



Mémoire de mastères

Pour l'obtentioп du diplôme de mastère professionnel

Option : Systèmes Embarqué

Installation et Configuration des outils Devops dans l'environnement de développement Informatique à la C.P.G .

Réalisé par :
Mme. WIJDEN Azaza

Encadreur academique : M.
WAJDI Saadoui
Encadreur industriel : M.
NOREDDIN Belgacem

FIG. 1 : .

Dédicace

“

Je dédie ce modeste travail à mes parents pour leurs soutiens, leurs patience, leur encouragement durant mon parcours scolaire. A mes chers frères et ma sœur ainsi toute ma famille. A tous mes chers professeurs de département Automatiques des Systèmes Industriels (ASI). A tous mes amies. Je le dédie de même au staff de CPG et à toutes personnes qui m'a supporté tout au long de la période de mon stage et de réalisation de ce projet.

Merci.

”

- Wijden

Remerciements

Je tiens à exprimer ma profonde gratitude et mon sincère remerciement à mon encadreur pédagogique Mr. Saadaoui Wajdi pour l'encadrement et pour avoir assuré la partie théorique, l'aide et les conseils concernant les missions évoquées dans ce projet, qu'il m'a apporté lors des différents suivis.

Je m'intéresse à exprimer ma profonde gratitude et mes sincères remercîments à Mr Noreddine Belgacem et toute l'équipe de CPG pour son encadrement industriel et ces conseils précieux avec beaucoup de respect et plaisir. Mes remerciements s'adressent aussi aux membres du jury, qui m'ont honoré en acceptant de juger ce travail.

Enfin, c'est avec une grande émotion et de sincérité que je voudrais adresser ma gratitude à toutes les personnes ayant participé, collaboré, soutenues et apprécié de près ou de loin au bon déroulement de ce projet ainsi que la réalisation de mon travail.

Résumé

Ce travail s'inscrit dans le cadre du projet de mémoire de mastère à l’Institut Supérieur des Sciences Appliquées Technologies de Gafsa pour l’obtention du Diplôme de Mastère Professionnel spécialité système embarqué (MPSE). Dans ce cadre, j’ai conçu à concevoir l’installation et la configuration des outils DevOps dans l’environnement de développement informatique au sein de la Compagnie de Phosphate Gafsa (CPG).

Mots clés : DevOps / Jenkins / Docker / Git / Ansible / CI, CD / MySQL / Apache.

Abstract

This work is part of the Master's thesis project at the Higher Institute of Applied Sciences Technologies of Gafsa for the obtaining of the Professional Master's Diploma specializing in embedded systems (MPSE). In this context, I designed to design the installation and configuration of DevOps tools in the IT development environment within the Gafsa Phosphate Company (CPG).

Keywords : DevOps / Jenkins / Docker / Git / Ansible / CI, CD / MySQL / Appache.

Table des matières

Dédicace	II
Remerciements	III
Résumé	IV
Abstract	V
Introduction générale	1
1 Présentation Générale de l'entreprise et Cadre général de projet	2
1.1 Introduction :	3
1.2 Présentation de l'entreprise :	3
1.2.1 Présentation de l'entreprise :	3
1.3 Historique de la Compagnie de Phosphate de Gafsa :	3
1.3.1 L'implémentation de la société	4
1.3.2 Les missions de la société :	5
1.3.3 Organigramme de CPG :	6
1.3.4 Organigramme de la direction de l'informatique de gestion :	7
1.3.5 Direction de Contrôle de Gestion, de l'Informatique et d'Organisation :	7
1.3.6 L'évolution de la société :	8
1.4 Objectif de projet :	9
1.5 Analyse de l'existant :	9
1.6 La critique de l'existant :	9
1.7 Solution :	10
1.8 Conclusion :	11
2 La technologie DevOps	12
2.1 Introduction	13
2.2 C'est quoi DevOps :	13
2.3 Description du besoin métier pour DevOps :	14
2.4 Chemins d'adoption de DevOps :	15
2.5 Les outils de Devops :	16
2.5.1 Le développement CI/CD :	16
2.5.2 GitHub :	16
2.5.3 SourceForge :	16
2.5.4 Les outils CI/CD alternatifs :	16

Table des matières

2.5.5	La gestion des configurations :	17
2.5.6	La conteneurisation :	17
2.5.7	Le GitOps	18
2.5.8	Ansible :	18
2.5.9	Jenkins :	20
2.5.10	L'infrastructure as code : (IaC) :	20
2.5.11	Doper la production :	20
2.5.12	Docker :	20
2.5.13	Kubernetes :	22
2.5.14	La supervision et le monitoring de l'infrastructure :	22
2.5.15	Grafana :	23
2.5.16	ELK (Elasticsearch, Logstash, Kibana) :	23
2.5.17	Prometheus :	23
2.6	Les trois processus DevOps :	23
2.6.1	L'intégration continue :	23
2.6.2	La livraison continue :	24
2.6.3	Le déploiement continu :	24
2.6.4	L'amélioration continue :	24
2.7	Création de pipeline de livraison :	25
2.8	Les dix mythes de DevOps :	25
2.9	Présentation d'une solution DevOps :	26
2.10	MLOps : Le DevOps appliqué aux projets de Machine Learning	27
3	Installation des outils DevOps	30
3.1	Introduction	31
3.2	Ansible :	31
3.2.1	Installation de Ansible :	31
3.2.2	Configuration d'Ansible :	32
3.2.3	Version d'Ansible :	32
3.3	Jenkins	33
3.3.1	Installation de Jenkins :	33
3.3.2	Configuration de Jenkins :	34
3.3.3	Pipline Jenkins	38
3.4	Création de clé SSH :	39
3.5	Kubernetes :	41
3.5.1	Installation de Kubernetes :	41
3.6	Git :	42
3.6.1	Installation git :	43
3.7	Installation XAMPP :	44
3.8	Visual Studio Code :	45
3.9	Machine Virtuel :	46
3.10	Virtual Box :	47
3.11	VMware :	48
4	Test et Résultat	49
4.1	Introduction	50

Table des matières

4.2 Cadre du travail :	50
4.2.1 Environnement matériel :	50
4.2.2 Environnement logiciel :	50
4.3 Interface de connexion :	51
4.3.1 Page de connexion :	51
4.3.2 Le tableau de bord :	52
4.4 Code header PHP :	53
4.5 Activation de Hyper-V :	54
4.6 Docker compose yml :	56
4.7 Docker file :	57
Conclusion générale	59
Annexes	61
A Code	62

Table des figures

1	I
1.1	Direction de Compagnie de Phosphate de Gafsa	3
1.2	Les étapes de la production de phosphate.	5
1.3	Organigramme de CPG.	6
1.4	Organigramme de la Direction Informatique.	7
1.5	Architecture de solution proposée.	11
2.1	La technologie DevOps.	13
2.2	DevOps.	14
2.3	L'architecture de référence DevOps.	16
2.4	Conteneur vs machine virtuelle.	17
2.5	GitOps workflow.	18
2.6	Architecture de Ansible.	19
2.7	L'architecture de Kubernetes.	21
2.8	L'architecture de Kubernetes.	22
2.9	Processus DevOps.	23
2.10	La pipeline CI/CD	24
2.11	MLOps de DataScientet.	27
2.12	Étapes du développement d'un modèle de machine learning.	28
3.1	Installation ansible	31
3.2	Configuration de l'ansible.	32
3.3	Version installée ansible.	32
3.4	Installation de Jenkins	33
3.5	Ecran unlock Jenkins.	34
3.6	L'option d'installer des plugins suggérés.	35
3.7	Jenkins Plugin.	35
3.8	Création d'un jenkins user.	36
3.9	Instance de configuration.	36
3.10	Le tableau de bord principal.	37
3.11	Création de clé SSH.	40
3.12	Installation de Kubernetes.	41
3.13	Git	43
3.14	XAMPP.	44
3.15	Machine virtuel Ubuntu	46
3.16	Virtuel Box.	47
3.17	VMware Workstation.	48

Table des figures

4.1	La page de connexion compilée.	51
4.2	La page du tableau de bord.	52
4.3	code header php part 1.	53
4.4	code header php part 2.	53
4.5	code header php part 3.	54
4.6	Hyper-V.	54
4.7	Docker compose yml.	56
4.8	Docker file.	57
4.9	How to build Docker.	57
4.10	How to pull Docker.	58

Liste des sigles et acronymes

DevOps	<i>Dévelopement Opérationel</i>
CPG	<i>Compagnie des Phosphates de Gafsa</i>
CI	<i>Intégration Continue</i>
CD	<i>Déploiement continu</i>
IaC	<i>Infrastructure as code</i>
IBM	<i>International Business Machines Corporation</i>
nées sur le Web <i>Sociétés qui proviennent d'Internet</i>	
VM	<i>Virtuel Machine</i>
XAMPP	<i>cross-platform, Apache, MySQL, PHP and Perl</i>
VS code	<i>Visual Studio Code</i>
ITIL	<i>Information Technology Infrastructure Library</i>
paaS	<i>Platform as a Service</i>
CLI	<i>Interface de ligne de commande</i>
AWS	<i>Amazon Web Service</i>
GCP	<i>Google Cloud Platform</i>
OS	<i>Operating System</i>
ELK	<i>Elasticsearch, Logstash, Kibana</i>
ML Ops	<i>Machine Learning Operations</i>

Table des figures

RAM	<i>Random Access Memory</i>
CPU	<i>Central Processing Unit</i>
API	<i>Application Programming Interface</i>
MDB	<i>Configuration Management Database</i>
IA	<i>Intelligence Artificielle</i>
ML	<i>Machine Learning</i>
SQL	<i>Structured Query Language</i>

Introduction générale

Suite aux innovations technologiques, le DevOps a connu une évolution importante au cours de la dernière année.

Notamment, l'informatique a facilité la nécessité d'adopter des cadres de gestion logicielle plus rapides et plus agiles. En conséquence, les cycles de vie informatiques se sont raccourcis à mesure que les applications deviennent plus complexes. Cela a à son tour facilité le besoin de collaboration croisée et d'intégration entre les différents composants informatiques tels que le développement, les opérations et l'assurance qualité.

Le résultat : une nouvelle discipline informatique appelée DevOps.

[DevOps-The-Ultimate-Beginners-Guide-to-Learn-DevOps-Step-by-Step]

De manière générale, DevOps est une approche qui repose sur les principes Lean et Agile dans lesquels les responsables métiers avec les services de développement, des opérations et d'assurance qualité collaborent pour délivrer le logiciel en continu dans l'objectif de permettre à l'entreprise de saisir plus rapidement les opportunités du marché et d'accélérer la prise en compte des retours clients. En effet, les applications d'entreprise sont si diverses et composées de tant de technologies, bases de données, d'équipements utilisateurs, etc., que seule une approche DevOps permet de gérer avec succès toute cette complexité.

Certains affirment que DevOps s'adresse aux professionnels uniquement. D'autres avancent qu'il tourne autour du Cloud. IBM propose une vue large et holistique et considère DevOps comme une approche de livraison du logiciel orientée métier : une approche qui porte une fonctionnalité nouvelle ou une amélioration de l'idée jusqu'à la mise en production, en fournissant de la valeur métier aux clients finaux de manière efficace et en capturant les retours des utilisateurs de cette nouvelle fonctionnalité.

Ce rapport contient une introduction générale, quatre principaux chapitres couvrant l'ensemble des aspects de notre travail et conclusion générale qui présente les perspectives sur lesquelles peut s'ouvrir le présent projet. Nous avons commencé dans un premier lieu à situer notre projet dans son cadre général et de présenter l'entreprise où j'ai fait mon stage. Nous avons fait une étude sur la méthode de travail afin de mettre en évidence le choix de l'entreprise pour le suivi et la planification du projet. Dans le second chapitre , nous avons présenté les nouveaux concepts du DevOps rencontré lors de ce stage. Au cours du troisième chapitre nous avons présenté une étude technique (installation des logiciels et la conception de l'applicatiopn) pour mettre l'accent sur les différents outils implémentés tout au long du projet afin de bien mettre en évidence les concepts de DevOps.

Chapitre 1

Présentation Générale de l'entreprise et Cadre général de projet

1.1 Introduction :

Le premier chapitre a pour objectif de situer le projet dans son cadre général en présentant dans une première partie l'organisme d'accueil au sein duquel nous avons effectué notre stage de mémoire.

Dans la deuxième partie nous introduisons le sujet tout en faisant une étude de l'existant pour déterminer et critiquer les points de faiblesse de l'ancien procédé puis proposer des solutions adéquate et une évaluation indépendante et neutre. Ensuite nous présentons la problématique qui a engendré ce travail ainsi que la solution proposée.

1.2 Présentation de l'entreprise :

1.2.1 Présentation de l'entreprise :

La Compagnie des phosphates de Gafsa ou CPG est une entreprise tunisienne d'exploitation des phosphates basée à Gafsa. Elle est rattachée en 1994 au Groupe chimique Tunisien. La CPG figure parmi les plus importants producteurs de phosphates, occupant la cinquième place mondiale avec une production de presque huit millions de tonnes en 2009. En 2014, la production a chuté à cinq millions de tonnes et la Tunisie est le huitième producteur mondial, avec 2,27 En 2010, la CPG exploite huit mines à ciel ouvert, situées dans les délégations de Redeyef, Moulares, Metlaoui, et Mdhila et onze laveries destinées au traitement du minerai.



FIG. 1.1 : Direction de Compagnie de Phosphate de Gafsa

1.3 Historique de la Compagnie de Phosphate de Gafsa :

La Compagnie de Phosphate de Gafsa a été créée en 1896 par le gouvernement tunisien pour exploiter les gisements de phosphate de la région de Gafsa. À cette époque, le phosphate était principalement utilisé comme engrais pour l'agriculture.

Chapitre 1. Présentation Générale de l'entreprise et Cadre général de projet

Au fil des ans, la production de phosphate de la CPG a connu une croissance rapide. Dans les années 1960, la Tunisie est devenue l'un des principaux exportateurs de phosphate dans le monde.

Cependant, la compagnie a connu des difficultés au cours des dernières décennies en raison de la baisse des prix du phosphate sur le marché mondial et de la concurrence accrue d'autres producteurs de phosphate.

En 2011, la révolution tunisienne a eu un impact important sur la production de phosphate de la CPG. Les troubles sociaux et politiques ont perturbé les activités de la compagnie, entraînant une baisse de la production et des exportations.

Malgré ces défis, la CPG reste l'un des principaux producteurs de phosphate dans le monde et un acteur clé de l'économie tunisienne.

1.3.1 L'implémentation de la société

Le bassin minier de Gafsa regroupe cinq secteurs d'extraction et de traitement de phosphate qui sont situés à Mdhilla, Metlaoui, Moularès et Redeyef. Le tableau suivant présente la localisation des différents secteurs, les mines et les usines de traitement de phosphate de la CPG.

Secteur	Mines	Usines de traitement
Metlaoui Kef Schfaier	Kef Schfaier - Tables de Metlaoui	4 unités de lavage 1 unité de flottation des rejets fins
Metlaoui Kef Eddour	Kef Eddour Centre	1 unité de lavage
Mdhila	Jallabia - Mzinda	3 unités de lavage Mzinda, 1 unité de flottation des rejets fins
Moulares	5Moulares - Kef ed-dour Ouest	2 unités de lavage
Redeyef	Redeyef	1 unité de lavage

TAB. 1.2 : L'implantation des différents secteurs de la CPG

1.3.2 Les missions de la société :

Les différentes missions de la société se devisent en quatre grands groupes tels que la préparation du terrain, l'extraction, la production et la commercialisation des phosphates.

La figure suivante récapitule les étapes essentielles de la production des phosphates.

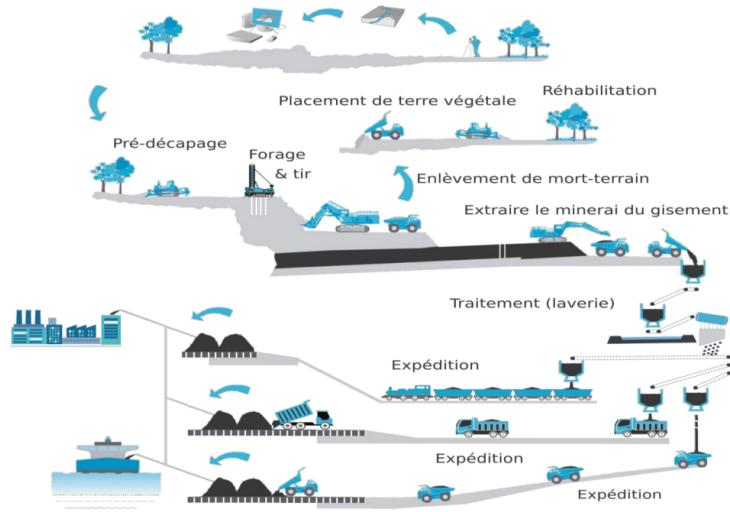


FIG. 1.2 : Les étapes de la production de phosphate.

