

"DevOps ne se résume pas à l'automatisation de la même façon que l'astronomie ne se résume pas aux télescopes. »

Christopher Little, The DevOps Handbook

Module 6 : automatisation et architecture des DevOps Toolchains

PERIODIC TABLE OF DEVOPS TOOLS (V3)

Os Open Source

Fr Free

Fm Freemium

Pd Paid

En Enterprise

Source Control Mgmt.

Database Automation

Continuous Integration

Testing

Configuration

Deployment

Containers

Release Orchestration

Cloud

AIOps

Analytics

Monitoring

Security

Collaboration

1 Gl GitLab																	2 Sp Splunk	
3 Gh GitHub	4 Dt Datacal																	10 Sg Sumo Logic
11 Sv Subversion	12 Db DBMaestro																	18 Fd Fluentd
19 Cw ISPW	20 Dp Delphix	21 Jn Jenkins	22 Cs Codeship	23 Fn FitNesse	24 Ju JUnit	25 Ka Karima	26 Su SoapUI	27 Ch Chef	28 Tf Terraform	29 XLd XebiaLabs XL Deploy	30 Ud UrbanCode Deploy	31 Ku Kubernetes	32 Cc CA CD Director	33 Pr Plutora Release	34 Al Alibaba Cloud	35 Os OpenStack	36 Ps Prometheus	
37 At Artifactory	38 Rg Redgate	39 Ba Bamboo	40 Vs VSTS	41 Se Selenium	42 Jm JMeter	43 Ja Jasmine	44 Sl Sauce Labs	45 An Ansible	46 Ru Rudder	47 Oc Octopus Deploy	48 Go GoCD	49 Ms Mesos	50 Gke GKE	51 Om OpenMake	52 Cp AWS CodePipeline	53 Cy Cloud Foundry	54 It ITRS	
59 Nx Nexus	60 Fw Flyway	61 Tr Travis CI	62 Tc TeamCity	63 Ga Gatling	64 Tn TestNG	65 Tt Tricentis Tosca	66 Pe Perfecto	67 Pu Puppet	68 Pa Packer	69 Cd AWS CodeDeploy	70 Ec ElectricCloud	71 Ra Rancher	72 Aks AKS	73 Rk Rkt	74 Sp Spinnaker	75 Ir Ironclad	76 Mg Moogsoft	
73 Bb BitBucket	74 Pf Perforce	75 Cr CircleCI	76 Cb AWS CodeBuild	77 Cu Cucumber	78 Mc Mocha	79 Lo Locust.io	80 Mf Micro Focus UFT	81 Sa Salt	82 Ce CFEngine	83 Eb ElasticBox	84 Ca CA Automate	85 De Docker Enterprise	86 Ae AWS ECS	87 Cf Codefresh	88 Hm Helm	89 Aw Apache OpenWhisk	90 Ls Logstash	

XebiaLabs

Enterprise DevOps

Follow @xebialabs

91 XLi XebiaLabs XL Impact	92 Ki Kibana	93 Nr New Relic	94 Dt Datadog	95 Ad AppDynamics	96 El ElasticSearch	97 Ni Nagios	98 Zb Zabbix	99 Zn Zenoss	100 Cm Checkmarx SAST	101 Wp Signal Sciences WPP	102 Bd BlackDuck	103 Sr SonarQube	104 Hv HashiCorp Vault	105 Os
106 Sw ServiceNow	107 Jr Jira	108 Tl Trello	109 Sk Stack	110 St Stride	111 Cn CollabNet VersionOne	112 Ry Remedy	113 Ac Agile Central	114 Og OpsGenie	115 Pd Pagerduty	116 Sn Snort	117 Tw Tripwire	118 Ck CyberArk	119 Vc Voracode	120 Ff Fortify SCA

Avantages de l'automatisation

L'automatisation permet :

- Délais plus courts
- Mises en production plus fréquentes
- Mises en production moins houleuses
- Moins d'erreurs
- Meilleure qualité
- Sécurité améliorée et atténuation des risques
- Rétablissements plus rapides
- Satisfaction de l'entreprise et des clients

L'automatisation donne les tâches de routine à l'ordinateur et permet aux individus de :

- Analyser les preuves
- Résoudre des problèmes
- Prendre des décisions en fonction du feedback
- Utiliser leurs compétences, leur expérience et leur jugement

“Vos outils seuls ne vous feront pas réussir.”

