

**MINISTERE DE L'ENSEIGNEMENT SUPERIEUR DE LA RECHERCHE  
SCIENTIFIQUE ET DE L'INNOVATION**

**INSTITUT SUPERIEUR DE TECHNOLOGIES (IST)**



**DOMAINE : Science et Technologies**

**MENTION :**

**SPECIALITE : Réseaux et Systèmes Informatiques**

**MEMOIRE DE FIN DE CYCLE**

**THEME : AUTOMATISATION DU DEPLOIEMENT DES SOLUTIONS SUR LE  
CLOUD : LA SOLUTION MEDIAWIKI.**

**Période de stage : de Novembre 2021 à Février 2022**

**Réalisé par : OUARO Hamidou**

**Pour l'obtention du diplôme de Licence en : Réseaux et Systèmes Informatique**

Maitre de stage

TASSEMBEDO A RAZACK

Enseignant à I.S.T

Directeur de rapport

TOPAN B ZAKARIA

Ingénieur de conception

DG de LOGO-SERVICES

**ANNEE ACADEMIQUE 2020-2021**



**MINISTERE DE L'ENSEIGNEMENT SUPERIEUR DE LA RECHERCHE  
SCIENTIFIQUE ET DE L'INNOVATION**

**INSTITUT SUPERIEUR DE TECHNOLOGIES (IST)**



**DOMAINE : Science et Technologies**

**MENTION :**

**SPECIALITE : Réseaux et Systèmes Informatiques**

**MEMOIRE DE FIN DE CYCLE**

**THEME : AUTOMATISATION DU DEPLOIEMENT DES SOLUTIONS SUR LE  
CLOUD : LA SOLUTION MEDIAWIKI.**

**Période de stage : de Novembre 2021 à Février 2022**

**Réalisé par : OUARO Hamidou**

**Pour l'obtention du diplôme de Licence en : Réseaux et Systèmes Informatique**

Maitre de stage

TASSEMBEDO A RAZACK

Enseignant à I.S.T

Directeur de rapport

TOPAN B ZAKARIA

Ingénieur de conception

DG de LOGO-SERVICES

**ANNEE ACADEMIQUE 2020-2021**

**Période de production du rapport Décembre et Février 2021**



## **SOMMAIRE**

## **AVANTPROPOS**

## **DEDICACES**

## **REMERCIEMENTS**

## **ABREVIATIONS**

## **LISTES DES FIGURES**

## **LISTES DES TABLEAUX**

## **RESUME**

## **ABSTRACT**

## **INTRODUCTION GENERALE**

### **Chapitre 1 : Présentation du contexte de stage**

#### Introduction

#### 1.1 Présentation de l’Institut Supérieur de Technologie (IST)

##### 1.1.1 Historique et missions

##### 1.1.2 Formations

#### 1.2 Présentation de la structure d’accueil (Logo Services)

##### 1.2.1 Objectif

##### 1.2.2 Organisation générale de Logo Services

#### 1.3 Présentation du projet

#### 1.4 Problématique

#### 1.5 Les objectifs

#### 1.6 Résultats attendu

#### 1.7 Organisation du projet

#### Conclusion

### **Chapitre 2 : Etude de l’infrastructure existant et procédure de résolution**

#### Introduction

#### 2.1 La présentation de l’infrastructure existant

#### 2.2 L’infrastructure ou l’outil utilisé pour le déploiement

## 2.3 Evaluation du système informatique

### 2.3.1 Vue positive

### 2.3.2 Vue négative

## 2.4 Quelques suggestions

### 2.4.1 Démarche de résolution du problème

#### 2.4.1 Formule utilisée

#### 2.4.2 Ressources logistiques utilisées ou mise à disposition

### Conclusion

## **Chapitre 3 : Généralité sur l'utilisation de l'outil d'automatisation ansible pour le déploiement de MediaWiki sur le cloud.**

### Introduction

## 3.1 Généralités sur le DevOps

### 3.1.1 Définition des concepts

### 3.1.2 Les concepts du DevOps

### 3.1.3 Outils du DevOps

## 3.2 Présentation et description de l'automatisation de déploiement

### 3.2.1 Présentation de la solution

### 3.2.2 Description de la solution

### 3.2.3 Utilisation de la solution

### 3.2.4 Les étapes d'automatisation du processus de déploiement des logiciels

## 3.3 Généralités sur quelques plateformes d'automatisation

### 3.3.1 Le choix des critères

### 3.3.2 Etude comparative et synthèse

### 3.3.3 La synthèse et le choix

## 3.4 Procédure d'amélioration de l'automatisation des déploiements

### 3.4.1 Avantage de l'automatisation de déploiement

### 3.4.2 Limite de l'automatisation du déploiement

## 3.5 Aperçu sur Media Wiki et sur le cloud

## 4.6 Estimation du Coûts

## 4.7 Architecture l'automatisation du déploiement de la solution

### Conclusion

## **Chapitre 4 : Présentation du bilan de la solution choisie**

### Introduction

#### 4.1 Définition des concepts

#### 4.2 Histoire de ansible

#### 4.3 Les composants de base de Ansible

#### 4.4 Présentation de ansible et de ses objectifs

##### 4.4.1 Présentation de Ansible

##### 4.4.2 Ansible Tower

##### 4.4 .3 Les objectifs de conception de ansible

#### 4.5 Listes des différentes versions de Ansible

#### 4.6 Avantages et désavantages de l'outil d'automatisation ansible

#### 4.7 L'apport du stage

#### 4.8 L'architecture de Ansible

### Conclusion

## Chapitre 5 : L'implémentation de l'automatisation du déploiement de la media wiki sur le cloud.

### Introduction

#### 5.1 Mise en place et configuration de ansible sur le node manager et dans l'environnement virtuel

##### 5.2 Installation des serveurs sur les nodes

##### 5.2.1 Installation du serveur web sur le http1

##### 5.2.2 Installation de la base de données sur le bdd1

##### 5.3 Le déploiement de Media Wiki sur le cloud à l'aide de ansible

### Conclusion

## **CONCLUSION GENERALE**

## **BIBLIOGRAPHIE**

## **WEBOGRAPHIE**

## **ANNEXE**

## **AVANT-PROPOS**

Pour se conformer aux exigences du système de formation de l'Institut Supérieur de Technologie, il est recommandé à tout étudiant en année de licence de passer un stage de formation en entreprise dans le but de mettre en pratique les connaissances théoriques acquises durant sa formation universitaire, et également de rédiger un rapport de stage qui fera l'objet d'une soutenance publique devant un jury. Ce rapport de stage, réalisé au sein de l'entreprise LOGO SERVICES à Ouagadougou est le fruit d'un stage de recherche au vu de valider ma Licence en Réseau et Système Informatique.

Les relations, les interventions et l'expérience de mon maître de stage et de mon professeur de suivi ont été précieux pour recentrer un sujet dont le champ est très vaste. Nous étions conviés à mener des recherches sur un thème afin d'apporter une contribution. Le choix du sujet vise à traiter les questions liées à l'actualité récente avec l'avènement du Dev Ops et surtout de l'orientation des entreprises vers l'automatisation dans le but de faciliter leurs activités quotidiennes. Cette étude porte sur un sujet d'actualité et encore peu exploré nous a amené à nous penser sur le thème : Utilisation d'un outil d'automatisation du déploiement de la solution Media Wiki sur le cloud. Evoluant dans plusieurs branches de l'informatique, l'entreprise nous a permis de nous pencher sur la question de l'automatisation et du déploiement des applications.

Je remercie toutes ces personnes pour leur modeste accompagnement pour la réalisation de ce document en leur exprimant ma gratitude. L'objectif de ce rapport de stage est de mettre en exergue les potentiels et les avantages de l'outil ansible dans l'automatisation et du déploiement des applications d'une structure informatique.



## *Dédicace*

C'est avec beaucoup d'enthousiasme que je dédie ce travail avec beaucoup de plaisir à toutes ces personnes qui me sont chères afin de leur prouver mon immense fierté.

À mes très chers parents,

Pour tout tous ceux que vous avez fait pour moi pendant tant d'années et surtout pour toute l'éducation que vous m'avez octroyée. Je me battrais pour maintenir ce principe de fierté tout en gardant espoir de ne jamais vous offensez. Que ce modeste travail soit le fruit de vos vœux tant formulées et de vos prières quotidiennes en mon égard.

À mes sœurs et frères,

Je suis fier d'être des vôtres grâce à votre grand sens de fraternité. C'est un grand honneur pour moi de vous exprimer ma gratitude, que Dieu vous le rend au centuple.

À son Excellence Mohamed S TOPAN et à toute sa famille,

Pour le goût à l'effort que vous avez suscité en moi, de par votre rigueur et qui m'avez toujours soutenu et encouragé durant ces années d'études

À mes très chers amis,

Vous qui m'avez prodigué les précieux conseils et en souvenir des bons moments. Au regard de toutes ces épreuves que nous avons franchi ensemble en traçant les signons d'un avenir radieux. J'ai bon espoir que notre amitié demeurera éternellement.

