49/229
eu.curia.suivi2.bean.java	50%	14/28	N/A	53% 10/14	19% 61/327	0% 0/22
eu.curia.suivi2.view.converter	0%	0/6	N/A	0% 0/6	0% 0/79	0% 0/22
eu.curia.suivi2.view.managedbean.popups	81%	35/43	N/A	85% 34/40	4% 166/4316	6% 56/947
eu.curia.suivi2.datamanager.suivi1.impl	100%	3/3	N/A	100% 3/3	58% 264/459	95% 42/44
eu.curia.common.logger	50%	1/2	N/A	50% 1/2	31% 14/45	50% 3/6
eu.curia.suivi2.service.impl	0%	0/4	N/A	0% 0/4	0% 0/43	0% 0/20
eu.curia.suivi2.service.xml	100%	1/1	N/A	100% 1/1	67% 22/33	82% 9/11
eu.curia.suivi2.view.tracker	0%	0/4	N/A	0% 0/4	0% 0/12	0% 0/16
eu.curia.suivi2.session.cipher.impl	67%	2/3	N/A	67% 2/3	74% 47/58	73% 11/15
eu.curia.suivi2.view.managedbean	85%	76/89	N/A	99% 67/68	4% 1018/25370	3% 145/4615
eu.curia.suivi2.bean.config	50%	1/2	N/A	50% 1/2	14% 6/43	21% 5/24
eu.curia.suivi2.model.suvi1	100%	9/9	N/A	100% 9/9	7% 49/702	9% 40/465
eu.curia.common.util	60%	3/5	N/A	60% 3/5	23% 21/93	35% 8/23

Alerts Unit test snapshot, 29 mai 2012 10:48

Comments 27,5%
46 115 lines ▲
98,4% docu API
107 undocu API
992 commented LOCs

Duplications 5,6%
14 395 lines ▼
646 blocks
112 files

Violations 1 084
Rules compliance 98,6%

Package tangle index 3,0%
> 54 cycles

Dependencies to cut 17 between packages
46 between files

LCOM 1,7 /class
24,8% files having LCOM4+1

Response for Class 40 class

Security violations 0
100% compliance
66 rules activated

Tags 60
0 mandatory
60 optional
0 nosnar

Technical Debt 13,2%
\$ 403 316 ▼
801 man days ▼

Design

Events All

Events All

Des processus automatisés

Continuous Testing



Equipe d'exploitation

- Configuration des environnements de tests et monitoring
- Test et validation des environnements
- Support pour les tests de performance

Equipe de développement

- Ecriture et automatisation des tests unitaires et conception des jeux de données

Des types de tests différents exécutés en continu:

- Tests unitaires
- Tests d'intégration
- Tests fonctionnels
- Tests non fonctionnels
 - Tests de performances, tests de sécurité (intrusion check OWASP), tests de robustesse (fuite mémoire, deadlocks, connection pools), tests d'infrastructure (vérification de l'installation sur l'ensemble des composants de la chaîne, vérification de la chaîne applicative)
- Tests d'acceptation

Equipe de test

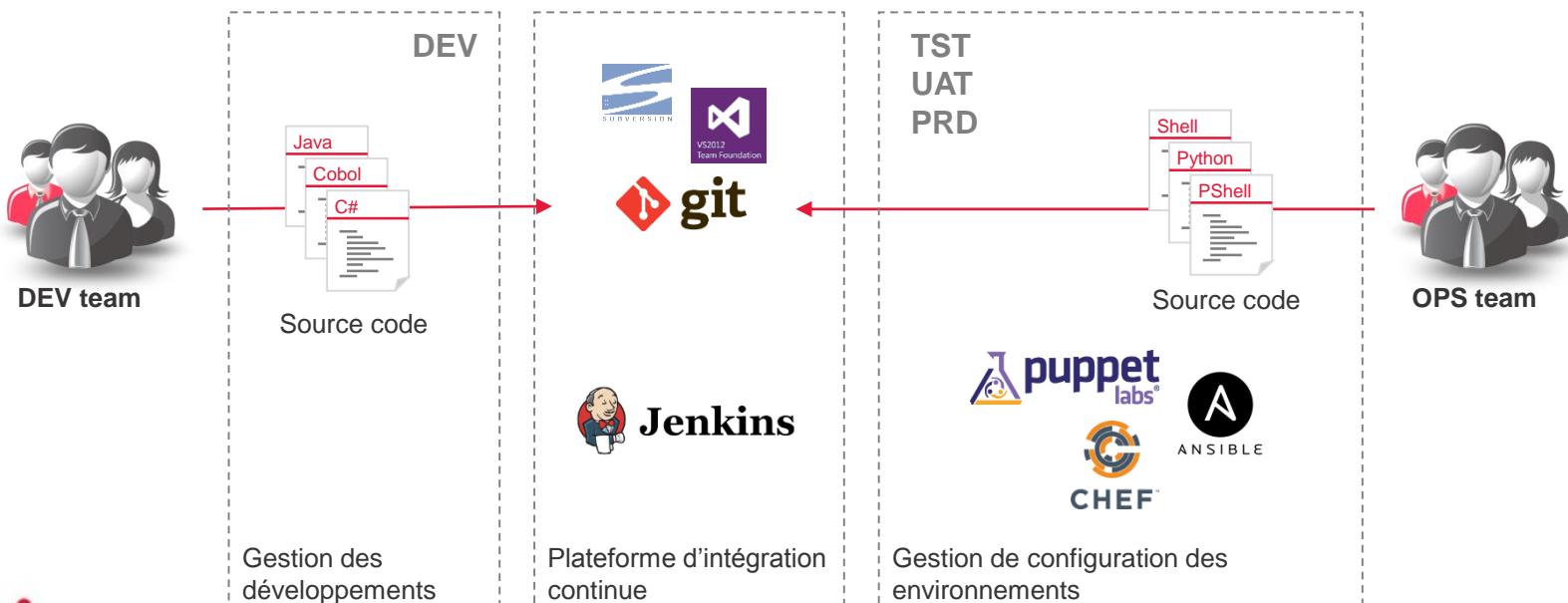
- Ecriture et automatisation des tests fonctionnels et non-fonctionnels et définition des jeux de données

Un projet d'automatisation des tests est un vrai projet de développement.
Elle se fait en scriptant les cas de tests dans le langage de l'automate choisi.

