



DevOps: quand les skills techniques et le mindset s'allient pour Agiliser le développement !

Séminaire UMONS 2019



Les présentations d'abord!



Un mot sur moi

- Ancien étudiant de la Faculté (promotion 2009)
- Consultant IT depuis lors
- Profil: DevOps & Agile Coach
- Actuellement en mission: DevOps Specialist @ BNPPF
- Fondateur de MYRMIDONS



Un mot sur Myrmidons

- IT Consulting & Coaching
- Très jeune structure en pleine évolution
- 3 grands axes:
 - Software Craftsmanship
 - Data Management
 - Incubation



L'idée générale

Dans la mythologie grecque, Les Myrmidons étaient connus pour leur capacité à renverser le cours des batailles auxquelles ils participaient, même s'ils étaient inférieurs en nombre à leurs opposants...

LA bonne personne à la bonne place: cela fait LA différence!



Quelques Références

- Clients



- Technologies / Outils

Agnostiques !

(... mais conscients des réalités du marché ... :^)

Notre Offre



PERMANENT-POSITION
STUDENT-JOB
TRAINEESHIP
Myrmidons

mindset
smart
commitment
culture
research
transparency
potential
incubator
passion
together
coaching
transformations
digital
beyond
career
start-up
human
difference
empowerment
opportunities
strengths
valuable
challenges
quality
fun
agile
communication
strengths
challenges
opportunities
empowerment
coaching
transformations
digital
beyond
career
start-up
human
difference
research
transparency
potential
incubator
passion
together



Un mot sur l'Agile

Un contexte pour DevOps...

Agilité ?



L' Agile pour ...

... une livraison rapide de
“Valeur Business”

Agilité ?

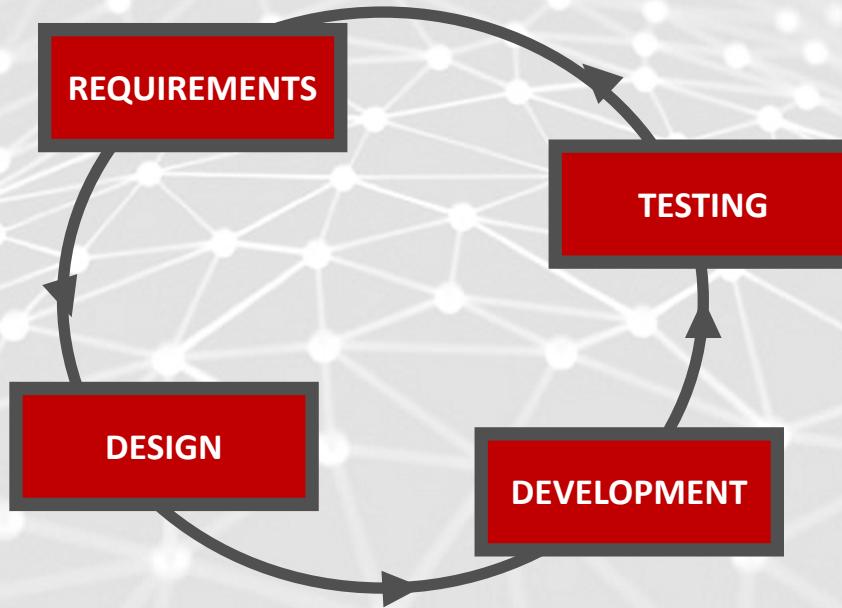


Ensemble de pratiques de gestion de projet ...

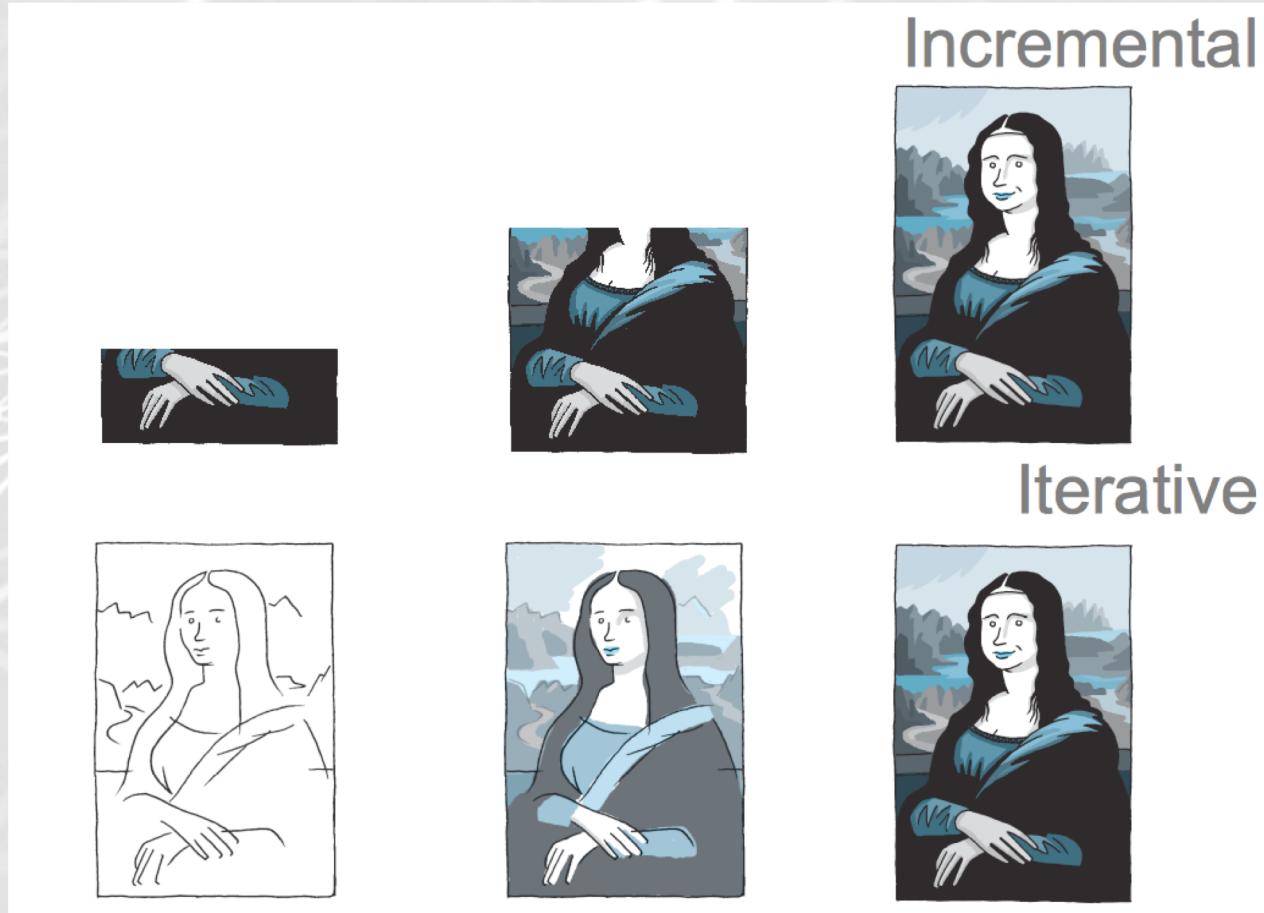
... se reposant sur une structure commune (itératrice,
incrémentale et adaptative) ...

... grâce auxquelles le client (impliqué activement) peut
retirer un maximum de valeur.

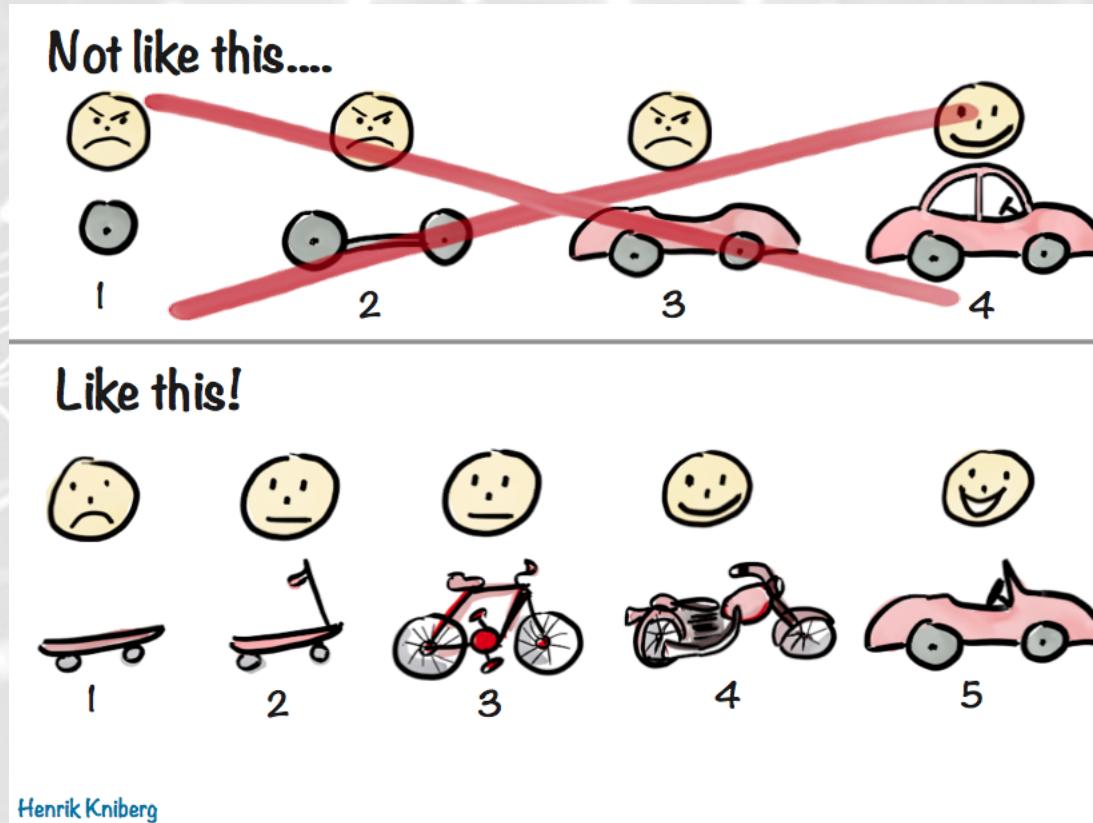
Agilité ?