À ma grand-mère,

Feue Binso Lemba, ceci est ma profonde gratitude pour ton éternel amour, que ce rapport soit le meilleur cadeau que je puisse t'offrir.

A Allah le tout puissant et l'omniscient, à qui nous rendons grâce particulièrement pour la santé et l'engouement au travail qu'il a daigné nous vouer tout au long de ce projet.

## **Remerciements**

Au terme de ce projet de fin d'étude, mes vifs remerciements sont dédiés à tous ceux qui ont contribué, directement ou indirectement à l'élaboration de ce projet.

Nos sincères remerciements vont à l'endroit de Monsieur Zakaria B TOPAN, mon encadrant coté entreprise, qui a toujours trouvé le temps de faire la surveillance de mon document et également d'être à l'écoute lorsque nous rencontrions des problèmes techniques dans le projet. Nos remerciements vont également à Monsieur Laurent NIKIEMA sa disponibilité, ses directives et son sens du détail nous ont permis de soigner et d'améliorer constamment la qualité de ce travail.

Nous saisissons cette opportunité pour adresser nos profonds remerciements tout particulièrement à M. Abdoul Razak TASSEMBEDO pour ses conseils, ses enseignements, sa transparence lors de la rédaction, de m'avoir éclairé et conseillé au quotidien. Également pour les valeurs qu'il nous a inculqué car ses conseils ont guidé nos choix et nos démarches et m'ont alors permis de tirer le meilleur profit de cette expérience professionnelle.

Nous exprimons notre gratitude à tous les responsables au sein de LOGO SERVICE qui m'ont donné l'opportunité de réaliser ce stage et les encouragements qu'ils nous ont prodigués.

C'est avec une immense fierté que j'adresse mes remerciements les plus distingués à toute l'équipe pédagogique de l'Institut Supérieur de Technologie (I.ST) qui nous ont transmis leur savoir et nous ont assuré la meilleure des formations pour un avenir radieux. Nous remercions également tous les camarades de l'Institut Supérieur de Technologie (I.ST) pour leur soutien et leur contribution.

Nous souhaitons par ce dernier message adresser un remerciement à l'endroit de tous les membres du jury pour l'honneur et pour leur disponibilité qu'ils nous font de participer à l'examen de ce travail.

## Liste des sigles et abréviations

**SSH** : Secure Shell

**DNS** : Domain Name System

**IP** : Internet Protocole

**PING** : Packet Internet ou Inter-Network Groper

**ECDSA** : Algorithme de signature numérique à courbe elliptique

**IBM** : International Business Machines Corporation

**Pip** : python installer packages

**Dev Ops** : Développement des Opérations

**GPL** : General Public License

**GUI** : Graphical User Interface

**YAML** : Yet Another Markup Language

**PHP** : Hypertext Preprocessor

**ERB** : Embedded Ruby

**DSL** : Langue Spécifique au Domaine

**OVH** : Oles Van Herman (On Vous Héberge)

**CLI** : Interface de Ligne de Commande

**AWS** : Cloud Computing Services

**GCP** : Google Cloud Platform

## **LISTES DES FIGURES**

Figure. 1

Figure. 2

Figure. 3

Figure. 4

Figure. 5

Figure. 6

Figure. 7

Figure. 8

Figure. 9

Figure. 10

Figure. 11

Figure. 12

Figure. 13

Figure. 14

Figure. 15

Figure. 16

Figure. 17

Figure. 18

Figure. 19

Figure. 20

Figure. 21

Figure. 22

Figure. 23

Figure. 24

Figure. 252

Figure. 26

## **LISTES DES TABLEAUX**

Tableau. 1

Tableau. 2

Tableau. 3

## RESUME

Avec l'émergence de l'avènement du cloud et du Dev Ops, une nécessité s'impose pour une gestion des configurations de manière automatique. Certaines entreprises restent incertaines quant à l'automatisation du déploiement des applications par rapport aux outils d'automatisation du système informatique. L'automatisation a pris une place figurante (autre tournure) dans l'avancée de la technologie. Et grâce au serveur nous pouvons contenir et résoudre nos différentes erreurs par le biais de plusieurs outils.

Nous disposons de multiples outils, dont d'autres exercent, paraît-il les mêmes fonctions. Notre document aborde ansible qui cadre avec le contexte de notre sujet. Ansible est cet outil dédié à l'automatisation du déploiement des applications et des sites web sur le cloud. Il détaille aussi les valeurs de notre outil ainsi que le domaine principal qu'il couvre sans oublier les avantages qu'il nous propose au sein du DevOps intégré.

## ABSTRACT

## INTRODUCTION GENERALE

L'avènement de l'automatisation engendre la révision profonde des méthodes de déploiement des solutions sur le cloud. De nos jours, il n'est plus réalisable d'accuser du temps pour le déploiement des applications et des sites web de nos clients. A l'égard de ces exigences de temps au marché, de l'entretien, de la sécurité et la croissance des objectifs (projets) dans les domaines de l'informatique comme dans les configurations logicielles et d'infrastructures de plus en plus difficile, engendrant des risques d'opérationnalisation, les besoins de traitement des tests et de fluidifier les déploiements en production se sont graduellement exprimés. Dans notre contexte, nous constatons que le DevOps répond exactement à cet immense défi du digital. L'entreprise, Logo Service a pris acte de cette évolution numérique, et a pris l'option de se conformer et de se mesurer à cette nouvelle exigence.

Le DevOps offre de nombreux avantages parmi ses pratiques les plus propagées dans le domaine de l'automatisation. Nous rencontrons le déploiement continue et la gestion des configurations œuvrent dans la résolution de nos préoccupations. Le déploiement continue est cette procédure logicielle qui octroie aux différentes équipes d'offrir de nouvelles fonctionnalités aux utilisateurs efficacement et vite. L'avantage de ce déploiement continue est de nous permettre d'automatiser les actions de déploiements qui, auparavant étaient réalisées manuellement. Le but de ce déploiement continue est d'autoriser le flux constant de changements en direction de la production à travers une ligne de production automatisée de logiciels grâce au pipeline.

C'est dans cette optique qu'adhère notre projet, qui aspire à développer l'automatisation de déploiement tout en utilisant Ansible, un outil d'automatisation et de gestion de configuration. L'entreprise Logo Service, au sein de laquelle ce projet a été développé, ambitionne de résoudre les multiples problématiques.

Notre rapport de stage synthétise les travaux que nous avons essentiellement menés, il est organisé en (05) chapitres. Le premier chapitre aborde la présentation du contexte de stage qui nous présente l'Institut Supérieur de Technologie et notre structure d'accueil Logo Service. Dans le deuxième chapitre nous parlons de l'étude de l'infrastructure existant et la procédure de résolution. Le troisième chapitre traite les généralités sur les outils d'automatisation pour le déploiement sur le cloud. L'avant dernier (quatrième) chapitre présente les détails de la solution choisie. Le cinquième chapitre évoque l'implémentation de la solution ansible pour le déploiement de media wiki.



## **Chapitre 1 : Présentation des structures d'accueil**

### **Introduction**

Dans ce chapitre nous donnons d'abord une brève présentation de notre Institut IST et de la structure d'accueil LOGO-SERVICES qui nous a guidée dans le choix de ce projet, ensuite nous aborderons la problématique du projet en définissant ainsi ses objectifs. Enfin nous montrerons les résultats en sans oublier l'organisation du projet.

### **1.1 Présentation de l'Institut Supérieur de Technologie (IST)**

#### **1.1.1 Historique et missions**

L'Institut Supérieur de Technologies (IST) a été créé sous la forme d'une SARL de droit privé burkinabè au capital de 10 000 000 FCFA. Il a obtenu son autorisation sous le numéro N°204/2000/MESSRS/DGSRS/SP du 14 mars 2001. Sa direction générale est située à Tampouy. Elle dispense des formations dans le domaine de la technologie et de la gestion.

L'Institut Supérieur de Technologie (IST) a pour mission de

- Assurer la formation des cadres compétents, dans les domaines technique et professionnels, capable de répondre aux exigences du marché de l'emploi ;
- Assurer la formation continue des travailleurs des entreprises afin d'élever le niveau de compétitivité de leurs entreprises ;
- Faciliter l'insertion des étudiants en fin de cycle en entreprise.

#### **1.1.2 Formations**

Les filières de l'IST relèvent du domaine technique et professionnel. L'IST compte plusieurs niveaux de formations en BAC+2, BAC+3, BAC+5 regroupées en deux départements : le département sciences et technologies et le département science de gestion.