Des processus automatisés

Infrastructure as code

- Ops « code » la configuration et l'automatisation du provisionnement des environnements (système, serveur d'applications, base de données, etc.)
- Ne consiste pas uniquement à écrire des scripts, mais plutôt de bénéficier des bonnes pratiques déjà éprouvées dans le développement d'application (gestion des versions, plateformes d'intégration continue, automatisation des tests)



Des processus automatisés

Continuous Delivery

Processus permettant de s'assurer qu'après chaque modification l'application est dans un état « prêt à être déployée en production »:

- Englobe le processus d'intégration continue
- Englobe le processus de tests en continu
- Assure la gestion de configuration des environnements
- Assure le provisionnement des environnements
- Automatisation du processus jusqu'en « Acceptance »

Le déploiement en production reste déclenché **manuellement**.

Objectifs

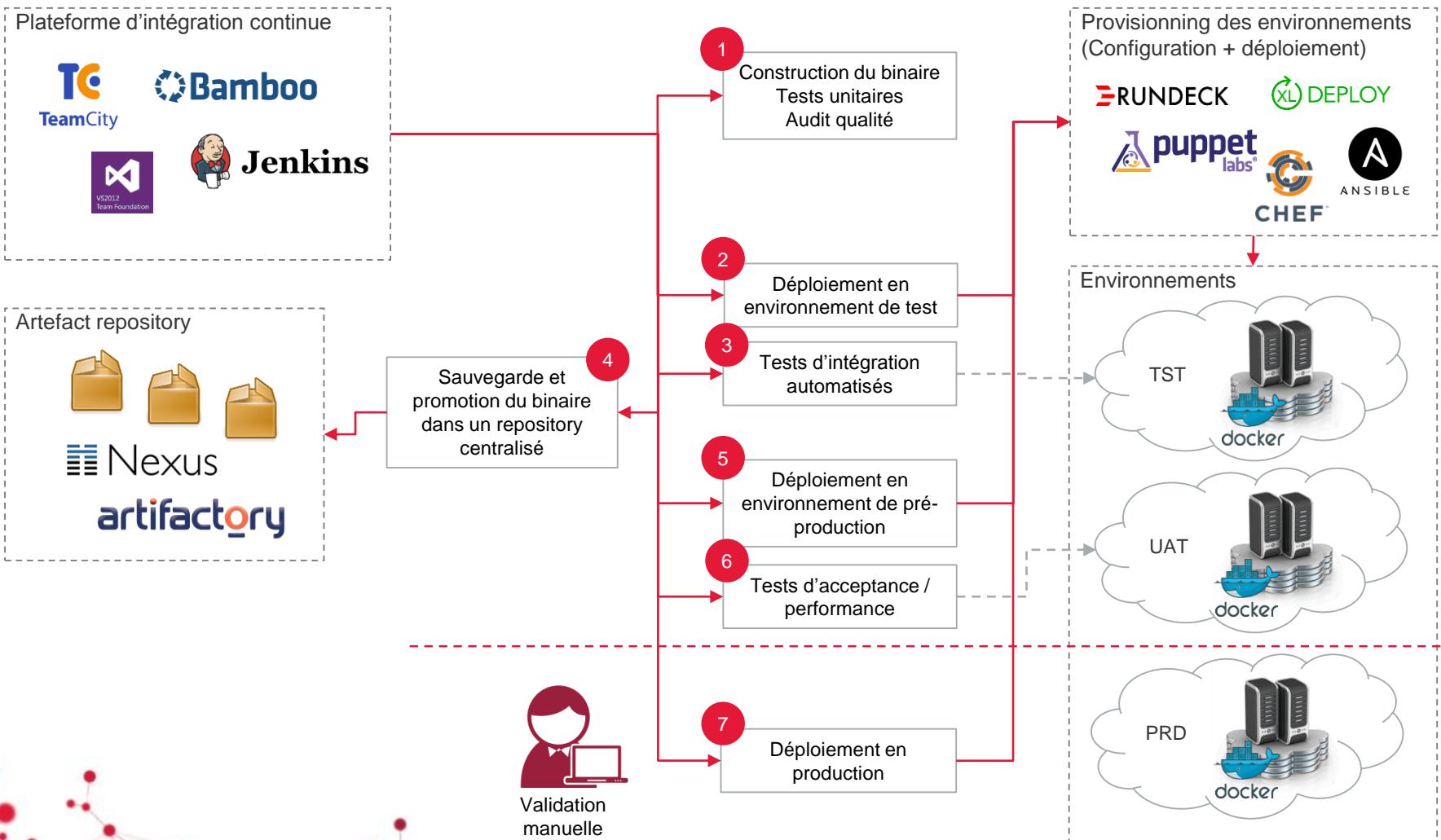
- Fournir un feedback rapide aux utilisateurs
- Améliorer la qualité de l'application
- Sécuriser les déploiements via une gestion de configuration versionnée
- Réduire le time to market



Ne pas confondre avec le « continuous deployment » qui consiste à automatiser jusqu'à la production

Des processus automatisés

Continuous Delivery - orchestration du processus



Pilotage & Gouvernance



CGI

La force de l'engagement^{MD}

Pilotage & gouvernance

Objectif : mesure et boucle de feedback rapide avec le métier

1. Priorisation



- Prioriser les projets au sein du portefeuille
- Prioriser les features au sein d'un projet
- Kanban pour managers