1.3.3 Organigramme de CPG :

La structure de la CPG est définie comme l'ensemble des dispositifs par lesquels la société répartit, organise, cordonne et contrôle ses activités. Cette répartition des domaines d'activité et de supervision des différents agents est représentée par l'organigramme de la figure ci-dessous qui montre aussi la direction de l'informatique de gestion où je vais réaliser mon stage de mémoire de mastère.

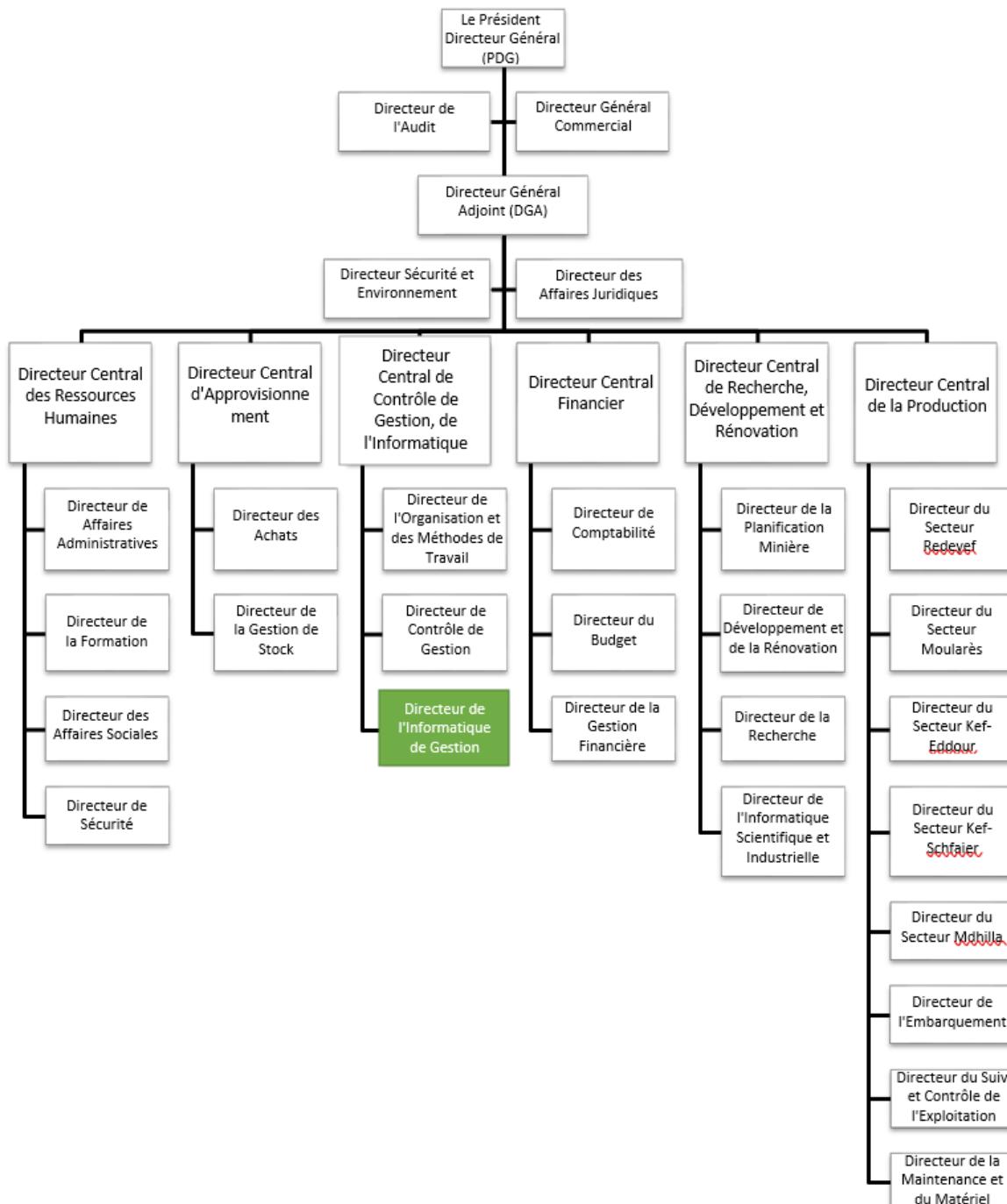


FIG. 1.3 : Organigramme de CPG.

1.3.4 Organigramme de la direction de l'informatique de gestion :

Ce organigramme illustre la direction informatique de la siège de C.P.G.

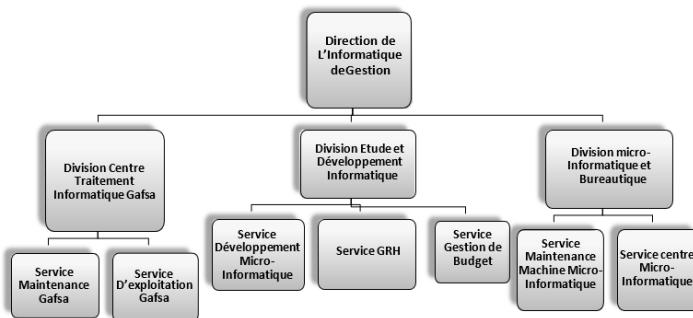


FIG. 1.4 : Organigramme de la Direction Informatique.

1.3.5 Direction de Contrôle de Gestion, de l’Informatique et d’Organisation :

La Direction Centrale du Contrôle de Gestion, Informatique et Organisation est composée de trois directions :

- La Direction de l’Organisation et des Méthodes de Travail.
- La Direction de Contrôle de Gestion.
- La Direction de l’Informatique de Gestion.

La Direction de l’Informatique de Gestion est un groupe de départements qui réalisent des travaux particuliers en C.P.G.

Le Direction Informatique se compose de trois départements, qui sont répartis comme suit :

Division d’étude et de développement informatique dont la mission est :

- La conception et le développement des projets informatiques.
- L’administration de la base des données.
- La maintenance des projets existants.

Division micro-informatique et bureautique dont la mission est :

- L’assistance des utilisateurs.
- La réparation et la maintenance de matériels informatique.

Chapitre 1. Présentation Générale de l'entreprise et Cadre général de projet

- La prévision des consommables (Matériels informatiques et bureautiques).
- Préparation et présentation multimédia de la direction informatique et les autres divisions.

Division centre traitement informatique Gafsa dont la mission est :

- La maintenance des applications comme l'application de la comptabilité générale analytique, la mutuelle, la paie, les déclarations fiscales.
- Un processus d'exploitation qui englobe :
 - La saisie des données ressources humaines, des données comptables.
 - La préparation et l'application des consignes indiquées dans le cahier d'exploitation.
 - Le contrôle des tâches réalisées.

D'autre part, il existe autres tâches que fait partie de la mission de la direction informatique mais n'appartient à aucune entité dans l'organigramme tels que :

- L'administration des systèmes d'exploitation, réseaux logiciels et base de données.
- La surveillance du réseau et la sécurité informatique.
- Identifier et résoudre les problèmes des utilisateurs.
- L'assistance technique des équipements du Data Center.

1.3.6 L'évolution de la société :

La CPG figure parmi les plus importants producteurs de phosphates, occupant la cinquième place mondiale avec une production de presque huit millions de tonnes en 2009. En 2014, la production a chuté à cinq millions de tonnes et la Tunisie devient le huitième producteur mondial. [Françoise Auvray et Francis Auvray, « Les gisements des phosphates de Gafsa, Tunisie », Saga Information, no 325, mars 2013, p. 8-16, consulté le 18 mai 2013.]

Durant ces dernières années, des mouvements de protestation organisés par des centaines de chômeurs dans les villes où l'implantation de différents secteurs de la CPG ont paralysé le système d'extraction du phosphate qui a causé une crise économique, un blocage de production, perturbations dans tous les secteurs de la CPG telles que l'informatique, l'électrique, le mécanique ...

Plus récemment, en 2020, la production de phosphates a chuté de 23 pourcent par rapport à 2019 atteignant un des niveaux les plus bas depuis 2011, avec seulement (3,14 pourcent millions de tonnes) produites, contre 3,85 millions en 2019, 2,8 millions en 2018, et 3,9 millions en 2017.

La société s'est trouvée dans une situation inconfortable à cause de cette descente aux enfers de la production des phosphates. En 2021 et avec l'apparition de la quatrième révolution industrielle, la CPG tend vers une évolution et un développement de secteurs avec une orientation vers DevOps, afin de récupérer leur place mondiale en production de phosphates et pour améliorer les performances du travail.

1.4 Objectif de projet :

Notre objectif à travers ce projet étant d'abord d'adapter à la culture DevOps en appliquant les bonnes pratiques de cette nouvelle approche de développement logiciel dans C.P.G vise à changer sa stratégie en passant vers cette nouvelle stratégie qui a comme objectifs d'accélérer et automatiser les différentes étapes du cycle de vie des logiciels pour garantir des logiciels de qualité et donc une meilleure satisfaction des utilisateurs finaux. De plus, nous avons comme objectif d'atteindre un meilleur développement à travers la mise en œuvre d'un système de déploiement continue. Ce chapitre consiste à définir le contexte du projet proposé par la société d'accueil où on exposera l'analyse des besoins et les spécifications du travail demandé. Cette partie est indispensable afin d'aboutir notre objectif. Nous commençons par une étude de l'existant pour déterminer et critiquer les points de faiblesse de l'ancien procédé puis proposer des solutions adéquate

1.5 Analyse de l'existant :

Actuellement, la Direction Centrale du Contrôle de Gestion, Informatique et Organisation applique Devops mais sans avoir de rapport de dockérisation. L'équipe opérationnelle intervient après afin de livrer une version finale sans avoir contrôlé le travail déployé. Il n'existe aucun moyen d'automatisation pour ce processus, alors l'automatisation de ces taches devient un besoin ainsi que la synchronisation de l'équipe de cette direction vue que le temps perdu est remarquable et il a un impact sur la productivité.

1.6 La critique de l'existant :

Le temps du cycle d'un projet, la date de livraison ainsi que la qualité de code sont des facteurs importants pour chaque entreprise. On trouve dans certaines entreprises des difficultés pour la gestion des équipes spécialement la relation entre l'équipe de développement et l'équipe opérationnelle (administrateur système) .L'objectif principal d'une équipe d'administrateurs est de garantir la stabilité des systèmes. La meilleure manière d'atteindre cet objectif est de contrôler sévèrement la qualité des changements qui sont apportés aux systèmes qu'ils maintiennent. De son côté, l'équipe développement a pour objectif d'apporter les changements nécessaires au moindre coût et le plus vite possible. Par suite, cette séparation des charges entre les deux types d'équipes a rapidement mené à un conflit perpétuel du fait de l'incompatibilité des objectifs respectifs. Ceci peut être illustré en considérant les 3 contraintes de la gestion de projet qui sont le coût, la qualité et le temps

1.7 Solution :

Dans cette entreprise, le souci primordial serait donc d'avoir un procédé qui permet d'automatiser le processus d'intégration, des tests et de contrôle de qualité de code ainsi que la livraison continue pour les applications, améliorer la communication et la collaboration dans la culture du développement logiciel. La culture du développement se concentre principalement sur la performance de l'entité plutôt que sur les objectifs individuels, qui sont communs à d'autres entreprises.

D'où la nécessité d'une méthode permettant, dans un premier temps, de concevoir et de déployer rapidement une application de qualité, puis de faire en sorte que des modifications plus ou moins importantes puissent être disponibles en quelques heures ou quelques jours. Donc les défis de mon projet sont de création des images et d'automatisation.

Dans une présentation à la conférence O'Reilly Velocity 2009, Allspaw et Hammond ont proposé d'intégrer le développement et les opérations dans une infrastructure automatisée unique avec un contrôle de version partagé et une construction et un déploiement en une seule étape.

[DevOps-The-Ultimate-Beginners-Guide-to-Learn-DevOps-Step-by-Step]

L'objectif de mon projet consiste instaurer et configuration des outils DevOps en la mise en place d'un pipeline d'intégration et de déploiement continu. Ce projet englobe 2 grands axes :

- Intégration continue.
- Déploiements continus.

En fait, DevOps est là pour rester dans un avenir prévisible jusqu'à ce qu'une nouvelle méthodologie ou technologie vienne le remplacer et perturber son espace. C'est la progéniture du développement logiciel agile né de la nécessité de suivre la vitesse accrue des logiciels grâce à des méthodologies agiles. Nous assistons déjà à des signes de maturation et d'évolution de DevOps vers quelque chose de plus avancé et à la mode.

[DevOps-The-Ultimate-Beginners-Guide-to-Learn-DevOps-Step-by-Step].

La figure ci-dessous illustre ma solution que j'ai proposée :

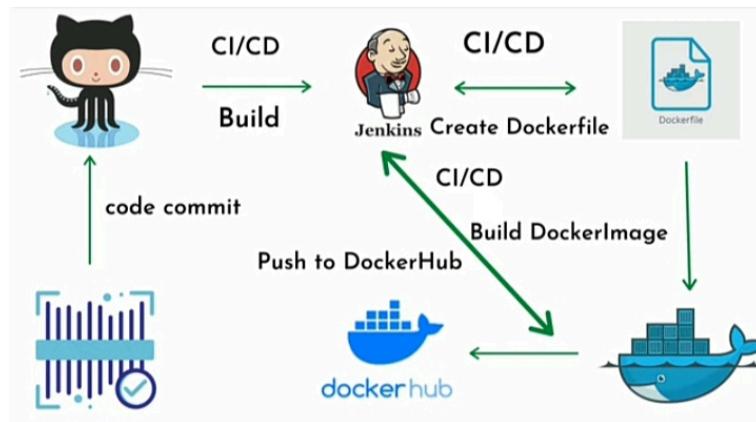


FIG. 1.5 : Architecture de solution proposée.

1.8 Conclusion :

Au cours de ce premier chapitre, Nous avons commencé par présenter l'organisme d'accueil C.P.G. Puis nous avons étudié l'existant en citant la critique de l'existant. Finalement, Nous avons expliqué solution proposée en présentant nos objectifs. Dans le chapitre suivant, nous détaillerons, clairement, la notion de DevOps, la virtualisation traditionnelle et la conteneurisation et nous détaillons la notion de l'intégration continue/ le déploiement continu et en fin nous critiquerons les outils utilisés.

Chapitre 2

La technologie DevOps

2.1 Introduction

Dans ce chapitre nous détaillerons, clairement, la notion de DevOps, la virtualisation traditionnelle et la conteneurisation et nous détaillons la notion de l'intégration continue/le déploiement continu et finalement nous critiquerons les outils utilisés.

2.2 C'est quoi DevOps :

DevOps est une méthodologie de développement de logiciels qui vise à combiner les équipes de développement (Dev) et d'exploitation (Ops) au sein d'une organisation pour permettre une livraison de logiciels plus efficace et rationalisée. Il s'agit d'un ensemble de pratiques et d'outils conçus pour automatiser et optimiser le processus de développement et de déploiement de logiciels.

DevOps cherche à briser les silos traditionnels entre le développement logiciel et les opérations en favorisant la collaboration et la communication entre les deux équipes. Il se concentre sur l'amélioration de la vitesse, de la qualité et de la fiabilité de la livraison de logiciels en automatisant les tâches répétitives, en permettant une intégration et une livraison continues et en utilisant des boucles de rétroaction pour identifier et résoudre les problèmes tôt dans le processus de développement.

Changer les habitudes de travail est toujours compliqué et demande un certain investissement.

En conséquence, lorsqu'une organisation adopte de nouvelles technologies, méthodologies ou approches, cette adoption doit être motivée par un besoin métier. Pour développer une étude de rentabilité à l'adoption de DevOps, vous devez comprendre le besoin métier associé.

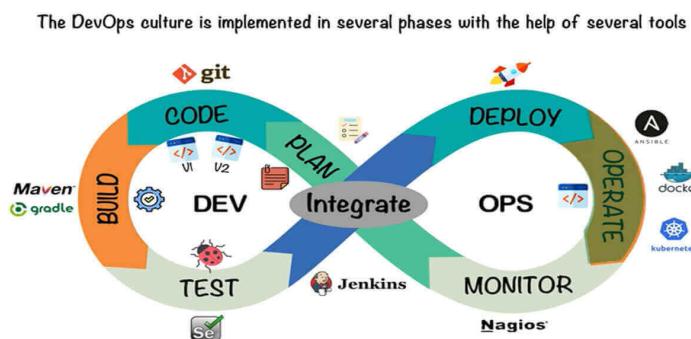


FIG. 2.1 : La technologie DevOps.

Certaines pratiques clés de DevOps incluent l'intégration continue (CI), la livraison continue (CD), l'infrastructure en tant que code (IaC), ainsi que la surveillance et la journalisation. DevOps met également l'accent sur une culture d'amélioration continue et d'apprentissage, en mettant l'accent sur la création de valeur pour les clients de manière rapide et efficace. Pour mener à bien ses missions, l'ingénieur DevOps doit avoir plusieurs outils à sa disposition :

- Les outils de gestion de code source.
- Les outils d'intégration et de déploiement continus CI/CD.
- Les outils de gestion de configuration.
- Les outils d'infrastructure comme code IaC.
- Les outils de conteneurisation.
- Les orchestrateurs de conteneurs logiciels.
- Les outils de monitoring et alerting.