##### **➤ Le département sciences et technologies**

Ce département prépare aux diplômes de BTS, DTS, d'Ingénieur, de Licence et de master dans les domaines suivants :

- Génie Minier
- Génie Biomédical
- Agroalimentaire
- Réseaux Informatiques et Télécoms
- Génie Electrique

### ➤ **Le département Science de Gestion**

Ce département prépare aux diplômes de DTS, BTS, de licence, et de Master dans les domaines de science de gestion. Ces domaines sont entre autres :

- Gestion des Ressources Humaines Transport Logistique
- Communication d'Entreprise
- Banque et Micro finance
- Marketing et Gestion Commerciale
- Secrétariat de Direction Bureauistique
- Gestion des Ressources Humaine
- Etc.

## 1.2 **Présentation de la structure d'accueil (Logo Services)**

LOGO SERVICES SARL est une société de services informatiques Burkinabè. Créée en Mars 2019, c'est une Société à Responsabilité Limitée (SARL) avec comme identifiant Registre du commerce (RCCM) N°BFOUA 2019B1837 et fiscale (N° IFU) : 00117182 B.

LOGO Services est reconnue intégrateur national de solutions informatiques à travers son agrément technique D5 : Intégrateur de solutions informatique.

Elle est joignable aux adresses suivantes 01 BP 5988 Ouagadougou 01, Secteur 19 ex 25.

Téléphones : Nous avons le fixe : +226 25 39 29 20 et les portables tels que le 76 40 32 14 et le 71 57 57 16.

Logo Service est également joignable sur cette adresse email : [infos@logo-services.com](mailto:infos@logo-services.com).

### 1.2.1 **Les prestations de Logo Services**

L'entreprise offre de nombreuses prestations dans de multiples secteurs d'activités comme :

La conception et la mise en place de solutions logicielles,

La conception et la mise en place de solutions SIG et d'analyse de données spatiales ;

La conception et la mise en place de solution d'aide à la décision (analyse et de reporting des données)

L'ingénierie réseau et télécom,

La Sécurité informatique,

L'organisation, le traitement et la numérisation des documents et Archives,

Le renforcement des capacités en informatique.

Elle dispose de compétences dans ses domaines d'actions et d'un réseau de partenaires qui lui ont permis de mener à bien des projets informatiques au profit de l'administration Burkinabè et des entreprises publiques et privées confère quelques références.

### 1.2.2 **Organisation générale de Logo Services**

Logo Services est une entreprise qui fonctionne grâce à la détermination de son Président Directeur Général, ses partenaires et ses différents consultants évoluant dans toutes les domaines d'activité. Leur efficacité et leur professionnalisme octroient une satisfaction aux clients.

## 1.3 **Présentation du projet**

### Problématique

Logo Services est une entreprise qui prône une nouvelle vision dans le domaine de l'informatique pour la conception des nouvelles solutions. L'entreprise a une nécessité d'héberger ses sites web et ses applications après la conception. L'entreprise fait face aux problèmes de déploiement manuels des applications et des mises à jour. Quelle solution faut-il apporter pour une résolution efficace de ce problème ? Comment mettre en place cette solution ? Comment faciliter le déploiement de ses applications et des sites web sur le cloud ? Quel outil ou quelle méthode faut-il se doter pour assurer ses déploiements sur le cloud ?

## 1.4 **Les objectifs**

### Objectifs généraux

Dans la lancée d'apporter une réponse, notre solution devra permettre de :

Faciliter l'hébergement et le déploiement des applications et des sites web de nos clients

Limiter les interventions humaines sur le cloud

Garantir la sécurité et la maintenance des applications et des sites web de nos clients.

Effectuer les mises à jour

Dans le but d'accroître l'efficace en limitant les interactions humaines ou en les remplaçant par les systèmes informatiques utilisant des outils pour exécuter les tâches. Cette manière de faire nous ramène dans l'automatisation qui nous assurera de réduire nos différentes interventions sur le serveur. Comprendre les besoins des clients et apporter une solution plus efficace.

Spécifiquement nous avons d'autres objectifs tels que :

- Comprendre l'intérêt du mouvement DevOps
- La maîtrise de plusieurs environnements du DevOps
- Faire une étude sur les différents outils d'automatisation
- La réduction de certaines erreurs suite au déploiement manuel.

## **1.5 Résultats attendus**

A l'issue de cette étude nous devons apporter une solution au problème en œuvrant à l'intégration continue et la capacité d'automatiser le déploiement de la solution sur le cloud. Nous allons apporter une satisfaction totale à la préoccupation des clients dans le but d'une bonne coopération.

## **1.6 Organisation du projet**

- Les acteurs du projet

Les acteurs du projet sont ces personnes qui se sont distinguées par leur contribution dans l'élaboration du projet. Dans cette répartition on peut retenir entre autres le groupe de pilotage et le groupe de projet.

- Le groupe de pilotage

Sous l'organisation technique et contrôle et la pertinence des solutions proposées, ce groupe est conduit par : M. Abdoul Razak TASSEMBEDO M. Laurent NIKIEMA et M. Bia Zakaria TOPAN

- Le groupe de projet

Pour une superbe réalisation de ce projet, la constitution de ce groupe est assurée par l'acteur principal du projet, la personne chargée à la réalisation dudit projet. Il est recommandé de mener d'abord à bien les travaux dans les délais impartis, ensuite d'assurer une excellente cohérence et une faisabilité des solutions retenues, et enfin soumettre au comité de pilotage en présentant les rapports d'évolution. Il est constitué en la personne de M. HAMIDOU OUARO, étudiant en 3<sup>ème</sup> année de Réseaux et Système d'Information à l'IST.

- Planning du travail

## Conclusion

Dans ce chapitre il a été essentiellement question de la présentation de l'IST et de Logo Services. Nous nous sommes également penchés sur la problématique à traiter, des objectifs, des résultats attendus et sur l'organisation de notre projet. Dans le chapitre suivant, nous élaborons un rapport de l'existant, nous exposerons une démarche de résolution du problème fondamentalement posé ainsi que les moyens mis à notre disposition pour la réalisation du projet

## Chapitre 2 : ETUDE DE L'INFRASTRUCTURE EXISTANT ET PROCEDURE DE RESOLUTION

### Introduction

L'implémentation de cette solution est implantée pour accroître l'efficacité des ressources informatique de de Logo Services. La présentation de l'infrastructure existant de Logo Services, de l'évaluation du système informatique et les démarches de résolution du problème constitueront le socle de notre travail dans ce chapitre.

### 2.1 La présentation de l'infrastructure existante

Pour un bon fonctionnement de l'entreprise il est nécessaire d'avoir un minimum d'infrastructure. Dans le but de réaliser et d'apporter des solutions aux préoccupations de ses clients, Logo Services s'est vu doté des matériels de premières nécessités afin de s'adapter aux réalités du terrain. En vue d'apporter son expertise à l'édifice de ses intérêts et à celles des autres il a réalisé des prouesses indéniables au grand bonheur de ses clients. Nous avons dressé le tableau ci-dessous qui illustre les différentes infrastructures de l'entreprise tout en précisant également leur nombre et leur caractéristique.

Désignation	Nombre	Caractéristiques
<b>Ordinateur portable</b>	10	HP Core i7
<b>Serveur physique</b>	01	HP DL 380 G9
<b>Serveur cloud</b>	01	4To en raid logiciel hébergé sur OVH
<b>Micro-Ordinateur de Bureau</b>	01	HP Core i5
<b>Vidéo Projecteur</b>	01	Epson série s 9
<b>Véhicule tout terrain rave 4</b>	0101	11 PP 3843 BF
<b>Fourgonnette utilitaire</b>	01	11 KJ 2980
<b>Véhicule avensis berline</b>	01	11 GP 6186 BF

Tableau. 2 : Présentation de l'infrastructure existante

L'infrastructure utilisée pour le déploiement

Pour le déploiement de notre solution nous avons utilisé un serveur cloud de **4To en raid logiciel hébergé sur OVH**. Pour le test de notre solution, il a été mis à notre disposition deux adresses IP.

### 2.2 Procédure actuelle de déploiement

Pour gérer le développement et le déploiement de ses projets, l'entreprise LOGO SERVICES possède une infrastructure de déploiement. Cette infrastructure se constitue d'un serveur cloud de qualimétrie logicielle.

Nos différents projets disposent des environnements de déploiement suivants :

Le développement : IL représente l'endroit où le projet est développé et où les tests unitaires sont performés (les machines du développeur).

L'intégration : Son objectif est d'assembler la tâche d'un groupe de développeurs sur un projet pour le tester.

Le test : Il permet le lancement des test fonctionnels sur le projet pour la validation de la qualité. Le test est également utilisé pour la reproduction des bogues.

La préparation (la préproduction) : C'est l'environnement de production pour l'exécution des tests autorisés avant le déploiement de production.

La production : C'est le lieu qui permet aux utilisateurs finaux d'interagir avec le projet.