2. Mesure



- Monitoring métier
- A/B testing

3. Expérimentation



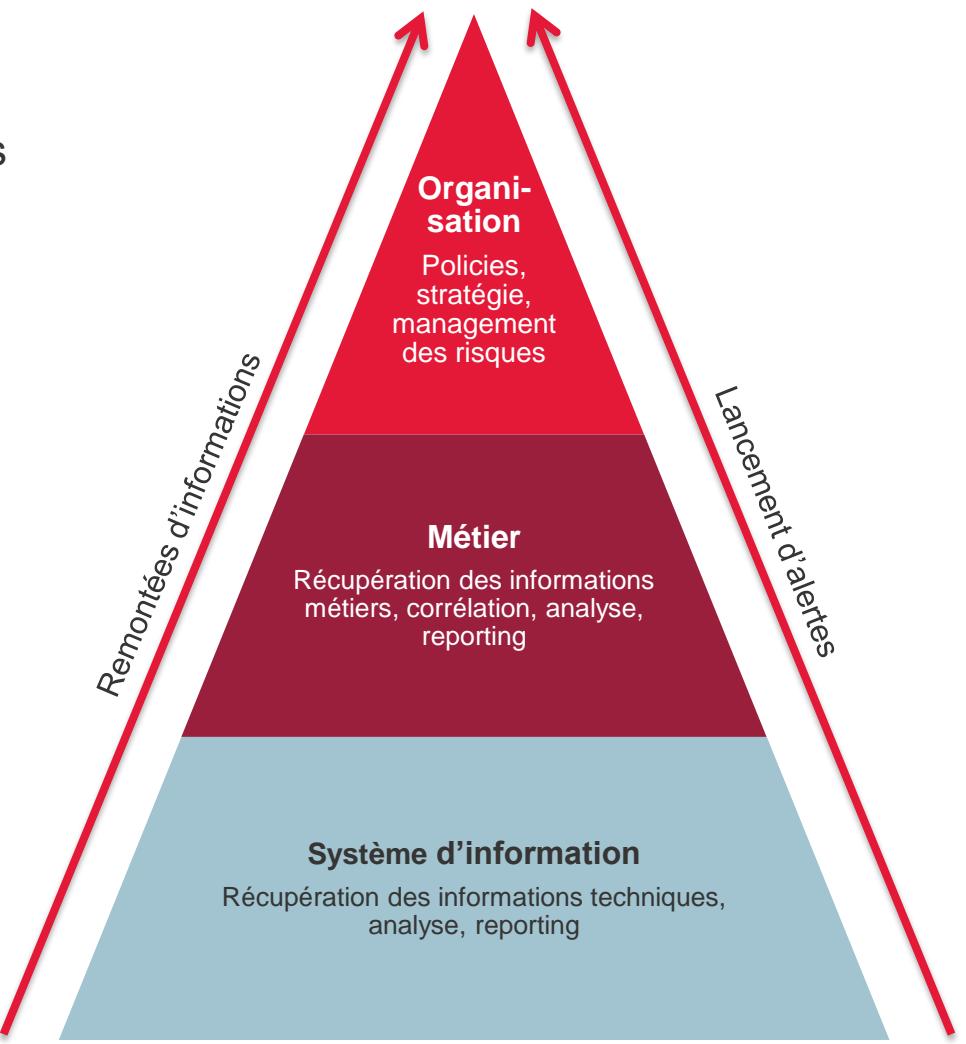
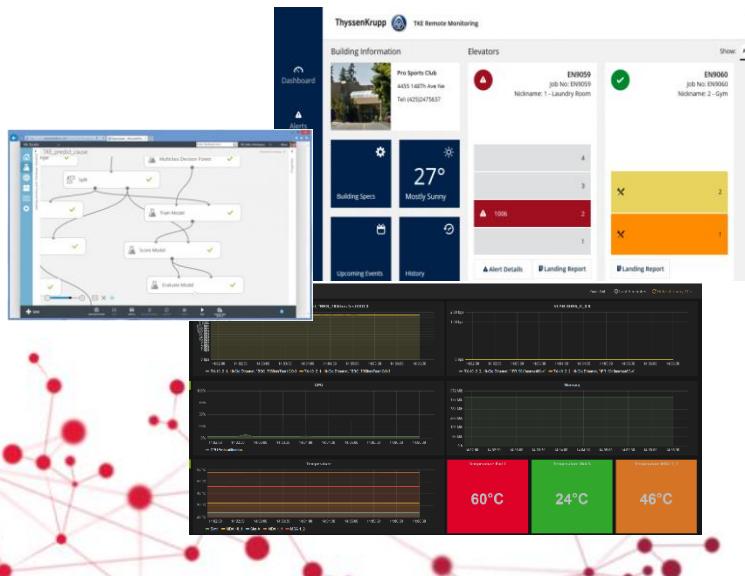
- End-user feedback
- Build / Measure / Learn (Lean Startup)