FIG. 2.2 : DevOps.

2.3 Description du besoin métier pour DevOps :

[devOps pour les Nuls®; 2ème Edition Limitée IBM]

Les organisations veulent créer des applications ou services innovants pour résoudre les problèmes métier. L'intention peut être de résoudre des problèmes métier internes (tels que créer un système de gestion de la relation client plus performant) ou d'aider leurs clients ou utilisateurs finaux (en fournissant une nouvelle application mobile). Cependant, de nombreuses organisations ne mènent pas à bien leurs projets logiciels, et leurs échecs sont généralement liés aux défis inhérents au développement et à la livraison des logiciels. Bien que la majorité des entreprises aient conscience que le développement et la livraison des logiciels sont des activités essentielles, une étude récente d'IBM dans le secteur indique que seulement 25 pourcent d'entre elles ont le sentiment que leurs équipes sont efficaces. Ces échecs dans la réalisation se traduisent par des pertes d'opportunités commerciales.

Ces difficultés sont amplifiées par une évolution majeure des types d'applications que les entreprises doivent fournir, des systèmes d'enregistrement aux systèmes d'engagement :

- Systèmes d'enregistrement : les applications logicielles traditionnelles sont de grands systèmes qui fonctionnent comme des systèmes d'enregistrement contenant d'énormes volumes de données et/ou de transactions, et qui sont conçus pour être très fiables et stables. Comme ces applications n'ont pas besoin d'être modifiées fréquemment, les entreprises peuvent satisfaire leurs clients ou leurs propres besoins métier en fournissant une ou deux nouvelles versions majeures chaque année.
- Systèmes d'engagement : avec l'avènement des communications mobiles et l'évolution des applications Web, les systèmes d'enregistrement sont complétés par des systèmes d'engagement auxquels les clients peuvent accéder directement et utiliser pour interagir avec l'entreprise. Ces applications doivent être simples à utiliser, très performantes et pouvoir être modifiées rapidement pour répondre à l'évolution des besoins des clients et des contraintes du marché.

Comme les systèmes d'engagement sont utilisés directement par les clients, l'expérience utilisateur, la rapidité de livraison et l'agilité revêtent une importance extrême. En d'autres termes, ils nécessitent d'adopter une approche DevOps. Les systèmes d'engagement ne sont pas des systèmes isolés et ils sont souvent liés à des systèmes d'enregistrement. Par conséquent, lorsque les systèmes d'engagement changent, les systèmes d'enregistrement doivent changer aussi. En fait, tout système nécessitant une mise à disposition rapide des innovations requiert DevOps. Ces innovations sont issues principalement des technologies émergentes, telles que le Cloud Computing, les applications mobiles, le Big Data et les réseaux sociaux, qui peuvent concerner tous les types de systèmes.

2.4 Chemins d'adoption de DevOps :

[[devOps pour les Nuls®; 2ème Edition Limitée IBM]]

Une architecture de référence fournit un modèle d'une solution éprouvée en utilisant un ensemble de méthodes et fonctionnalités préférées. Une architecture de référence peut fournir des fonctionnalités via divers composants. Chacune de ces fonctionnalités peut être couverte par un seul composant ou un groupe de composants fonctionnant conjointement. Par conséquent, vous pouvez voir l'architecture de référence DevOps, représentée dans figure 4, du point de vue des principales fonctionnalités qu'elle est supposée fournir. Au fur et à mesure que l'architecture définie se concrétise, ces fonctionnalités seront supportées par un groupe de personnes compétentes, de pratiques définies et d'outils d'automatisation.

L'architecture de référence DevOps de l'illustration propose les quatre groupes de chemins d'adoption suivants :

- Pilotage.
- Développement/test.
- Déploiement.
- Opération.

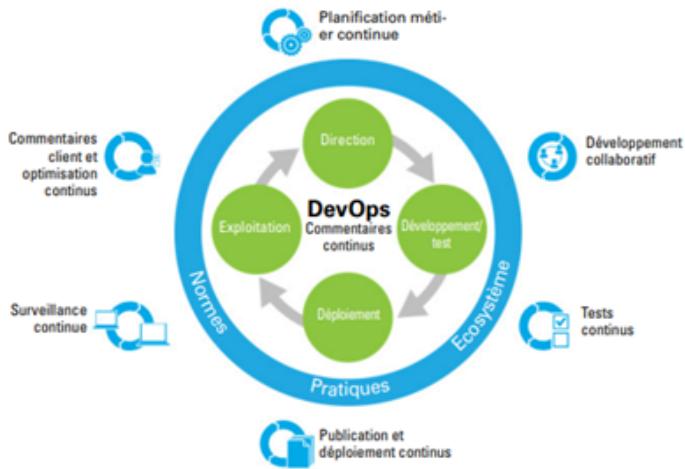


FIG. 2.3 : L'architecture de référence DevOps.

2.5 Les outils de Devops :

2.5.1 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.

2.5.2 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

2.5.3 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.

2.5.4 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.

2.5.5 La gestion des configurations :

La gestion de la configuration consiste à automatiser les activités importantes et répétitives dans un environnement IT. La gestion de la configuration traite les tâches qui s'étendent à plusieurs machines. Ces tâches peuvent inclure des installations de logiciels, des mises à niveau et des mises à jour, la gestion des correctifs, la conformité à la sécurité, la gestion des utilisateurs, etc.

2.5.6 La conteneurisation :

Un conteneur c'est un environnement d'exécution léger, et une alternative aux méthodes de virtualisation traditionnelles basées sur les machines virtuelles. Pour exécuter les applications, il est nécessaire d'exploiter des packages, des bibliothèques et divers composants logiciels. Pour exploiter ces ressources en isolant une application, nous utilisons depuis longtemps les machines virtuelles. Les machines virtuelles permettent de séparer les applications entre elles sur un même système, et de réduire les conflits entre les composants logiciels et la compétition pour les ressources. La nouvelle alternative qui est le conteneur partage le kernel de son OS avec d'autres conteneurs. C'est une différence avec une machine virtuelle, qui utilise un hyperviseur pour distribuer les ressources hardware. Le conteneur regroupe tous les composants systèmes nécessaires à l'exécution du code, pour ne pas peser aussi lourd d'un OS complet. De même, un conteneur est plus léger et plus simple qu'une machine virtuelle et peut donc démarrer et s'arrêter rapidement. Il est donc plus réactif, et adaptable aux besoins fluctuants liés au scaling d'une application. Dernier point fort de la conteneurisation : contrairement à un hyperviseur, un moteur de conteneur n'a pas besoin d'émuler un système d'exploitation complet. Le conteneur offre donc de meilleures performances qu'un déploiement sur machine virtuelle. La figure ci-dessous, illustre la différence au niveau de l'architecture entre les machines virtuelles et les conteneurs.

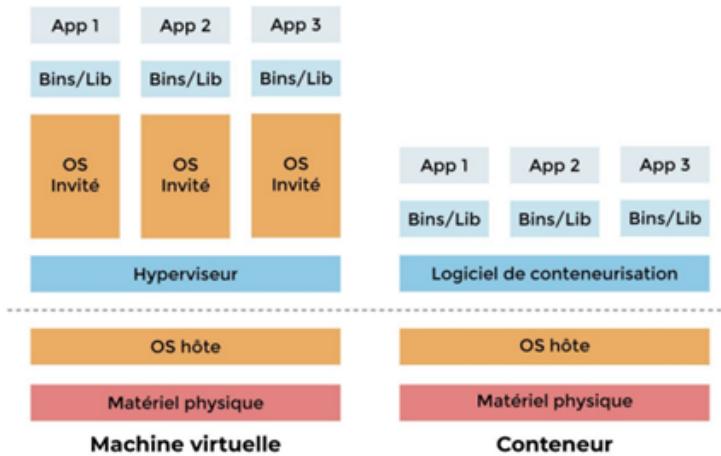


FIG. 2.4 : Conteneur vs machine virtuelle.

2.5.7 Le GitOps

Le GitOps désigne un ensemble de pratiques de gestion de l'infrastructure et des configurations d'applications qui reposent sur l'utilisation de Git. Git est utilisé comme unique source de vérité pour la formalisation déclarative de l'infrastructure et des applications.

Selon les principes de GitOps, l'approvisionnement et le déploiement de l'infrastructure sont gérés automatiquement par des requêtes « pull » dans Git. Avec le GitOps, nous devons aussi déclarer l'état souhaité du système. Nous pouvons en outre contrôler dans Git les versions de l'ensemble du code source et des fichiers de configuration de notre infrastructure.

Le déploiement d'une nouvelle version dans le cadre d'un workflow GitOps s'effectue à l'aide d'une requête « pull » qui vient modifier l'état déclaré du cluster. L'opérateur GitOps, situé entre le pipeline GitOps et le système d'orchestration, détecte la validation et déclare le nouvel état reçu de Git.

Cet opérateur compare l'état souhaité qui a été déclaré dans le référentiel avec l'état réel de l'infrastructure déployée, et met à jour l'infrastructure en cas de différence. La figure ci-dessous, représente la boucle de feedback et de contrôle continu pour des différents états d'une application dans un modèle GitOps.

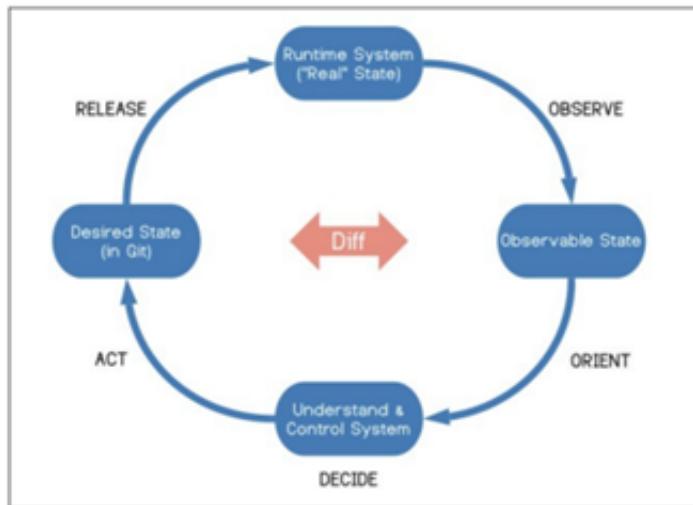


FIG. 2.5 : GitOps workflow.

2.5.8 Ansible :

Ansible automatise des tâches courantes telles que le provisioning dans le cloud, 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.

Vous trouverez ci-dessous le schéma simple d'une architecture d'Ansible :

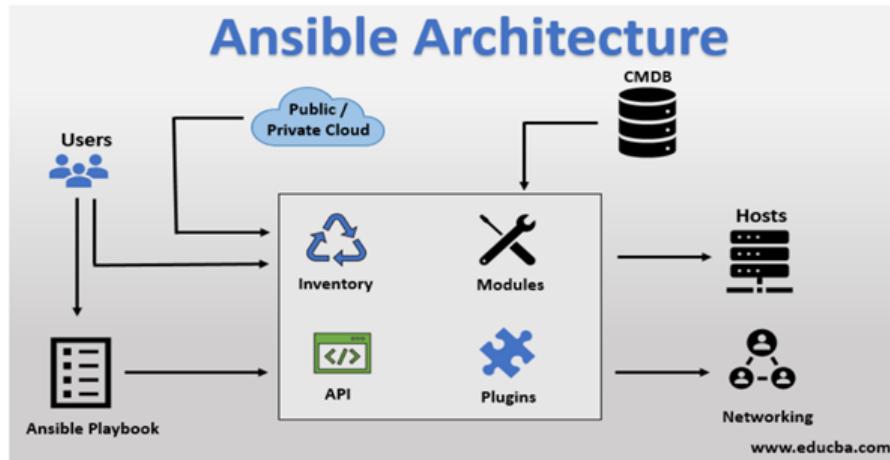


FIG. 2.6 : Architecture de Ansible.

À partir du diagramme ci-dessus, la visibilité entre dans l'image :

- Users : les utilisateurs qui créent un playbook Ansible ont une connexion directe avec le moteur d'automatisation ansible.
- Ansible Playbook : il interagit également avec le moteur d'automatisation ansible et la base de données de gestion de la configuration
- Public or Private Cloud : ils aident à interagir avec tous les modules et API avec cela, mais aussi avec l'ensemble du cloud, ce qui prouve qu'il dispose également de mesures de sécurité.
- Inventory : L'inventaire qui fait partie du moteur d'automatisation aide à l'approvisionnement et à l'approvisionnement interne à l'aide de l'automatisation.
- API : Cela aide à créer l'API nécessaire à l'interaction des modules de bout en bout.
- Modules : les modules sont directement exécutés à l'aide de playbooks, les modules peuvent contrôler tous les services, packages, formation de cloud AWS, etc.
- Plugins : Tout le cache nécessaire, l'objectif de journalisation, le fonctionnement d'ansibles aident à créer le noyau d'ansible augmenté.
- Networking : elle aide à automatiser différents réseaux qui utilisent toutes les trames sans agent et génèrent des configurations utiles.
- Hosts : les hôtes font ici référence aux machines telles que Linux ou Unix qui sont automatisées à l'aide d'Ansible.
- CMDB (Configuration Management Database) : C'est une sorte de référentiel qui consiste en un réseau entier d'ordinateurs d'infrastructure opérationnelle ou informatique.

2.5.9 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.

2.5.10 L'infrastructure as code : (IaC) :

L'Infrastructure comme Code, ou IaC est un type de configuration informatique permettant aux ingénieurs DevOps de gérer et d'approvisionner automatiquement l'infrastructure informatique par un code sans passer par des processus manuels.

2.5.11 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.

2.5.12 Docker :

Docker est un système en vogue dans le monde DevOps. Historiquement, la montée en puissance des serveurs a fait émerger des systèmes de virtualisation comme VMWare ou encore HyperV. L'approche était donc de créer au sein d'un même serveur plusieurs machines virtuelles, puis de dédier ces machines virtuelles à un fonctionnement particulier : telle VM servirait pour le serveur SQL, telle autre comme serveur de messagerie, etc. Le problème c'est le "gaspillage" des ressources que cela induit. D'une part, cela oblige à installer autant de systèmes d'exploitation qu'il y a de machines virtuelles. Nous ne parlons pas du coût des licences que cela suppose également (pour le monde Windows en premier, mais aussi Linux si nous optons pour des OS payants). Ces OS prennent de la place en disque et ont besoin d'un minimum de mémoire et de processeurs pour fonctionner. Autant de ressources qui auraient pu n'être allouées qu'une fois à la machine source et être libérées pour nos applications métier. Il faut aussi pouvoir gérer des parcs entiers de machines physiques, ce qui ajoute à la lourdeur du système. Docker aborde les choses de façon différente en introduisant l'idée de conteneur. Cela ne remplace pas le besoin de virtualisation bien entendu, mais on pourrait qualifier cette approche de complémentaire (plutôt que concurrente) à celle-ci.

Vous trouverez ci-dessous le schéma simple d'une architecture Docker.

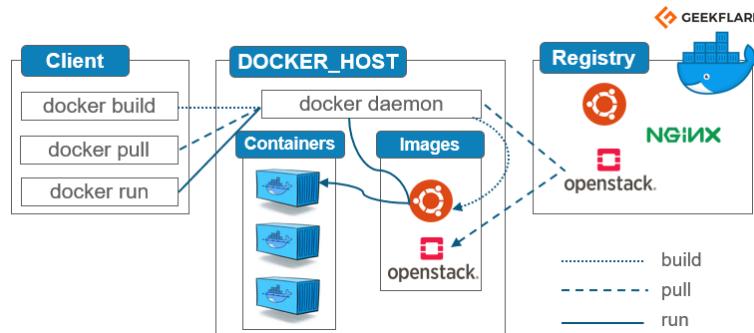


FIG. 2.7 : L'architecture de Kubernetes.

Les composants d'une architecture docker :

- Docker Engine : C'est la partie centrale de tout le système Docker. Docker Engine est une application qui suit architecture client-serveur. Il est installé sur la machine hôte. Il existe trois composants dans le moteur Docker :
 - Server : C'est le démon docker appelé docker. Il peut créer et gérer des images de docker, Conteneurs, réseaux, etc.
 - API REST : Il est utilisé pour indiquer au démon docker ce qu'il doit faire.
 - Interface de ligne de commande (CLI) : C'est un client qui sert à entrer commandes docker.
- Client Docker : Les utilisateurs de Docker peuvent interagir avec Docker via un client. Lorsqu'une commande docker s'exécute, le client les envoie au démon dockerd, qui les exécute. L'API Docker est utilisée par les commandes Docker. Le client Docker peut communiquer avec plus d'un démon.
- Registres Docker : C'est l'emplacement où les images Docker sont stockées. Il peut s'agir d'un registre docker public ou d'un registre docker privé. Docker Hub est l'emplacement par défaut des images docker, le registre public de ses magasins. Vous pouvez également créer et exécuter votre propre registre privé.

