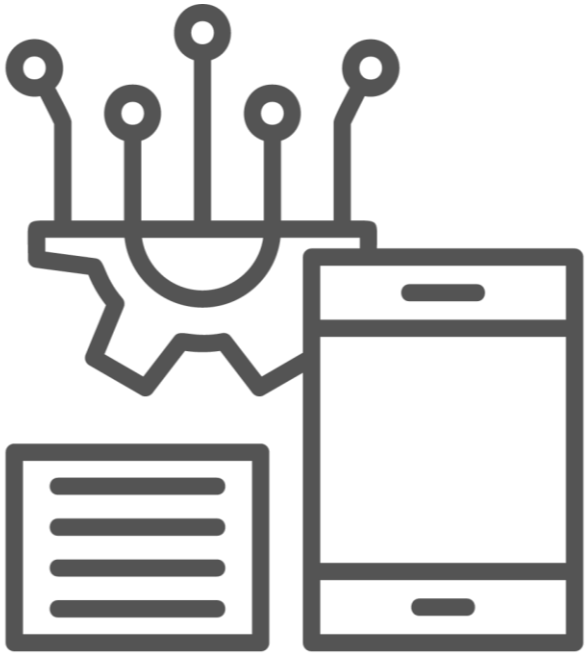
VEILLE TECHNOLOGIQUE

Automatisation d’infrastructure

SIO 2023 – Option SISR



Winness RAKOTOZAFY

TABLE DES MATIERES

I- QU'EST-CE QUE LA VEILLE TECHNOLOGIQUE ?....3

II- SUJET DE MA VEILLE TECHNOLOGIQUE....3

III- COMMENT S'ORGANISE MA VEILLE TECHNOLOGIQUE ?....3

1) Outils utilisés....3

1.1) Google Alert....3

1.2) YouTube....5

1.3) Abonnement magazine....6

1.4) CVE – CERT-FR....6

2) Méthodologie appliquée....7

Etape 1 : Collecte des informations....7

Etape 2 : Tri et traitement des informations....7

Etape 3 : Analyse des informations....7

Etape 4 : Diffusion et production du document de veille....7

Etape 5 : Mise à jour régulière....7

IV- QU'EST-CE QUE L'AUTOMATISATION D'INFRASTRUCTURE ?....8

V- COMPRENDRE LE PROCESSUS D'AUTOMATISATION....8

1) Les outils de versionning....9

2) Les outils de build....10

3) Les outils de test....11

4) Les outils de déploiement....11

5) Les outils de surveillance....12

VI- LES ENJEUX DE L'AUTOMATISATION D'INFRASTRUCTURE....13

VII- BILAN DE LA VEILLE TECHNOLOGIQUE....14

REFERENCES....15

I- Qu’est-ce que la veille technologique ?

La veille technologique est un processus de surveillance et d'analyse des nouvelles avancées technologiques dans un domaine spécifique. Elle consiste à collecter, à trier et à évaluer l'information pertinente pour anticiper les évolutions technologiques futures.

L’objectif de la veille technologique ?

Permettre à une entreprise ou à un individu de rester à jour avec les dernières tendances et innovations dans son domaine. En faisant cela, il est possible de prendre des décisions éclairées en matière de développement de produits, de stratégie commerciale et d'investissement, ce qui peut permettre une meilleure compétitivité et une croissance durable.

II- Sujet de ma veille technologique

Pour ma veille technologique, j’ai décidé de traiter le thème de **l’automatisation d’infrastructures**, regroupant les différentes technologies qui permettent de réaliser des tâches automatisées et des déploiements continues.

Mon choix s’est porté sur ce thème car c’est actuellement une tendance émergente dans le domaine informatique, et présente également de nombreuses opportunités dans le métier.

Cet intérêt s’est surtout manifesté par mes débuts en scripting shell et PowerShell dans les tâches d’administration système et de déploiement, et que par la suite, j’ai pu découvrir Ansible au cours de ma formation, ce qui a suscité ma curiosité sur les outils similaires et à quoi ils pourront servir dans mes tâches quotidiennes.

Aussi, au cours de ma veille et du processus d’apprentissage, j’apprends plusieurs technologies de part et d’autre me permettant ainsi d’élargir mon panel de compétence et me permettra de répondre aux besoins actuels des entreprises.

Toutefois, pour ne pas s’y perdre dans la grande diversité et les technologies à savoir dans le domaine de l’automatisation, il est important de se définir une sorte de roadmap et un objectif de ce que l’on souhaite faire et concrétiser en termes d’automatisation.

III- Comment s’organise ma veille technologique ?

Pour organiser ma veille technologique, j’ai utilisé plusieurs outils me permettant de traiter de l’information de manière qualitative, découvrir les nouvelles technologies et fonctionnalités, et construire une habitude de traitement d’information.

La stratégie de veille se construit donc surtout par l’utilisation des outils qui récupère des informations et génère des alertes dans ma boite de messagerie, puis le traitement des informations qualitatives pour en générer le contenu suivant, et de rester à jour sur les différentes technologies utilisées dans le monde de l’automatisation d’infrastructure.