Continuous Monitoring

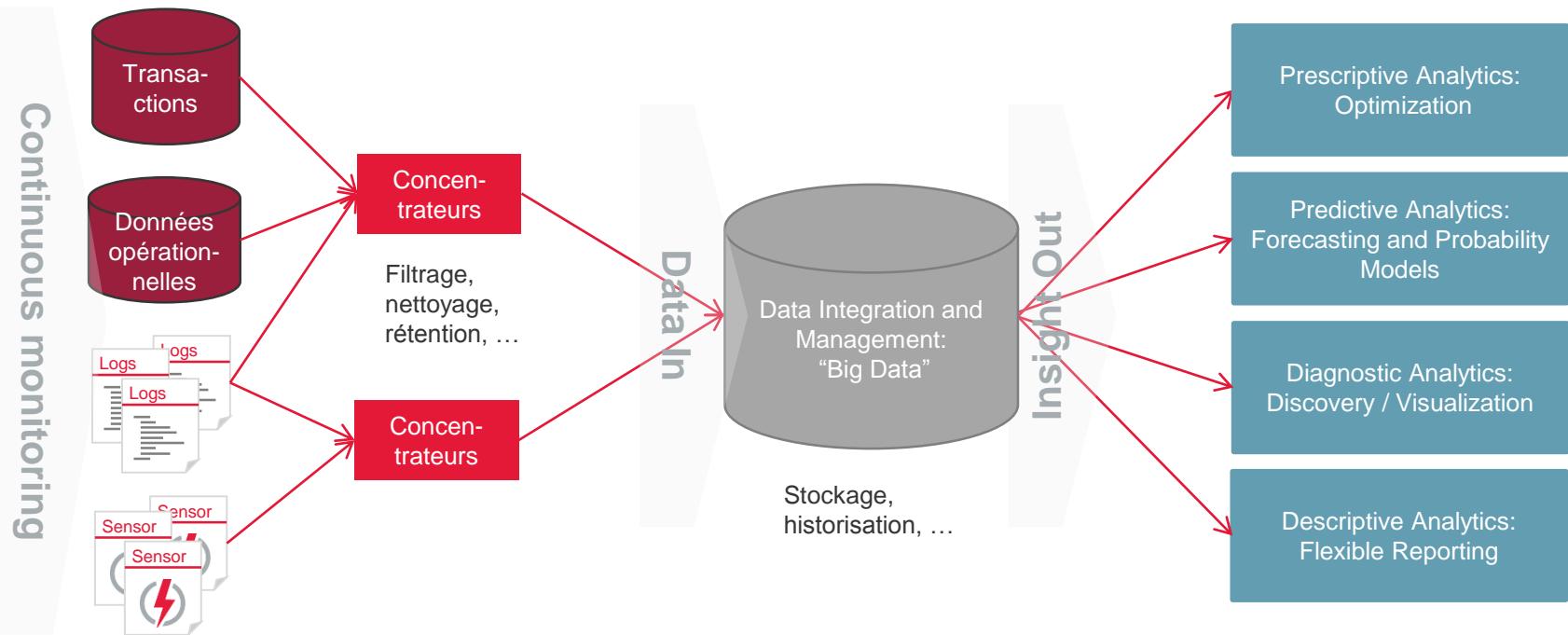
Pilotage

« On ne peut améliorer que ce que l'on peut mesurer »

- Remontée d'information et/ou d'alertes dans un but:
 - D'analyse prédictive
 - Optimisations
 - Prévisions, machine learning
 - De diagnostic
 - Investigation suite à incident
 - Reporting



Gagner en valeur et réduire la complexité



Créer des métriques regroupant plusieurs sources :

- Mesures techniques (informations JVM, CPU ...)
- Mesures applicatives (état d'un pool de connexion, temps réponses des transactions ...)
- Mesures métiers (tracer l'activité d'un utilisateur, ...)

Regrouper et corrélérer ces différents critères pour réaliser des **indicateurs partagés** par toute l'entreprise (dev, ops, métier)

Utilisation du visual management pour exposer des **indicateurs métier** permettant de focaliser l'attention sur les objectifs principaux de l'entreprise

Retours d'expérience



CGI

La force de l'engagement^{MD}

DevOps Secteur Public



CGI

La force de l'engagement^{MD}

Continuous Delivery

Plateforme d'industrialisation

Challenge

- Améliorer le time to market en automatisant au maximum les processus de développement et de livraison des applications
- Sécuriser les mises livraisons
- Améliorer la qualité des applications
- Faciliter la montée en compétence et l'intégration de nouveaux développeurs

Solution CGI

- Mise en place d'une plateforme d'intégration continue (Git, Jenkins, Sonarqube, Elasticsearch/Kibana)
- Automatisation du provisioning des environnements de développement et intégration via Docker et Puppet

Bénéfices

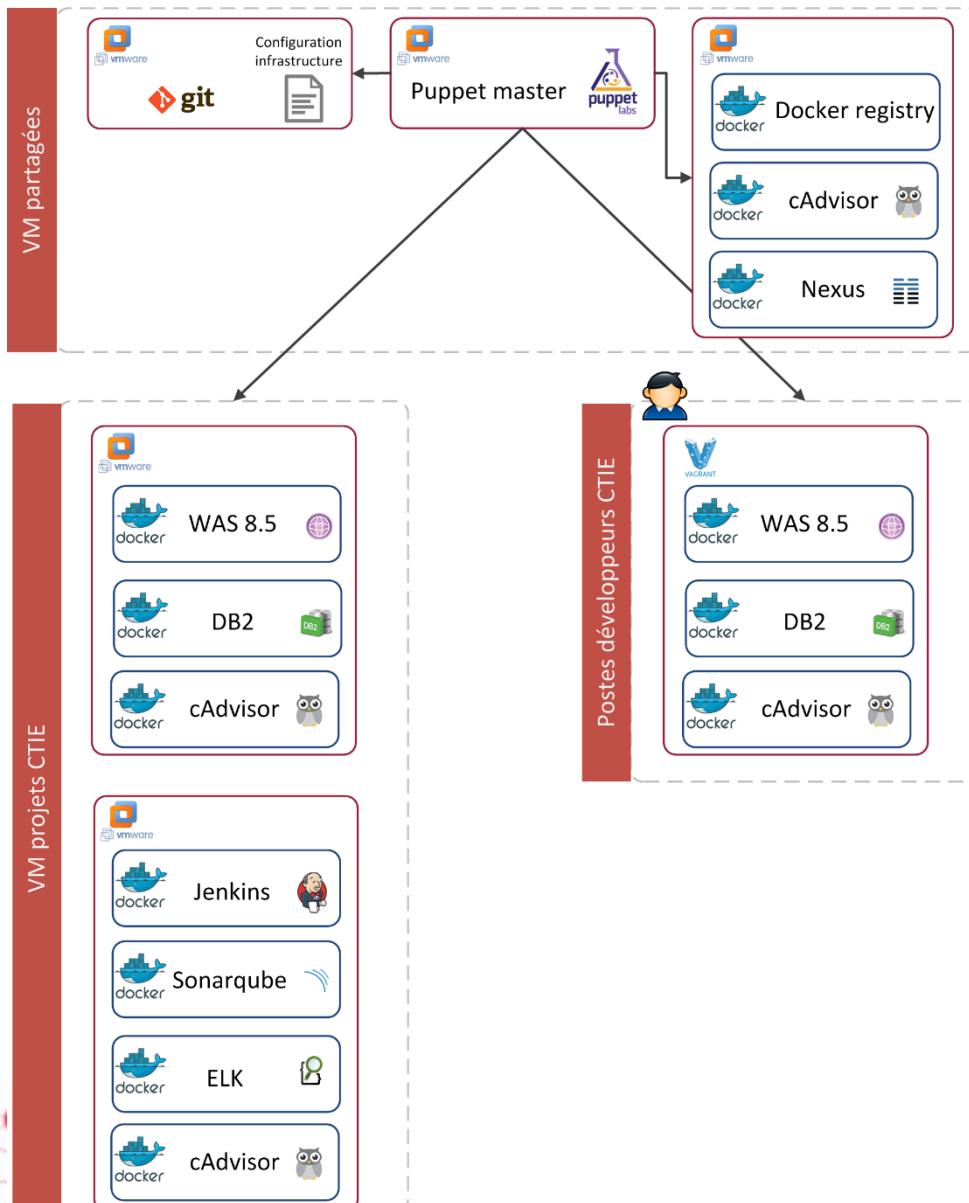
- Processus de développement et de livraison simplifiés et documentés
- Amélioration de la productivité de l'équipe de développement
- Vision complète des indicateurs qualité des applications
- Automatisation des tests
- Maîtrise totale de la configuration des environnements
- Déploiement en environnement d'homologation chez le client sécurisé