Lorsque vous exécutez des commandes docker pull ou docker run, l'image docker requise est extraite du registre configuré. Lorsque vous exécutez la commande push docker, l'image docker est stockée dans le registre configuré.

Il existe trois composants dans le moteur Docker : Server : C'est le démon docker appelé docker. Il peut créer et gérer des images de docker. ... API REST : Il est utilisé pour indiquer au démon docker ce qu'il doit faire. Interface de ligne de commande (CLI) : C'est un client qui sert à entrer commandes docker.

2.5.13 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. Il existe plusieurs systèmes d'orchestration de conteneurs disponibles aujourd'hui, mais Kubernetes est devenu plus populaire car il est rentable et offre de nombreuses options pour personnaliser les déploiements et prend en charge de nombreux fournisseurs différents. Il est pris en charge par tous les principaux fournisseurs de services de cloud public tels que GCP, Azure, AWS, Oracle Cloud, Digital Ocean, etc.

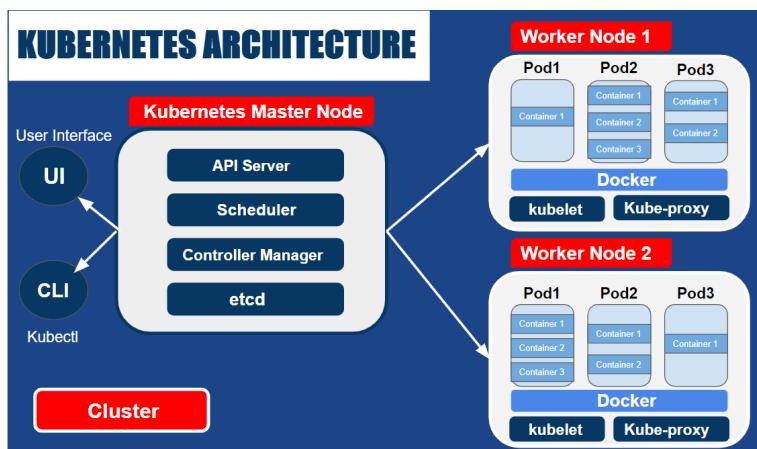


FIG. 2.8 : L'architecture de Kubernetes.

Kubernetes suit l'architecture maître/esclave. Nous avons donc les nœuds maîtres et les noeuds travailleurs. Les noeuds maîtres gèrent les nœuds de travail et forment ensemble un cluster. Un cluster est un ensemble de machines appelées nœuds. Un cluster Kubernetes a au moins un nœud maître et un nœud de travail. Cependant, il peut également y avoir plusieurs clusters.

2.5.14 La supervision et 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.

La supervision et le monitoring consistent à récolter un maximum de données, de mettre et de suivre des indicateurs précis. L'analyse de ces données va permettre de mesurer l'impact de celles-ci sur notre application et notre infrastructure. La supervision et le monitoring permettent de surveiller en temps réel l'évolution et les performances de l'ensemble.

L'objectif pour l'entreprise est de pouvoir anticiper toute perte de productivité, de gagner en agilité et de mettre en place rapidement des mesures correctives en cas d'anomalies ou de dysfonctionnement des applications. La supervision dans le Devops est essentielle

pour réagir rapidement en cas d'erreurs. Tout comme le Devops, la supervision doit être automatisée pour pouvoir récolter efficacement toutes les données à chacune des étapes.

2.5.15 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.

2.5.16 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 Logstash, le stack rassemble les données et rend possibles les alertes mails ou SMS.

2.5.17 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.

2.6 Les trois processus DevOps :

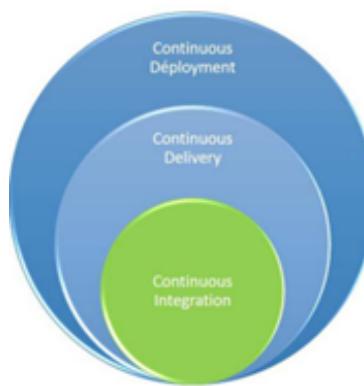


FIG. 2.9 : Processus DevOps.

2.6.1 L'intégration continue :

L'intégration continue ou Continuous Integration processus orienté consistant à compiler, tester et déployer sur un environnement d'intégration. Le but est de tester aussi

souvent et autant que possible les non-régressions du livrable pour détecter les bugs le plus tôt possible. La plupart du L'intégration continue ou Continuous Integration est un orienté études compiler, tester et sur un environnement est de tester uvent et autant que régressions du tecter les bugs le plus tôt possible. La plupart du travail est réalisé par des outils de test. Le déploiement sur la plateforme d'intégration devient simple et peut être réalisé par les études sans faire intervenir l'exploitation.

2.6.2 La livraison continue :

La livraison continue ou Continuous Delivery est un processus d'intégration et de production. Le but est de compiler, tester et livrer une application à chaque étape de son cycle de vie (recette, pré-production, répétition, production). Cette étape est réalisée après validation des tests effectués en intégration. La phase de test correspond aux tests fonctionnels du livrable. Le passage d'un état à l'autre est entièrement automatisé, c'est pourquoi le livrable doit être constitué de tel sorte qu'il soit déployable en production dès la mise en recette.

2.6.3 Le déploiement continu :

Le déploiement continu ou Continuous Deployment est un processus de production. Le but est de compiler, tester, et déployer une application en production. Le Continuous Deployment nécessite que les processus de Continuous Integration et de Continuous Delivery aient été réalisés avec succès. Le déploiement est réalisé par un simple « press button ». Après le déploiement, il doit être possible de mesurer les éventuels impacts à l'aide d'outils de mesure de la performance et d'outils de supervision. En cas de problème, un processus automatisé de retour arrière peut être exécuté.

2.6.4 L'amélioration continue :

Ce processus ne fait pas partie des processus DevOps mais il semble essentiel pour POESI. Le processus d'amélioration continue consiste à améliorer continuellement les trois processus précédents. Ce processus est représenté comme un ensemble d'indicateurs capable de mesurer le « Time to market » et la qualité des processus précédents dans le but de faire ressortir des axes d'amélioration.

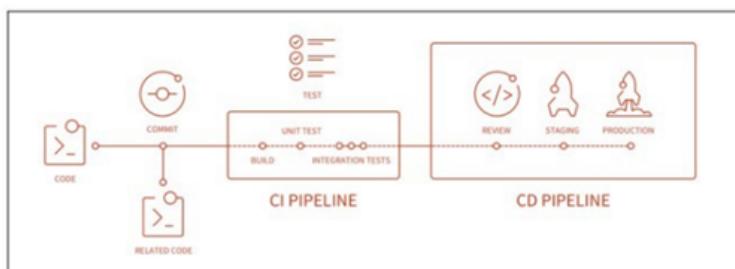


FIG. 2.10 : La pipeline CI/CD

2.7 Creation de pipeline de livraison :

[devOps pour les Nuls®; 2eme Edition Limitee IBM] Lequipe IBM SWG decida de creer un pipeline de distribution ameliorant lusage des outils comme ‘services’ afin de permettre aux developpeurs de valider le code, de le tester et de le deployer dans un environnement de production en quelques 60 minutes au lieu d’un ou deux jours auparavant. Ce processus a permis de reduire les retours en arriere et d’optimiser la productivite. Lequipe de deploiement determina qu’un pipeline de distribution continue necessitait d’appliquer les meilleures pratiques suivantes :

- Executer les tests au plus tot (« Shift-left ») et automatiser au maximum.
- Utiliser le meme mecanisme de deploiement sur tous les environnements.
- S’efforcer de maintenir constamment lapplication en d’tre livre aux utilisateurs.
- Traiter linfrastructure comme une infrastructure programmable.

Une meilleure pratique essentielle de limplemetation d’un pipeline de livraison continue est de “traiter linfrastructure comme une infrastructure programmable”. Cela implique que le developpeur peut crevoir des scripts pour configurer linfrastructure requise pour lapplication dans le code de cette dernire. Auparavant, cela tait ralised generallement par un administrateur systeme ou une personne chargee des operations, mais desormais avec le controle et les gains dfficacite que cela apporte, le developpeur peut le faire directement. Puppet, Chef et IBM UrbanCode Deploy with Patterns sont des exemples des nouvelles categories doutils dautomatisation qui font de linfrastructure programmable une ralite concrete. Desormais, lequipe dIBM SWG traite son infrastructure comme une infrastructure programmable et suit ces meilleures pratiques :

- Traiter les definitions de modoles de deploiement, les packages de scripts et les services comme du code.
- Versionner tout.
- Automatiser le deploiement des modoles de topologie vers le Cloud.
- Gerer les versions de modoles dans plusieurs environnements Cloud.
- Automatiser les tests des modoles.
- Nettoyer les ressources de catalogue pour eviter la propagation.

2.8 Les dix mythes de DevOps :

[devOps pour les Nuls®; 2eme Edition Limitee IBM]

- DevOps s’adresse uniquement aux entreprises « nees sur le Web ».

- DevOps vise apprendre aux équipes des opérations à programmer.
- DevOps s'adresse uniquement aux équipes de développement et des opérations.
- DevOps n'est pas fait pour les entreprises ITIL.
- DevOps n'est pas adapté aux secteurs réglementés.
- DevOps n'est pas fait pour les développements externalisés.
- Das de DevOps sans Cloud.
- DevOps ne s'applique pas aux grands systèmes complexes.
- DevOps est uniquement une affaire de communication.
- DevOps est synonyme de déploiement continu des modifications.

2.9 Présentation d'une solution DevOps :

On peut imaginer une solution composée de trois outils qui permettent de suivre un déploiement applicatif tout au long de son cycle de vie.

Elle permet :

- D'automatiser le processus de « Continuous Delivery ».
- De rendre répétable tout déploiement d'application et quel que soit la technologie.
- De faciliter la collaboration entre les développeurs et les opérationnels.
- D'établir des standards, des normes et des protocoles pour assurer la gouvernance et maîtriser les coûts.
- D'afficher une série d'indicateurs et des rapports personnalisables en fonction des besoins de chacun.

2.10 MLOps : Le DevOps appliqué aux projets de Machine Learning

[<https://datascientest.com/mlops-le-devops-applique-aux-projets-de-machine-learning>]

Dans cette partie nous allons nous intéresser à l'application de cette approche dans le cadre de problématiques de Machine Learning : on parle de MLOps.

Le DevOps permet une réduction du time to market ainsi qu'une amélioration de la qualité des produits logiciels grâce à 3 principes :

- Désilotage : les équipes de développement et les opérations doivent travailler de concert, vers un objectif commun, en partageant toute l'information.
- Automatisation : Tout ce qui est automatisable doit l'être : build, test, déploiement. Ceci dans un double objectif de réduction du temps nécessaire au déploiement et du nombre de non-qualités, les étapes manuelles étant propices aux erreurs humaines.
- Monitoring : récolter et surveiller les métriques importantes, à la fois côté métier (nombre d'utilisateurs connectés, nombre de commandes, etc.) et côté opérations (pourcentage d'utilisation CPU / RAM des serveurs, nombre d'erreurs, etc.)

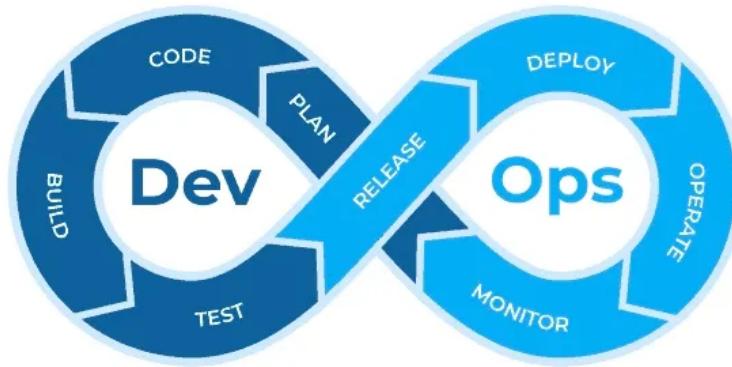


FIG. 2.11 : MLOps de DataScientet.

Suite à l'adoption massive du DevOps et à l'augmentation significative de l'utilisation du Machine Learning et de l'IA en entreprise, les concepts DevOps ont été repris dans les projets data. Si les objectifs et principes restent globalement les mêmes, certaines spécificités propres aux machine learning ont nécessité la création d'une approche propre : le MLOps.

MLOps est une culture et une pratique d'ingénierie ML qui vise à unifier le développement (Dev) et les opérations (Ops) des systèmes de ML.

Afin de comprendre ce qu'est le MLOps, nous allons nous attacher à saisir les différences avec le DevOps classique :

- Développement expérimental : par nature, tous les projets ML suivent un développement non linéaire : différents preprocessing, feature engineering et algorithmes sont testés, jusqu'à ce qu'une performance suffisante soit atteinte. Garder une trace de ce qui a été expérimenté et de ce qui a ou non fonctionné peut s'avérer complexe. C'est pourtant crucial pour réduire le temps de développement, notamment dans le cas de grosses équipes, afin d'éviter de tester à nouveau des approches qui auraient déjà été expérimentées sans succès par un autre data scientist.

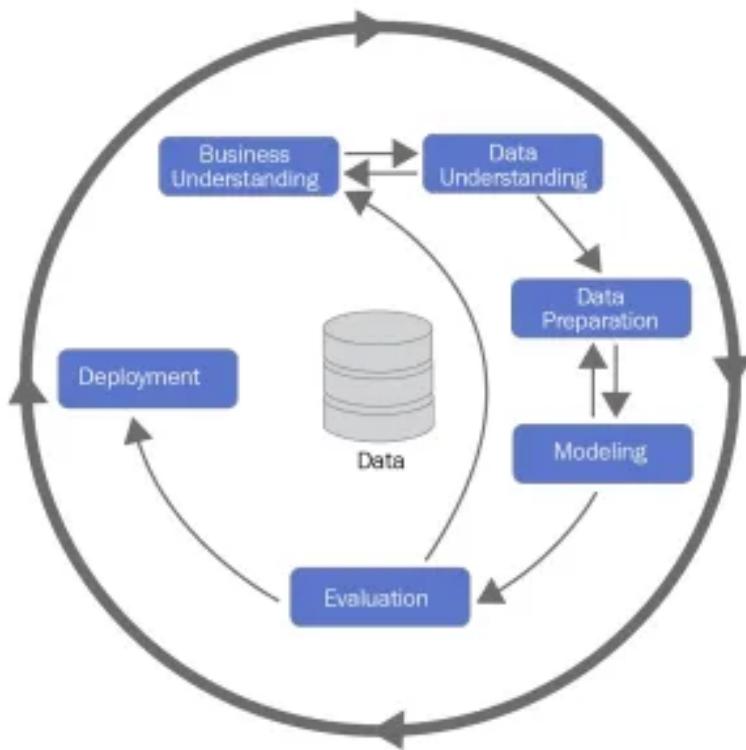


FIG. 2.12 : Étapes du développement d'un modèle de machine learning.

- Dérive conceptuelle : à la différence d'un produit logiciel classique, un système de ML peut voir sa performance varier (et la plupart du temps décliner) au cours du temps, sans qu'il y ait eu de modification dans le code ou l'infrastructure. Ceci est dû au phénomène de dérive conceptuelle, dans lequel les relations entre la variable à prédire et les variables explicatives vont évoluer de manière imprévue. Il est donc indispensable de suivre l'évolution de la performance des modèles de ML en production, et potentiellement d'implémenter une stratégie de réapprentissage automatique afin de « re-calibrer » le modèle lorsque sa performance diminue. On peut notamment citer l'exemple de Google Flu Trends, un projet de Google qui devait permettre d'estimer le nombre de cas de grippe à partir des recherches effectuées sur le moteur de recherche, mais qui montra ses limites en surestimant largement le nombre de cas au bout de quelques années d'exploitations.
- Équipe : Dans la plupart des organisations, les équipes data transverses ou intégrées dans différents départements métier sont relativement jeunes et manquent

de ressources qualifiées pour gérer le déploiement et le maintien en condition opérationnelle de systèmes ML complexes. En effet, ces équipes se composent généralement principalement de data scientists qui se concentrent sur le développement des modèles de machine learning, mais n'ont pas les compétences nécessaires pour gérer le déploiement et la maintenance d'applications complètes. De plus, les équipes data évoluent encore trop souvent en silo, sans communiquer avec les différentes équipes techniques avec lesquelles elles devraient interagir pour mettre en production leurs modèles.

Ces spécificités et complexités inhérentes aux projets data expliquent en partie pourquoi, alors que de plus en plus d'entreprises investissent massivement dans le machine learning et l'IA, très peu de modèles sont effectivement déployés en production. Selon une étude menée par Algorithmia en 2020, 55% des MLOps a donc pour objectif de pallier ces difficultés afin de permettre le déploiement et le run des systèmes ML, et ce en appliquant des principes similaires au DevOps :

- Monitoring : Le monitoring prend encore de l'importance dans la philosophie MLOps, en run (notamment en collectant les métriques de performance des modèles afin de détecter les dérives conceptuelles), comme au stade de développement (pour traquer les différentes expérimentations et leurs résultats).
- Culture : les équipes data doivent également travailler avec les autres équipes techniques (les opérations mais également les équipes développant les produits logiciels avec lesquels les modèles de ML doivent s'intégrer)
- Automatisation : En plus des étapes classiques (build, test, deploy), le MLOps prône l'automatisation des tâches propres à la maintenance des systèmes ML, comme le réapprentissage des modèles. Après que les modèles sont entraînés, ils doivent également être automatiquement déployés.

