**Table des matières**

[1. Devops 2](#_Toc121245253)

[1.1. Les 8 avantages de la démarche devops 2](#_Toc121245254)

[1.2. Les débouchés du devops sur le marché de l’emploi 2](#_Toc121245255)

[1.3. Les outils indispensables 3](#_Toc121245256)

[Le développement CI/CD 3](#_Toc121245257)

[GitHub 3](#_Toc121245258)

[SourceForge 3](#_Toc121245259)

[Les outils CI/CD alternatifs 3](#_Toc121245260)

[La gestion des configurations 3](#_Toc121245261)

[Ansible 3](#_Toc121245262)

[Jenkins 3](#_Toc121245263)

[Infrastructure as code 4](#_Toc121245264)

[Terraform 4](#_Toc121245265)

[Doper la production 4](#_Toc121245266)

[Docker 4](#_Toc121245267)

[Kubernetes 4](#_Toc121245268)

[Le monitoring de l’infrastructure 4](#_Toc121245269)

[Grafana 4](#_Toc121245270)

[ELK (Elasticsearch, Logstash, Kibana) 4](#_Toc121245271)

[Prometheus 4](#_Toc121245272)

## 1. Devops

DevOps est un ensemble de pratiques qui met l’accent sur la collaboration et la communication entre les développeurs de logiciels et les professionnels des opérations informatiques, en automatisant le processus de livraison de logiciels et les changements d’infrastructure. Le terme DevOps est né de l’union du « développement » et des «opérations» dont l’objectif est favoriser une meilleure communication entre les deux équipes.

DevOps vise à créer une culture et un environnement dans lesquels la conception, les tests et la diffusion de logiciels peuvent être réalisés rapidement, fréquemment et efficacement.

DevOps n’est pas seulement une méthodologie, c’est une véritable philosophie de travail.

### 1.1. Les 8 avantages de la démarche devops

1. Amélioration de la qualité du code, des produits et des services (réduction des anomalies, taux de réussite des changements plus important, etc.)
2. Efficacité accrue
3. Amélioration du délai de mise en place sur le marché
4. Meilleur alignement entre l’informatique et les métiers
5. Des versions de plus petite taille fournies très rapidement et très fréquemment
6. Amélioration de la productivité, satisfaction du client, satisfaction du personnel
7. Moins de risques et moins de retours arrière
8. Réduction des coûts à long terme

### 1.2. Les débouchés du devops sur le marché de l’emploi

1. Le poste d’ingénieur en informatique industrielle
2. Les [compétence DevOps](https://www.qrpinternational.fr/blog/methode-agile/les-competences-devops/)
3. Consultant fonctionnel
4. Agent de contrôle qualité logicielle
5. Ingénieur IT applicatif

### 1.3. Les outils indispensables

Les outils indispensables sont catégorisés comme suit :

1. Développement ci/cd
2. Gestion des configurations
3. Infrastructure as code
4. Monitoring d’infrastructure

## Le développement CI/CD

Le développement CI/CD traduit la phase de test qui challenge le code. L’acronyme CI/CD signifie « Intégration Continue/Déploiement Continu ». Plutôt que les équipes de développement livrent leur code source aux opérationnels en un seul bloc et à un seul moment, le code est testé tout au long de son écriture ou de ses modifications. Les outils d’intégration et de déploiement réduisent le temps nécessaire pour créer de nouvelles fonctions. Le code, fonctionnel puisque vérifié, servira de base aux opérationnels.

### GitHub

Le service d’hébergement de projet et de programmation est depuis quelque temps la star des outils CI/CD. En enregistrant toutes les modifications du code source, GitHub permet de suivre l’avancement des travaux et d’avoir un aperçu sur l’historique de travail. Très avantageux pour une équipe, GitHub fusionne les codes sources.

### SourceForge

L’adversaire de GitHub permet de créer des projets en open source grâce à une palette d’outils très complète : wikis de projets, accès à une BDD MySQL, url de sous-domaines uniques, métriques et analyses.

### Les outils CI/CD alternatifs

Si vous voulez éviter les deux premières sources, tournez-vous vers GitLab et son accès facilité à Azure DevOps ; Bitbucket et son environnement complet intégrant un puissant outil de suivi des problèmes ou Beanstalk qui permet de coder, tester et déployer directement depuis un navigateur.

## La gestion des configurations

Elle rassemble les processus qui valident la conformité d’un produit face aux diverses exigences durant tout son cycle de vie.

### Ansible

Ansible automatise des tâches courantes telles que le déploiement d’application et l’orchestration IT. Il facilite l’installation, la configuration et la modification d’un ou plusieurs serveurs de manière prévisible et déclarative.

### Jenkins

Jenkins permet aux équipes de surveiller la réalisation des tâches répétées. L’outil s’appuie sur ses nombreux plugins pour délivrer une solution simple d’utilisation pratique dans les configurations et les tests de régression d’automatisation.

## Infrastructure as code

Elle chiffre l’ensemble des processus afin d’automatiser les tâches reproductibles telles que la configuration ou les déploiements. Les plateformes répondent, pour la plupart d’entre elles, à la philosophie UNIX : elles ne font qu’une chose, mais elles le font bien.

### Terraform

L’environnement logiciel Terraform se concentre principalement sur le provisioning d’infrastructure. Il automatise la construction des ressources et facilite la définition des typologies cloud pour les principaux fournisseurs d’infrastructure.

## Doper la production

Grâce aux conteneurs, les applications sont empaquetées avec leur contexte d’exécution. Elles sont livrées avec tous leurs logiciels, bibliothèques et documents nécessaires. Ces paquets immuables traversent les équipes du développement à la production.

### Docker

Le logiciel libre en open source se concentre sur la normalisation et la spécification des containers afin de construire un écosystème sain. Son objectif ? Ne faire tourner qu’une seule application. Son système repose sur une utilisation minimaliste des fichiers, une gestion simple du démarrage des services et des contenairs immuables.

### Kubernetes

Kubernetes se présente comme une plateforme d’orchestration des containers, elle abrite une quantité importante de paquets. Au-delà de son service de plateforme, Kubernetes automatise le processus de gestion d’une large palette de ces containers.

## Le monitoring de l’infrastructure

Utilisé en fin de parcours, le monitoring permet d’analyser et de tirer les enseignements du déploiement des architectures et développement. Son but : trouver et extraire les pistes d’amélioration.

### Grafana

Le logiciel libre multiplateformes se concentre sur la visualisation de données. Via un tableau de bord et ses graphiques, il facilite la lecture des informations et accélère l’analyse des bugs, ainsi que leur résolution.

### ELK (Elasticsearch, Logstash, Kibana)

ELK stack collecte et analyse les journaux les plus courants du monde IT. Il centralise les données afin de résoudre les problèmes, de surveiller les applications ou de réaliser des audits. Via Logstach, le stack rassemble les données et rend possibles les alertes mails ou SMS.

### Prometheus

Grâce à l’enregistrement des métriques en temps réel, Prometheus répond aux demandes en s’appuyant sur un langage de requête simple et génère des alertes. En un seul outil, les équipes rassemblent une base de données multidimensionnelle, un mécanisme de collecte évolutif et un langage de requête puissant.