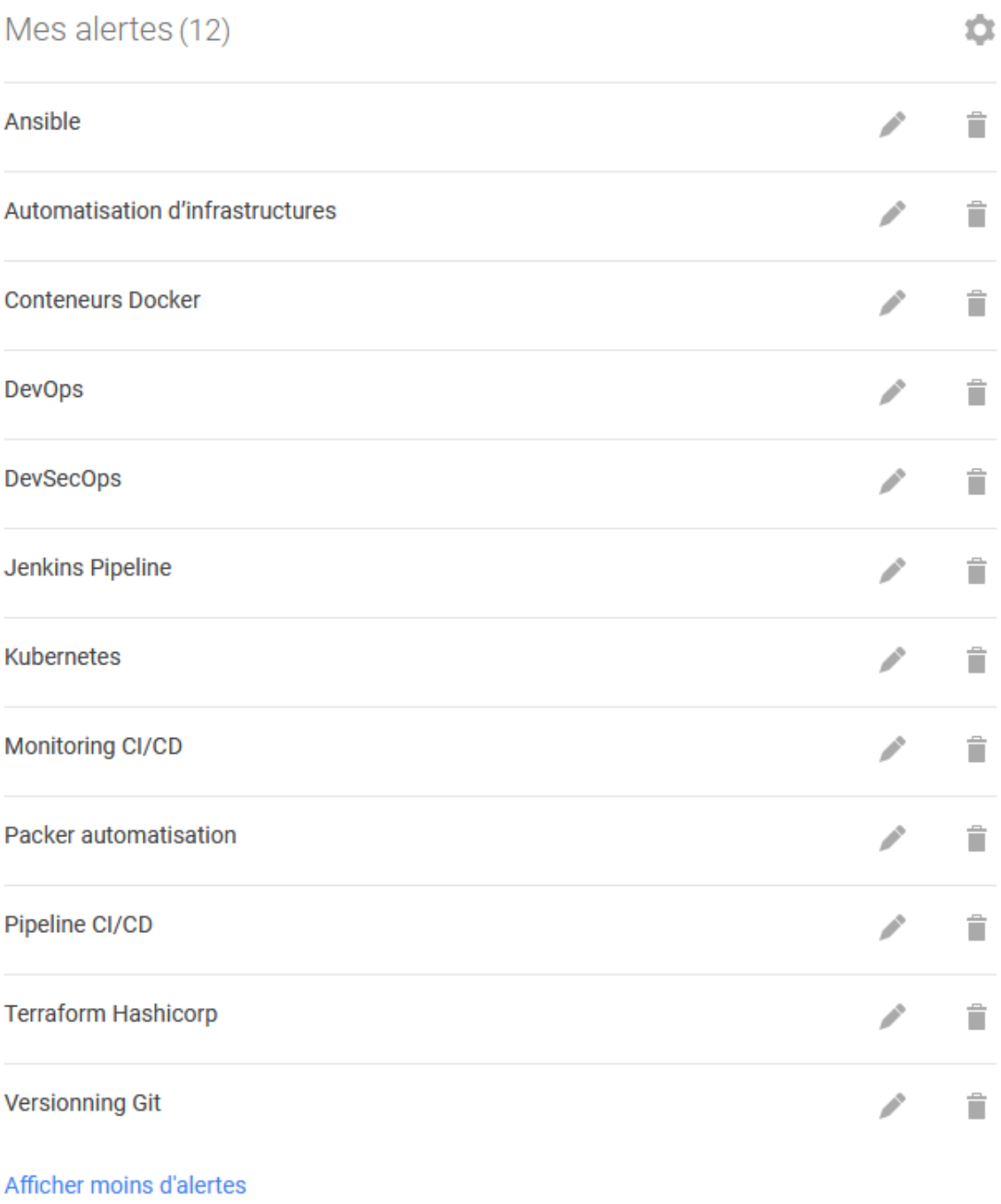
Ainsi, les principaux outils pour la collecte d’information sont : **Google Alert, YouTube, Magazine IT, CVE – CERT FR.**

1) Outils utilisés

1.1) Google Alert

Google Alert nous permet de définir par le biais de mots clés de faire remonter des informations dans notre boite de messagerie sur les sujets, thématiques et technologies qui peuvent s’apparenter à l’automatisation d’infrastructure et culture DevOps.

Pour la veille technologique, les mots clés utilisés sont : **Ansible, Automatisation d'infrastructures, Conteneurs Docker, DevOps, DevSecOps, Jenkins Pipeline, Kubernetes, Packer automatisation, Pipeline CI/CD, Supervision des réseaux informatiques.**