## **2.3 Evaluation du système informatique**

### **2.3.1 Vue positive**

Nous avons apprécié positivement le système informatique de Logo Services sur plusieurs aspects. Nous pouvons essentiellement retenir :

L'efficacité des différents matériels dans la gestion et dans la sécurité des données

La vitesse d'exécution et la gestion des travaux

Une meilleure hiérarchisation du travail

Un amour pour un travail bien fait

### **2.3.2 Vue négative**

En se basant sur cette théorie qui rappelle qu'aucune œuvre humaine n'est parfaite et que le risque zéro n'existe pas, nous avons relevés certains incohérences ou incohérence qu'il faut songer à combler.

Une connexion limitée pour les utilisateurs

## **2.4 Quelques suggestions**

Pour remédier aux problématiques abordées, nous proposé après plusieurs discussions avec le personnel de Logo Services de :

Assurer la continuité de la mise en place de l'outil d'automatisation du déploiement qui permettra de gérer les problèmes majeurs.

#### **2.4.1 Démarche de résolution du problème**

Dans l'élan d'une meilleure amélioration du système de LOGO SERVICES, je formule ces suggestions suivantes :

Redéfinir une politique de sécurité plus sophistiquée

Mise en œuvre et le test de la solution

Mettre en place un wiki pour un échange avec les différents clients

La confidentialité dans certains domaines limite le champ d'apprentissage des stagiaires

Impliquer encore plus les stagiaires dans les travaux ;

Mettre à leur disposition une connexion de haute qualité pour leurs différentes recherches.

#### **2.4.2 Ressources logistiques utilisées**

La réalisation de notre projet nécessite des moyens pour son élaboration et sa mise en place tant en termes de connexion qu'également de moyen financier.

L'accès au cloud

Un local bien équipé

Les équipements informatiques

### **Conclusion**

En récapitulatif, dans ce chapitre il a été question de la présentation de l'existant de Logo Services. Nous avons évoqué les différentes critiques et les démarches de résolution du problème. La question sur les généralités de l'automatisation du déploiement sera évoquée dans le chapitre suivant.



## **Chapitre 3 : sur les outils d'automatisation pour le déploiement sur le cloud.**

### **Introduction**

Dans ce chapitre, nous allons évoquer les généralités du DevOps, des outils d'automatisation et le choix notre outil pour l'automatisation du déploiement. Nous allons également présenter et décrire l'automatisation de déploiement. Ce sont ces démarches qui contribueront à la résolution de problème. Nous parlerons également de MediaWiki sans oublier leurs avantages et leurs limites.

### **3.1 Généralités sur le DevOps**

L'expression “ DevOps” est composée des termes “développement” et “opérations” qui est une approche regroupant le développement et les opérations. Auparavant les développeurs écrivaient le code et les responsables de son intégration et de son déploiement étaient les administrateurs système. Cependant avec le flux de travail continue, un changement s'est donc imposé pour répondre aux nouvelles exigences. L'approche DevOps a vu le jour lorsque des nouveaux défis et rôles s'imposaient. Nous sommes de nos jours tous unanimes que cette approche de développement logiciel est la plus populaire. Le DevOps repose sur un ensemble de principe que sont à travers l'acronyme “CAMES” qui signifie Culture, Automatisation, Mesure, Partage.

- Une culture, parce qu'il est doté d'un esprit de collaboration entre les équipes de développement et d'opérations. Elle porte sur la collaboration et la communication perpétuelle, le partage de responsabilité, les variations progressives et la résolution prématurée des problèmes.
- La mesure des KPIs (Indicateurs Clés de Performances) autorise d'observer en permanence l'évolution des tâches composant le flux DevOps. Il nous revient de prendre des initiatives basées sur les informations réelles, de ce qui marche ou pas et aussi d'optimiser les performances.
- Le troisième principe montre que les équipes doivent partager le feedback, les meilleures manières de faire, et le savoir afin de favoriser la clarté, de construire ensemble une pensée collective et d'écarter les contraintes.
- Nous avons un autre processus du DevOps qui porte sur l'automatisation systématique des procédures, de test, de configuration et de déploiement. L'automatisation nous permet de supprimer les tâches redondantes et de nous concentrer sur les activités essentielles qui ne peuvent pas être automatisées.

#### **3.1.1 Définition des concepts**

Nous définissons l'automatisation comme étant le fonctionnement d'un système sans intervention humaine.

Un déploiement se définit comme étant la promotion de composants d'une application depuis un environnement vers l'environnement suivant.

L'automatisation du déploiement est un procédé qui permet de lancer de nouvelles fonctions et applications plus rapidement et fréquemment, sans une intervention humaine. Elle est cette méthode qui nous permet de déployer nos applications dans notre environnement de manière continue. Son importance se justifie également dans la mise en place des pratiques de DevOps et la gestion du pipeline CI/CD (Intégration Continue/ Distribution Continue).

### **3.1.2 Les concepts du DevOps**

Nous avons plusieurs concepts qui expliquent le mouvement du DevOps

- Nous pouvons dire que l'Intégration Continue (CI) repose sur une étape de tests automatisés intégrée aux flux de déploiement. La CI est un ensemble de pratiques utilisé par les développeurs qui consiste à vérifier à chaque modification de codes sources que le résultat des changements n'affecte pas la baisse de l'application développée. L'intégration continue présente parfois une phase de création ou d'intégration du processus de publication logiciel, suscite un aspect automatisé et un aspect culturel. Ses objectifs sont de trouver et de réparer plus vite les bogues, de perfectionner la qualité des applications et de faire de nouvelles mises à jour de logiciels.
- La livraison continue, connue comme étant la deuxième étape de DevOps, est une extension de l'intégration continue. Au-delà des simples tests d'unités, elle assure aux développeurs d'automatiser les tests dans le but de vérifier autres aspects d'une mise à jour d'application avant de la déployer auprès des utilisateurs. Cela peut concerner les tests de fiabilités de l'API, d'intégration et d'interface qui permettront de détecter les possibles problèmes à rectifier avant le déploiement.
- Le déploiement continue est une progression avancée de la livraison continue permettant de déployer toutes les voies en production sans aucune intervention humaine. Cette phase du processus, le déploiement continue, déclenche automatiquement le produit logiciel chez les utilisateurs finaux. Les outils utilisés par l'automatisation permettent de transférer le produit logiciel à des serveurs publics.

### **3.1.3 Outils du DevOps**

La mise en œuvre de l'intégration continue, de la livraison continue et du déploiement continue nécessitent des outils différents et parfois aussi les mêmes. Cela prend en compte les outils de

test, de gestion de la configuration et de création de systèmes et de développement d'application. Sa mise en œuvre s'étale sur quatre étapes que sont :

- Le dépôt du code source : Les développeurs s'en servent pour enregistrer et modifier le code. Le référentiel de code source conduit autres versions du code qui est classer afin d'attester que les développeurs écrivent le code de chacun. Sont quelques-uns des outils de référentiel de codes célèbres Git, Eclipse, IBM Rational et Apache
- La création d'un serveur : C'est l'outil d'automatisation qui est exploité dans le but de compiler le code en code source compréhensible dans la base de code exécutable. Nous avons entre autres Jenkins et Artifactory.
- La gestion de la configuration : Elle assiste à la gestion de la configuration de l'application et de l'infrastructure pour permettre le maintien de la cohérence entre les environnements. Nous avons comme outils Ansible, Puppet et Chef.
- L'infrastructure virtuelle : Nous nous pouvons retenir qu'Azure, AWS et GCP sont les principaux fournisseurs de cloud public qui octroient presque tous les services d'infrastructures. Ils fournissent une API qui nous aide à provisionner l'infrastructure et de gérer comme un code.

### **3.2 Présentation et description de l'automatisation de déploiement**

#### **Présentation de la solution**

L'automatisation du déploiement nous permet de déployer nos applications entre les environnements de tests et de production grâce aux mécanismes automatisés. L'automatisation de déploiement est ce processus qui permet de lancer de nouvelles fonctions et applications le plus rapidement et chaque fois sans l'utilisation ou l'intervention humaine. Notre solution permettra à l'entreprise de déployer ses différentes applications et ses sites web sur son cloud avec plus de tranquillité et sans quiétude. Nous avons réalisé en testant le déploiement de MediaWiki pour le déploiement sur le cloud. Le choix de ansible est de nous permettre créer les playbooks pour nos différents déploiements.

#### **3.2.2 Description de la solution**

L'automatisation est cruciale de nos jours, avec des environnements informatiques trop complexes et qui doivent souvent évoluer trop rapidement pour que les administrateurs systèmes et les développeurs puissent suivre s'ils devaient tout faire manuellement. Elle consiste à réaliser les tâches avec un intervention humaine très réduite et moindre pour faciliter les flux des feedbacks entre les équipes d'exploitation afin d'accélérer le déploiement en production des mises à jour itératives apportées aux applications. Nous avons jugé qu'il est nécessaire d'apporter notre contribution pour le bien-être de l'entreprise pour l'amélioration des différentes infrastructures. Dans le but de simplifier les tâches pour rendre la fluidité de gestion aux administrateurs qui pourrons nous éviter les risques d'erreur lors de nos déploiements.