Pourquoi CGI?

CGI propose une démarche personnalisée à chaque client en les accompagnant dans leur démarche DevOps. Notre connaissance des solutions techniques et des méthodes est le fruit d'une vaste expérience en réalisation de plateforme d'industrialisation des développements ainsi que d'une volonté constante d'innover sur les projets que nous réalisons. Cette expérience acquise nous permet d'aider au mieux nos clients à automatiser leurs processus et ainsi leur permettre d'améliorer leur productivité tout en diminuant les risques et les coûts.



Architecture infrastructure



- Configuration infrastructure sous Git
- Provisioning et maintien de l'état des VM via Puppet
- Images Docker customisées stockées dans la docker registry
- Applicatifs s'exécutant dans des containers Docker issus de la registry
- VM sur les postes développeurs provisionnées via Puppet et Vagrant (Utiliser pour les tests d'intégration)
- cAdvisor : Container Docker monitorant les containers de la VM

DevOps Opérateur Luxembourgeois

Continuous delivery



CGI

La force de l'engagement^{MD}

Continuous Delivery

Service technologies de transport xDSL & FTTx

Challenge

- Améliorer le time to market en automatisant au maximum les processus de développement et de livraison des applications
- Sécuriser les mises en production
- Améliorer la qualité des applications
- Faciliter la montée en compétence et l'intégration de nouveaux développeurs

Solution CGI

- Animation de workshops pour définir les objectifs du projet ainsi que les besoins nécessaires pour les réaliser. Associer le client à la conception d'une vision cible de la solution
- Mise en place d'une plateforme d'intégration continue (Git, Jenkins, Sonarqube, Artifactory)
- Automatisation du provisioning des environnements de test et de production via Docker et Ansible

Bénéfices

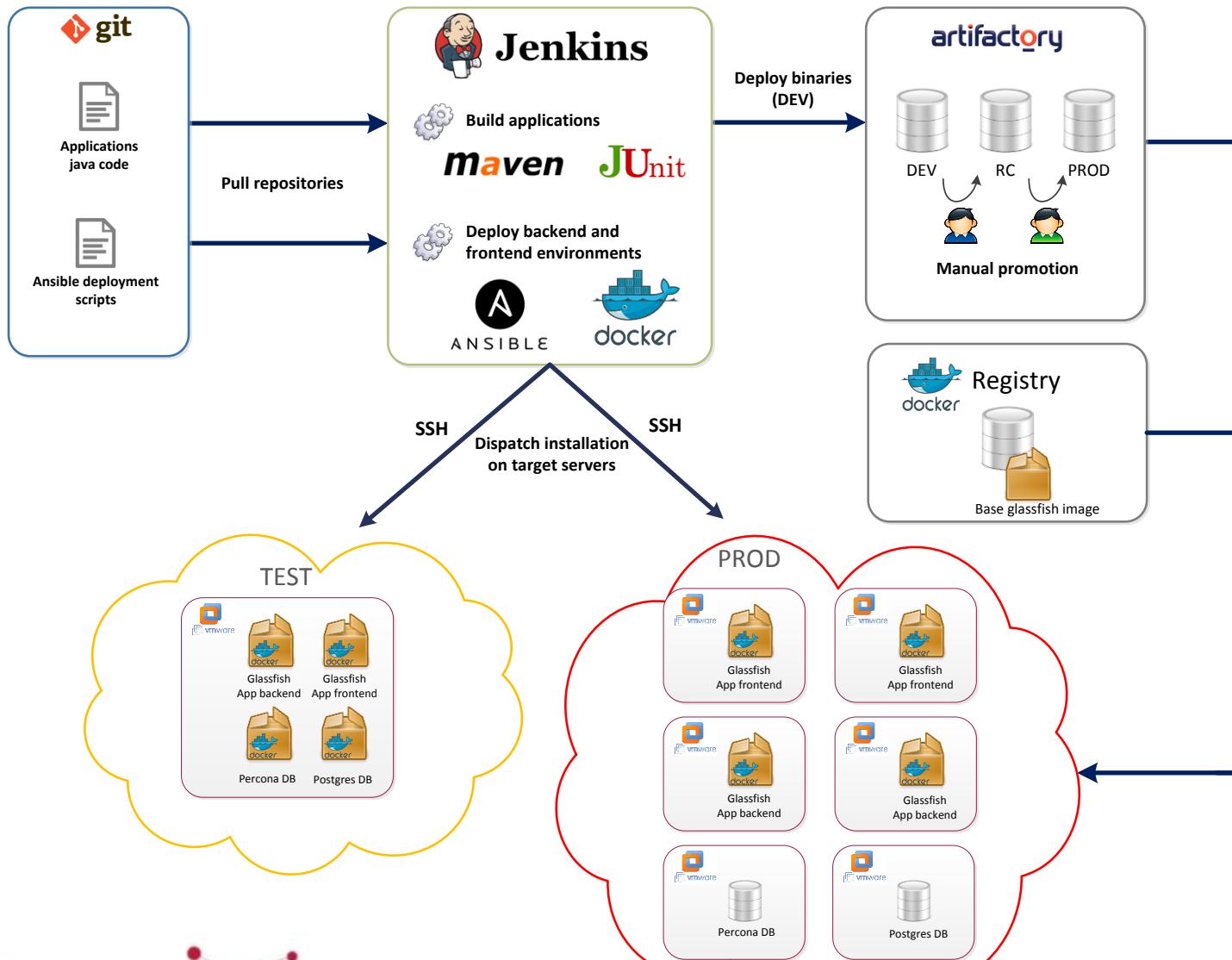
- Processus de développement et de livraison simplifiés et documentés
- Amélioration de la productivité de l'équipe de développement
- Vision complète des indicateurs qualité des applications
- Automatisation des tests
- Maîtrise totale de la configuration des environnements
- Création de copie d'environnement de production à la demande
- Déploiement en production sécurisé