Patrick Debois

Terminologie importante

- **Artefact**

Tout élément d'un projet de développement logiciel, y compris la documentation, les plans de test, les images, les fichiers de données et les modules exécutables

- **Interface de programmation d'application (API)**

Ensemble de protocoles utilisés pour créer des applications pour un système d'exploitation spécifique ou comme interface entre des modules ou des applications

- **Microservices**

Une architecture logicielle composée de modules plus petits qui interagissent via des API et peut être mise à jour sans affecter l'ensemble du système.
Ceci est connu comme un couplage lâche

- **Virtualisation de système d'exploitation (OS)**

Une méthode pour diviser un serveur en plusieurs partitions appelées "conteneurs" ou "environnements virtuels" afin d'empêcher les applications d'interférer les unes avec les autres

- **Conteneurs**

Une façon de regrouper des logiciels dans des packages légers, autonomes et exécutables, incluant tout le nécessaire pour l'exécuter (code, exécution, outils système, bibliothèques système, paramètres) aux fins de développement, d'envoi et de déploiement.

- **Open source**

Logiciel distribué avec son code source afin que les organisations d'utilisateurs finaux et les fournisseurs puissent le modifier à leurs propres fins

- **Machine learning**

Analyse de données utilisant des algorithmes qui apprennent des données

Cloud, conteneurs et microservices

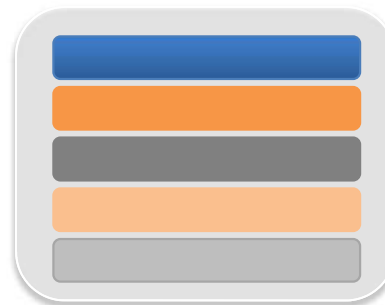
Cloud computing : pratique consistant à utiliser des serveurs distants hébergés sur Internet, pour héberger des applications plutôt que des serveurs locaux dans un centre de données privé.



docker est un outil conçu pour faciliter la création, le déploiement et l'exécution d'applications à l'aide de conteneurs. Les conteneurs permettent à un développeur d'embarquer une application avec toutes les parties dont il a besoin, telles que des bibliothèques et autres dépendances, et de l'envoyer dans un seul package.



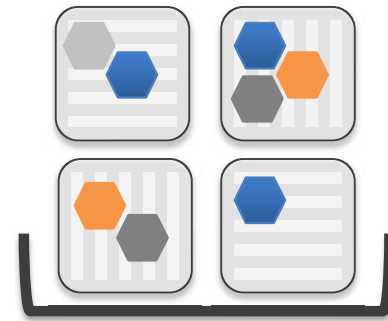
Les équipes qui adoptent les caractéristiques essentielles du Cloud ont **24 fois** plus de chances d'être des élites.



Monolithic



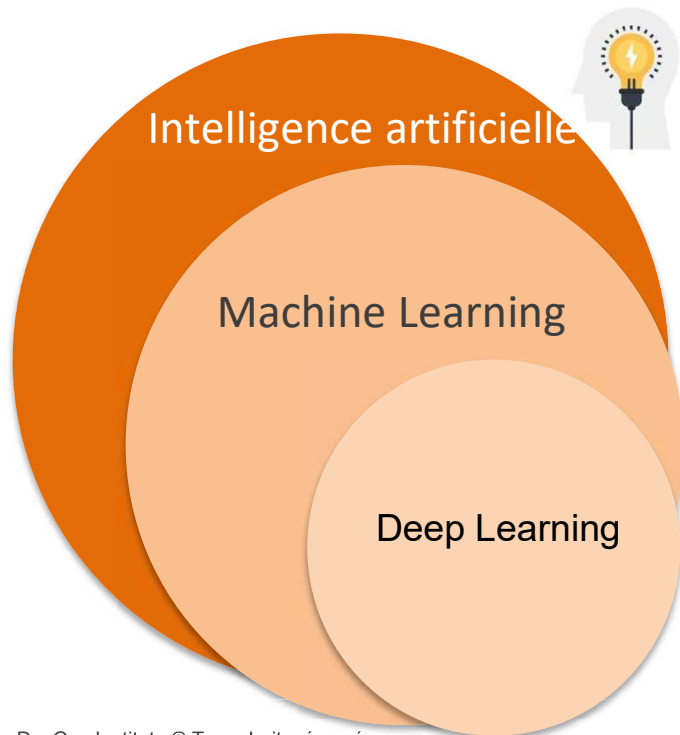
kubernetes est un système open source permettant de gérer des applications conteneurisées sur plusieurs hôtes, fournissant des mécanismes de base pour le déploiement, la maintenance et la mise à l'échelle des applications