L'automatisation augmente l'efficacité de notre structure et nous libère en temps concernant les tâches répétitives. En somme notre solution consiste à automatiser le déploiement des applications sur le cloud en réalisant un test de déploiement avec MediaWiki en utilisant ansible comme outils.

### **3.2.3 Utilisation de la solution**

Il est à la portée de l'entreprise d'automatiser le déploiement de leurs applications et leurs sites web sur son cloud tout accédant et gérant ses différentes interfaces. De nos jours plusieurs entreprises tendent vers l'automatisation du déploiement.

### **3.2.4 Les étapes d'automatisation du processus de déploiement des logiciels**

Il y'a trois étapes principales nécessaires qui caractérisent le pipeline de déploiement. Il nous aide à automatiser le processus de déploiement et garantissant un accès diligent du code source en déploiement. Ces étapes sont :

La création : Dans cette étape, le développeur expédie du code dans un emplacement de stockage pour les logiciels (référentiel de logiciel). Toutes les modifications portées au code doivent être introduites dans des environnements qui représentent l'environnement de création.

Le test : Après la création, l'outil d'automatisation de déploiement comme par exemple Ansible ou Jenkins repère le nouveau code et lance une vague de tests. Lorsque le code a franchi tous les tests alors il peut être mis en production.

Le déploiement : Cette étape est justifiée par le déploiement de l'application en production et mise à la portée des utilisateurs.

## **3.3 Généralités sur quelques plateformes d'automatisation**

Dans le domaine du déploiement de nos solutions nous disposons de plusieurs outils d'automatisation. Nous avons fait le choix d'énumérer quelques-uns.

Ansible est un outil d'automatisation de la gestion de la configuration, de la création de machine virtuelle ou du déploiement d'application dans nos différentes infrastructures informatiques. Ansible utilise des clés SSH, pour authentifier et gérer les machines cibles étant donné que c'est un modèle sans agent. Les playbooks assurent les différentes tâches de configuration définies sans oublier les modules de Ansible qui permettent d'effectuer des tâches spécifiques.

Puppet est une plate-forme d'automatisation ou un outil de gestion de configuration logicielle utilisé par les entreprises qui gère le processus de diffusion et de déploiement d'application. Les agents sont installés sur les machines cibles pour permettre à Puppet Master d'exécuter des manifestes qui définissent la configuration souhaitée de l'infrastructure Azure et des machines virtuelles. Il est Open source et la fois professionnelle. Puppet peut s'intégrer à d'autres solutions

telles que Jenkins et GitHub pour un flux de travail DevOps amélioré. Son objectif est de permettre de partager, d'appliquer des changements à travers un datacenter, de tester, tout en préservant la visibilité et le reporting pour la prise de décision et la conformité.

Chef est un Framework open source de gestion de configuration, de déploiement et d'intégration de systèmes dans le cloud écrit en Ruby. IL transforme l'infrastructure en code, la rendant souple, lisible et testable, quelle que soit la plateforme où il s'exécute. Chef nécessite l'écriture d'une multitude de recettes expliquant une série de ressources qui manifeste un état particulier : les packages devant être installés, les services devant être exécutés, ou les fichiers devant être écrits. Disposant des composants comme Chef Habitat qui assure l'automatisation du cycle de vie de l'application au lieu de l'infrastructure, il présente aussi Chef InSpec qui permet d'automatiser la conformité avec les exigences de sécurité et de stratégie.

### 3.3.1 Le choix des critères

Les critères ci-dessous orienteront notre choix d'outil d'automatisation du déploiement et de gestion de configuration.

Contributeur : Nous parlons ici du nombre de contributeur dans le code source de l'outil.

Sans Agent : La latitude de pouvoir contrôler des machines

Langage : Le langage de programmation

Dépendance client : Les logiciels prérequis pour configurer une machine

Installation : La complexité de mise en place de l'outil d'automatisation dans notre environnement de production ;

Mécanisme de partage de configuration, push ou pull ;

Approche de description d'état de machine, déclaratif ou procédural.

Le système de communication des outils d'automatisation

Déploiement : L'outil le plus approprié au déploiement de nos applications ;

Maintenance : Assurer l'entretien des infrastructures,

La sécurité des différents environnements.

### 3.3.2 Etude comparative et synthèse.

Outils	Ansible	Puppet ou marionnette	Chef
--------	---------	-----------------------	------

Critères			
Contributeur	3432	492	522
La langue de configuration	Python	Ruby	Ruby
Sans Agent	Maître uniquement (sans agent)	Maître-Agent(intermédiaire)	Maître-Agent(intermédiaire)
Dépendance client	Python	Ruby	Ruby, sshd, bash
Approche	Procédural	Déclarative	Procédural
Installation	Facile	Difficile	Peu Facile
Maintenance	Sa maintenance n'est pas une tâche difficile	Il nécessite une maintenance de tous les serveurs supplémentaires	Il a besoin d'un entretien élevé.
Déploiement	Avec ansible, il est facile de déploiement des applications.	Un déploiement d'applications sur Puppet n'est pas simple.	Un chef n'effectue pas de déploiement d'application.
Sécurité	Sécurité efficace	Sécurité moyenne	Sécurité moyenne
Système de communication	Plus rapide	Lent	Très lent

Tableau. 2 : Etudes comparatives et synthèse

### 3.3.3 La synthèse et le choix

A la fin de cette étude comparative, nous avons défini notre choix technologique parmi les outils d'automatisation cités, mais nous allons résumer.

Nous avons opté Ansible pour l'automatisation de déploiement qui est plus facile par rapport aux autres outils d'automatisation. Nous pouvons également l'utiliser pour la gestion de nos configurations.

Ansible est un outil de déploiement extrêmement simple et efficace à utiliser par rapport à Puppet ou autre. C'est un outil qui est assez adapté aux problématiques du Cloud. Ansible nous permet de gagner en temps dans la gestion de nos infrastructures en nous assurant de par sa rapidité et sa transparence la création des bibliothèques de rôles. Ansible est un outil qui est très prolifique aux utilisateurs quotidiens. Il décrit l'interrelation de tous nos systèmes informatiques en utilisant un langage écrit en YAML sous la forme d'Ansible playbooks. Le choix de ansible est dû au fait qu'il efficace pour la résolution de notre problème à savoir le déploiement de nos solutions.

### **3.4 Procédure d'amélioration de l'automatisation des déploiements**

D'abord il est important de documenter dans un outil qui est accessible à tout le monde le processus de déploiement existant. Nous pouvons utiliser par exemple GitHub ou Google Docs et pourquoi pas un wiki.

Ensuite, nous devons nous donner la peine de manière progressive d'essayer de simplifier et automatiser le processus de déploiement. Cette démarche prend en compte généralement quelques tâches telles que :

L'automatisation du déploiement et la configuration des intergiciels

Le redémarrage des serveurs, des applications ou des services

La création des machines virtuelles ou des conteneurs préconfigurés

Effectuer des tests de déploiement automatisé pour s'assurer que le système fonctionne et est correctement configuré

L'exécution des procédures de test

Développer des scripts pour automatiser la migration de bases de données

Enfin pour mieux accomplir il est nécessaire de supprimer les configurations manuelles pour maintenir pour une exploitation maximale des capacités de votre entreprise.

#### **3.4.1 Avantage de l'automatisation de déploiement**

Productivité : Cette intégration continue offre un gain en temps à l'équipe de développeurs tout en automatisant les tâches obsolètes et en diminuant les tâches manuelles. Elle limite la quantité d'erreur et les anomalies liés à la version publiée au moment des tests.

La mise en œuvre de l'automatisation du déploiement est rapide et efficace que parfois il n'est pas nécessaire de vérifier les validations.

Des études ont montré les trois bienfaits qui exhortent les entreprises à s'orienter vers l'automatisation dans leur service informatique. Ce sont tels que : le gain d'efficacité estimé à 78%, la réduction des menaces informatiques d'ordre de 40% et la restriction des coûts informatiques d'une valeur de 58%.

#### **3.4.2 Limite de l'automatisation du déploiement**

L'automatisation du déploiement de la solution présente quelques insuffisances que sont :

Sa complicité quant à sa configuration en termes de ressources.

La mystification de l'automatisation qui selon certains préjugés supprimera plusieurs postes mettent ainsi certaines personnes au chômage.

Les coups de l'automatisation de déploiement sont élevés.

Nous pouvons également dire que tout n'est pas automatisable et il faut souvent la présence humaine dans certaines situations complexe.

### **3.5 Aperçu sur Media Wiki et sur le cloud**

- Définitions et aperçu sur Media Wiki

MediaWiki peut-être défini comme étant un moteur qui assure la gestion du wiki qui fréquemment utilisé par les entreprises comme une solution de gestion des connaissances et des contenus. Il peut être utilisé pour les plateformes de formation à distance. MediaWiki dispose plusieurs fonctionnements pour faciliter son utilisation par les différents utilisateurs.