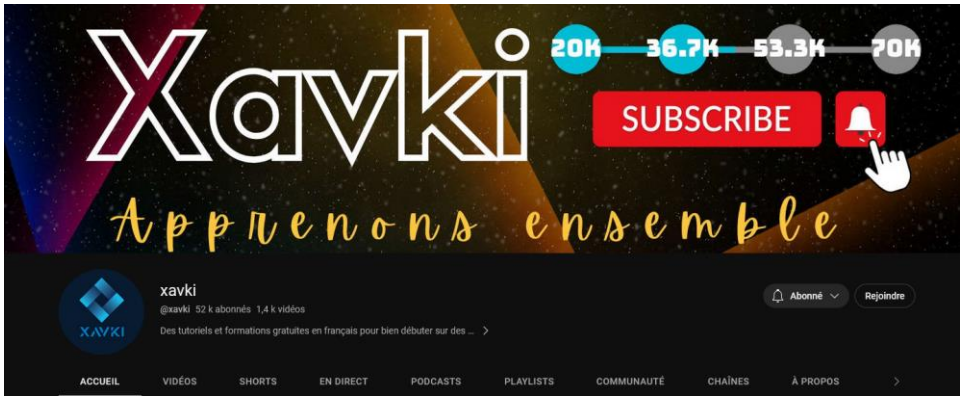
Lorsque ces mots-clés sont renseignés, des articles disponibles en ligne qui traiteront ces mot-clés seront envoyés dans ma boite de messagerie. Par la suite, je filtre les contenus qui m'intéressent et les enregistrent dans un onglet de ma messagerie.

Enfin, les contenus filtrés, lecture des articles établie, j'approfondis les sujets traités soit par le biais d'une expérimentation technique, soit par l'établissement d'une analyse des sujets rencontrées.

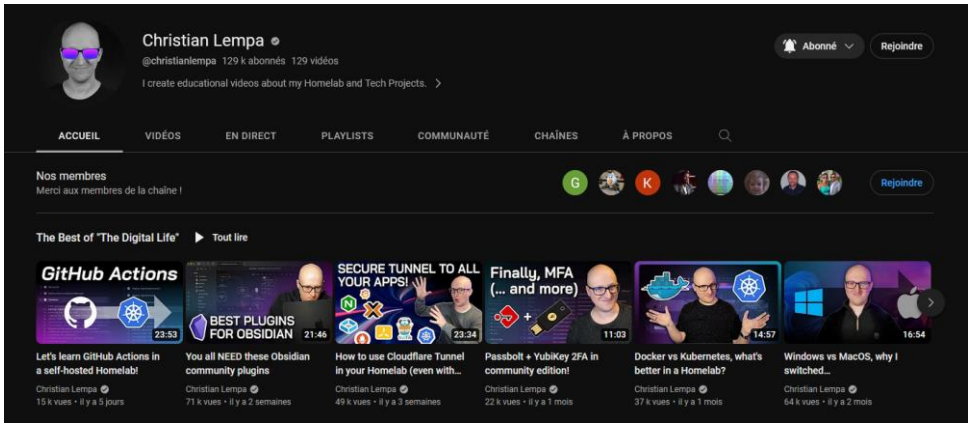
1.2) YouTube

Quelques chaînes YouTube intéressantes sont des sources d'informations qualitatives, mais permettent également la prise en main des outils, notamment sous forme de guide. Pour ma veille donc, les principales chaînes qui m'ont permis d'approfondir la compréhension sur les outils d'automatisation de déploiement d'infrastructure sont :

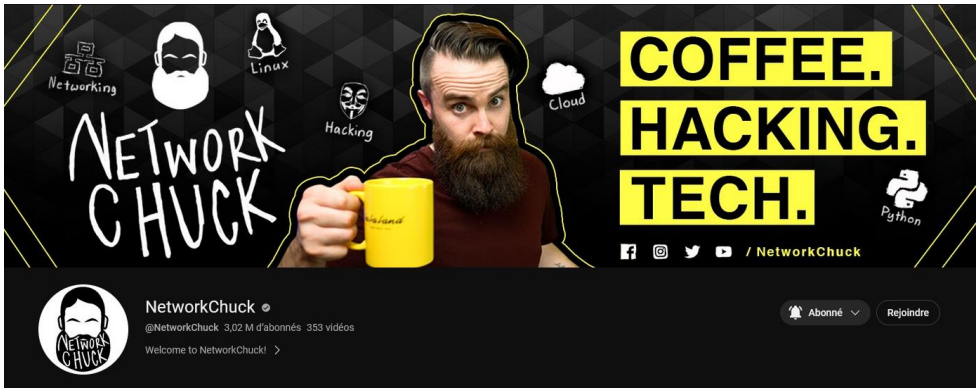
@xavki : chaîne française qui produit des contenus orientés DevOps et une belle référence dans de domaine, allant du versionning jusqu'au déploiement continue. Plusieurs tutos sur des technos sont disponibles (Terraform, Kubernetes, Ansible, Docker, GitLab, Grafana & Prometheus, Jenkins, …), avec également quelques podcasts. Sa philosophie : l'**autoformation**.