Pour aider les équipes data à adopter une approche MLOps, de plus en plus d'outils ont été développés. On pourra notamment citer MLflow, Metaflow, Kubeflow.

Mais pour pleinement réussir son virage MLOps il faudra avant tout casser les silos entre les opérations et les équipes data, et s'assurer que celles-ci disposent de ressources nécessaires (Data Engineers, Machine Learning Engineers). En effet, les Machine Learning Engineers sont des profils expérimentés qui disposent à la fois d'une expertise en machine learning (modélisation statistique, deep learning, etc.) mais aussi de compétences en développement logiciel, data engineering, et en passage en production des modèles de ML.

Chapitre 3

Installation des outils DevOps

3.1 Introduction

Dans ce chapitre, on passe à l'aspect pratique. Nous avons plusieurs possibilités pour installer notre serveur d'intégration, il peut s'installer indifféremment sur Windows ou du Linux.

3.2 Ansible :

Ansible est une plate-forme open source qui automatise le provisionnement de l'infrastructure, le déploiement de logiciels et l'exécution de tâches sur plusieurs hôtes simultanément.

3.2.1 Installation de Ansible :

Il y a plusieurs façons d'installer Ansible :

- Via les packages logiciels sur un système Linux.
- Via pip de Python dans un virtualenv ou pas.
- Via les sources officielles (Archives ou Git) maintenues par Red Hat.

On peut installer le logiciel Ansible avec la cmd qui est intégrée dans la figure ci-dessous :

```
wijden@wijden-VirtualBox: ~
Lecture des listes de paquets... Fait
wijden@wijden-VirtualBox:~$ sudo apt install ansible
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets suivants seront mis à jour :
  ansible
  1 mis à jour, 0 nouvellement installés, 0 à enlever et 8 non mis à jour.
  Il est nécessaire de prendre 15.5 Mo dans les archives.
  Après cette opération, 87.6 Mo d'espace disque seront libérés.
Récception de :1 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu kinetic
/main amd64 ansible all 7.3.0-1ppa-kinetic [15.5 MB]
15.5 Mo réceptionnés en 21s (721 ko/s)
(Lecture de la base de données... 250395 fichiers et répertoires déjà installés
.)
Préparation du dépaquetage de .../ansible_7.3.0-1ppa-kinetic_all.deb ...
Dépaquetage de ansible (7.3.0-1ppa-kinetic) sur (5.5.0-1) ...
Paramétrage de ansible (7.3.0-1ppa-kinetic) ...
wijden@wijden-VirtualBox:~$ sudo apt-key adv --keyserver keyserver.ubuntu.com -
--recv-keys 93C4A3FD7BB9C367
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
Executing: /tmp/apt-key-gpghome.auzXoYlZ57/gpg.1.sh --keyserver keyserver.ubuntu.com --recv-keys 93C4A3FD7BB9C367
gpg: clef 93C4A3FD7BB9C367 : « Launchpad PPA for Ansible, Inc. » n'est pas modifiée
gpg: Quantité totale traitée : 1
gpg:           non modifiées : 1
wijden@wijden-VirtualBox:~$
```

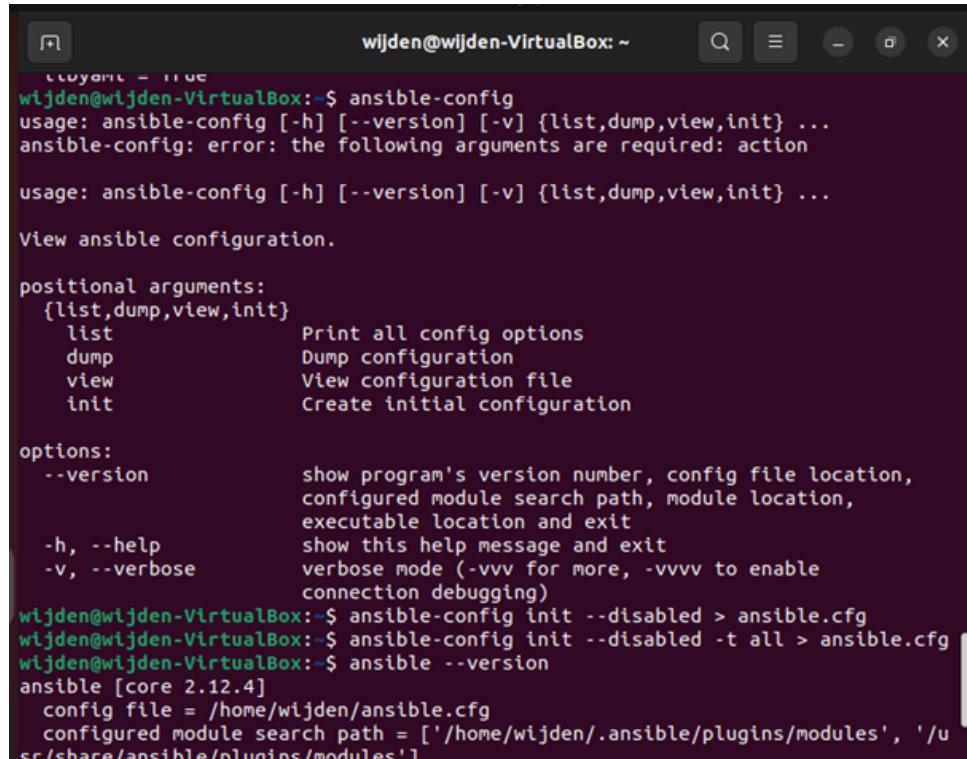
FIG. 3.1 : Installation ansible

Appuye sur Y lorsque on est invité à confirmer l'installation. Notre noeud de contrôle Ansible dispose maintenant de tous les logiciels nécessaires pour administrer vos hôtes. Ensuite, nous verrons comment mettre en place un fichier d'inventaire, afin qu'Ansible puisse communiquer avec vos noeuds gérés.

3.2.2 Configuration d'Ansible :

Pour configurer Ansible il faut exécuter la commande suivante :

- ansible-config



```
wijden@wijden-VirtualBox:~$ ansible-config
usage: ansible-config [-h] [--version] [-v] {list,dump,view,init} ...
ansible-config: error: the following arguments are required: action

usage: ansible-config [-h] [--version] [-v] {list,dump,view,init} ...

View ansible configuration.

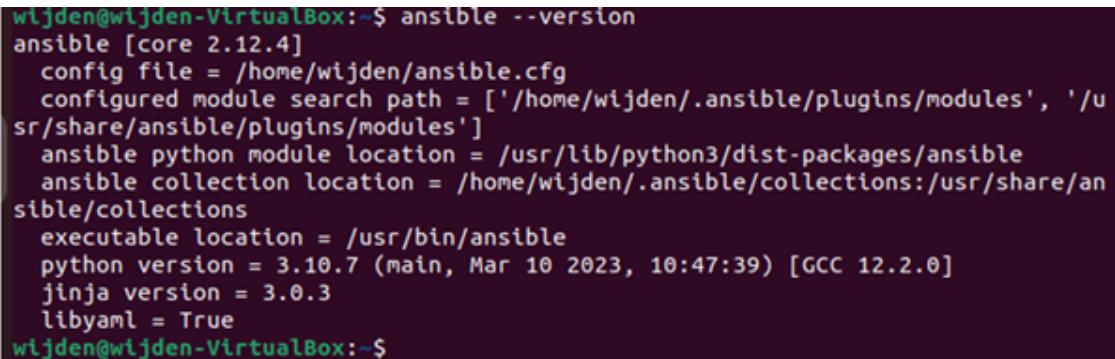
positional arguments:
  {list,dump,view,init}
    list          Print all config options
    dump          Dump configuration
    view          View configuration file
    init          Create initial configuration

options:
  --version      show program's version number, config file location,
                 configured module search path, module location,
                 executable location and exit
  -h, --help     show this help message and exit
  -v, --verbose  verbose mode (-vvv for more, -vvvv to enable
                 connection debugging)
wijden@wijden-VirtualBox:~$ ansible-config init --disabled > ansible.cfg
wijden@wijden-VirtualBox:~$ ansible-config init --disabled -t all > ansible.cfg
wijden@wijden-VirtualBox:~$ ansible --version
ansible [core 2.12.4]
  config file = /home/wijden/ansible.cfg
  configured module search path = ['/home/wijden/.ansible/plugins/modules', '/u
ser/share/ansible/plugins/modules']
```

FIG. 3.2 : Configuration de l'ansible.

3.2.3 Version d'Ansible :

On vérifie la version de Ansible avec la commande suivante :



```
wijden@wijden-VirtualBox:~$ ansible --version
ansible [core 2.12.4]
  config file = /home/wijden/ansible.cfg
  configured module search path = ['/home/wijden/.ansible/plugins/modules', '/u
ser/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/wijden/.ansible/collections:/usr/share/an
sible/collections
  executable location = /usr/bin/ansible
  python version = 3.10.7 (main, Mar 10 2023, 10:47:39) [GCC 12.2.0]
  jinja version = 3.0.3
  libyaml = True
wijden@wijden-VirtualBox:~$
```

FIG. 3.3 : Version installée ansible.

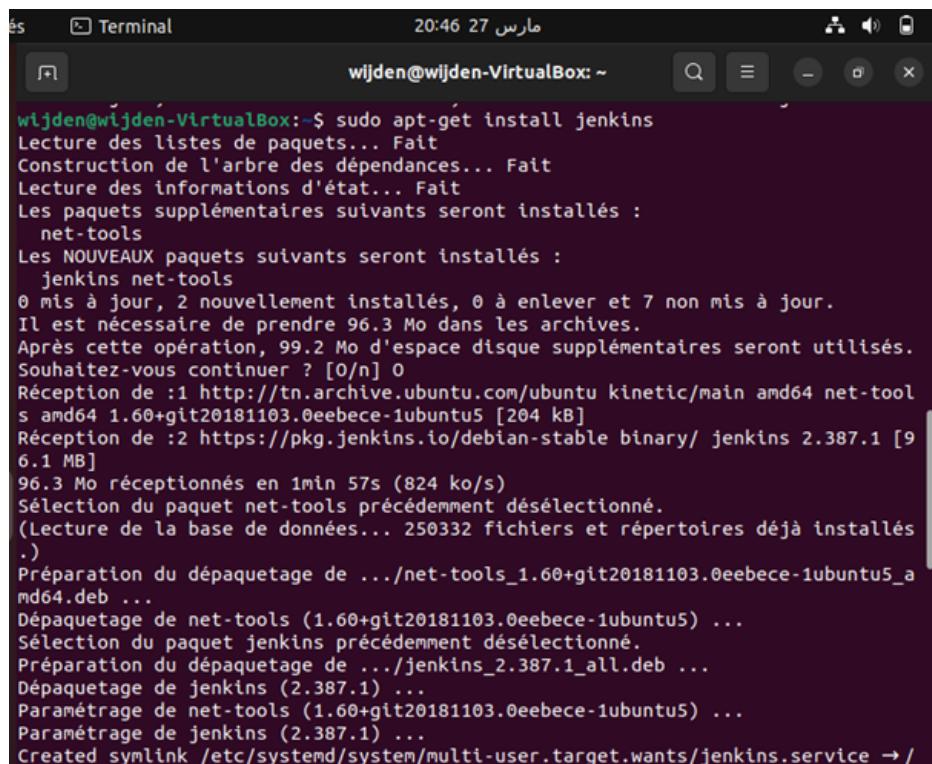
3.3 Jenkins

C'est un outil d'intégration continue. Il fonctionne dans un conteneur de service ou avec son propre serveur Web embarqué. Il s'interface avec des systèmes de gestion de versions tel que Git, CVS, Subversion.

3.3.1 Installation de Jenkins :

Nous installerons Jenkins et ses dépendances avec la commande suivante :

- sudo apt install jenkins



The screenshot shows a terminal window titled "Terminal" with the command "sudo apt-get install jenkins" being run. The output of the command is displayed, showing the progress of the package installation. The terminal window has a dark background and light-colored text. At the top, it shows the user "wijden@wijden-VirtualBox" and the date and time "مارس 27 20:46". The terminal interface includes standard Linux window controls like maximize, minimize, and close buttons.

```
wijden@wijden-VirtualBox:~$ sudo apt-get install jenkins
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  net-tools
Les NOUVEAUX paquets suivants seront installés :
  jenkins net-tools
0 mis à jour, 2 nouvellement installés, 0 à enlever et 7 non mis à jour.
Il est nécessaire de prendre 96.3 Mo dans les archives.
Après cette opération, 99.2 Mo d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [0/n] 0
Réception de :1 http://tn.archive.ubuntu.com/ubuntu kinetic/main amd64 net-tools amd64 1.60+git20181103.0eebece-1ubuntu5 [204 kB]
Réception de :2 https://pkg.jenkins.io/debian-stable binary/ jenkins 2.387.1 [96.1 MB]
96.3 Mo réceptionnés en 1min 57s (824 ko/s)
Sélection du paquet net-tools précédemment désélectionné.
(Lecture de la base de données... 250332 fichiers et répertoires déjà installés .)
Préparation du dépaquetage de .../net-tools_1.60+git20181103.0eebece-1ubuntu5_amd64.deb ...
Dépaquetage de net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Sélection du paquet jenkins précédemment désélectionné.
Préparation du dépaquetage de .../jenkins_2.387.1_all.deb ...
Dépaquetage de jenkins (2.387.1) ...
Paramétrage de net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Paramétrage de jenkins (2.387.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /
```

FIG. 3.4 : Installation de Jenkins

3.3.2 Configuration de Jenkins :

Pour configurer notre installation, consultons Jenkins sur son port par défaut, 8080 en utilisant notre nom de domaine ou l'adresse IP de notre serveur :

http://your_server_ip_or_domain:8080.

Vous devriez voir apparaître l'écran Unlock Jenkins qui affichera l'emplacement du mot de passe initial :

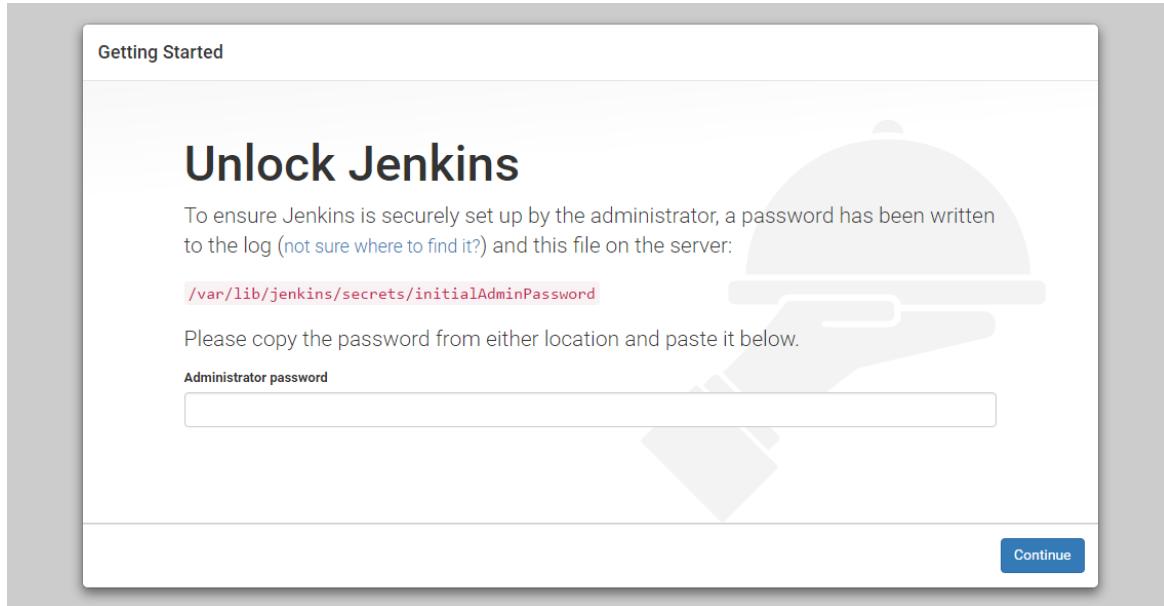


FIG. 3.5 : Ecran unlock Jenkins.

Dans la fenêtre du terminal, utilisez la commande cat pour afficher le mot de passe :

sudo cat /var/lib/jenkins/secrets/initialAdminPassword

Copiez le mot de passe alphanumérique composé de 32 caractères du terminal et collez-le dans le champ Administrator password, puis cliquez sur Continue.