Agilité ?



Agilité ?



Agilité ?



- Contrairement aux pratiques traditionnelles, l'Agilité presuppose que:
 - Le client découvre ce qu'il souhaite
 - Le développeur découvre comment le construire
 - Le contexte change continuellement

Le Manifeste



4 Piliers et 12 Principes sous-jacents

Le Manifeste



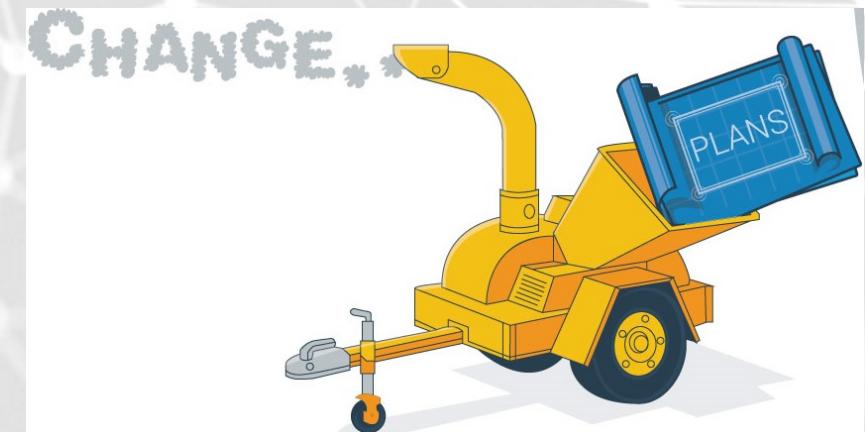
Individus & Interactions plutôt que des outils et des processus



Collaboration client plutôt que négociation de contrat



Un logiciel qui fonctionne plutôt qu'une documentation exhaustive



Répondre au changement plutôt que suivre un plan

Les « approches » les plus communes



- Lean IT
- Scrum
- Kanban
- ScrumBan
- XP



Méthodes ET Pratiques

- Les approches précédentes cherchent à créer un « **contexte propice** » à l'Agilité
- Elles sont nécessaires mais pas suffisantes !



Méthodes ET Pratiques

- Il faut également des pratiques techniques cohérentes afin de:
 - Maintenir un haut niveau de **qualité**
 - Garantir le **contrôle** de la méthode choisie
 - D'éviter les **pertes** en tout genre
 - Fournir du livrable **rapidement** et de manière **flexible**



Méthodes ET Pratiques

DevOps ...

... les pratiques techniques au service de l'Agilité !



DevOps: c'est quoi ?

Définissons les choses...



Généralités

- Terme récent: 2009
- Proposé par Patrick Debois (Cocorico pour la Belgique)
- L'idée générale était d'Agiliser la partie « Infrastructure » d'un projet (cf. Conférence Agile, Toronto 2008)



DevOps: c'est quoi ?

Contraction de « Development » et « Operations »



DevOps: c'est quoi ?

Development, responsable de:

- Produire du code source
- Le transformer en une application exécutable
- Tester le résultat
- Fixer les problèmes

Operations, responsable de:

- Déployer l'application sur l'infrastructure
- Configurer les environnements applicatifs
- S'assurer que les aspects de sécurité/de qualité/... soient appliqués/respectés



DevOps: c'est quoi ?

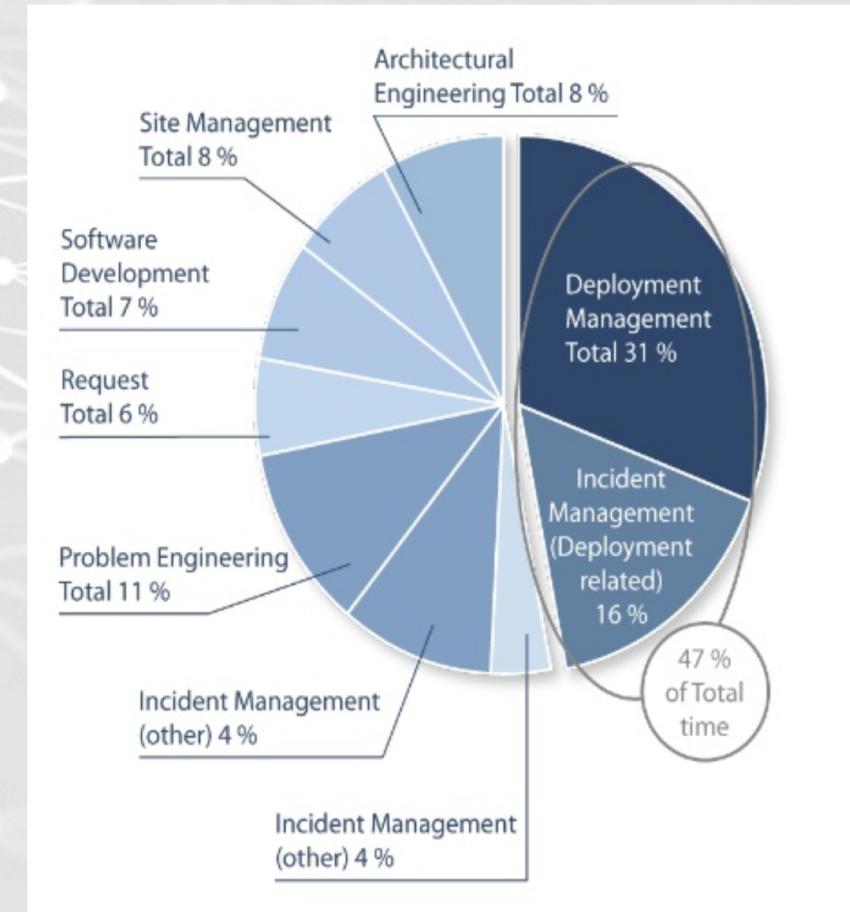
- Problèmes ?
 - Les objectifs des 2 sont diamétralement opposés:
 - Development vise le « Change »
 - Operation vise la « Stability » (le contrôle)
 - Les Operations ne sont pas spécialistes de l'applicatif
 - Ne savent toujours pas comment le déployer/le configurer/... (engendre des « ping-pong »)
 - Dépendances fortes
 - Si les opérations n'avancent pas, les développeurs sont coincés
 - Valeur ajoutée relative des operations
 - Development peut certainement s'en sortir seul
 - Qui travaille souvent en mode « ticketing »
 - Gaspillages et procédures dûs aux silos
 - Pertes de temps importantes
 - Quand on en fait la somme!



DevOps: c'est quoi ?

- Ops en pratique...

47% du temps Ops est gaspillé en problématiques liées au déploiement !





DevOps: c'est quoi ?

DevOps - Mouvement visant à:

- Remettre dans les mains des développeurs les tâches opérationnelles
- En garantissant les aspects de sécurité et de qualité

Dans le but de diminuer le « Time to Market »



DevOps: c'est quoi ?

DevOps cherche à:

- Obtenir un optimum **global** pour un(e) département/entreprise plutôt qu'un optimum localisé à une équipe
- Réduire le **Time to Market** en favorisant la politique du « Tout en continu »
- Changer l'organisation pour que **DevOps** soit applicable



DevOps & Agile

- DevOps = Diminuer le Time to Market via des pratiques techniques
- Agile = Livrer le plus de valeur, le plus rapidement possible



Pour être réellement Agile, il faut adopter DevOps



DevOps core

Rentrans dans le vif du sujet...



Avant toute chose

- Comprendons bien que:
 - Mindset avant tout !
 - La technique/les outils servent le mindset, pas l'inverse !
 - On n'achète pas DevOps ! (outils, licences)
 - Il n'existe pas de profils DevOps !
 - Ce n'est pas une méthode !