Conteneurs & Microservices

IA & Machine Learning

Donner aux ordinateurs la possibilité "d'apprendre" avec des données, sans être explicitement programmé.



- Les organisations collectent plus de données que jamais
- Il est difficile d'extraire pleinement la valeur de ces données.
- La science des données est une discipline de plus en plus populaire
- L'intelligence artificielle et l'apprentissage automatique permettent une analyse prédictive
- Peut trouver des tendances et des corrélations que les humains n'auraient pu trouver
- Augmente la contribution humaine
- Augmente la productivité
- Boucles de feedback automatisées

Définition : Analyse de données qui utilise des algorithmes pour apprendre des données.

Pratiques d'automatisation DevOps

Une philosophie de toolchain implique l'utilisation d'un ensemble intégré d'outils complémentaires spécifiques aux tâches pour automatiser les processus de livraison et de déploiement de bout en bout.

- Toolchain (par opposition à une solution à fournisseur unique)
- Outils partagés
- En libre service
- Prévoir l'architecture du logiciel de manière à permettre :
 - Automatisation des tests
 - Surveillance
- Infrastructure en tant que code
- Expérimentation

Eviter les outils qui renforcent les silos !

L'automatisation de la communication et de la collaboration

Des outils et des plates-formes innovants facilitent et accélèrent la communication et la collaboration à travers le spectre Dev et Ops.

Comment	Outils
<ul style="list-style-type: none">• Émettre des alertes et des alarmes• Améliorer la réponse• Fournir des aperçus de mises à jour de statut• Améliorer le flux de travail• Améliorer le flux d'informations• Permettre la collaboration virtuelle• Permettre l'auto-organisation d'équipe cross-fonctionnelle par le partage de compétences.	<ul style="list-style-type: none">• Plateformes de communication• Tableaux de bord• Tableaux Kanban• Groupes de discussion (ChatOps)• Flux de travail et outils de gestion de projet• Partage de documents• Wikis et systèmes de gestion des connaissances• Outils ITSM• Outils sociaux• Backlogs partagés

Premières étapes pour améliorer l'automatisation DevOps

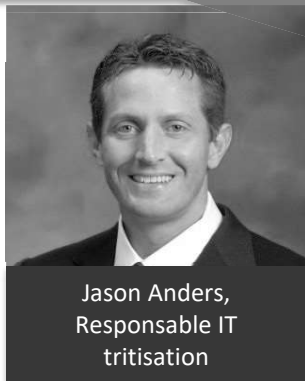
- Définir l'architecture avant d'automatiser
- Évaluer vos outils existants et vos capacités d'automatisation
- Simplifier d'abord – ne pas automatiser les processus défectueux
- Identifier les lacunes critiques
- Rechercher des fournisseurs qui peuvent répondre à vos exigences
- Automatiser le travail à haute valeur ajoutée, répétitif et sujet aux erreurs
- Optimiser les goulots d'étranglement du flux de travail et la communication
- Améliorer les pratiques de surveillance et de notification automatisées
- S'attendre à un processus itératif - votre toolchain évoluera avec le temps

Ne sous-estimez pas l'effort et le coût de la construction des toolchains à partir d'applications open source. L'open source n'est pas nécessairement gratuit. Cela signifie que vous pouvez modifier la source pour répondre à vos besoins.