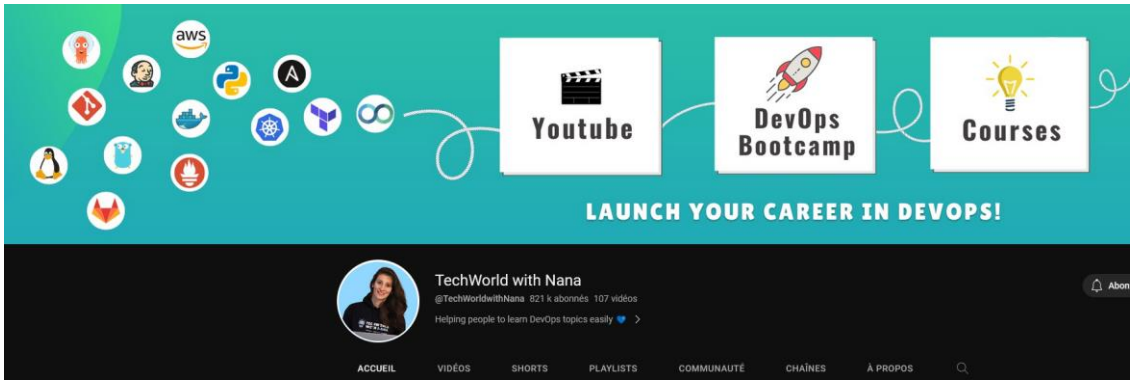
@Cristian Lempa : une chaîne qui traite de nombreuses technologies avec une appétence sur les conteneurs, notamment avec Docker, et l'orchestration avec Kubernetes.



@NetworkChuck : sans grande surprise, comme son nom l'indique, une chaîne dédiée au réseau de base, mais touche actuellement des sujets sur la cybersécurité (ethical hacking principalement), mais aussi tout ce qui est automatisation avec Ansible et Kubernetes.



@Techworld with Nana : chaine anglophone, proposant un large panel de tutoriel et d'informations sur le métier du DevOps, et aussi sur les différents outils utilisés et de leur tendance dans le cadre du déploiement automatisé.



1.3) Abonnement magazine

Dans le cadre de mon alternance actuel, j'ai eu l'opportunité d'avoir accès à des magazines mensuels traitant des sujets de l'informatique :

Linux Pratique, magazine francophone qui traite de tous les sujets sur le système d'exploitation GNU/Linux, et du kernel Linux, ainsi que les outils de déploiement et de configuration automatique comme **Ansible, Packer, Puppet**.

MISC, le magazine de la cybersécurité offensive et défensive, qui aborde la thématique du DevSecOps, permettant d'intégrer la sécurité dans le cadre de l'automatisation d'infrastructure.

1.4) CVE – CERT-FR

Pour compléter le lot des informations à prendre en compte et dont se tenir informé, les informations concernant la sécurité informatique sont toutes aussi importantes.

Les informations tirées des CVE permettent de se tenir à jour sur les différentes vulnérabilités présentes au sein d'un système ou d'une application, ce qui permettra en finalité d'optimiser toute la chaine de production de l'automatisation d'infrastructure.

Aussi, les informations publiées par le **Centre Gouvernemental de veille l'alerte et de réponses informatiques (CERT – FR)**, constitue également une source d'information qualitative pour se tenir à jour des différentes menaces relevés dans le contexte francophone.

Ainsi, voici les deux principaux liens source d'information pour la mise en place de la veille sécurité :

- https://www.cve.org/Media/News/AllNews

- https://www.cert.ssi.gouv.fr/

A présent, nous allons schématiser l'organisation, la mise en place et la méthodologie appliquée pour la veille technologique.

2) Méthodologie appliquée

Les sources d'informations définies, ci-dessous la méthodologie appliquée pour la mise en place de la veille technologique :

Etape 1 : Collecte des informations

Cette étape consiste à recueillir les diverses informations sur le thème choisi, ou d'autres (comme la sécurité informatique) de manière passive comme les **alertes de messagerie** avec Google Alert ou active, par le biais de recherche sur les différentes sources comme YouTube, Reddit, Journaldunet, les magazines de l'IT.

Etape 2 : Tri et traitement des informations

Parmi les informations collectées, nous effectuerons un tri pour prendre en considération les informations qui nous sont nécessaire pour rester à jour, et d'en approfondir les connaissances technologiques. C'est dans ce processus que nous évaluerons la crédibilité et pertinences des sources, d'éliminer les quelconques doublons d'informations et recueillir que les informations importantes.

Etape 3 : Analyse des informations

Les informations triées, nous allons ensuite analyser les données recueillis pour identifier les tendances, innovations et les différents enjeux des technologies rencontrées au cours de la veille technologique.

Etape 4 : Diffusion et production du document de veille

Rédaction et diffusion du rapport d'analyse ci-présente en synthétisant les informations pertinentes et en découler un bilan de la veille technologique.

Etape 5 : Mise à jour régulière

La mise à jour régulière de la veille technologique rend la veille plus efficace, ainsi, il est nécessaire de mettre à jour régulièrement les sources d'informations, de suivre les évolutions des technologies, pour pouvoir s'adapter en conséquence.

IV- Qu’est-ce que l’automatisation d’infrastructure ?

L'automatisation d'infrastructure consiste à utiliser des technologies pour réaliser des tâches tout en limitant l'intervention humaine, de manière à contrôler les composants matériels, logiciels et réseau, le système d'exploitation ainsi que les systèmes de stockage des données utilisés pour fournir des services et solutions informatiques.[1]

C'est un élément clé des processus d'optimisation de l'environnement informatique et de transformation numérique. Ainsi, pour réussir, une entreprise a besoin d'environnements efficaces, évolutifs et fiables. L'automatisation de l'infrastructure peut aider votre entreprise à rationaliser l'exploitation, améliorer l'agilité accroître la productivité, renforcer la sécurité et augmenter la disponibilité.

Nombreux sont les domaines pouvant être gérés par l'automatisation informatique tel que le Cloud, Sécurité Informatique, Surveillance.