4 Piliers à découvrir

- Culture, méthode et organisation
 - Méthodes, rituels et attitudes
 - L'organisation typique ?
 - L'obsession de la mesure
- Architecture et patterns
 - Patterns de scalabilité et de disponibilité
 - Patterns de déploiement
 - Le Cloud : facilitateur de l' « architecture DevOps »
- L'infrastructure par le code
 - Définition
 - Cartographie des outils
 - Docker et son écosystème
- Construction et déploiement continu
 - Définition
 - Usine du « continu »

Culture, méthode et organisation



- Méthodes, rituels et attitudes
- L'organisation typique ?
- L'obsession de la mesure



Culture, méthodes et organisation

- La culture doit être Agile:
 - « Fail fast »
 - Eviter les pertes
 - Amélioration Continue
- Les rituels sont ceux amenés par Scrum:
 - Planifier à - très - court terme (*aka Sprint Planning*)
 - Alignements récurrents (*aka Daily Scrum*)
 - Récolter le feedback (*aka Sprint Review*)



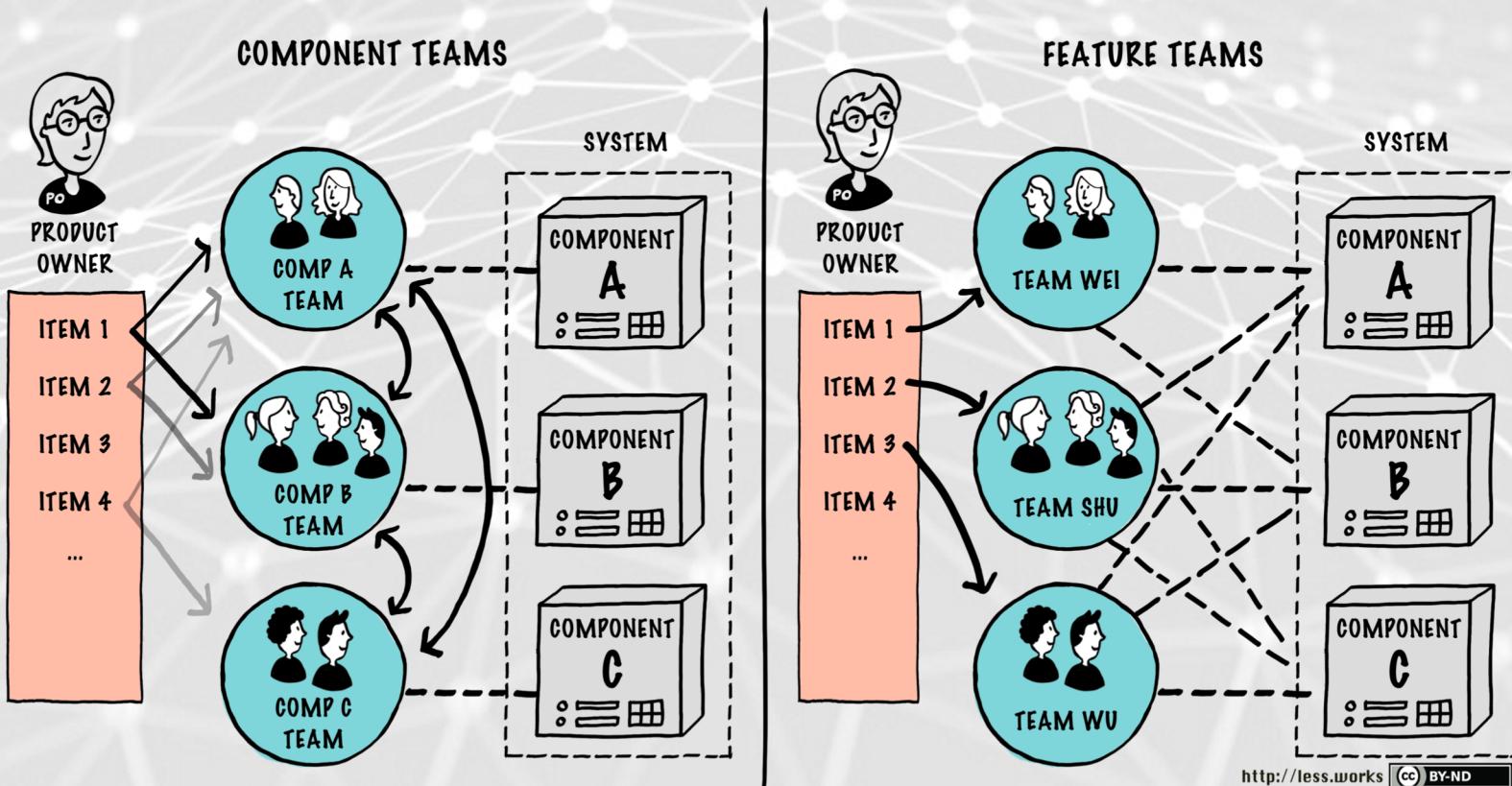
Culture, méthodes et organisation

- L'organisation typique ?
 - « Feature Teams »: orientées produits (fonctionnel)
 - « Component Teams »: orientées architecture (technique)



Culture, méthodes et organisation

- L'organisation typique ?





Culture, méthodes et organisation

- L'obsession de la mesure !

Si on n'a pas de mesure... on ne sait pas s'améliorer !



Culture, méthodes et organisation

- Quels métriques pour mesurer ?

- **Performance:**

- Automated test pass %
 - Deployment time
 - ...

- **Système:**

- Availability
 - Mean time to recovery (MTTR)
 - ...

- **Process:**

- Customer Ticket Volume
 - Usability
 - Defect Age
 - Lead Time (from Dev to Prod)
 - ...

- **Humain:**

- Bus Factor
 - Team Motivation Barometer
 - ...

Architecture et patterns



- Patterns de scalabilité et de disponibilité
- Patterns de déploiement
- Le Cloud : facilitateur de l'architecture DevOps

Architecture et patterns



- Patterns de scalabilité et de disponibilité

Comment modifier l'infrastructure, sans impact client ?