ETUDE DE CAS : Fannie Mae

« Nous avons choisi une mise en adoption application par application. Cela fait maintenant environ un an et demi et nous avons atteint un point de masse critique. Nous pouvons être beaucoup plus flexibles, beaucoup plus dynamiques et fournir à nos clients et partenaires les outils dont ils ont besoin pour interagir beaucoup plus facilement avec nous. Comme tout le monde, nous devons mettre les idées en production beaucoup plus rapidement que ce que nous faisons aujourd'hui. »

“Nous faisons en sorte qu’il soit facile de travailler avec nous”

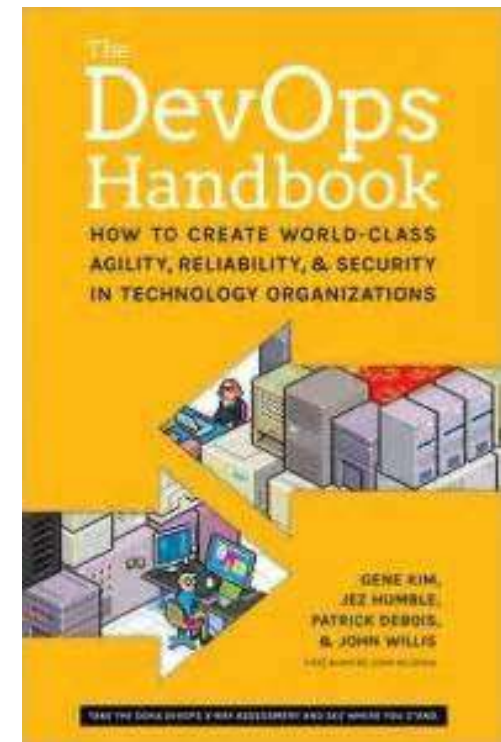


Avantages

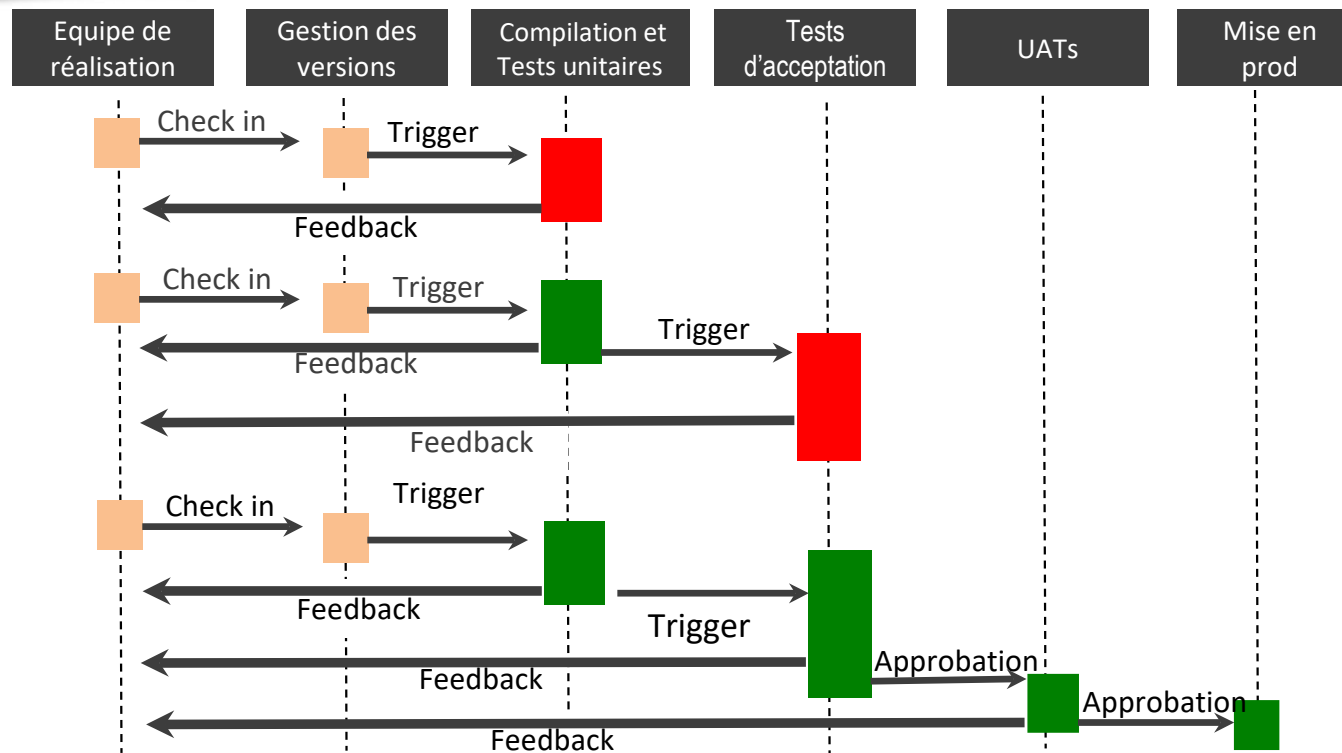
- Déploiement de 3 jours réduit à 45 minutes
- Déployé sept ou huit fois par jour
- 40 à 75% d'économie sur les coûts de stockage grâce à la virtualisation des données