L'écran suivant présente l'option d'installer des plugins suggérés ou de sélectionner des plugins spécifiques :

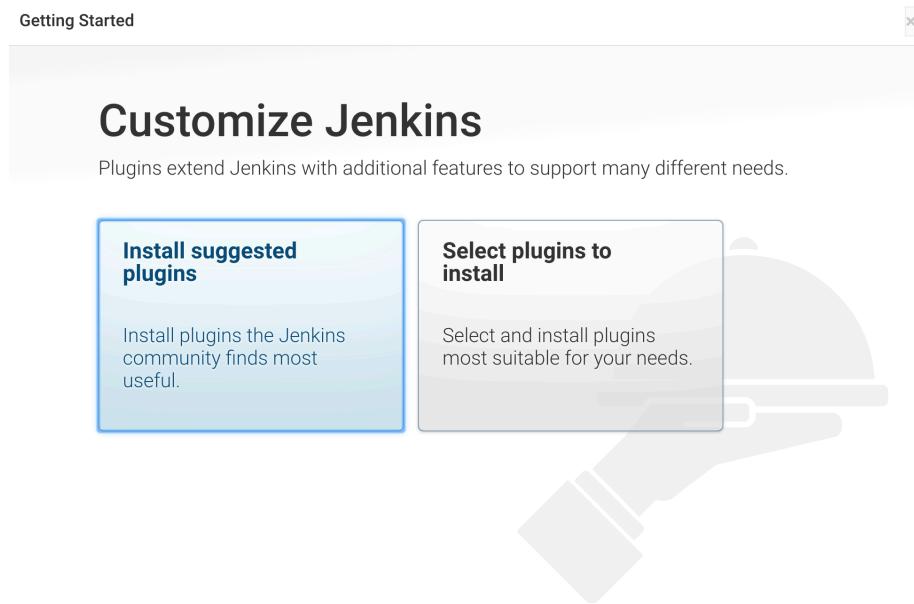


FIG. 3.6 : L'option d'installer des plugins suggérés.

Nous allons cliquer sur l'option Installer les plugins suggérés, qui lancera immédiatement le processus d'installation.

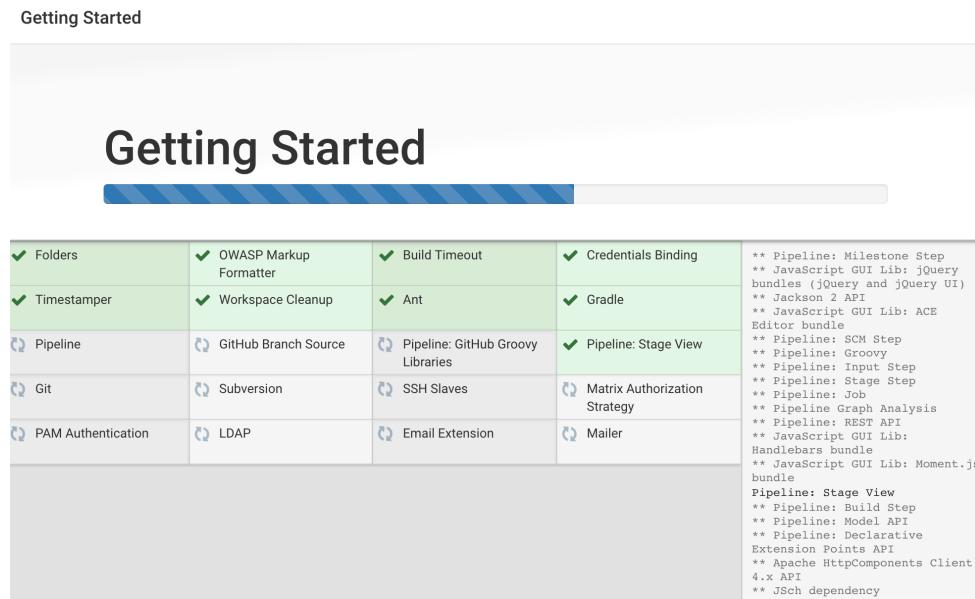


FIG. 3.7 : Jenkins Plugin.

Une fois l'installation terminée, vous serez invité à configurer le premier utilisateur administratif. Vous pouvez sauter cette étape et continuer comme admin en utilisant le mot de passe initial utilisé ci-dessus, mais nous allons prendre quelque minutes pour créer l'utilisateur.

The screenshot shows the 'Create First Admin User' form within a 'Getting Started' wizard. It contains five input fields: 'Username' (filled with a placeholder), 'Password' (empty), 'Confirm password' (empty), 'Full name' (empty), and 'E-mail address' (empty). At the bottom, it shows 'Jenkins 2.121.1' and two buttons: 'Continue as admin' (disabled) and 'Save and Continue'.

FIG. 3.8 : Crédit d'un jenkins user.

Saisissez le nom et le mot de passe de l'utilisateur. Vous verrez apparaître une page Instance configuration qui vous demandera de confirmer l'URL préférée de votre instance Jenkins. Vérifiez qu'il s'agit bien du nom de domaine ou de l'adresse IP de votre serveur.

The screenshot shows the 'Instance Configuration' form within a 'Getting Started' wizard. It has one main field, 'Jenkins URL' (empty). Below it is a detailed description of what the Jenkins URL is used for. At the bottom, it shows 'Jenkins 2.121.1' and two buttons: 'Not now' and 'Save and Finish'.

FIG. 3.9 : Instance de configuration.

Chapitre 3. Installation des outils DevOps

Après avoir confirmé l'exactitude des informations, cliquez sur Save and Finish. Vous verrez apparaître une page de confirmation confirmant que « Jenkins est prêt ! ». Cliquez sur Start using Jenkins pour consulter le tableau de bord principal de Jenkins :



FIG. 3.10 : Le tableau de bord principal.

À ce stade, vous avez terminé d'installer correctement Jenkins.

3.3.3 Pipline Jenkins

Le pipeline Jenkins est une autre partie importante de l'outil Jenkins. Elle affiche un aperçu des différents travaux en cours d'exécution sur les builds après les commits effectués par les développeurs. Il informe sur les tâches jenkins est actuellement l'exécution. Il existe 2 syntaxes de Pipelines :

- Declarative Pipeline : est une approche plus récente et plus structurée pour définir des pipelines dans Jenkins. Il utilise une syntaxe déclarative basée sur YAML pour décrire les étapes du pipeline, ce qui le rend plus facile à lire et à comprendre. Le Declarative Pipeline favorise également une approche plus orientée objet en divisant le pipeline en différentes sections telles que les paramètres, les environnements, les options, les étapes, etc. Il simplifie la syntaxe et fournit une abstraction plus élevée, permettant aux développeurs de se concentrer sur la logique métier plutôt que sur les détails de l'implémentation.
- Scripted Pipeline : est l'approche plus ancienne et plus flexible pour définir des pipelines dans Jenkins. Il utilise une syntaxe basée sur le langage de script Groovy, offrant ainsi un contrôle plus granulaire sur les étapes et les actions du pipeline. Le Scripted Pipeline permet d'utiliser des boucles, des conditions et d'autres structures de contrôle de Groovy pour une personnalisation plus avancée du flux de travail. Cependant, cela rend également le Scripted Pipeline plus complexe à écrire et à maintenir, car il nécessite une connaissance plus approfondie de Groovy et peut conduire à une duplication de code si les pipelines deviennent complexes.

Voici un exemple de pipeline : voir l'annexe.

3.4 Création de clé SSH :

SSH (Secure Shell) est un protocole de réseau sécurisé largement utilisé pour établir des connexions sécurisées et chiffrées entre des ordinateurs distants. Il est couramment utilisé pour l'administration à distance de systèmes informatiques et pour l'accès sécurisé à des serveurs.

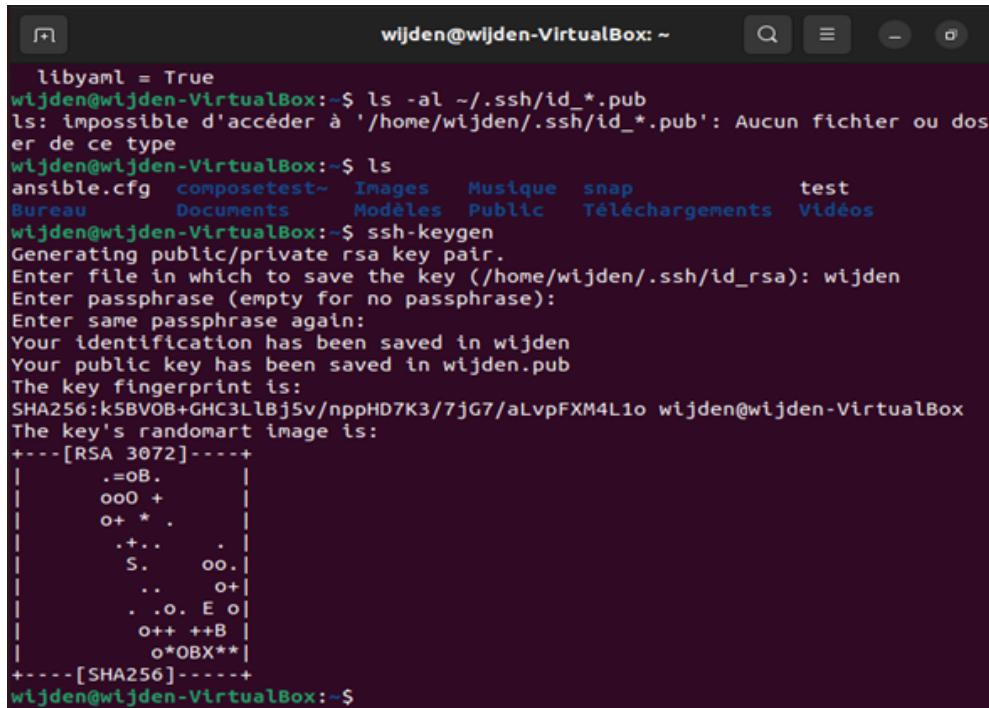
Voici comment fonctionne SSH :

- Établissement de la connexion : Lorsqu'un client SSH tente de se connecter à un serveur distant, une demande de connexion est envoyée. Le serveur distant répond en présentant son identité.
- Vérification de l'identité : Le client SSH vérifie l'identité du serveur distant en utilisant les clés publiques et privées. Si le serveur est reconnu et approuvé, la connexion se poursuit.
- Négociation de la session : Une fois l'identité du serveur vérifiée, le client et le serveur négocient les paramètres de la session SSH, tels que les algorithmes de chiffrement et les clés de chiffrement à utiliser.
- Authentification : Le client fournit ses informations d'identification (nom d'utilisateur et mot de passe ou clé privée) pour s'authentifier auprès du serveur distant. L'authentification par clé publique/privée est souvent préférée car elle est plus sécurisée et ne nécessite pas de mot de passe.
- Session établie : Une fois que l'authentification est réussie, une session sécurisée est établie entre le client et le serveur. Toutes les données échangées entre les deux parties sont chiffrées pour assurer la confidentialité.

SSH est utilisé pour diverses tâches, notamment :

- Connexion à distance : SSH permet aux administrateurs système de se connecter à distance à des serveurs et d'exécuter des commandes, configurer des services ou transférer des fichiers de manière sécurisée.
- Transfert de fichiers sécurisé (SFTP) : SSH prend également en charge SFTP, qui permet de transférer des fichiers de manière sécurisée entre des ordinateurs distants.
- tunneling (port forwarding) : SSH permet de créer des tunnels sécurisés entre des ordinateurs, permettant ainsi de rediriger le trafic réseau à travers des connexions chiffrées.
- Exécution de commandes distantes : SSH permet d'exécuter des commandes sur des ordinateurs distants de manière sécurisée, ce qui est utile pour l'automatisation des tâches ou la gestion de clusters.

SSH est une technologie essentielle pour sécuriser les communications et les accès à distance, et elle est largement utilisée dans le domaine de l'administration système et du développement logiciel.



```
libyaml = True
wijden@wijden-VirtualBox:~$ ls -al ~/.ssh/id_*.pub
ls: impossible d'accéder à '/home/wijden/.ssh/id_*.pub': Aucun fichier ou dossier de ce type
wijden@wijden-VirtualBox:~$ ls
ansible.cfg  componetest~  Images  Musique  snap          test
Bureau        Documents   Modèles  Public   Téléchargements  Vidéos
wijden@wijden-VirtualBox:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/wijden/.ssh/id_rsa): wijden
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in wijden
Your public key has been saved in wijden.pub
The key fingerprint is:
SHA256:kSBV0B+GHC3LlBj5v/nppHD7K3/7jG7/aLvpFXM4L1o wijden@wijden-VirtualBox
The key's randomart image is:
+--[RSA 3072]----+
|    .=oB.      |
|    ooO +     |
|    o+ * .    |
|    .+...     . |
|    S.    oo.|  |
|    ..    o+|  |
|    . .o. E o|  |
|    o++ ++B |  |
|    o*OBX**|  |
+---[SHA256]----+
wijden@wijden-VirtualBox:~$
```

FIG. 3.11 : Crédit de clé SSH.

SSH permet également la redirection de port (port forwarding) : le port SSH d'un client ou d'un serveur est utilisé par un autre abonné dans un réseau local pour établir une connexion sécurisée sur Internet.

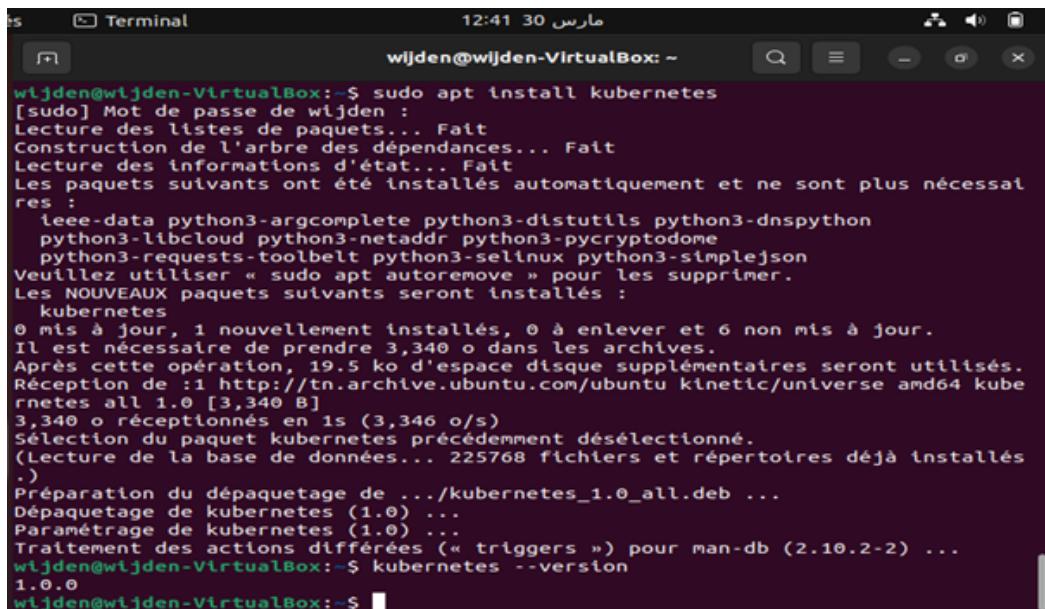
3.5 Kubernetes :

C'est un environnement open source permettant de déployer, faire évoluer et gérer des applications conteneurisées, sur n'importe quelle plate-forme.

3.5.1 Installation de Kubernetes :

Nous installerons Kubernetes avec la commande suivante :

- sudo apt install kubernetes



The screenshot shows a terminal window titled "Terminal" with the command "sudo apt install kubernetes" being run. The output indicates that the installation was successful, showing the download of packages from the Ubuntu archive, the preparation of the deb file, and the final command "kubernetes --version" which returns "1.0.0". The terminal interface includes a header with the date "مارس 30" and time "12:41".

```
wijden@wijden-VirtualBox:~$ sudo apt install kubernetes
[sudo] Mot de passe pour wijden :
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets suivants ont été installés automatiquement et ne sont plus nécessaires :
  libe-data python3-argcomplete python3-distutils python3-dnspython
  python3-libcloud python3-netaddr python3-pycryptodome
  python3-requests-toolbelt python3-selinux python3-simplejson
Veuillez utiliser « sudo apt autoremove » pour les supprimer.
Les NOUVEAUX paquets suivants seront installés :
  kubernetes
0 mis à jour, 1 nouvellement installés, 0 à enlever et 6 non mis à jour.
Il est nécessaire de prendre 3,340 o dans les archives.
Après cette opération, 19.5 ko d'espace disque supplémentaires seront utilisés.
Réception de :1 http://tn.archive.ubuntu.com/ubuntu kinetic/universe amd64 kube
rnetes all 1.0 [3,340 B]
3,340 o réceptionnés en 1s (3,346 o/s)
Sélection du paquet kubernetes précédemment désélectionné.
(Lecture de la base de données... 225768 fichiers et répertoires déjà installés
.)
Préparation du dépaquetage de .../kubernetes_1.0_all.deb ...
Dépaquetage de kubernetes (1.0) ...
Paramétrage de kubernetes (1.0) ...
Traitement des actions différées (« triggers ») pour man-db (2.10.2-2) ...
wijden@wijden-VirtualBox:~$ kubernetes --version
1.0.0
wijden@wijden-VirtualBox:~$
```

FIG. 3.12 : Installation de Kubernetes.