Architecture et patterns



- Patterns de scalabilité et de disponibilité

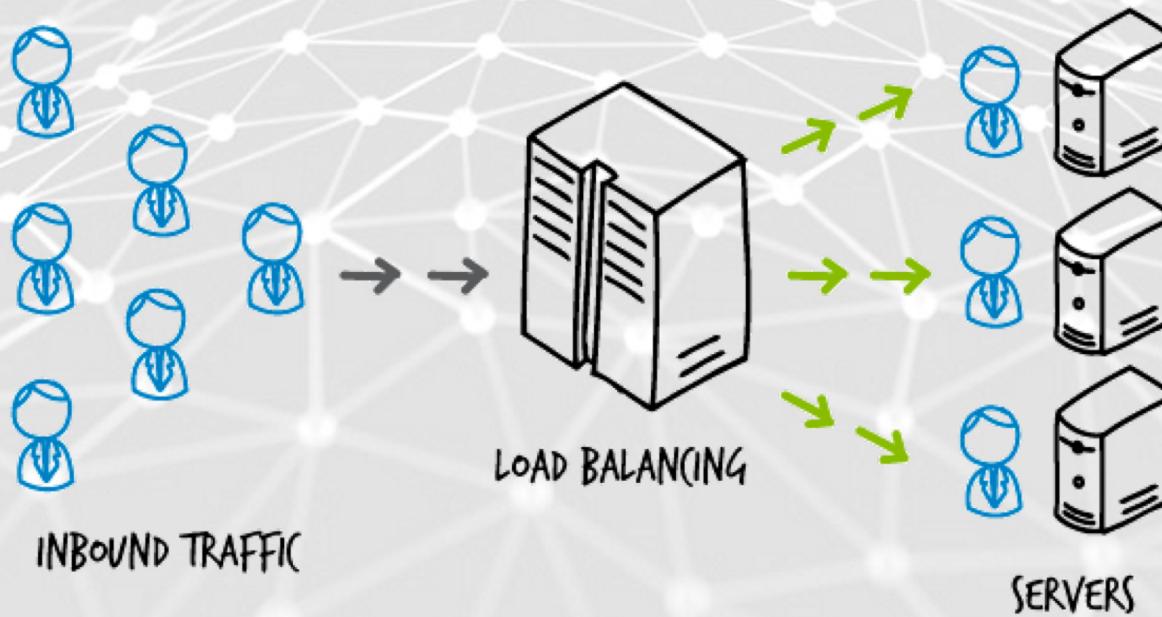
- Load Balancing

- High-Availability



Architecture et patterns

- Load Balancing



Architecture et patterns

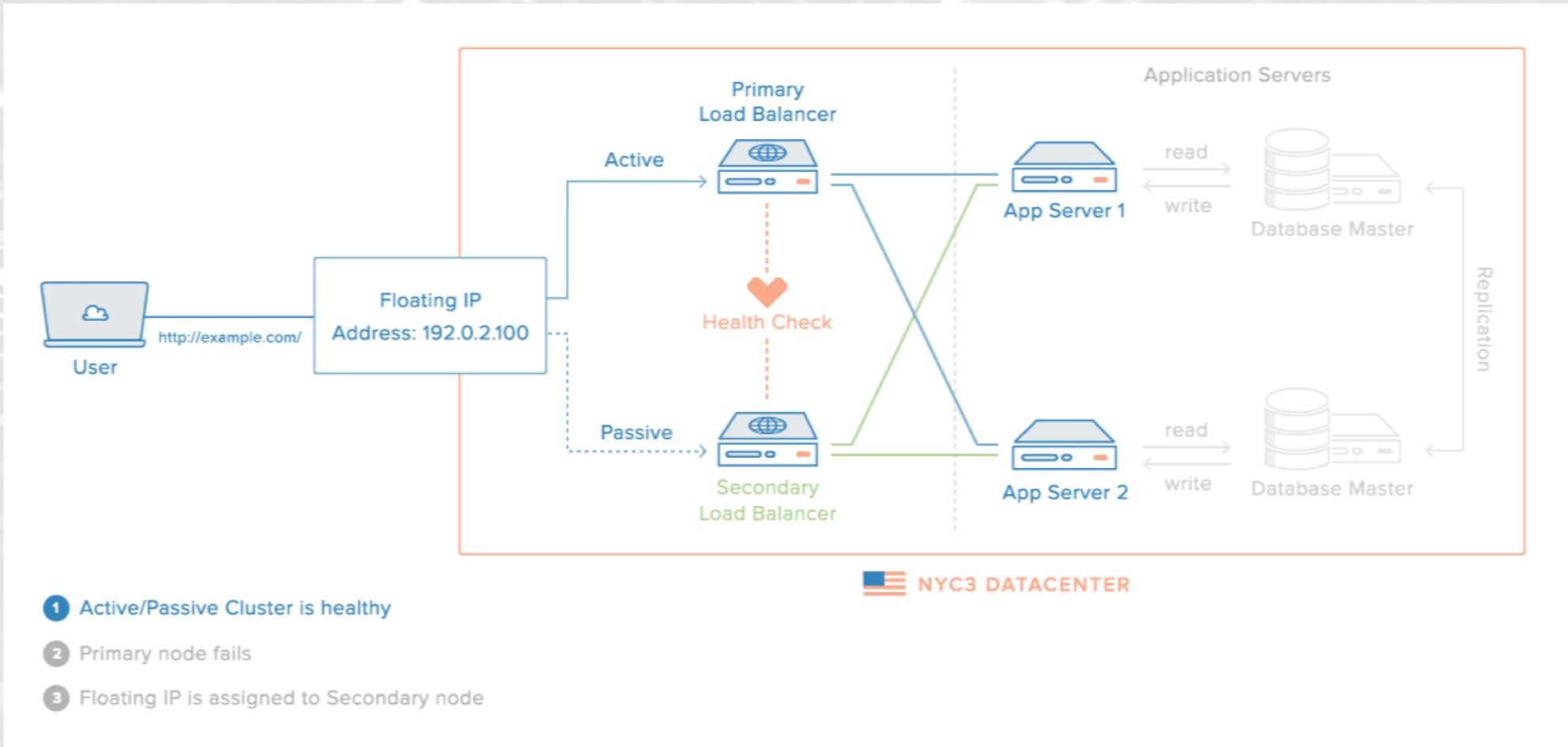


- Load Balancing
 - Round Robin
 - Weighted Round Robin
 - Least Connection
 - Agent-based Adaptative
 - ...



Architecture et patterns

- High-Availability



Architecture et patterns



- Patterns de déploiement

**Comment livrer de nouvelles fonctionnalités
rapidement et à chaud ?**

Architecture et patterns

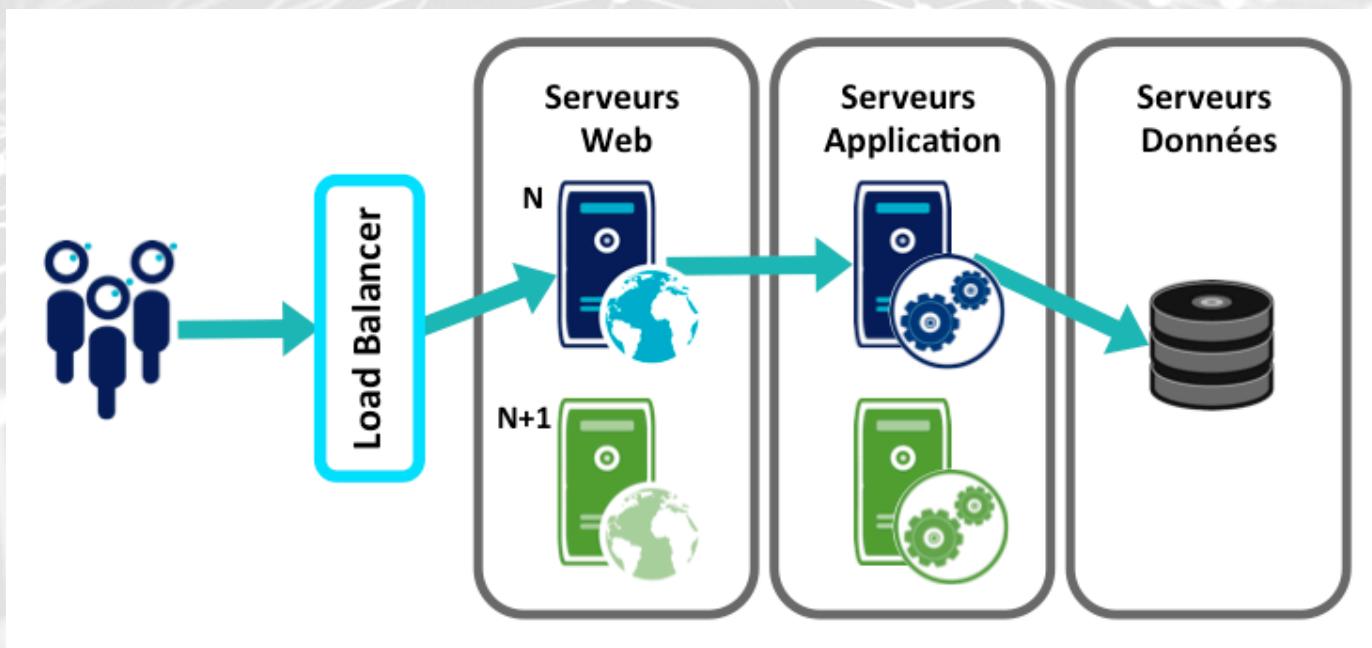


- Patterns de déploiement
 - Blue/Green
 - Canary
 - Feature Toggle & Dark Launch