Pourquoi CGI?

CGI propose une démarche personnalisée à chaque client en les accompagnant dans leur démarche DevOps. Notre connaissance des solutions techniques et des méthodes est le fruit d'une vaste expérience en réalisation de plateforme d'industrialisation des développements ainsi que d'une volonté constante d'innover sur les projets que nous réalisons. Cette expérience acquise nous permet d'aider au mieux nos clients à automatiser leurs processus et ainsi leur permettre d'améliorer leur productivité tout en diminuant les risques et les coûts.

Continuous Delivery

Service technologies de transport xDSL & FTTx



Quelques chiffres pour la production...

4 serveurs physiques

4 serveurs glassfish en loadbalancing

15 datasources

plus de 30 queues JMS

2 security realms (file / database)

25 applications déployées

processus de déploiement < 10 min

downtime < 1min



Plateforme microservices pour la VoIP

Service Applications Télécom Voix

Challenge

- Améliorer le time to market du déploiement des services VoIP dédiés aux « entreprises »
- Plateforme stable et opérationnelle 24/7
- Pro-activité dans le monitoring et l'alerting
- Standardiser les processus de développement

Solution CGI

- Workshops et association du client à la conception de la solution
- Mise en place d'une plateforme d'intégration continue (Git, Jenkins, Sonarqube, Artifactory)
- Automatisation du provisioning de nouveaux services via Rundeck, Ansible, Spring Boot et Docker
- Monitoring opérationnel et applicatif

Bénéfices

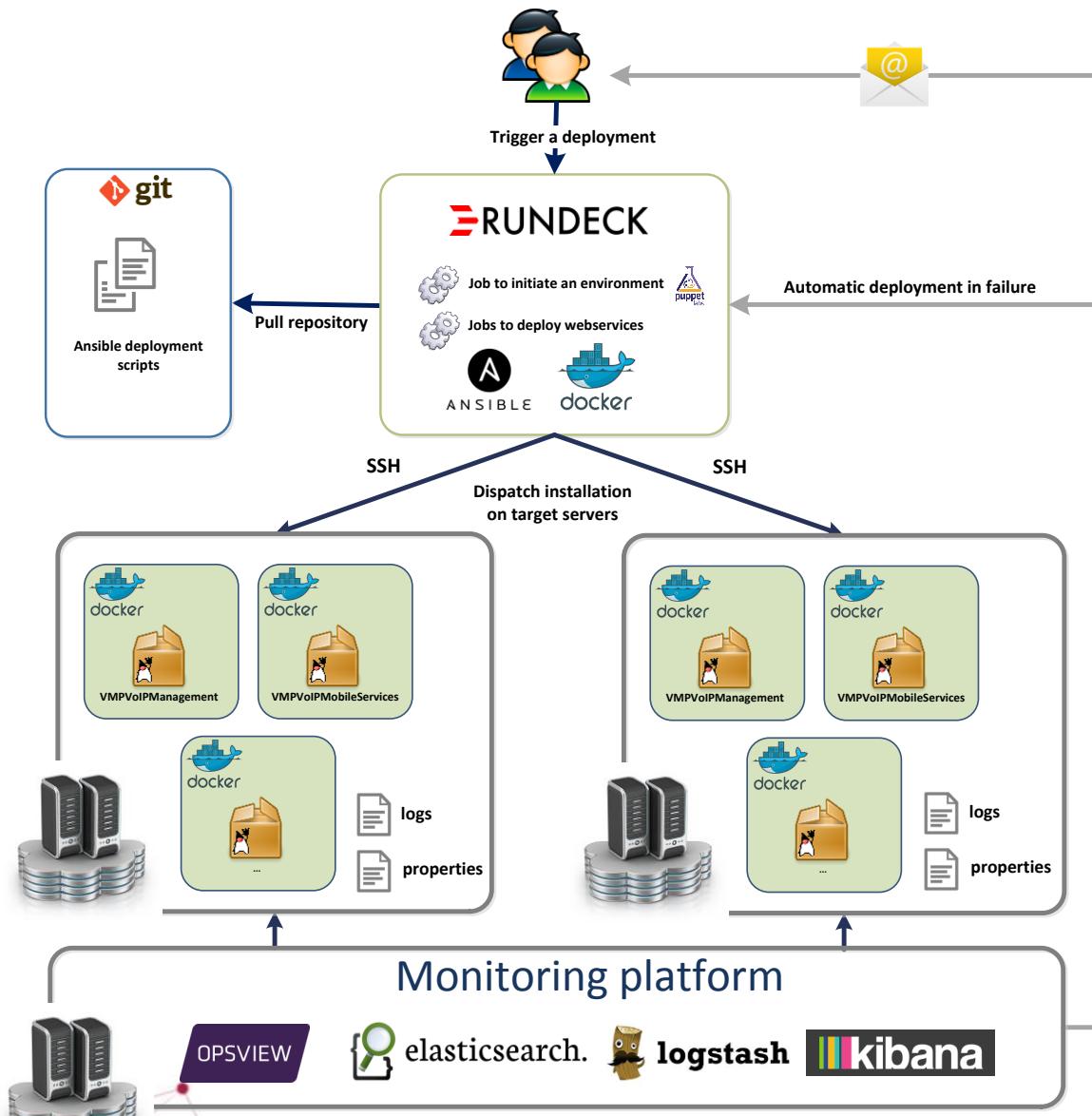
- Processus de livraison industrialisé
- Haute disponibilité de la plateforme
- Système d'auto-déploiement des services en cas d'erreur
- Vision complète des indicateurs de santé de la plateforme
- Statistiques métiers sur l'utilisation de la plateforme

Pourquoi CGI?