« L'un des axes afin d'obtenir des résultats rapides pilotés par le marché consiste, pour les opérations, à créer des plates-formes et des outils pouvant être utilisés par toutes les équipes de développement afin de devenir plus productifs... une plateforme qui s'appuie sur un système de gestion de version de référence, associé à des bibliothèques de sécurité vérifiées, un pipeline de déploiement qui exécute automatiquement des outils d'analyse de sécurité et de qualité du code, qui déploie nos applications dans des environnements connus et performants sur lesquels des outils de surveillance de la production sont déjà installés. »

Le DevOps Handbook



Le pipeline de déploiement



Le pipeline de déploiement est un processus automatisé permettant de gérer tous les changements, du check-in à la mise en production. Les toolchains couvrent des silos et automatisent le pipeline de déploiement.

Source : *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*

DevOps Toolchains

La DevOps toolchain est composée des outils nécessaires pour prendre en charge l'intégration continue, la livraison continue ainsi que de mise en production continue et les opérations. (Gartner)

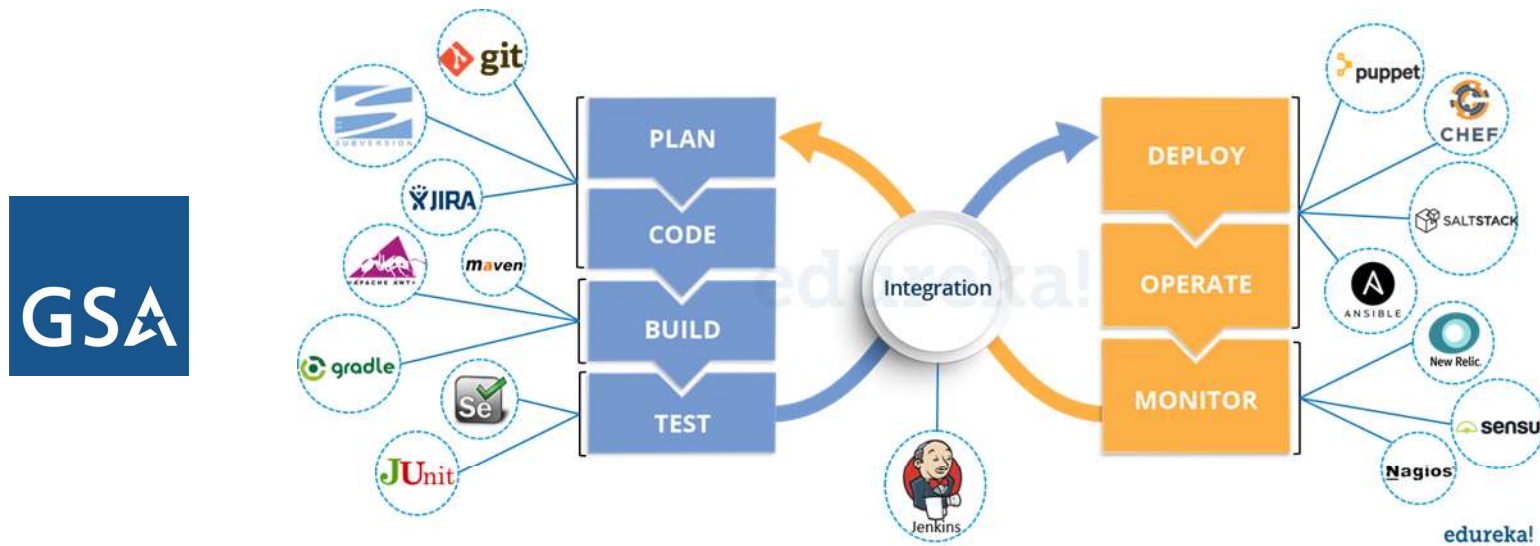
- Les toolchains automatisent les tâches dans le pipeline de déploiement
- Chaque élément de la toolchain sert un objectif spécifique
- Les applications dans les toolchains sont connectées via des API
- Elles ne doivent pas nécessairement être homogènes ou provenir d'un seul fournisseur
- Les toolchains sont généralement construites autour d'écosystèmes sources ouverts
- Exiger une conception architecturale pour assurer l'interopérabilité et la cohérence