Architecture et patterns

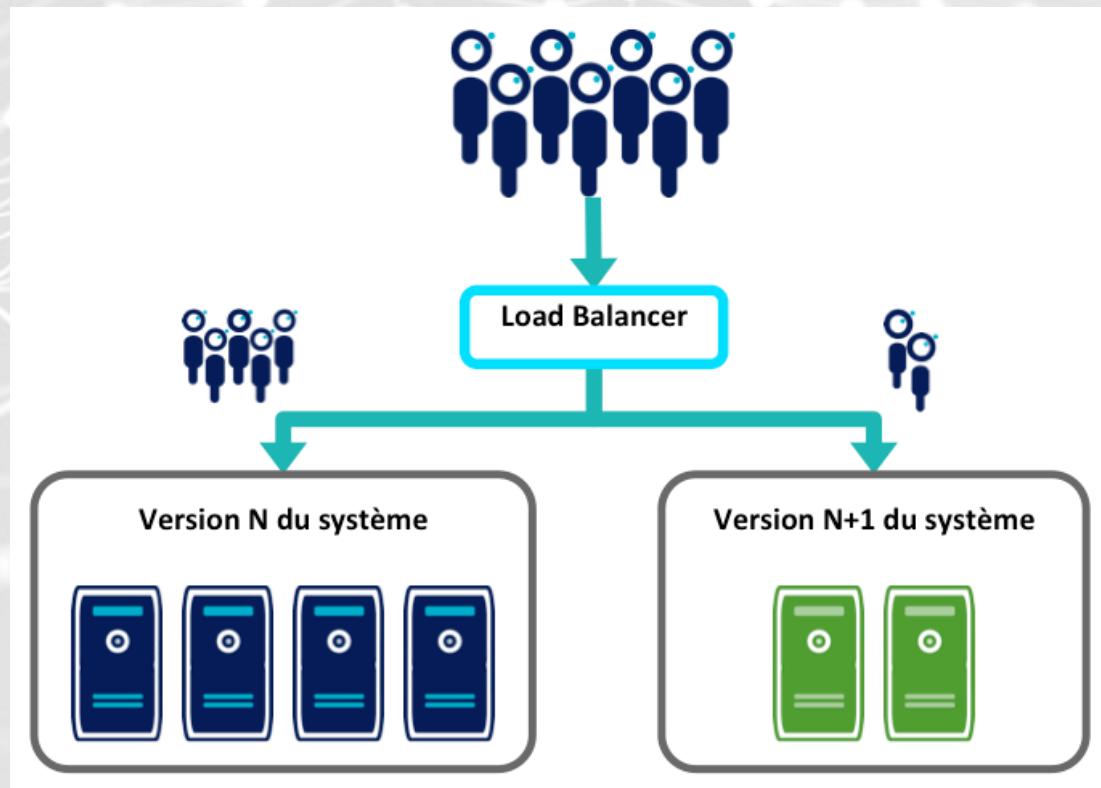
- Blue/Green Deployment





Architecture et patterns

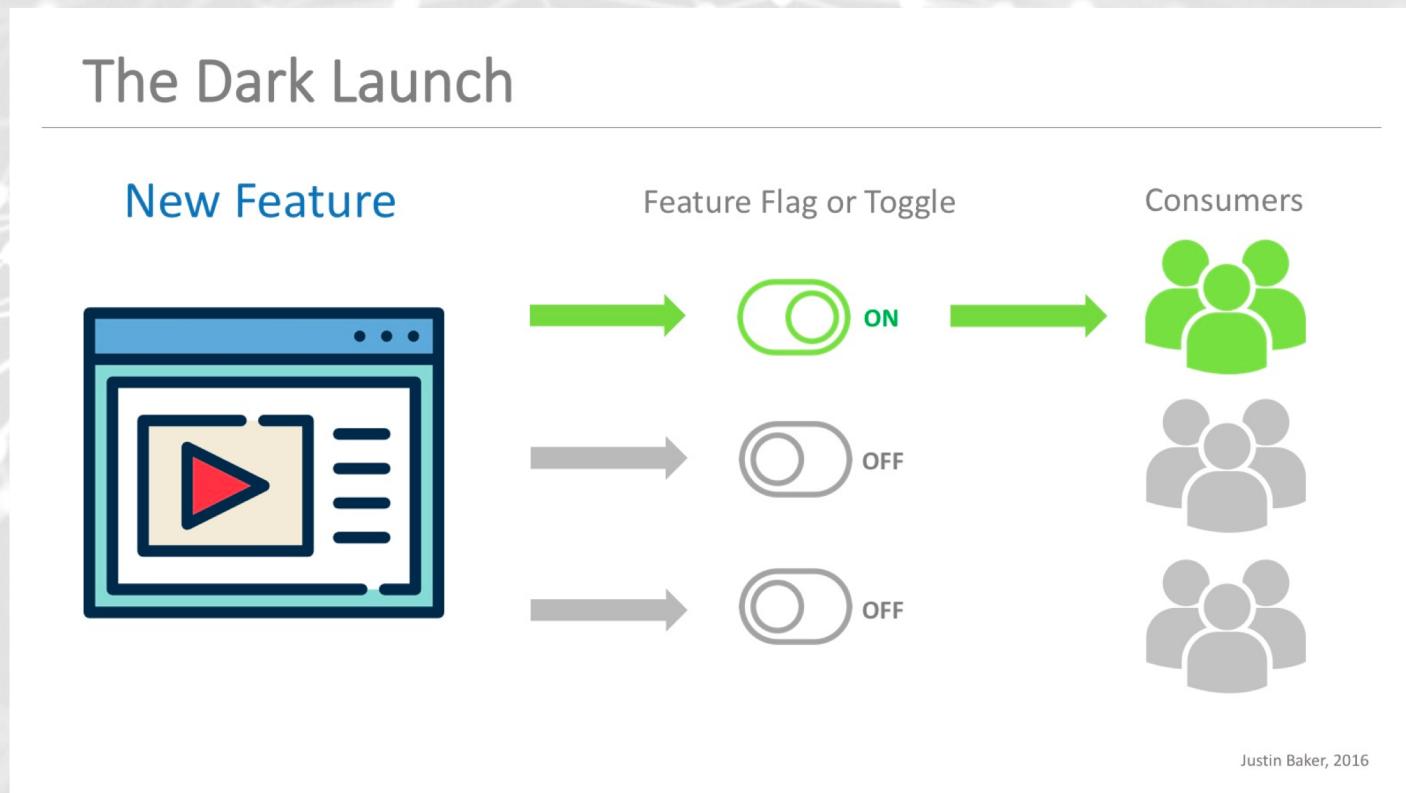
- Canary Deployment





Architecture et patterns

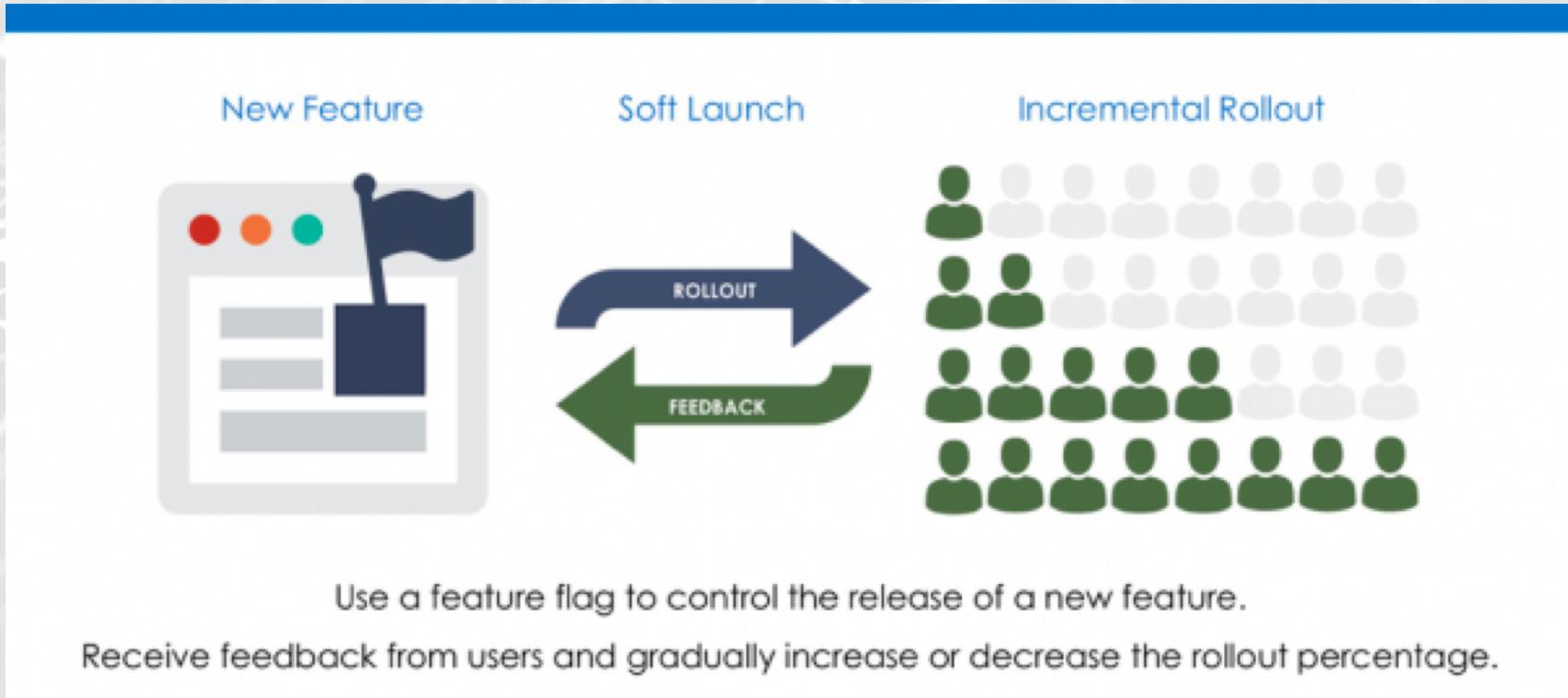
- Feature Toggle & Dark Launch





Architecture et patterns

- Feature Toggle & Dark Launch





Architecture et patterns

- Feature Toggle en pratique

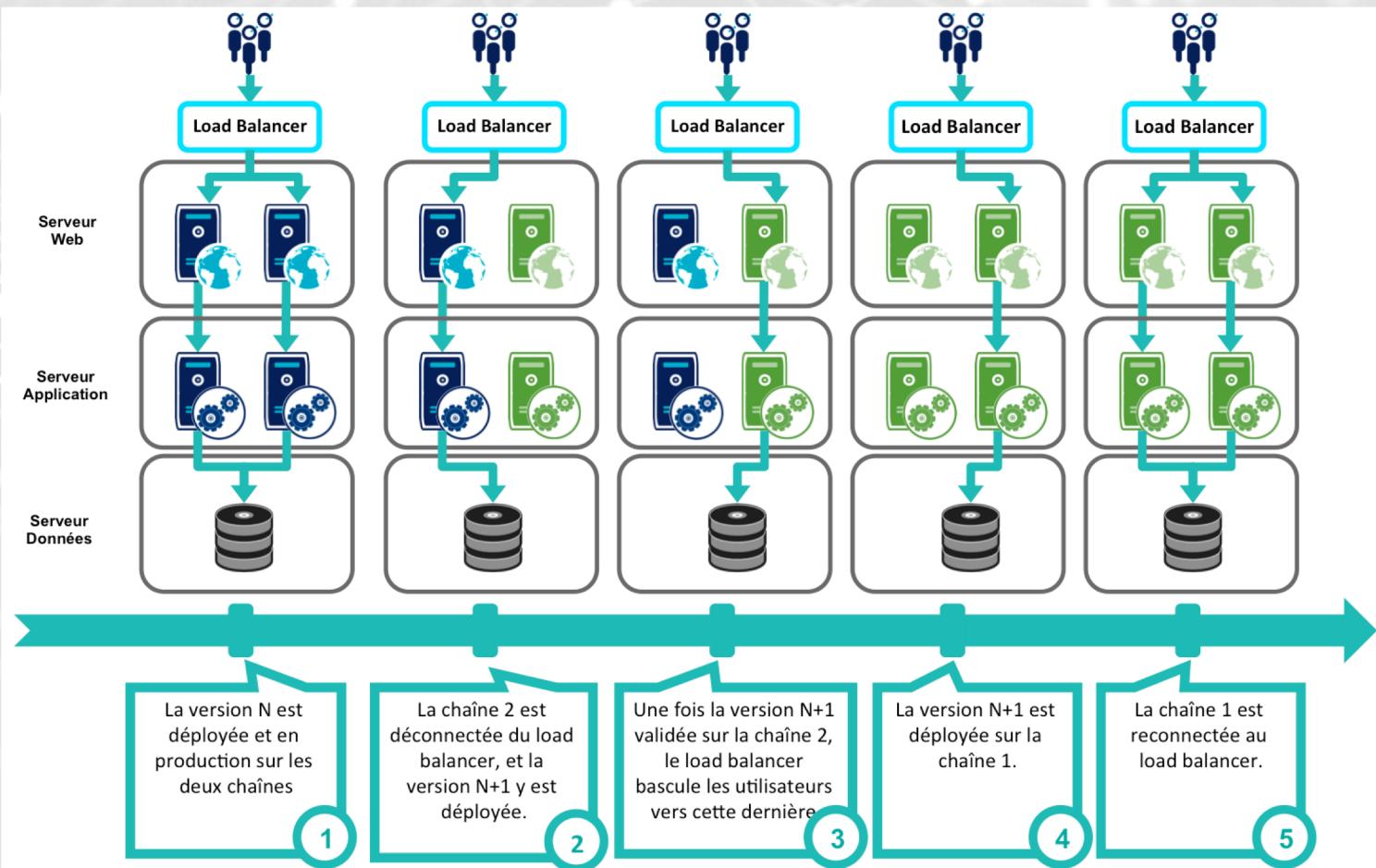
```
1 | if Feature.is_enabled('new_feature')
2 | # do something new
3 | else
4 | # do same as before
5 | end

1 | if Feature.is_enabled_for('new_feature', current_user)
2 | # do something new
3 | else
4 | # do same as before
5 | end
```