CGI propose une démarche personnalisée à chaque client en les accompagnant dans leur démarche DevOps. Notre connaissance des solutions techniques et des méthodes est le fruit d'une vaste expérience en réalisation de plateforme d'industrialisation des développements ainsi que d'une volonté constante d'innover sur les projets que nous réalisons. Cette expérience acquise nous permet d'aider au mieux nos clients à automatiser leurs processus et ainsi leur permettre d'améliorer leur productivité tout en diminuant les risques et les coûts.

Plateforme microservices pour la VoIP

Service Applications Télécom Voix



Continuous monitoring

Services Access Services / xDSL / Backbones

Challenge

- Collecter des informations provenant de diverses sources
- Conserver les données pendant une longue période sans perte de précision
- Automatiser la configuration des agents sur des milliers de ressources
- Interfaces graphiques totalement adaptables
- Remontée de notifications en temps réel

Solution CGI

- Workshops pour définir les exigences fonctionnelles et techniques
- Etude de faisabilité technique sur les outils (agent/base de données/UI)
- Mise en place d'un PoC automatisant la configuration des éléments sur un périmètre cible

Bénéfices

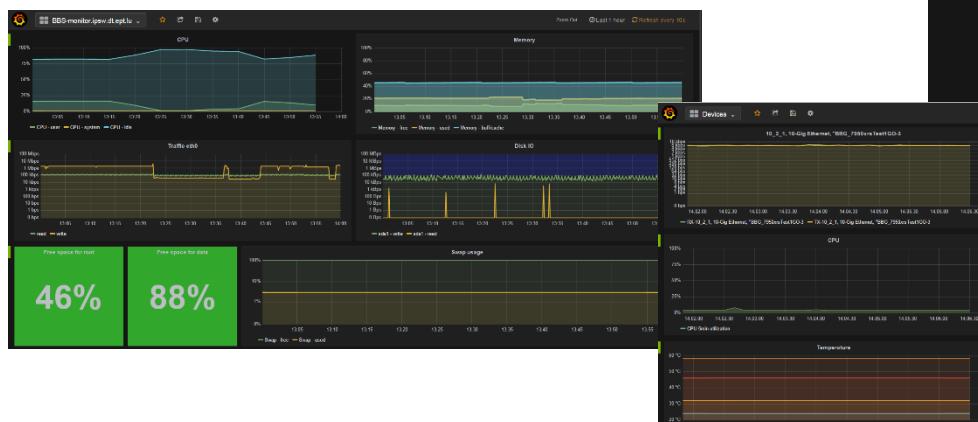
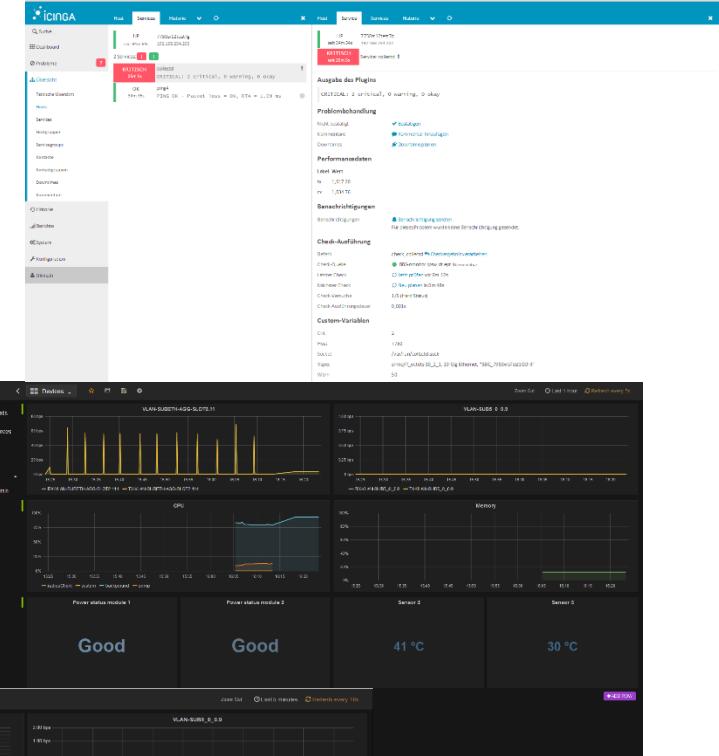
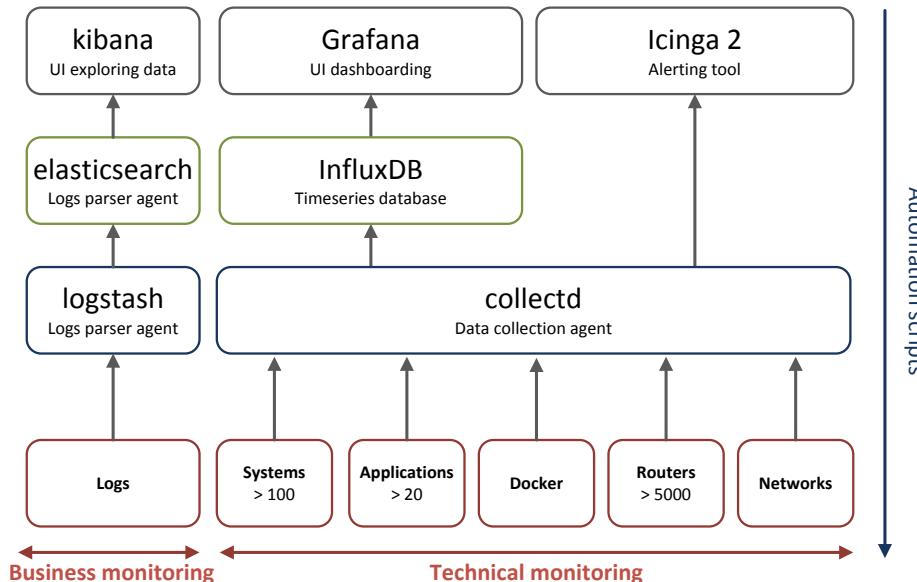
- Configuration simplifiée du monitoring lors de la mise en place de nouveaux éléments (VM, applications, routeurs ...)
- Affichage temps réel sur un gros volume de données
- Peu ou pas de perte de précision sur l'analyse des données
- Vision complète des indicateurs techniques et métiers afin d'établir des corrélations