V- Comprendre le processus d’automatisation

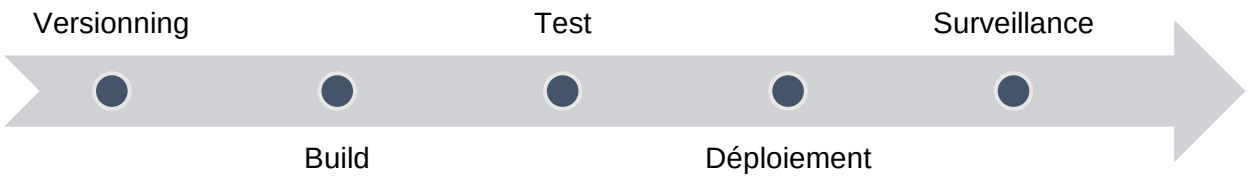
Dans le cadre d’automatisation d’infrastructure, nous nous retrouvons rapidement immergée par la grande diversité des outils disponibles pour automatiser nos tâches. Ainsi, avant de se lancer, il faut distinguer l’utilité de chaque pour ne pas en finalité être dans l’usine à gaz.

Aussi, dans la culture DevOps et l’automatisation, nous retrouvons ce qu’est la **Pipeline CI/CD,** pour l’intégration continue et déploiement continu.

La pipeline CI/CD

La pipeline CI/CD regroupe tout le cycle de vie du déploiement et automatisation d’une application et/ou infrastructure. Mais concrètement, qu’est-ce qu’un pipeline ? Un pipeline va surtout s’intéresser au déploiement d’une applicatif de manière automatisée sur toutes les tâches : les dépôts, le versioning, les tests et l’installation/déploiement finale.

Plusieurs outils sont utilisés dans le cadre du pipeline, allant du collaboratif dans le cadre d’une ITSM (Information Technology Service Management), jusqu’au déploiement d’une infrastructure. Mais comme les outils sont très nombreux, et que nous ne voudrions pas trop rentrer dans le détail, les outils peuvent être catégorisés selon le schéma suivant, illustrant également le pipeline CI/CD où chaque étape sera automatisée.



[1] « What is automation? », Red Hat Enterprise, www.redhat.com

1) Les outils de versionning

Comme outils de versionning les plus rencontrés sur le marché, on y retrouve majoritairement les outils basés sur **Git,** comme **Gitlab**, une solution open source que nous pouvons héberger soi-même sur nos propres serveurs, et **Github**, une plateforme SaaS, Software as a Service, permettant d'héberger des codes sources en ligne avec un suivi de version.

GIT est un système de contrôle de version le plus utilisé aujourd'hui, développé en 2005 par Linus Torvalds. De plus en plus de projets logiciels reposent sur Git pour le contrôle de version, y compris des projets commerciaux et en open source.

GITLAB est une plateforme de développement logiciel basée sur le système de contrôle de version Git, largement utilisé. Elle offre une gamme complète d'outils pour la gestion du cycle de vie des applications, y compris l'hébergement de dépôts Git, le suivi des problèmes, la planification des tâches, l'intégration continue, la livraison continue, les tests automatisés et la collaboration en équipe.

En tant qu'alternative à GitHub, GitLab est disponible en deux versions principales : GitLab Community Edition, qui est une version open source et gratuite, et GitLab Enterprise Edition, qui est une version propriétaire avec des fonctionnalités supplémentaires pour les grandes entreprises

GITHUB est une plateforme SaaS (Software as a Service) qui permet aux développeurs de développer et de déposer leur code source facilitant la collaboration et la gestion des projets logiciels.

Github dispose de plusieurs fonctionnalités dont : **l'hébergement de dépôts, le suivi des problèmes, demandes d'extraction par pull, collaboration en équipe, intégration continue et gestion de versions**.