Architecture et patterns

- ZDD : mise en oeuvre



Architecture et patterns

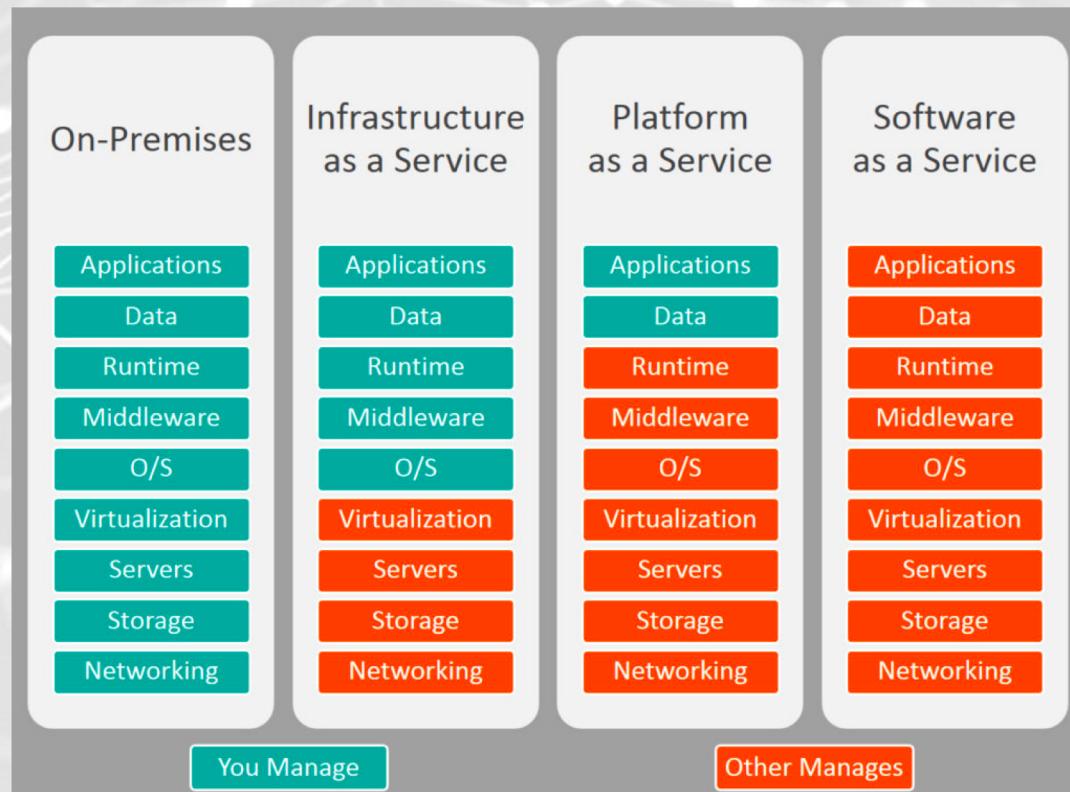


- Le Cloud : facilitateur de l'architecture DevOps
 - IaaS: Infrastructure as a Service
 - PaaS: Platform as a Service
 - SaaS: Software as a Service



Architecture et patterns

- On-Premises vs IaaS vs PaaS vs SaaS



L'infrastructure par le code



- Définition
- Cartographie des outils
- Docker et son écosystème



L'infrastructure par le code

- Définition

Considérer le provisionnement et la configuration de systèmes informatiques en tant que software !

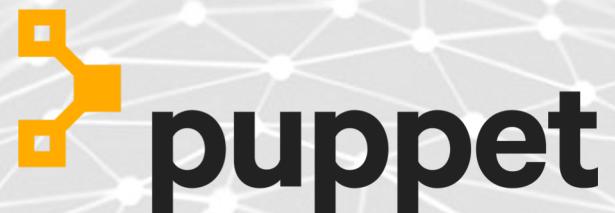


L'infrastructure par le code

- Nécessite de:
 - Faire l'usage de SCM (SVN, Git, ...)
 - Tester son IaC en continu
 - Refactorer et Intégrer en continu

L'infrastructure par le code

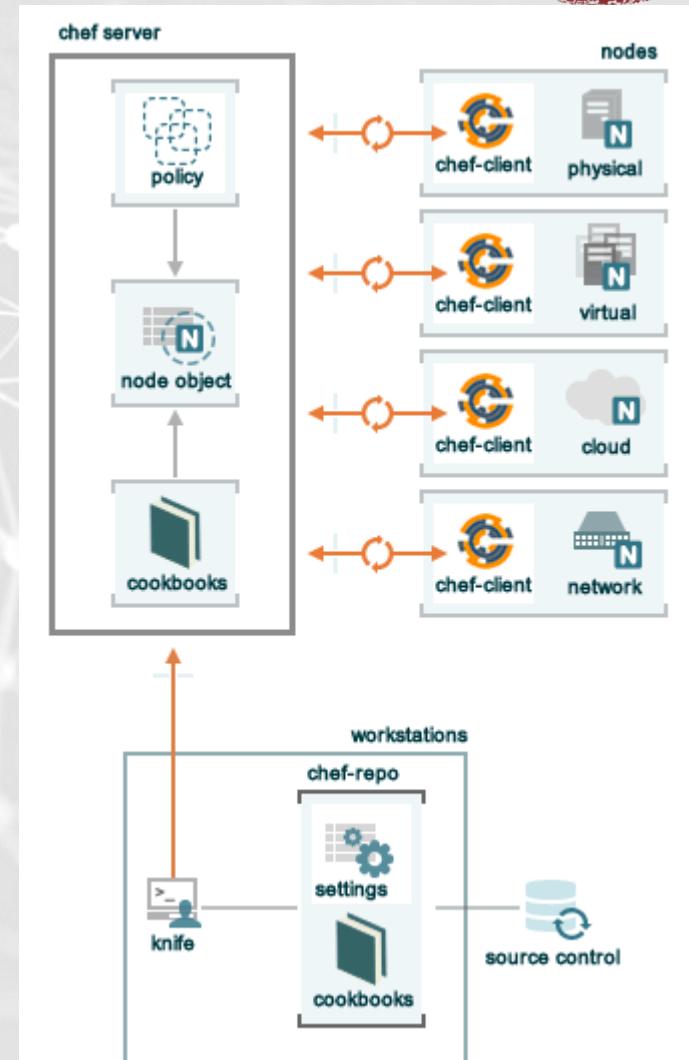
- Cartographie des outils



L'infrastructure par le code



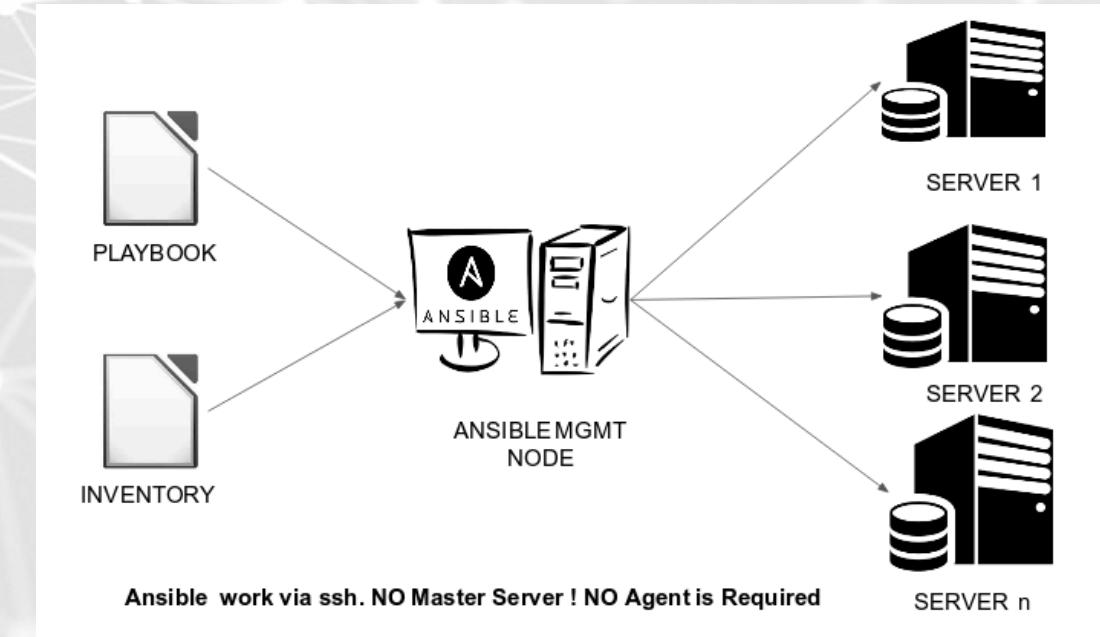
- Cartographie des outils
 - Ruby DSL
 - « Recipes » & « Cookbooks »
 - Approche procédurale pour décrire l'état du système
 - REST API (server – client)