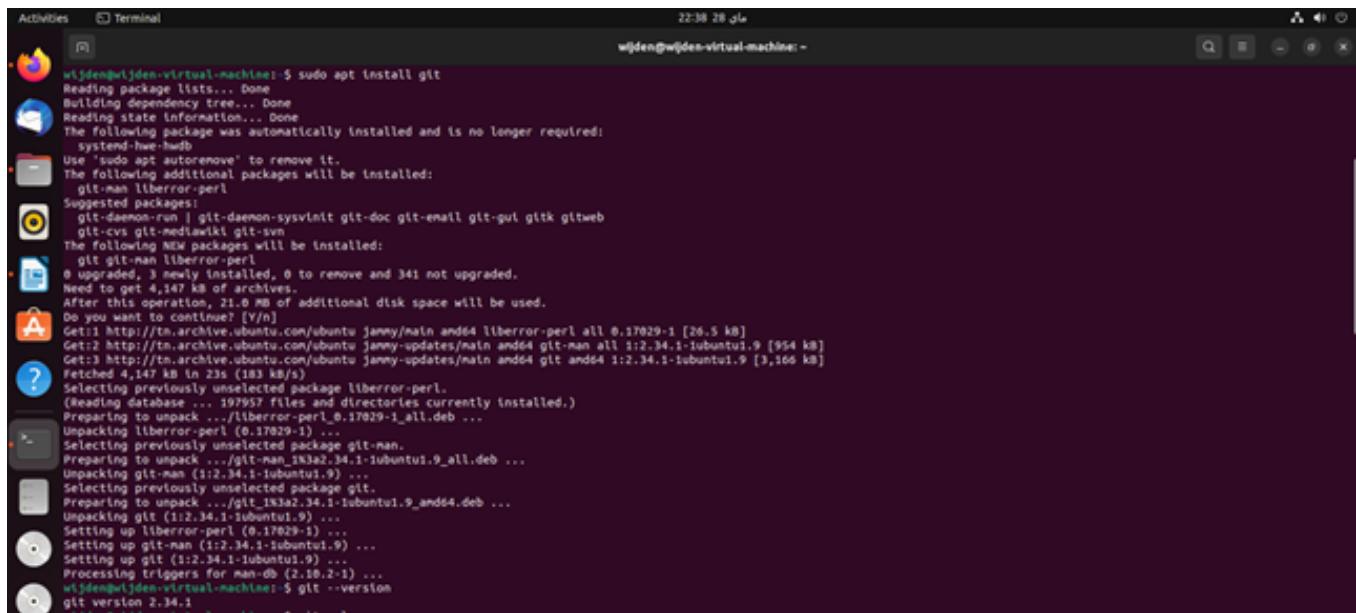
La conteneurisation permet aux développeurs de logiciels de créer et de déployer des applications de façon plus rapide et plus sécurisée. Kubernetes opérant au niveau des conteneurs plutôt qu'au niveau du matériel, il fournit une partie des fonctionnalités des offres PaaS, telles que le déploiement, la mise à l'échelle, l'équilibrage de charge (load balancing), la journalisation (logging) et la surveillance (monitoring).

3.6 Git :

Git est un puissant système de contrôle de version qui vous permet de suivre les modifications, de collaborer avec d'autres et de gérer efficacement votre code source. Il est couramment utilisé dans l'industrie du développement logiciel pour faciliter le travail d'équipe et assurer un suivi précis des modifications apportées aux projets. Voici quelques commandes Git courantes et leur utilisation :

- git init : Initialise un nouveau dépôt Git dans le répertoire courant.
- git clone <repository-url> : crée une copie locale d'un référentiel distant.
- git status : affiche l'état actuel du référentiel, y compris tous les fichiers modifiés, ajoutés ou supprimés.
- git add <file> : ajoute un ou plusieurs fichiers à la zone de préparation, les préparant à être validés.
- git commit -m "Commit message" : enregistre les modifications dans le référentiel, en créant un nouveau commit avec un message descriptif.
- git push : télécharge vos commits locaux vers un référentiel distant.
- git pull : récupère et intègre les modifications d'un référentiel distant dans votre branche actuelle.
- git branch : répertorie toutes les branches du référentiel.
- git checkout <branch> : Bascule vers une autre branche.
- git merge <branch> : fusionne les modifications d'une branche spécifiée dans la branche actuelle.
- git log : affiche un historique des commits, indiquant l'auteur, les dates et les messages de commit.
- git diff : affiche les différences entre les modifications actuelles et le dernier commit.
- git remote : répertorie les référentiels distants associés à votre référentiel local.
- git stash : enregistre temporairement les modifications qui ne sont pas prêtes à être validées, vous permettant de basculer vers une autre branche ou d'appliquer les modifications ultérieurement.
- git reset <file> : supprime un fichier de la zone de préparation, mais conserve les modifications dans le répertoire de travail.

3.6.1 Installation git :



```
wijden@wijden-virtual-machine:~$ sudo apt install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  systemd-hwdb
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb
  git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  git git-man liberror-perl
0 upgraded, 3 newly installed, 0 to remove and 341 not upgraded.
Need to get 4,147 kB of archives.
After this operation, 21.0 MB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://tm.archive.ubuntu.com/ubuntu jammy/main amd64 liberror-perl all 0.17029-1 [26.5 kB]
Get:2 http://tm.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git-man all 1:2.34.1-1ubuntu1.9 [954 kB]
Get:3 http://tm.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git amd64 1:2.34.1-1ubuntu1.9 [3,166 kB]
Fetched 4,147 kB in 23s (183 kB/s)
Selecting previously unselected package liberror-perl.
(Reading database ... 197957 files and directories currently installed.)
Preparing to unpack .../liberror-perl_0.17029-1_all.deb ...
Unpacking liberror-perl (0.17029-1) ...
Selecting previously unselected package git-man.
Preparing to unpack .../git-man_183a2.34.1-1ubuntu1.9_all.deb ...
Unpacking git-man (1:2.34.1-1ubuntu1.9) ...
Selecting previously unselected package git.
Preparing to unpack .../git_183a2.34.1-1ubuntu1.9_amd64.deb ...
Unpacking git (1:2.34.1-1ubuntu1.9) ...
Setting up liberror-perl (0.17029-1) ...
Setting up git-man (1:2.34.1-1ubuntu1.9) ...
Setting up git (1:2.34.1-1ubuntu1.9) ...
Processing triggers for man-db (2.10.2-1) ...
git version 2.34.1
```

FIG. 3.13 : Git

3.7 Installation XAMPP :

XAMPP est l'environnement de développement PHP le plus populaire. Apache est le serveur web. Son rôle est d'écouter les requêtes émises par le navigateur, de chercher la page demandée et de renvoyer.

XAMPP est une distribution Apache entièrement gratuite et facile à installer contenant MySQL, PHP et Perl.

On utilise XAMPP pour créer un serveur de test local et pour le but de fournir une distribution facile à installer.

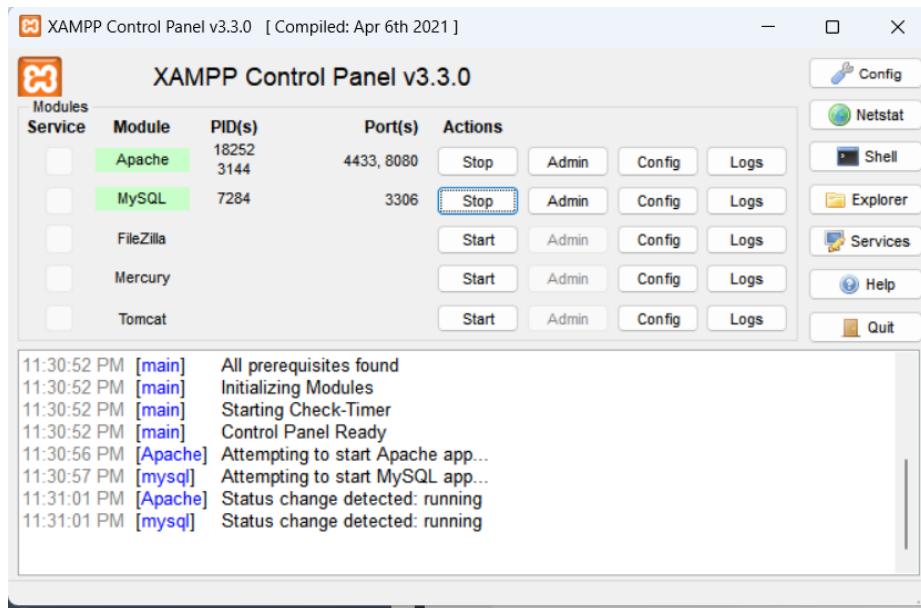


FIG. 3.14 : XAMPP.

Dans le panneau de configuration de XAMPP, on clique sur 'Start' ou 'Actions' pour le module Apache et MySQL. Cela indique à XAMPP de démarrer le serveur de votre choix. Les modules démarrés seront surligné en vert.

3.8 Visual Studio Code :

Est un éditeur de code extensible développé qui peut être utilisé avec une variété du langage de programmation.

Il fonctionne sous Windows, mac OS et Linux. Il fournit aux développeurs à la fois un environnement de développement intégré avec des outils permettant de faire avancer les projets techniques, de l'édition, à la construction, jusqu'au débogage. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré.

3.9 Machine Virtuel :

La machine virtuelle (VM) est un ordinateur virtuel créé par VirtualBox et utilise un système invité. Une machine virtuelle, ou « virtual machine », est « le client » créé dans un environnement informatique, « l'hôte ». Plusieurs machines virtuelles peuvent coexister sur un seul hôte. Dans ce projet , on utilise l'Ubuntu.

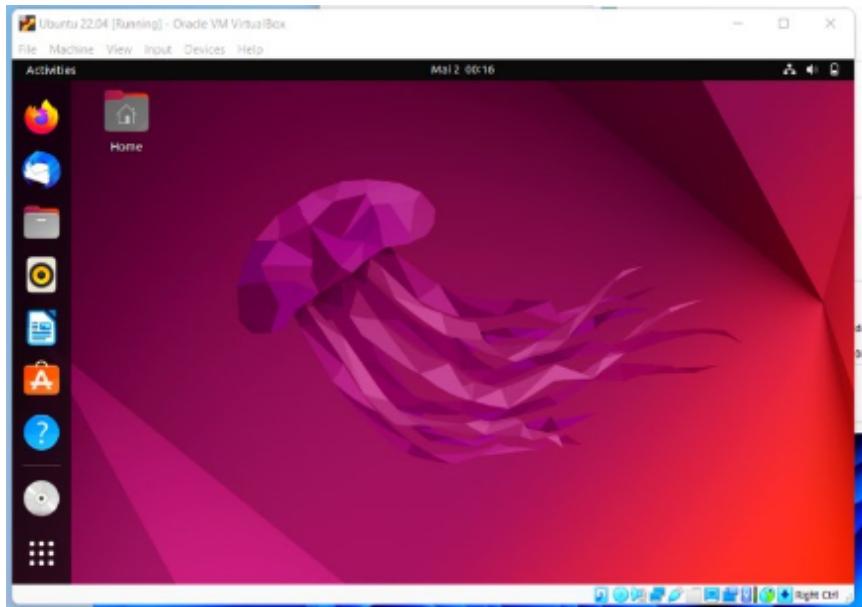


FIG. 3.15 : Machine virtuel Ubuntu

Dnas ce projet on install la machine virtuelle Ubuntu 22.04Lts.

C'est l'une des meilleures versions de Canonical à ce jour. Il est un système d'exploitation GNU/Linux basé sur la distribution Debian. Il est libre, gratuit, et simple d'utilisation.

3.10 Virtual Box :

VirtualBox vous permet d'exécuter un nombre illimité de machines virtuelles, avec pour seules limites l'espace disque et la mémoire de votre ordinateur. Utilisable sur tous supports, le logiciel s'adapte aussi bien aux systèmes embarqués qu'aux ordinateurs, aux datacenters ou encore aux environnements cloud.

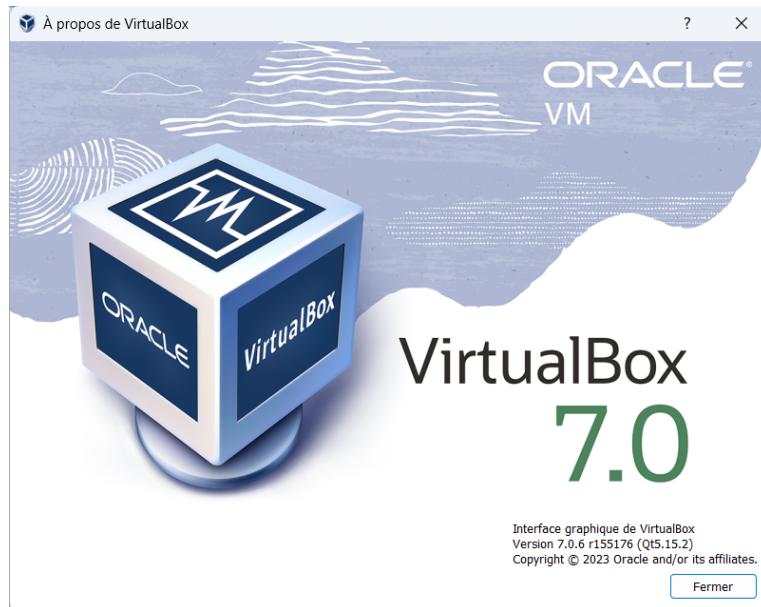


FIG. 3.16 : Virtuel Box.

VirtualBox permet de virtualiser un PC qui fonctionne exactement comme un PC physique. On y installe un OS que l'on peut ensuite démarrer. L'utilisateur peut alors piloter l'OS avec le clavier à la souris. En utilisant une combinaison de touches, il sort de la VM pour revenir sur l'OS hôte.

3.11 VMware :

VMware Workstation Pro est une référence pour la virtualisation sur des postes de travail sous Windows ou Linux et permet notamment de :

- Supporter les applications les plus gourmandes en ressources (graphiques).
- Créer et gérer vos réseaux virtuels VMware.

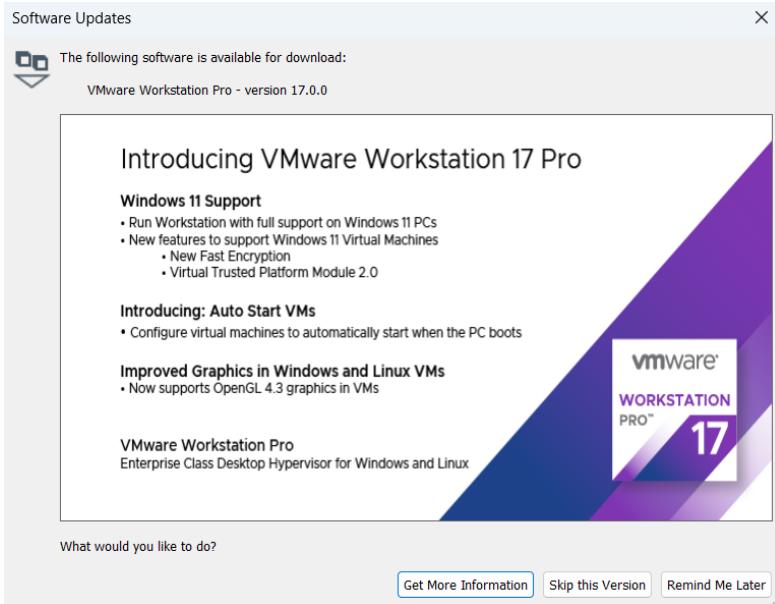


FIG. 3.17 : VMware Workstation.

On utilise VMware Workstation car il permet aux utilisateurs d'exécuter des machines virtuelles, des conteneurs et des clusters Kubernetes.

Chapitre 4

Test et Résultat

4.1 Introduction

Dans ce chapitre on utilise les outils DevOps sur l'application que l'on crée. Notre application est conçue pour offrir une interface utilisateur intuitive qui vous permettra de naviguer facilement dans notre collection et de trouver les informations que vous recherchez. Vous pourrez effectuer des recherches, consulter des documents numérisés, explorer des photographies et des enregistrements audio et vidéo, et bien plus encore.

4.2 Cadre du travail :

Dans cette section, nous présentons l'ensemble des équipements utilisés ainsi que les technologies adoptées pour mener à bien la réalisation de notre projet de fin d'études.

4.2.1 Environnement matériel :

Afin de bien mener le travail, j'ai utilisé un PC portable, dont les caractéristiques sont les suivants :

- Processeur Intel® Core™ i5.
- 24 Go de mémoire vive.
- Système d'exploitation Ubuntu 22.10 LTS.
- 512 Go SSD.