Notion de git flow

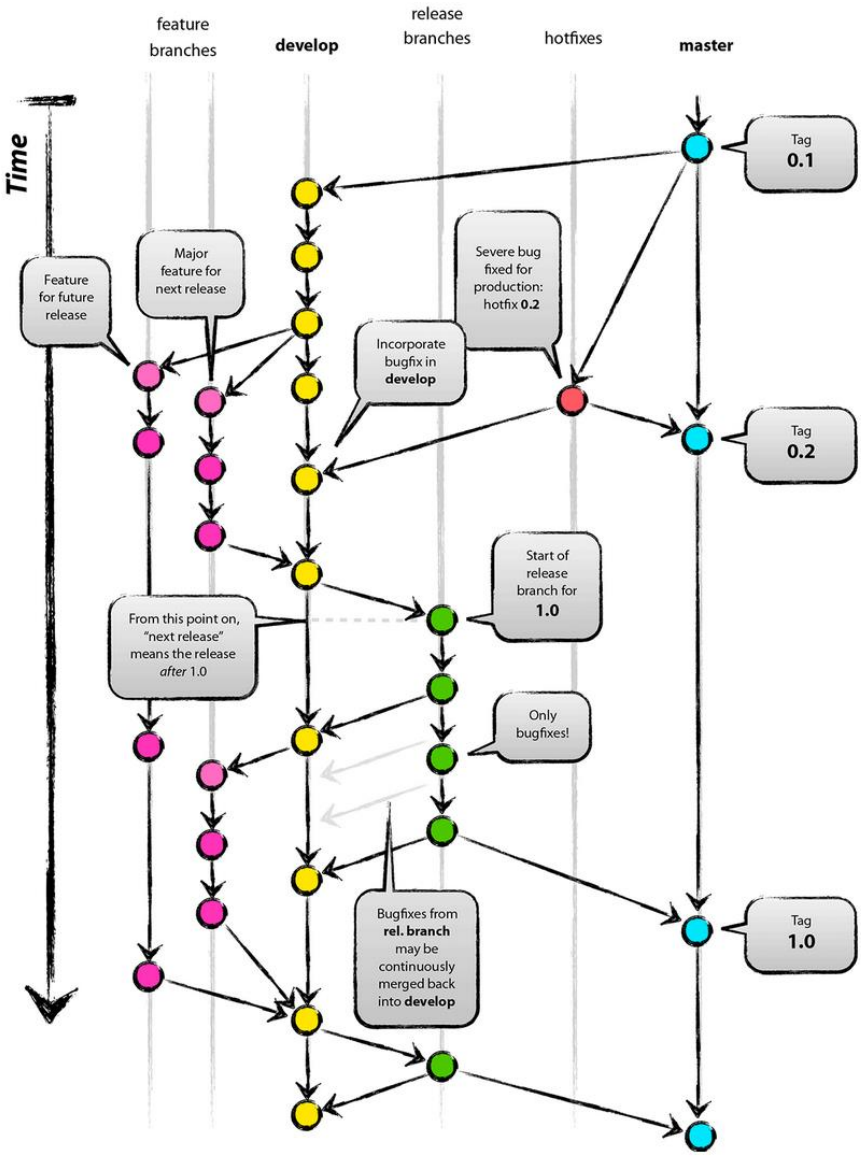
Git flow est une organisation de développement avec multibranches grâce à l'outil git, comme une branche **features → dev → release → master.**

Il y existe de nombreuses variantes du git flow selon les besoins d'une entreprise, mais en somme, le git flow permet d'automatiser le plus possible tout le process de développement et déploiement, notamment les **merges, tests, build** et **déploiement**.

Selon les informations recueillies, et illustrer cette notion, nous pouvons considérer la fonctionnalité de **Github Actions**[2]**.**

A titre indicatif, ci-dessous un schéma illustrant cette notion proposée par Jeff Kreeftmeijer, disponible sur https://jeffkreeftmeijer.com/git-flow (avec les explications adéquates)

[2] Plateforme de Github qui permet d'automatiser tout le flux opérationnel pour le déploiement d'un produit/infrastructure.



2) Les outils de build

L'objectif principal d'un outil de build est de faciliter et d'automatiser la construction cohérente d'une application, en minimisant les erreurs humaines et en simplifiant le processus pour les développeurs. Il permet également de **gérer efficacement les dépendances externes**, de **gérer les configurations spécifiques à l'environnement de développement** et de **générer des rapports de build détaillés**. Comme outils de build, nous pouvons citer **Jenkins** et **Docker**, mais d'autre outils tel que **Make** ou **Maven** peuvent être prises en compte et sont disponible sur le marché.

JENKINS, en plus d'être une grande référence dans l'intégration continue au côté de GitLab CI, est également un outil de build populaire et très appréciés par les développeurs. Il est open source, gratuit, et prend en charge une large gamme de fonctionnalités comme : **l'automatisation de build, la gestion des dépendances, .la compilation de code source,** mais surtout **l'intégration avec d'autres outils,** ce qui permet automatiser et orchestrer les différentes étapes du processus de build dans un projet de développement

DOCKER est une plateforme open source qui permet de créer, distribuer et exécuter des applications dans des conteneurs légers et isolés. Il n'est pas strictement un outil de build, mais étroitement lié au processus de construction d'une application. Docker est également une grande référence dans la **gestion de conteneurs** qui permet de lancer dans des **environnements isolés** des instances, et facilite grandement le déploiement de divers applicatifs grâce à une **gestion efficace des dépendances**.

3) Les outils de test

Les outils de test permettent d'assurer qu'aucune erreur est présente dans le code d'une application, et garantissent la qualité du code et en détectant les erreurs et les problèmes potentiels à chaque étape du processus. Il permet de valider les modifications apportées au code source, de s'assurer que l'application fonctionne correctement et de réduire les risques d'introduire des bugs dans la version déployée.

Selon les besoins, nous pouvons procéder à des **tests unitaires**, selon le code d'application comme ou pour valider l'état de configuration d'une infrastructure. Ainsi comme outil étudié nous pouvons citer **ServerSpec** ou **TestInfra.**

SERVERSPEC est un outil de test pour valider l'état et la configuration de l'infrastructure automatiquement construite par les outils de build. Ecrit en Ruby, il permet de tester l'infrastructure avec ces différentes fonctionnalités dont : **linting des configurations** qui permet de valider les fichiers de configuration comme les fichiers d'Ansible, Chef ou Puppet avant déploiement, les **tests d'intégration** pour vérifier que chaque couche de l'infrastructure fonctionne comme prévu et enfin les **tests de déploiement** pour vérifier que les mises à jour ou nouveau déploiement sont correctement configurés.