L'infrastructure par le code

- Cartographie des outils
 - Syntaxe YAML
 - « Playbooks » & « Inventories »
 - Approche descriptive du système
 - SSH

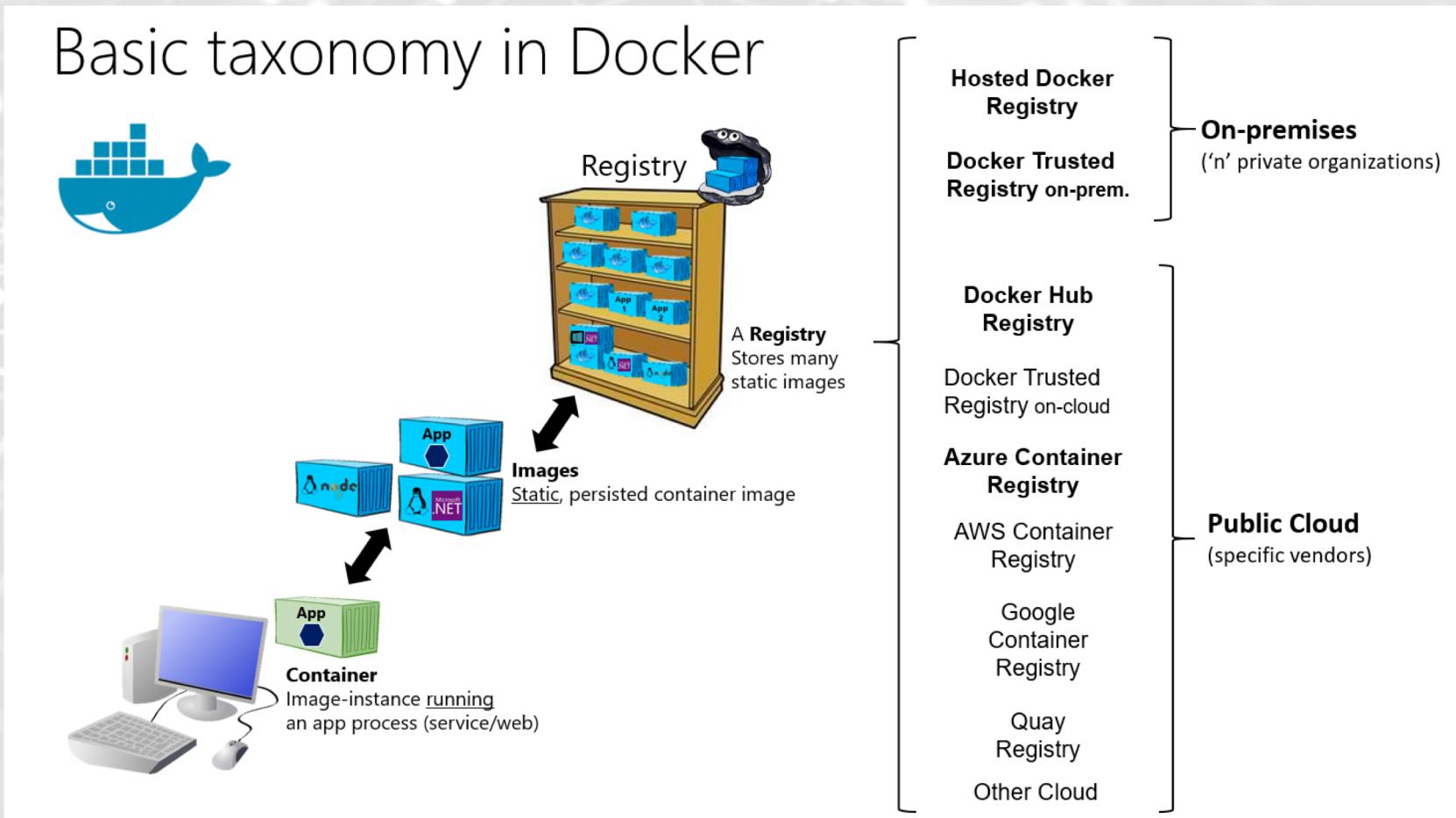


ANSIBLE



L'infrastructure par le code

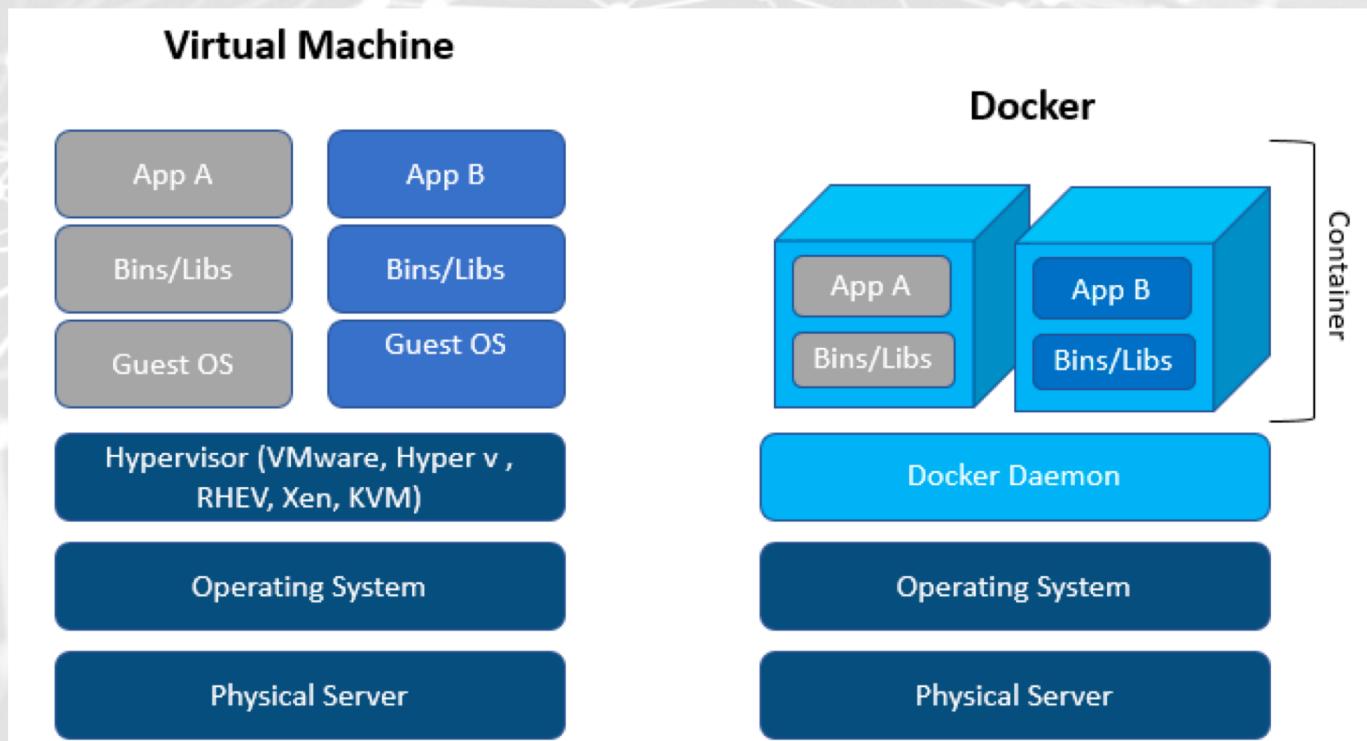
• Docker et son écosystème





L'infrastructure par le code

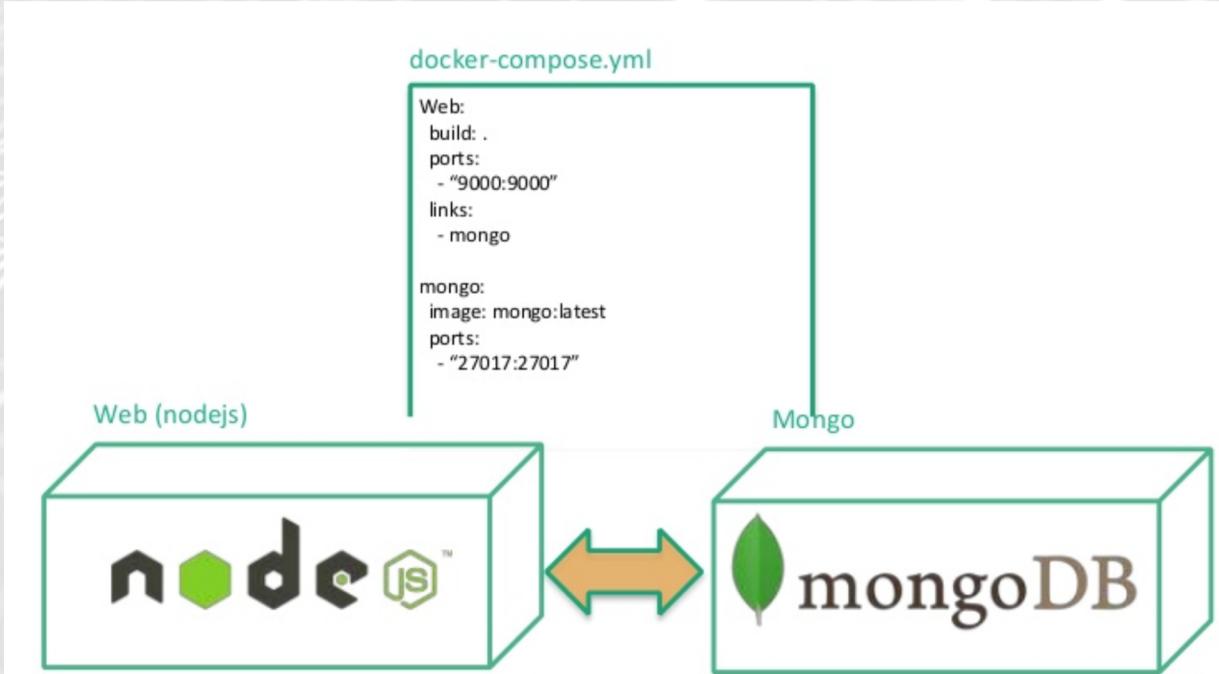
- Docker et son écosystème





L'infrastructure par le code

- Docker et son écosystème





L'infrastructure par le code

- Docker et son écosystème





L'infrastructure par le code

- Docker et son écosystème
 - L'orchestration c'est:
 - Provisionnement et déploiement de conteneurs