4.2.2 Environnement logiciel :

Les outils qui nous ont aidé à accomplir ce travail sont :

- Docker : c'est une plateforme permettant de lancer certaines applications dans des conteneurs logiciels lancé en 2013, créée par Solomon Hykes.
- Ansible : c'est une plateforme logicielle libre pour la configuration et la gestion des ordinateurs lancé en 2012, créée par Michael DeHaan.
- Jenkins : c'est un outil open source de serveur d'automatisation. Il aide à automatiser les parties du développement logiciel liées au build, aux tests et au déploiement, et facilite l'intégration continue et la livraison continue, lancé en 2011, développé par Kohsuke Kawaguchi.
- Kubernetes : est un système open source qui vise à fournir une plate-forme permettant d'automatiser le déploiement, la montée en charge et la mise en œuvre de conteneurs d'application sur des grappes de serveurs, lancée en 2014
- Star UML : C'est une plateforme en ligne gratuite permettant d'éditer du texte en LA TEX. Nous avons rédigé mon rapport avec Latex en utilisant Overleaf.

- Overleaf : C'est un logiciel de modélisation UML, qui a été « cédé comme open source » par son éditeur, sous une licence modifiée de GNU GPL. Nous avons utilisé StarUML pour construire les diagrammes présents dans le rapport.
- Git : c'est un logiciel de gestion de versions décentralisé, libre et gratuit, créé en 2005 par Linus Torvalds.

4.3 Interface de connexion :

4.3.1 Page de connexion :

Dans cette partie, nous allons créer un formulaire de connexion qui permet aux utilisateurs de se connecter en remplissant un formulaire Web.

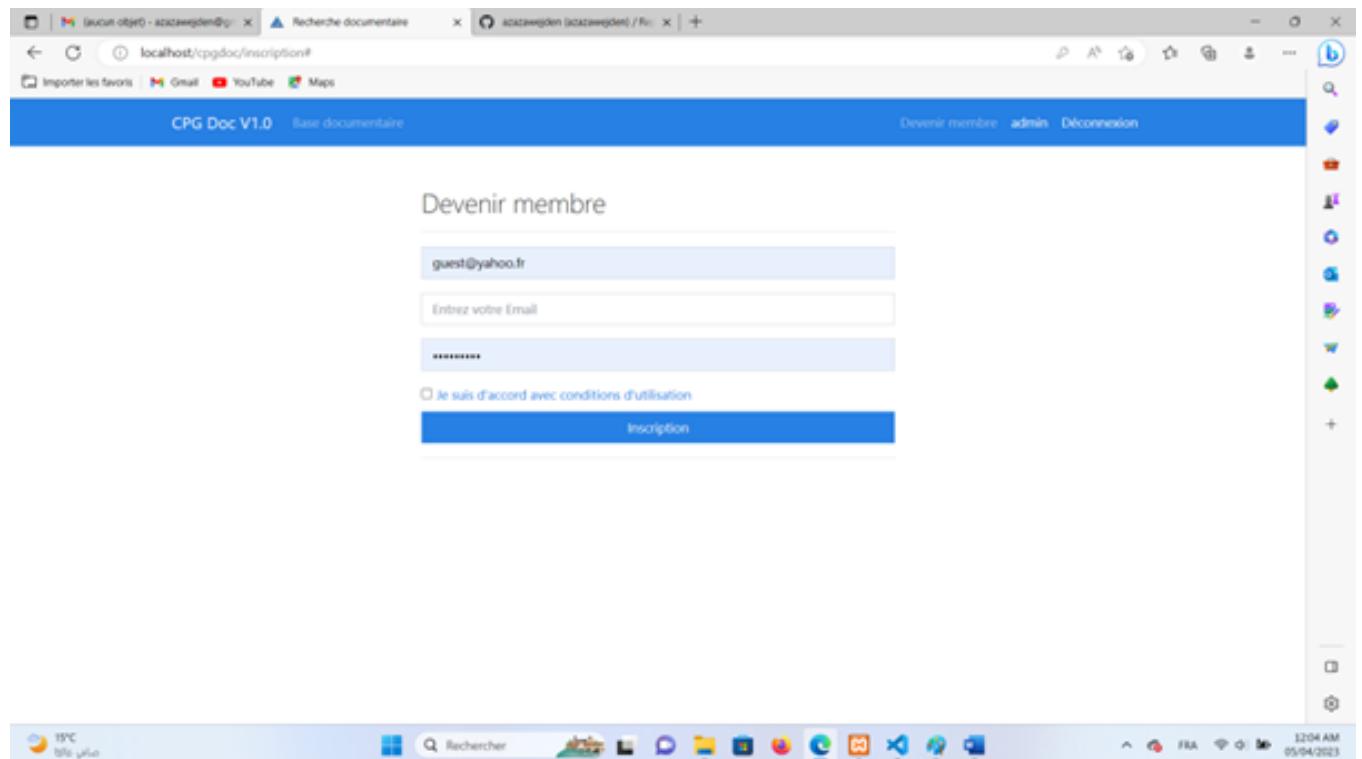


FIG. 4.1 : La page de connexion compilée.

4.3.2 Le tableau de bord :

Le Tableau de bord (CPD DOC) affiche tous les éléments qui permettent d'apprécier et de juger la situation dans laquelle se trouve l'entreprise. Il permet d'avoir une vue générale du système, car il résume toutes les informations en quelques indicateurs essentiels.

The screenshot shows a Microsoft Edge browser window with two tabs open: 'Recherche documentaire' and 'Recherche documentaire / fil'. The main content area is titled 'Recherche documentaire CPG' and contains a search bar with placeholder text 'Saisissez des termes à rechercher séparés par des espaces'. Below the search bar is a table listing documents. The columns are: N° boîte, Intitulé, Direction, Division, Service, Date début, Date fin, Écriture, and Remarque. The data in the table is as follows:

N° boîte	Intitulé	Direction	Division	Service	Date début	Date fin	Écriture	Remarque
9000	الجريدة الاجتماعية لسنة 2003	Direction Centrale des Ressources Humaines			2003	2003		
9000	[إسنادات 1990-2001] إدارة الموارد البشرية ومواعظ الملة	Direction Centrale des Ressources Humaines			1990	2001		
9000	مذكرة تأسيس حول تنشيط الشركة في موعد سبتمبر 2003	Direction Centrale des Ressources Humaines			sept-03	sept-03		
9000	التشهيد التقني لأكتوبر 2001 لإقليم المتوسط ثان الدور	Direction Centrale des Ressources Humaines			oct-01	oct-01		
9000	الجريدة الاجتماعية لسنة 2002	Direction Centrale des Ressources Humaines			2002	2002		
9000	مذكرة مراجعة الأداء وإستئصال واستئمان لسنة 2004	Direction Centrale des Ressources Humaines			2004	2004		
9000	لجنة الصياغات 2002-07-23 تزكيت الشروط	Direction Centrale des Ressources Humaines			2002	2002		
9000	مشروع حملة التحسيم سنة	Direction Centrale des Ressources Humaines			févr-02	févr-02		
9000	2001/2002 شفاعة إدارة المرافق	Direction Centrale des Ressources Humaines			2001	2002		
9000	بيان مواعظ الملة	Direction Centrale des Ressources Humaines			oct-01	oct-01		

FIG. 4.2 : La page du tableau de bord.

4.4 Code header PHP :

FIG. 4.3 : code header php part 1.

A screenshot of a code editor interface. The top navigation bar includes File, Edit, Selection, View, Go, Run, Terminal, Help, and a search bar. Below the navigation bar, there's a search panel with a 'SEARCH' button and a dropdown menu showing results from 50 files. The main workspace contains several tabs: 'header.php', 'bootstrap-theme.css', 'internet_cgfdocu', 'logins', 'cgfdocu.html', 'miscard', 'workspace.xml', 'index.php', and 'logos'. The 'header.php' tab is currently active, displaying PHP code for a navigation bar. The code includes logic for the current page, a collapsed navbar button, and a dropdown menu for 'internet_cgfdocu'. The 'PROBLEMS' and 'OUTPUT' tabs are visible at the bottom left, and the 'TERMINAL' tab is open at the bottom right, showing a command-line session for cloning a GitHub repository and switching branches.

FIG. 4.4 : code header php part 2.

```

<div class="nav navbar-nav navbar-right">
    <ul class="dropdown">
        <li><a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false">
            <span>Connexion</span>
        </li>
        <li class="dropdown-menu">
            <ul>
                <li><a href="login.php?logout">Déconnexion</a></li>
                <li><php if($SESSION['login']=='admin') {><li><php echo $page=='register' ? 'class="active"' : '' ;><a href="index.php?register">Créer un compte</a></li></ul>
            </li>
        </ul>
    </div>
</div>

```

FIG. 4.5 : code header php part 3.

4.5 Activation de Hyper-V :

Hyper-V est le produit de virtualisation de matériel de Microsoft. Il vous permet de créer et d'exécuter une version logicielle d'un ordinateur : une machine virtuelle. Chaque machine virtuelle agit comme un ordinateur complet, exécutant un système d'exploitation et des programmes. On fait la virtualisation par la commande suivante :

- > Enable-WindowsOptionalFeature -Online -FeatureName ("Microsoft-Hyper-V", "Containers") – All

```

Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Installez la dernière version de PowerShell pour nouvelles fonctionnalités et améliorations ! https://aka.ms/PSWindows

PS C:\Windows\system32> Enable-WindowsOptionalFeature -Online -FeatureName $($("Microsoft-Hyper-V", "Containers") -All

```

FIG. 4.6 : Hyper-V.

Chapitre 4. Test et Résultat

L'avantage de la virtualisation des serveurs réside dans les économies qu'elle permet de réaliser grâce à l'apport de services d'hébergement dans le Web et l'utilisation efficace des ressources existantes de l'infrastructure informatique.

Après l'activation de hyper-V, puisque la machine host est Windows on faire la switch de Docker avec la Windowes par l'exécution de commande suivante en Power Shell tant que administrateur :

– > DockerCli.exe –SwitchDaeamon

4.6 Docker compose yml :

Dans le fichier Docker Compose, vous pouvez définir les images Docker à utiliser, les ports à exposer, les volumes à monter, les variables d'environnement à définir et bien d'autres options. Vous pouvez également définir des dépendances entre les services, afin qu'ils démarrent dans le bon ordre. Docker Compose peut être intégré dans les pipelines CI/CD pour créer, tester et déployer des applications multi-conteneurs, rationalisant ainsi le processus de développement.

The screenshot shows a code editor interface with several files open. In the center, the `docker-compose.yml` file is displayed:

```
version: "3.9"
services:
  php:
    container_name: php
    image: php
    restart: always
    build:
      context: .
      dockerfile: Dockerfile
      target: base
    ports:
      - PORT=8080
```

Below it, the `Dockerfile` is shown:

```
FROM alpine:latest
COPY . /var/www/html
EXPOSE 8080
```

The left sidebar shows a project structure with various PHP files like `index.php`, `header.php`, and `login.php`. The bottom status bar indicates the path is `C:\xampp\htdocs\appdoc`.

FIG. 4.7 : Docker compose yml.

En utilisant un fichier yml, généralement nommé `docker-compose.yml`, Docker Compose vous permet de définir les services, les réseaux et les volumes de votre application. Ce fichier agit comme un plan détaillé pour l'ensemble de votre application, simplifiant la gestion, la reproduction et le partage de la configuration de l'application.

4.7 Docker file :

Un Docker-file est un document texte qui contient toutes les commandes qu'un utilisateur peut appeler sur la ligne de commande pour assembler une image.

FIG. 4.8 : Docker file.

On build Docker par ce commande :

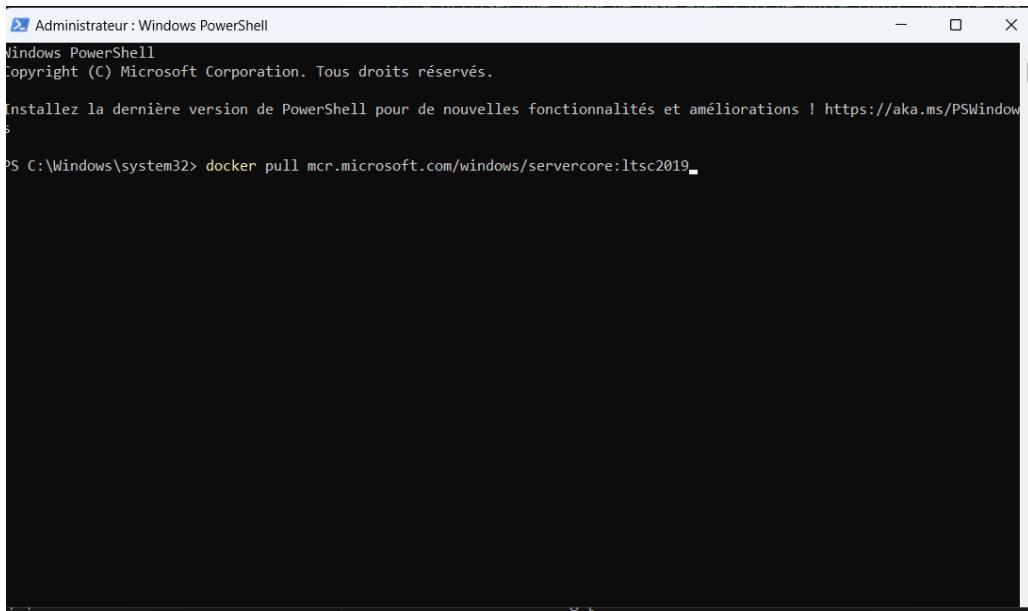
-> docker build -t cpgdoc .

FIG. 4.9 : How to build Docker.

On pull Docker par l'exécution de la commande suivante :

Chapitre 4. Test et Résultat

– > docker pull mcr.microsoft.com/windows/servercore :ltsc2019



The screenshot shows a Windows PowerShell window titled "Administrateur : Windows PowerShell". The window contains the following text:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations ! https://aka.ms/PSWindows
>
PS C:\Windows\system32> docker pull mcr.microsoft.com/windows/servercore:ltsc2019.
```

FIG. 4.10 : How to pull Docker.

Conclusion générale

Les DevOps ne cessent de croître et de développer leurs compétences logicielles dans l'industrie. De plus, les équipes de développement et d'exploitation déploient leurs ailes dans un monde de livraison plus développé, plus fiable et plus performant, ce qui va faire adopter la culture DevOps par de nombreuses entreprises qui souhaitent bénéficier de leurs compétences et connaissances technologiques pour les améliorer. Développer leur compétence logicielle. Ainsi, dans un avenir proche, nous assisterons à la pratique de la culture DevOps dans presque toutes les entreprises technologiques, les établissements institutionnels et même dans le secteur de la santé pour améliorer l'efficacité de la prestation de services.

L'utilisation intensive de l'infrastructure en tant que code assurera une plus grande flexibilité lors de la détermination de l'emplacement du serveur en modifiant la description du code.

Celles-ci peuvent être effectuées dès que l'application est installée dans le système d'organisation pour s'assurer qu'il y a efficience et efficacité dans le fonctionnement. Cependant, les gens sont censés écrire des codes afin qu'ils puissent tous interroger les grands corps de rapports de journal pour leur permettre de prédire ce qui est nécessaire ensuite. Ce faisant, ils peuvent créer l'efficacité du système où les changements futurs peuvent être démontrés suffisamment tôt pour éviter tout défi qui pourrait empêcher l'entreprise de bien fonctionner. De plus, le code produit à l'aide de services automatisés, bien implantés par l'équipe DevOps, assure la flexibilité du système. On ne préférerait pas avoir un système vulnérable aux changements et incapable de s'adapter aux nouvelles demandes. Pour cette raison, les développeurs prospéreront en produisant le meilleur des codes d'organisation qui résisteront à toutes les situations. Le code stable sera facilement avancé par les experts chaque fois que le besoin s'en fera sentir.

Les développeurs de logiciels seront obligés d'adopter la philosophie DevOps dans leurs procédures de travail. Cela est dû au nombre croissant d'applications et d'algorithme basés sur l'intelligence artificielle qui sont devenus monnaie courante dans le monde de la haute technologie. Pour contrer la concurrence entrante, ils doivent suivre le rythme et évoluer avec les changements qui peuvent survenir dans le monde de la technologie. De plus, les stratégies et méthodologies DevOps devraient être leur option universelle pour gérer les pipelines automatisés, maintenir et tester de nombreux modèles intégrés dans la chaîne de production de logiciels. Cela doit être intégré dans le système par l'équipe de développement et d'exploitation DevOps.

Conclusion générale

pour s'assurer que toutes les autres entrées et sorties correspondent aux buts et objectifs de l'organisation. À son tour, cela augmentera la production de produits logiciels, augmentant ainsi leur niveau de ventes. Ce faisant, la marge de vente augmentera, ce qui fera prospérer l'organisation et générera une marge bénéficiaire élevée.

Annexes

Annexe A

Code

A.1 Index

A.1.1 Index-base.php

A.1.2

A.1.3

A.1.4

A.1.5 db-connect.php

A.1.6 header.php

A.1.7 Docker file

A.1.8 Docker-compose-yml