La motivation principale qui nous a conduit à porter notre choix sur le test de MediaWiki sur le cloud est la suivante :

MediaWiki est un logiciel wiki largement utilisé pour animer Wikipédia et à cet effet plusieurs personnes seront plus motivées pour utiliser MediaWiki.

C'est un logiciel GPL c'est à dire open source qui provient du projet GNU.

Il présente des enjeux tels que la liberté parce qu'il nous permet d'utiliser le programme à notre fin. Il n'incombe à personne d'imposer ses choix pour faire obstacles aux autres.

MediaWiki est un logiciel qui prône l'égalité car nul n'est aussi au-dessus de l'autre, ils partagent les mêmes valeurs et possèdent les mêmes contenus.

- Définitions et aperçu sur le cloud

Le cloud est une solution qui offre une réduction des coûts de l'infrastructure informatique qui est décentralisée et rendue accessible peu importe l'endroit dans le monde. Le cloud, pour la gestion des applications et des données fait appel à internet et aux serveurs. Son rôle n'est plus à démontrer car elle bannit la notion de machine physique en fournissant des services sur des machines virtuelles allouées dans des plates-formes de grandes capacités.

Nous sommes poussés à faire un choix très important parmi les différents types de cloud pour une meilleure implémentation. Nos choix sont portés sur d'innombrables critères que nous pouvons retenir entre autres les niveaux de sécurité et de personnalisation variables. Nous citons ici quelques types de cloud.



Le cloud public : les ressources informatiques de l'entreprise sont stockées sur un serveur mutualisé, autrement dit partagé entre plusieurs clients, et accessibles par Internet. Ces serveurs sont partitionnés de manière à interdire les fuites de données.

Le cloud privé : comme son nom l'indique, il est dédié à un seul utilisateur. L'avantage du cloud privé est son important niveau de sécurité, renforcé par une connexion VPN. Le cloud privé est administré par l'entreprise elle-même ou un prestataire de services.

Le cloud hybride : l'entreprise utilise à la fois le cloud privé et le cloud public pour mettre en œuvre certaines activités. Par exemple, le cloud public est utilisé par les collaborateurs pour les tâches opérationnelles, tandis que le cloud privé sert à héberger le site web e-commerce de l'entreprise ou ses données financières, pour réduire le risque de piratage.

Le multicloud : le multicloud est un type de déploiement cloud qui implique l'utilisation de plusieurs cloud publics. Autrement dit, une organisation disposant d'un déploiement multiclouds loue des serveurs et des services virtuels auprès de plusieurs fournisseurs externes). Les déploiements multiclouds peuvent aussi être des cloud hybrides, et vice-versa.

Le cloud communautaire : plus rarement utilisé, il consiste à partager un espace donné entre plusieurs entreprises ayant les mêmes exigences en matière de sécurité et de confidentialité. Il s'apparente donc à un cloud privé partagé.

### 3.6 Estimation du Coûts

Composant	Quantité	Capacité	Prix
Serveur Cloud	1		
Adresses IP	2		

Tableau. 4 : tableau des Coûts

## 4.7 Architecture l'automatisation du déploiement de la solution

### Conclusion

Pour conclure ce chapitre, nous pouvons nous réjouir d'avoir évoqué la généralité sur le DevOps, les raisons qui nous ont poussé à recourir à ansible afin de répondre aux aspirations de LOGO SERVICES. Alors pour parvenir à nos fins il nous revient de mener plus de réflexion sur la solution choisie.

## Chapitre 4 : Présentation de la solution choisie

### Introduction

Cette partie de notre travail consistera d'abord par définir les concepts, à présenter son histoire, et en précisant tous ses composants. Ensuite, nous évoquerons la présentation et les objectifs de notre solution choisie et en dressant quelques versions. Enfin, nous montrerons les avantages et les limites de cette solution et son architecture sans oublier l'apport de ce stage.

### 4.1 Définition des concepts

Ansible est un outil d'automatisation open source et est utilisé pour les tâches informatiques telles que la gestion de la configuration, le déploiement d'application, l'orchestration interservices et l'approvisionnement.

### 4.2 Histoire de ansible

Lorsque nous examinons l'histoire d'Ansible, il est facile de comprendre la motivation qui a poussé les inventeurs de l'outil, à créer ce système d'automatisation, atteignant aujourd'hui un seuil de maturité assez exceptionnel.

L'outil Ansible a été développé par Michael DeHaan (il n'est pas le seul bien sûr, il faut tout une armée d'informaticiens pour construire un tel outil, mais l'idée de base vient de lui), c'est aussi l'auteur de l'application de serveur de provisionnement [Cobbler](#) (l'un de ses premiers projets qu'il a commencé à Redhat). Red Hat était très intéressé par un outil de gestion de configurations, plus particulièrement après l'arrivée de Puppet et Chef dans le marché des outils d'automatisation.

Pendant ce temps, le mot DevOps a commencé à être beaucoup utilisé, et diverses discussions ont lieu sur les chaînes d'outils DevOps. Les discussions sont entièrement axées sur la culture mais surtout sur le fait de rendre les outils (souvent Open Source) plus faciles à utiliser, plutôt que d'acheter un outil universel à un grand fournisseur.

Auparavant Michael DeHaan travaillait pour Puppet en tant que chef de produit pendant un certain temps, ou il y a eu quelques divergences. Pour lui la simplicité de l'outillage, aurait dû être plus prioritaire et la chose la plus importante dans un outillage d'automatisation.

Après avoir quitté Puppet et travaillé dans de nouveaux projets, il n'a pas réussi à convaincre les gens et ses nouveaux collègues à utiliser Puppet en tant qu'outillage d'automatisation principal. Il trouvait aussi que les utilisateurs n'étaient pas non plus en mesure d'adopter des pratiques d'automatisation de style DevOps en raison de la complexité des outils

alternatifs à Puppet. Les scripts internes étaient considérés comme plus faciles, mais ce n'était pas plus fiable.

L'idée d'un outil d'automatisation simple a commencé à émerger. Ansible n'a pas vraiment commencé non plus, même si les idées se formaient rapidement. Finalement, c'était inévitable, En février 2012 Ansible a donc commencé en tant que projet. Il a décollé assez rapidement grâce à de nombreux administrateurs systèmes et développeurs travaillant d'arrache-pied sur le projet. Le but principal consistait à montrer au monde qu'il y avait un moyen plus simple d'automatisation.

Pari réussi, puisque l'entreprise Red Hat rachète Ansible en octobre 2015, Ansible tourne actuellement à plein régime, et le projet est pris en charge par une entreprise de bonne taille (Red Hat). Ansible est actuellement l'outil de gestion de configuration le plus étoilé et le plus forké sur [GitHub](#).

Ce document a été tiré dans « Introduction au cours complet de Ansible » et est retrouvable sur ce lien : [Introduction au cours tutoriel complet sur Ansible \(devopssec.fr\)](#).

### **4.3 Les composants de base de Ansible**

Inventaire : C'est le fichier ayant le format INI qui contient la liste des machines à gérer qui seront ciblées par les playbooks. Nous avons comme exemple un conteneur hébergé chez OVH. Nous pouvons également d'autres formats comme le YAML pour le représenter.

Nodemanager ou le poste de gestion des machines : Elle est la machine sur laquelle ansible est installé. Dans notre cas nous l'avons installé dans notre environnement virtuel. Aucun logiciel n'est déployé sur les serveur géré car ansible est sans agent.

Un node : C'est le poste connecté au poste de control en ssh. Ansible n'est pas installé sur ce node.

Les tâches (tasks) : C'est un ensemble de procédure ou d'actions à exécuter sur une machine. Cela peut être l'installation d'un paquet logiciel ou de la création d'un utilisateur ou d'un groupe.

Un playbook un fichier YAML décrivant la configuration à déployer. Nous pouvons citer comme exemple le déploiement des serveurs. Il est aussi idempotent du fait qu'il peut être exécuté sur un système à chaque instant sans avoir un impact négatif sur celui-ci.

Les groups : permettent de classifier et de ranger les différentes machines tout en assurant une mise en hiérarchie.

Les groups\_vars : sont quant à eux un fichier qui contiennent leur propre configuration.

Les host\_vars : sont un fichier de machine possédant une configuration propre à la machine

Les rôles : représentent un ensemble de fichiers hiérarchisé dans une structure arborescente.

Un module : Ce sont des morceaux de codes généralement en python pouvant être utilisés sur une CLI ou pour modifier l'état d'une propriété d'un hôte. Il peut être écrit dans n'importe quel langage pouvant conduire au JSON. Apt, Template et copy sont par exemple des modules.

Les handlers : Ils assurent la gestion des tâches qui doivent être effectuées en cas de changement lors d'un arrêt ou d'un redémarrage d'un service.