Le pipeline de déploiement est un processus automatisé permettant de gérer tous les changements, du check-in à la mise en production. Les toolchains embrassent des silos et automatisent le pipeline de déploiement.

Comment les DevOps toolchains doivent-elles s'interfacer avec des outils opérationnels tels que des applications de surveillance ou de support ?

Exemple DevOps Toolchain (gouvernement américain - GSA)

Il existe de nombreux outils DevOps open-source ou non avec des écosystèmes dynamiques.



La manière dont ces outils sont adaptés et intégrés dans votre pipeline de déploiement déterminera leur valeur.

Éléments dans une DevOps Toolchain

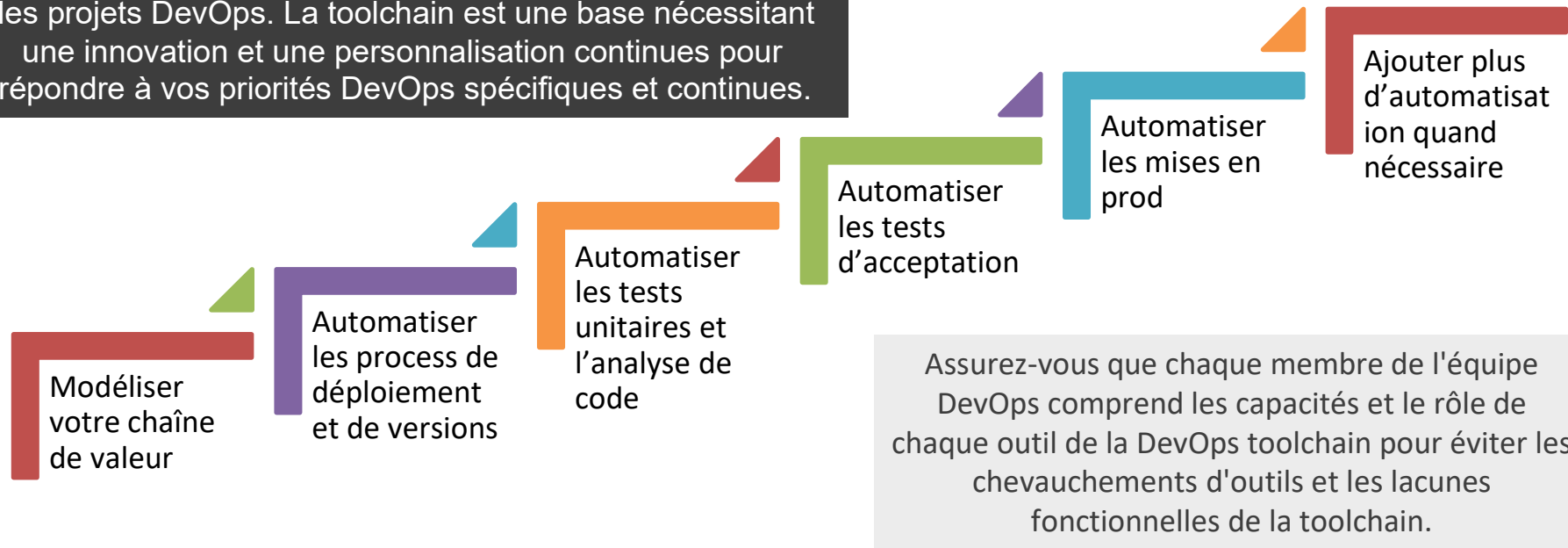
- Le pipeline de déploiement divise le cycle de vie de livraison du logiciel en étapes logiques
- Chaque étape fournit :
 - L'occasion de vérifier la qualité des nouvelles fonctionnalités sous un angle différent
 - Du feedback rapide à l'équipe
 - Une visibilité quant au flux des changements
- Les DevOps toolchains fournissent les fonctionnalités nécessaires pour automatiser et accélérer chaque étape