TESTINFRA est une bibliothèque Python qui permet de réaliser des tests sur l'infrastructure. Il offre des fonctionnalités pour exécuter des commandes sur des machines distantes, inspecter les fichiers et les services, et vérifier les résultats attendus. De ce fait, TestInfra assure ainsi que l'infrastructure est conforme aux spécifications et aux attentes. Cela contribue à garantir la stabilité, la sécurité et la fiabilité de l'infrastructure déployée.

4) Les outils de déploiement

Les outils de déploiement permettent d'**automatiser la mise en production** des applications ou des services après qu'ils ont passé avec succès les étapes de construction, de test et de validation. Ces outils facilitent le **déploiement continu** en réduisant les erreurs humaines, en accélérant le processus et en garantissant la cohérence de l'environnement de déploiement.

Dans le cadre de cette veille technologique, et dans les étapes du pipeline, ce sont ces outils que j'ai le plus apprécié et dans lequel j'ai effectué quelques tests de lab.

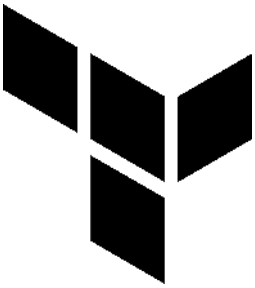
On y retrouve des outils comme **Ansible** pour la gestion et déploiement de configuration sur les serveurs, **Kubernetes** pour le déploiement et gestion des applications conteneurisées, et **Terraform**, dans le cadre d'une **Infrastructure as a Code**[3], pour automatiser le déploiement et la configuration des ressources d'infrastructure, telles que des serveurs, des réseaux, des bases de données, des services de cloud, etc.

[3] Type de configuration informatique permettant aux développeurs et aux techniciens d'exploitation de gérer et d'approvisionner automatiquement l'infrastructure informatique par le biais du code sans passer par des processus manuels. *Source : lebigdata.fr*

ANSIBLE est un outil d'automatisation open source qui peut être utilisé pour déployer des applications et des configurations sur des serveurs distants. Il permet de décrire l'état souhaité du système dans des fichiers YAML et d'exécuter des tâches pour atteindre cet état sur les machines cibles.

Ansible est populaire dans l'administration système, notamment par sa simplicité de mise en œuvre, mais aussi le fait qu'il nécessite aucune installation supplémentaire sur les serveurs distants, uniquement quelques configurations de SSH à effectuer.

KUBERNETES est un outil d'orchestration de conteneurs qui permet de gérer le déploiement, la mise à l'échelle, la gestion des configurations et la surveillance des applications conteneurisées. Dans le cadre d'une automatisation d'infrastructure, et d'un pipeline CI/CD, Kubernetes peut être utilisé de plusieurs manières dont : **déploiement des applications,** notamment si l'outil de build choisi est Docker, **rollbacks** pour revenir à une version précédente, **gestion des configurations, surveillance** et **gestion des erreurs.**



TERRAFORM est un outil d'infrastructure as code, utilisé pour automatiser le déploiement et la configuration des ressources d'infrastructure, telles que des serveurs, des réseaux, des bases de données, des services de cloud, etc.

Les avantages qu'offrent Terraform dans le cadre de l'automatisation d'infrastructure sont : **automatisation déploiement** avec les fichiers manifest de l'infrastructure, **gestion de l'état de l'infrastructure**, **intégration avec d'autres outils** de la chaine d'automatisation, et la **prise en charge de différents fournisseurs d'infrastructure.**

5) Les outils de surveillance

Les outils de surveillance jouent un rôle essentiel dans le cadre de l'automatisation d'infrastructure.Ils permettent de suivre l'état, les performances et les erreurs de l'infrastructure automatisée, et d'alerter en cas de défaillance ou d'anomalie. Ils sont essentiels pour garantir une surveillance proactive de l'infrastructure, identifier les problèmes rapidement et prendre des mesures pour les résoudre afin d'assurer la disponibilité et la performance du système.

Comme outil de surveillance et de monitoring dans le cadre de l'automatisation et du pipeline CI/CD, nous pouvons prendre en considération : la **pile ELK** pour la gestion et collecte de logs et **Prometheus** pour la supervision et collecte des métriques**.**

ELK Stack, pour **E**lasticsearch **L**ogstash **K**ibana, est une suite d'application pour permettre la gestion de logs des composants de l'infrastructure. En effet, Elasticsearch permet l'indexation et la recherche des logs traités et collecté par Logstash pour enfin être visualisé grâce au moteur de Kibana. Ainsi, cela facilite le dépannage, l'optimisation et l'amélioration continue du processus d'automatisation de développement et de déploiement.

PROMETHEUS est un outil de surveillance conçu pour la collecte et l'analyse des métriques, et offre des fonctionnalités puissantes pour surveiller l'état et les performances d'une infrastructure automatisée. En effet, il permet de collecter des métriques à partir de diverses sources et prend en charge plusieurs protocoles de collecte de données comme SNMP, JMX ou l'agent Prometheus. De plus, pour optimiser la surveillance de l'infrastructure, Prometheus intègre également des fonctionnalités d'alerte et notifications qui permettent de définir des règles d'alerte basées sur les métriques surveillées. Enfin, cet outil peut s'intégrer avec d'autres outils comme **Grafana**, pour optimiser la visualisation des données des métriques.