Le cloud : C'est le serveur distant hébergé en ligne et utilisé pour traiter, stocker et gérer les données.

Les plugins : Ils aident à exécuter les tâches ansible avec facilité.

Il faut également les différentes applications et sites web à déployer. Ces déploiements se feront à l'aide du logiciel ansible avec moins d'intervention humaine.

Il est important également de savoir sur quels types de cloud il faut pour le déploiement des solutions. Ainsi en fonction des différents critères il serait convenable de choisir le cloud qui répond et qui respecte ses critères.

Les utilisateurs : Ce sont eux qui créent le playbooks Ansible ont une liaison directe avec ansible,

Le cloud privé : Il sert de corrélation avec les APIs et tous les modules et également avec l'ensemble du cloud,

L'API : il permet de créer l'API nécessaire pour l'interdépendance des modules

La base de données de gestion de la configuration : Elle représente un ensemble qui est constituée de tout un réseau d'ordinateurs d'infrastructure informatique.

Exécution des tâches parallèles Ad Hoc : Utiliser Ansible en mode Ad Hoc montre que nous n'allons pas utiliser les scripts mais plutôt des commandes manuelles dans la mise en place des bases. Ceci nous aidera de les automatiser et de les appliquer au node dans le même instant. Elle ne nécessite pas de configuration supplémentaire car on peut lui communiquer immédiatement lorsque nous avons accès à l'instance.

Les clés SSH : Ansible prend en compte les mots de passe, mais l'un des moyens le plus efficace pour utiliser ansible reste la gestion des clés ssh grâce ssh-agent. Elles nous permettent de nous connecter directement en tant qu'utilisateur sans passer par la connexion "root" tout en utilisant le "su" ou le "sudo". Pour contrôler les accès ssh, Ansible dispose d'un module appelé "authorized\_key".

## **4.4 Présentation de ansible et de ses objectifs**

### **4.4.1 Présentation de Ansible**

La première version de Ansible est sortie le 20 février 2012 par Michael DeHaan. Ansible évolue constamment et une nouvelle est proposée approximativement tous les deux mois. Ansible a été racheté en 2015 par RedHat qui lui a racheté par IBM en 2018.

#### **4.4.2 Ansible Tower**

C'est un service web, un API et une console développer pour rendre Ansible utilisable pour les groupes informatiques avec des volets différents. C'est une console principale de gestion des tâches d'automatisation. Il présente une interface simple.

Ansible Tower a été annoncé par Red Hat, à qui la prise en charge lui revient. Il l'a annoncé à l'AnsibleFest 2016 qu'il passait en Open Source. Tower est apparu sous le nom de AWX. Son alternative qui est Open Source et écrit en GO n'est d'autre que Sémaphore.

#### **4.4.3 Les objectifs de conception de ansible**

Avoir une bonne compréhension du fonctionnement de Ansible

La capacité d'installer et configurer Ansible

Pouvoir créer des scripts de déploiement avec Ansible

Gérer les infrastructures de serveurs à partir de ansible

La sécurité nos infrastructures : Dans le cadre du déploiement, Ansible ne déploie pas d'agent sur le nœud. OpenSSH ou HTTPS est seulement nécessaire pour débiter une gestion.

#### **4.5 Listes des différentes versions de Ansible**

Les changements de version en version interviennent 4 à 6 mois lorsque la version d'Ansible est publiée. Chaque nouvelle est d'abord testée avant sa sortie officielle disponible.

La version antérieure de Ansible à 1.9

Plus d'une trentaine de nouveaux modules ont été ajoutés dans Ansible 1.6. L'ajout des rôles dans la version 1.2 d'Ansible.

La version 1.9

L'amélioration apportée aux modules de service web Amazone et de nouvelles améliorations dockers ont été ajoutées. Ajout de nouveaux modules et de filtres.

La version 2.0

Dans cette version il y'a un amortissement de certains « ssh » comme suit : « ansible\_ssh\_user » à ansible\_user, « ansible\_ssh\_host » à « ansible\_host » et « ansible\_ssh\_port » à « ansible\_port ».

#### La version 2.1

Dans cette version nous assistons à une réimplémentation de de la fonctionnalité de fichier de nouvelle tentative. Il y'a également l'ajout des nouveaux modules à Azure.

#### La version 2.2

Pour les tâches de longue durée, il y' a eu la prise en charge de Windows 'async :'. Les tâches 'meta' peuvent dorénavant utiliser les conditions.

#### La version 2.3

L'ajout des « métadonnées » aux modules pour l'activation de la classification. La mise à jour du nom de l'hôte afin de pouvoir travailler avec les versions les plus récentes de RHEL7.

#### La version 2.4

Le remplacement par de nouveaux mots-clés les anciens mots-clés. La division de la classe inventaire afin d'offrir une meilleure gestion.

#### La version 2.5

La version 2.5 a permis d'ajouter 'loop' pour améliorer l'utilisation des boucles de tâches. Nous pouvons utiliser le nouveau filtre qui vient d'être crée qui convertit la réponse XML à l'objet JSON à partir d'un réseau.

#### La version 2.6

Dans cette version le « check\_mode : non » a fait place à « always\_run ». Nous constatons également la correction de 'AnsibleParserError' qui était absent dans les versions précédentes.

#### La version 2.7

La prise en charge de simplejson a été retirée. Pour ignorer l'échec des tâches lié par l'inaccessibilités un nouveau mot-clé a été ajouté qui 'ignore\_unreachable'.

#### La version 2.8

Dans cette version l'architecture de plugin a pris la place de '- become'. Nous avons l'avantage de pouvoir utiliser certains arguments comme -sudo, -sudo-su, etc et peuvent actuellement être employés comme -devenir, -devenir-utilisateur, etc.

#### La version 2.9

La correction des fautes de frappe au niveau des différents modules de Ansible. Nous avons également l'ajout de certaines fonctionnalités de support pour Zabbix, le docker et d'autres IDE.

Le développement actuel nous oriente vers d'autres améliorations et l'ajout de nouvelles fonctionnalités à Ansible pour une meilleure utilisation. De nos jours nous constatons que son

usage est plus propagé parmi les outils déploiement continu. Cela s'explique par la convivialité de son langage utilisé dans le script qui s'avère facile.

#### **4.6 Avantages et désavantages de l'outil d'automatisation ansible**

- **AVANTAGES**

Nous disons que ansible est gratuit parce qu'il est un outil open source.

La puissance de ansible réside dans sa capacité à modéliser des workflows informatiques même les plus difficiles

De par sa flexibilité il nous permet de gérer toutes les configurations des applications quel que soit l'endroit où il est déployé tout en les personnalisant en fonction de nos besoins.

Ansible est un meilleur facilitateur dans la répétition des tâches fréquentes et également dans l'exécution des mêmes commandes sur plusieurs systèmes à la fois.

Ansible est un outil très efficace car nous n'avons pas besoin d'une installation supplémentaire dudit logiciel. Le serveur dispose du nécessaire pour toutes les ressources d'application.

- **Limites**

Ansible n'a pas complètement grandi et est livré avec des fonctionnalités limitées.

Ansible mise plus sur l'orchestration que sur la configuration.

L'interface graphique de ansible est sous développée

Il n'a pas d'architecture principale et alors pas de serveurs supplémentaires.

L'apprentissage de ansible est très difficile pour les apprenants et cela nécessite d'énormes efforts.

#### **4.7 L'apport du stage**

Nous avons commencé ce stage avec beaucoup de convivialité afin d'aboutir à un résultat très satisfaisant. Grâce à ce stage nous avons engrangé beaucoup d'acquis sur plusieurs plans. Nous nous réjouissons d'avoir collaborer avec tous ses cadres et toutes ses personnes qui nous ont permis de nous améliorer sur le plan professionnel. Pour nous, ce stage a également été prolifique dans la gestion de résolutions des problèmes.