 - Redondance et disponibilités des conteneurs
 - Scale up/down à chaud
 - Monitoring
 - ...

Construction et déploiement continu



- Définition
- Usine du « continu »



Construction et déploiement continu

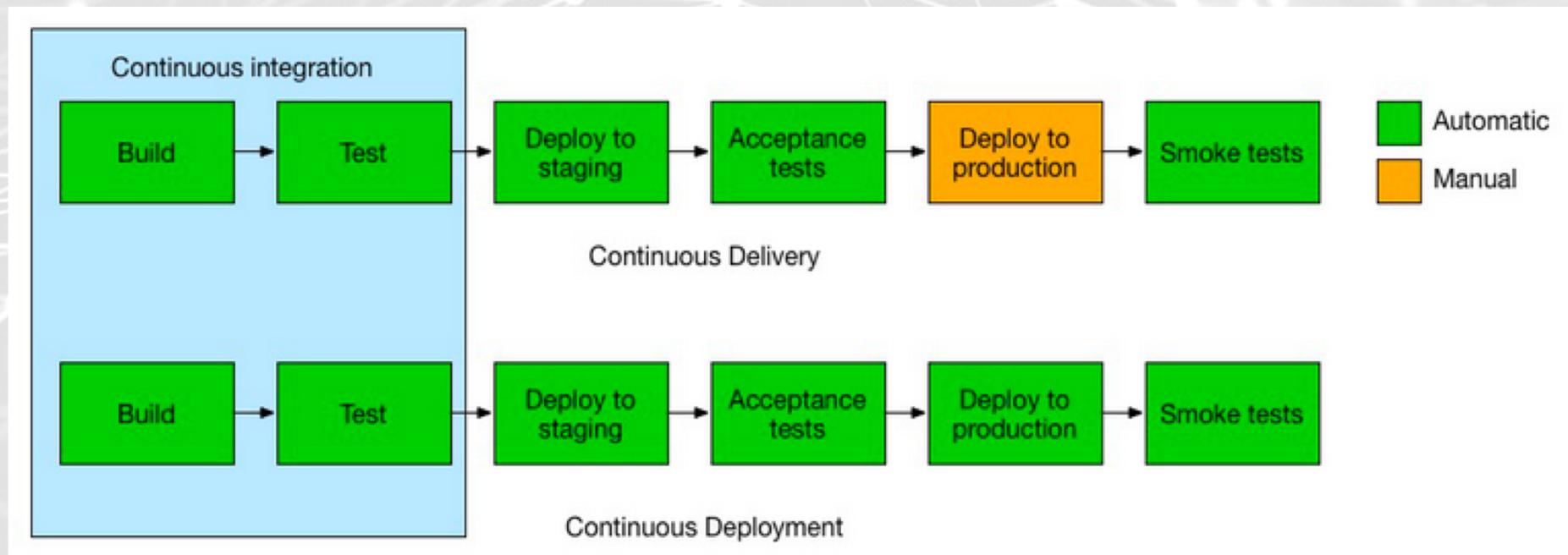
- Définition

Intégrer et déployer son code en continu: à chaque « commit », on lance une batterie de tests donnant suite à un/des déploiement(s) !



Construction et déploiement continu

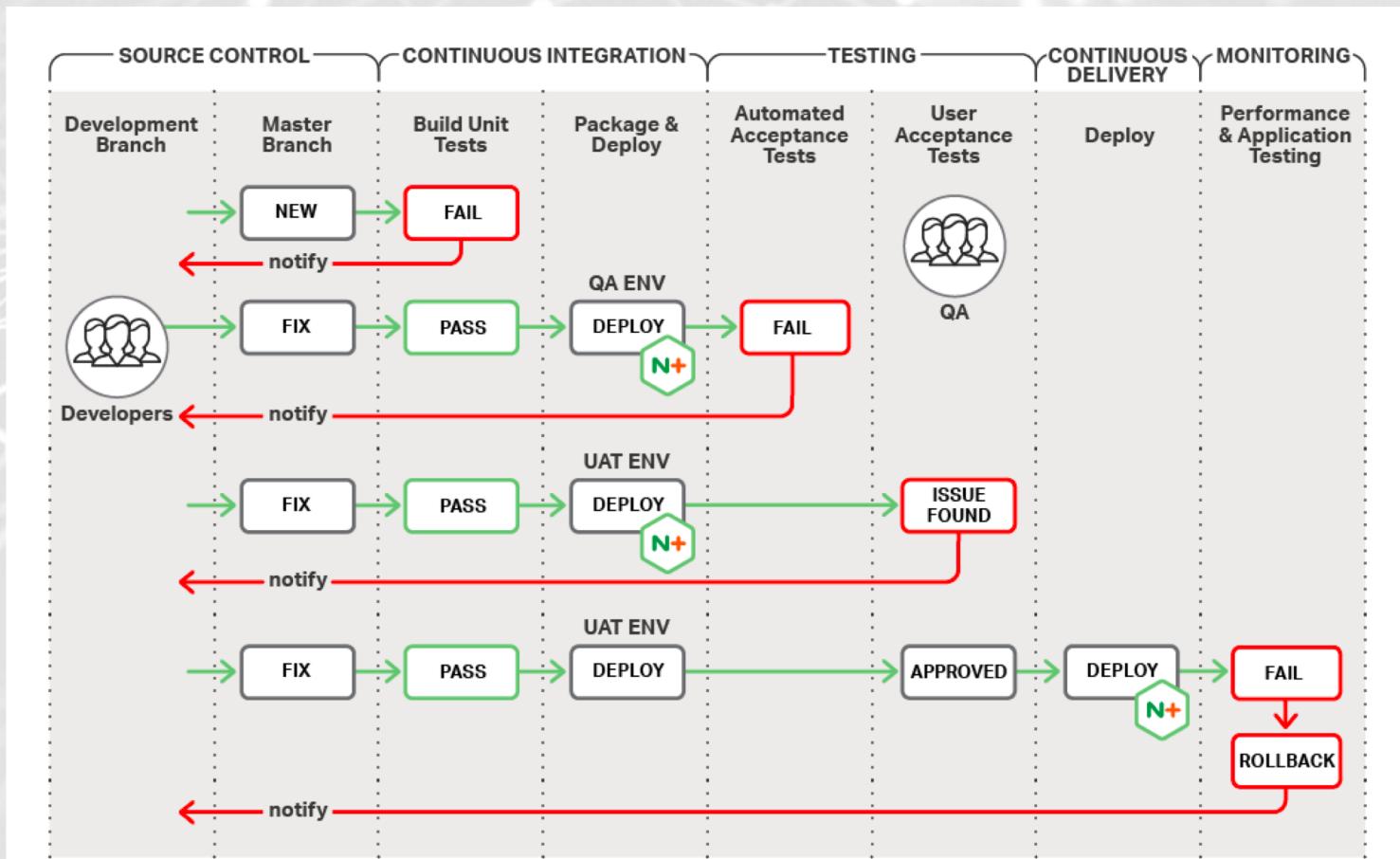
- Définition





Construction et déploiement continu

- Usine du « continu »



Construction et déploiement continu

- Quelques outils



Jenkins



Travis CI



CI CD

 Bamboo





Q & A

C'est votre tour ! :P