VI- Les enjeux de l'automatisation d'infrastructure

Ainsi, selon les notions vues et informations recueillies, l'automatisation de l'infrastructure présente plusieurs enjeux importants, notamment :

- **Efficacité opérationnelle** : L'automatisation d'infrastructure permet de réaliser des tâches répétitives et chronophages de manière plus rapide et précise, ce qui augmente l'efficacité des opérations. Cela réduit également les erreurs humaines potentielles, améliorant ainsi la fiabilité globale du système.
- **Évolutivité et flexibilité** : elle permet de également de mettre en place des infrastructures évolutives et adaptables aux besoins changeants, et permet de provisionner rapidement des ressources, de les allouer et de les configurer en fonction des besoins spécifiques, ce qui facilite la scalabilité de l'infrastructure.
- **Gestion du cycle de vie** : L'automatisation facilite la gestion du cycle de vie complet des infrastructures, y compris la création, la configuration, la surveillance, la mise à jour et la suppression des ressources. Cela permet de réduire les délais de déploiement, d'améliorer la cohérence et de simplifier la maintenance.
- **Réduction des coûts** : L'automatisation peut aider à réduire les coûts opérationnels en optimisant l'utilisation des ressources, en évitant les erreurs coûteuses, en minimisant les temps d'arrêt et en permettant une utilisation plus efficace du personnel technique.
- **Sécurité et conformité** : L'automatisation permet de mettre en place des politiques de sécurité et de conformité de manière cohérente et reproductible. Elle facilite la gestion des correctifs de sécurité, la configuration des pares-feux, la surveillance des journaux d'audit, etc., contribuant ainsi à renforcer la posture de sécurité globale.
- **Collaboration et communication** : L'automatisation peut favoriser la collaboration entre les équipes opérationnelles et de développement, en fournissant des processus standardisés et des outils partagés. Cela permet de faciliter la communication, de réduire les frictions et d'améliorer la transparence entre les équipes.

Il convient de noter que l'automatisation de l'infrastructure doit être mise en œuvre avec soin et être accompagnée d'une planification adéquate. Une gestion inappropriée de l'automatisation peut entraîner des problèmes tels que des erreurs systémiques, une perte de contrôle ou une dépendance excessive vis-à-vis de la technologie.

VII- Bilan de la veille technologique

Cette veille technologique nous a permis d'apprendre à recueillir les informations importantes sur le thème choisi, mais aussi d'apprendre de nouvelles technologies liées à l'automatisation d'infrastructure.

Aussi, nous avons constaté qu'actuellement les conteneurs commencent à prendre de l'ampleur par rapport aux machines virtuelles, et que le **Cloud** s'apparente beaucoup à l'automatisation d'infrastructure.

L'automatisation d'infrastructure regroupe l'utilisation de nombreuses technologies, nécessitant donc une grande curiosité sur les nouvelles technologies sur le marché et leurs fonctionnalités, et une bonne connaissance des différents systèmes.

Enfin, avec l'émergence de l'intelligence artificielle avec la popularisation de **ChatGPT**, l'automatisation d'infrastructure pourrait prendre un grand pas, en accélérant encore plus les processus d'automatisation par le biais de l'IA, comme exemple le déploiement d'un service ou d'un applicatif par uniquement une requête en langage humain, ou encore mieux, par la voix.

Parmi les informations recueillies au cours de la veille technologique, nous rencontrons de plus en plus l'**intersection** entre **les technologies DevOps** et l'**intelligence Artificielle**, qui pourrait en un sens façonner et innover le mode de fonctionnement des entreprises, mais dans l'autre sens, mettre en péril la sécurité d'un système d'information car nécessite des mesures appropriées cyber pour y faire face.

Cependant, cette corrélation pourrait mettre en alerte les équipes de sécurité informatique sur les menaces présentes dans de telles pratiques, et devraient donc envisager de prendre différentes mesures à mettre en place par rapport à cela.

Références

- https://www.redhat.com/fr/topics/automation/whats-it-automation
- https://www.journaldunet.com/solutions/dsi/1522259-quelles-sont-les-tendances-technologiques-qui-faconnent-les-entreprises-aujourd-hui/
- https://itsocial.fr/enjeux-it/enjeux-securite/cybersecurite/la-cybersecurite-nest-pas-toujours-a-la-hauteur-des-enjeux/
- https://www.itforbusiness.fr/devsecops-et-si-vous-faisiez-fausse-route-62585
- https://www.programmez.com/actualites/kubernetes-127-35283
- https://www.techmeup.fr/15097/pourquoi-ingenieurs-devops-importants-entreprises/
- https://www.lemondeinformatique.fr/actualites/lire-pourquoi-ansible-est-devenu-le-favori-des-devops-68328.html
- https://www.lebigdata.fr/docker-definition
- https://geekflare.com/fr/terraform-best-practices/
- https://jeffkreeftmeijer.com/git-flow
- https://www.chakray.com/fr/6-capacites-et-tendances-devops/