Éléments Toolchain typiques :

- Gestion des exigences
- Orchestration et visualisation
- Gestion des versions
- Intégration continue et versions
- Gestion des artefacts
- Virtualisation de conteneurs et de systèmes d'exploitation
- Automatisation des tests et de l'environnement
- Configuration et déploiement du serveur
- Gestion de la configuration du système
- Alertes et alarmes
- Monitoring

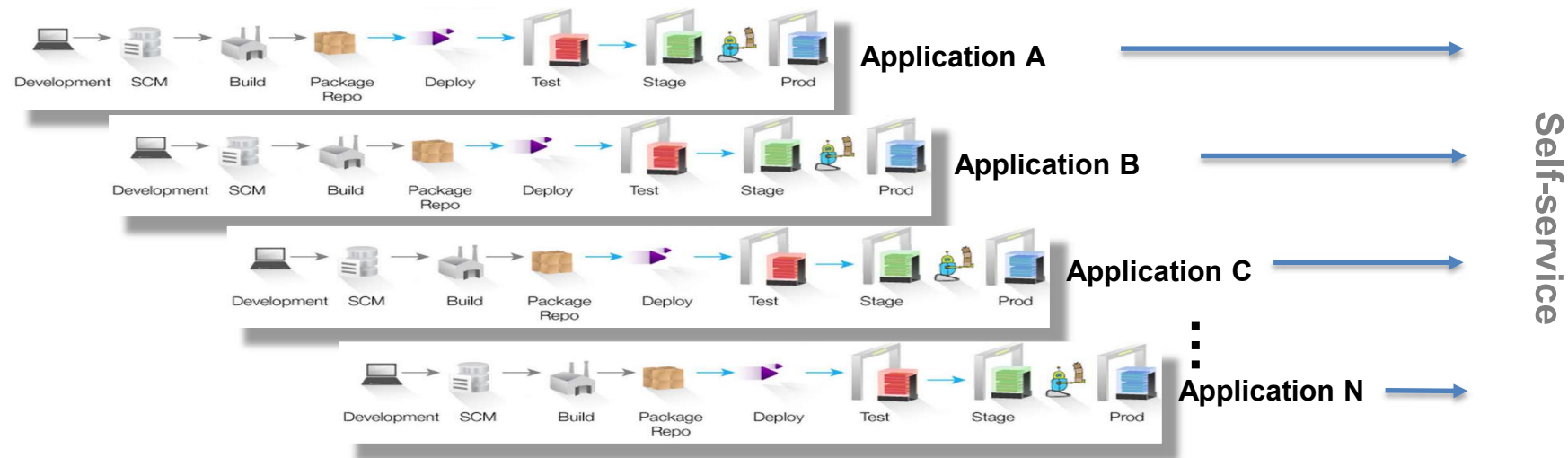
Construire votre DevOps Toolchain petit à petit

Ne créez pas une toolchain définitive qui s'applique à tous les projets DevOps. La toolchain est une base nécessitant une innovation et une personnalisation continues pour répondre à vos priorités DevOps spécifiques et continues.



Source : Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation par Jez Humble et Dave Farley

Des applications professionnelles multiples nécessitent des toolchains multiples



Source : Sanjeev Sharma, IBM

Éviter de créer davantage de silos entre les pipelines en adoptant une approche d'architecture d'entreprise