Pourquoi CGI?

CGI propose une démarche personnalisée à chaque client en les accompagnant dans leur démarche DevOps. Notre connaissance des solutions techniques et des méthodes est le fruit d'une vaste expérience en réalisation de plateforme d'industrialisation des développements ainsi que d'une volonté constante d'innover sur les projets que nous réalisons. Cette expérience acquise nous permet d'aider au mieux nos clients à automatiser leurs processus et ainsi leur permettre d'améliorer leur productivité tout en diminuant les risques et les coûts.

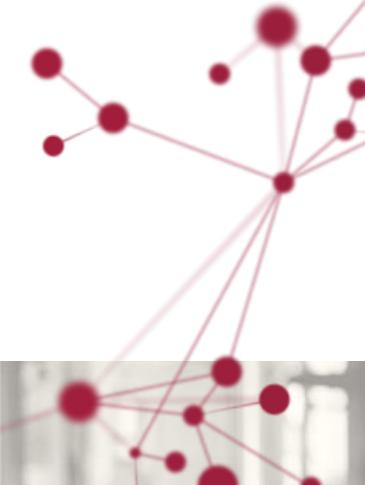
Continuous monitoring

Services Access Services / xDSL / Backbones



DevOps

Banque Luxembourgeoise



CGI

La force de l'engagement^{MD}

Core banking

Infrastructure as Code

Challenge

- Pour la création d'une nouvelle banque, le client a souhaité optimiser au maximum les temps de mise à disposition du Core Banking et des l'ensemble des environnements techniques nécessaires aux opérations de la banque.

Solution CGI

- Workshops pour définir les exigences et définir l'architecture cible (haute dispo, performances, tolérance de pannes communications inter applicatives, sécurité, ...)
- Développement de l'infrastructure sur base des composants techniques Nutanix, VMWare, Windows, IBM Websphere App Server, ESB, API Mngt et Gateway, pour tous les environnements

Bénéfices

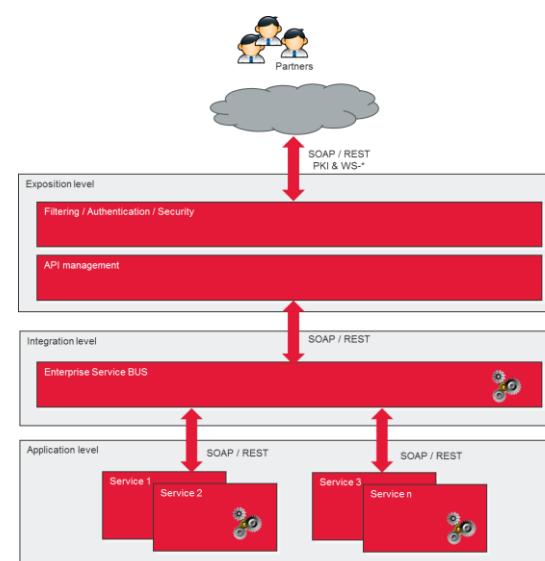
- Time to market, construction et mise à disposition d'un environnement complet en moins de 35 min.
- Réduction des erreurs lors des déploiements applicatifs ou de nouveaux flux.
- Sécurisation des déploiements en production

Pourquoi CGI?

CGI propose une démarche personnalisée à chaque client en les accompagnant dans leur démarche DevOps. Notre connaissance des solutions techniques et des méthodes est le fruit d'une vaste expérience en réalisation de plateforme d'industrialisation des développements ainsi que d'une volonté constante d'innover sur les projets que nous réalisons. Cette expérience acquise nous permet d'aider au mieux nos clients à automatiser leurs processus et ainsi leur permettre d'améliorer leur productivité tout en diminuant les risques et les coûts.

Core Banking System

Automatisation du déploiement du Core Banking System Basé sur Temenos T24



Quelques chiffres pour la mise en œuvre...

13 serveurs / env

8 serveurs Websphere en loadbalancing

2 serveurs MQ en failover

2 serveurs IBM IIB en failover

6 datasources

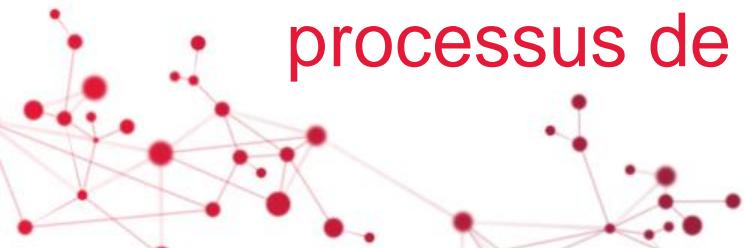
29 queues JMS

Intégration active directory

CBS et une dizaine de flux

déployés et opérationnels

processus de déploiement < 35 min (hors DB)



Notre engagement

Nous réalisons chaque mandat dans un seul but : contribuer au succès de nos clients.



CGI

La force de l'engagement^{MD}