#### **4.8 L'architecture de Ansible**

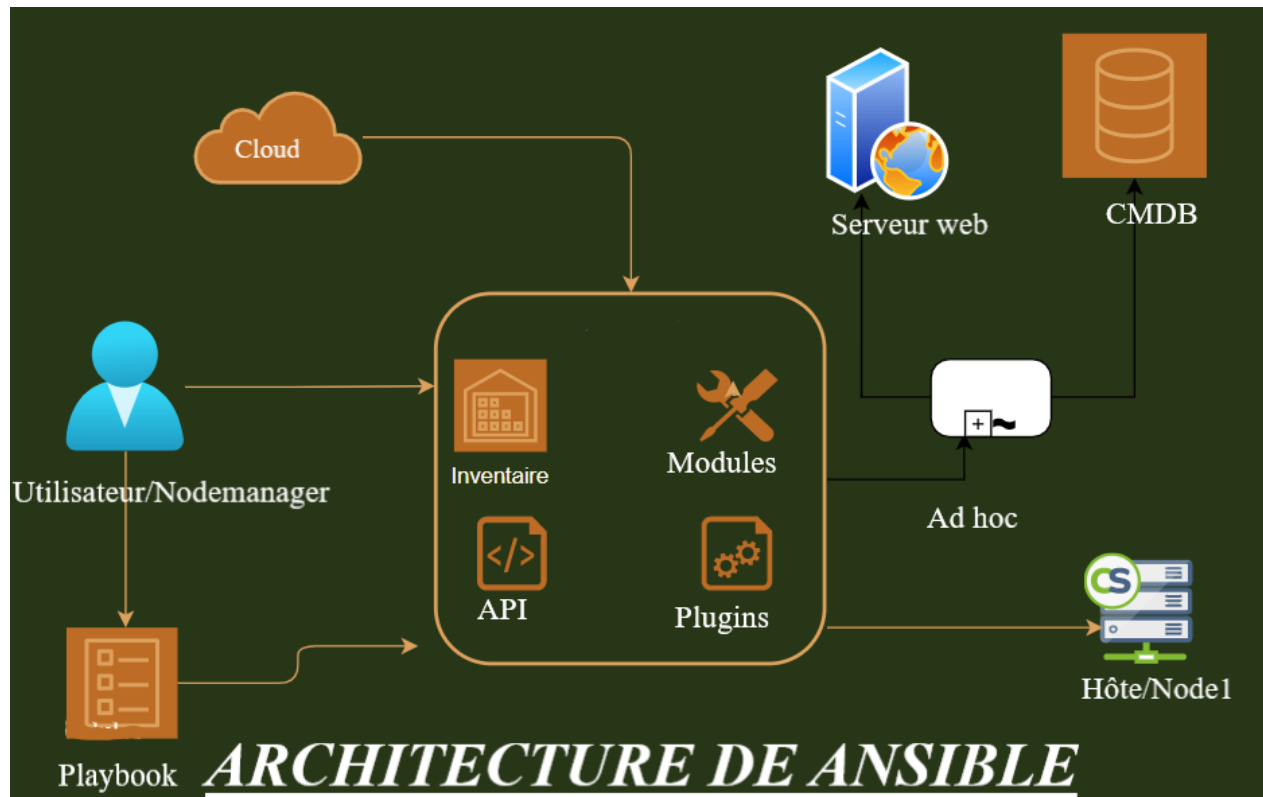


Figure 1 : L'architecture de Ansible

### Conclusion

En somme nous retenons que les déploiements de nos applications et de nos sites web, l'amoindrissement de l'intervention humaine sont les services fournis par notre outil d'automatisation qui est Ansible. Notre solution est limitée du fait que les coups de l'automatisation de déploiement et de sa jeunesse. Nous avons également évoqué les acquis engrangés grâce à ce stage.



## Chapitre 5 : L'implémentation de la solution ansible pour le déploiement de media wiki sur le cloud.

### Introduction

#### 5.1 Procédure d'implémentation de la solution ansible pour le déploiement de media wiki sur le cloud.

Mise en place et configuration de ansible sur le node manager et dans l'environnement virtuel

Pour la mise en place de notre solution nous avons utilisé un système d'exploitation Unix en utilisant la distribution Debian. Nous allons utiliser un serveur pour contrôler toutes les opérations que nous dénommons le nodemanager et un serveur pour le déploiement que nous appellerons le node1 ou http1.

L'installation et la configuration de ansible sur DEBIAN.

Instauration de la communication avec les nodes.

Dans cette partie il sera essentiellement question d'établir la communication avec les nodes.

#### 5.2 Préparation de la communication

Etant donné que nous disposons de deux serveurs, alors nous attribuerons à chacun de nos nodes des rôles bien spécifiques.

Sur le serveur 1 nous installerons Apache, PHP, Maria DB et MediaWiki ; nous dénommerons donc http1.

Nous allons de configurer la résolution de nom via le fichier */etc/hosts* sur le node manager du fait que nous n'avez pas accès au DNS de l'entreprise.

A défaut d'utiliser les adresses IP nous utiliserons les noms de machine qui sont sauvegardés dans le DNS. Dans ansible nous prenons un raccourci en utilisant un fichier */etc/hosts* car il nous permettra de mettre en place des correspondances entre les noms des serveurs et leurs adresses IP. Pour cela nous allons nous connecter sur le node manager en mode root en faisant appel à cette commande *sudo su*.

Nous allons enregistrer le node dans le fichier */etc/hosts*

Dans cette étape nous allons Installer le paquet **python-virtualenv**, ce qui permettra de créer des environnements de travail virtuel : **virtualenv**

**Création d'un utilisateur sur le node manager**

```
user you are currently creating.
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for user-ansible
Enter the new value, or press ENTER for the default
    Full Name []: user-ansible
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
```

Figure 2: Création d'un utilisateur sur le node manager

Nous allons nous connecter avec notre utilisateur que nous venons de créer.

```
root@nodemanager:~#
root@nodemanager:~# su - user-ansible
user-ansible@nodemanager:~$
```

Figure 3: Connection avec l'utilisateur user-ansible

### Création de mon environnement virtuel

```
user-ansible@nodemanager:~$ sudo virtualenv ansible2.7.10
[sudo] password for user-ansible:
Running virtualenv with interpreter /usr/bin/python2
New python executable in /home/user-ansible/ansible2.7.10/bin/python2
Also creating executable in /home/user-ansible/ansible2.7.10/bin/python
Installing setuptools, pkg_resources, pip, wheel...done.
user-ansible@nodemanager:~$
```

Figure 4: Création de mon environnement virtuel

### Activation de l'environnement

Pour activer notre environnement nous utiliserons la commande *source ansible2.7.10/bin/activate*.

Dans cette étape nous installerons ansible dans notre environnement virtuel

Lorsque nous utilisons cette commande *sudo pip install ansible==2.7.10* pour installer ansible dans notre environnement virtuel, nous obtenons le résultat suivant que montre la figure ci-dessous.

```

root@nodemanager: ~
(ansible2.7.10) user-ansible@nodemanager:~$ sudo pip install ansible==2.7.10
Collecting ansible==2.7.10
  Downloading https://files.pythonhosted.org/packages/9a/9d/5e3d67bd998236f32a72f255394eccd1e22b3e2843aa60dc30dd164816d0/ansible-2.7.10.tar.gz (11.8MB)
    100% |#####| 11.8MB 152kB/s
Collecting PyYAML (from ansible==2.7.10)
  Downloading https://files.pythonhosted.org/packages/ba/d4/3cf562876e0cda0405e65d351b835077ab13990e5b92912ef2bf1a2280e0/PyYAML-5.4.1-cp27-cp27mu-manylinux1_x86_64.whl (574kB)
    100% |#####| 583kB 3.2MB/s
Requirement already satisfied: cryptography in /usr/lib/python2.7/dist-packages (from ansible==2.7.10)
Collecting jinja2 (from ansible==2.7.10)
  Downloading https://files.pythonhosted.org/packages/7e/c2/1eece8c95ddbc9b1aeb64f5783a9e07a286de42191b7204d67b7496ddf35/Jinja2-2.11.3-py2.py3-none-any.whl (125kB)
    100% |#####| 133kB 14.1MB/s
Collecting paramiko (from ansible==2.7.10)
  Downloading https://files.pythonhosted.org/packages/ba/e5/16e51472617cdcc90a4b9019543087f584a73b76d501d4b165abb7893fe9/paramiko-2.8.1-py2.py3-none-any.whl (206kB)
    100% |#####| 215kB 9.4MB/s
Requirement already satisfied: setuptools in /usr/lib/python2.7/dist-packages (from ansible==2.7.10)
Collecting MarkupSafe>=0.23 (from jinja2->ansible==2.7.10)
  Downloading https://files.pythonhosted.org/packages/fb/40/f3adb7cf24a8012813c5edb20329eb22d5d8e2a0ecf73d21d6b85865dall/MarkupSafe-1.1.1-cp27-cp27mu-manylinux1_x86_64.whl
Collecting pynacl>=1.0.1 (from paramiko->ansible==2.7.10)
  Downloading https://files.pythonhosted.org/packages/de/63/bb36279da38df643c6df3a8a389f29a6ff4a8854468f4c9b9d925b27d57d/PyNaCl-1.4.0-cp27-cp27mu-manylinux1_x86_64.whl (964kB)
    100% |#####| 972kB 2.0MB/s
Collecting bcrypt>=3.1.3 (from paramiko->ansible==2.7.10)
  Downloading https://files.pythonhosted.org/packages/ad/36/9a0227d048e98409f012570f7bef8a8c2373b9c9c5dfbf82963cbae05ede/bcrypt-3.1.7-cp27-cp27mu-manylinux1_x86_64.whl (59kB)
    100% |#####| 61kB 16.7MB/s
Requirement already satisfied: six in /usr/lib/python2.7/dist-packages (from pynacl>=1.0.1->paramiko->ansible==2.7.10)
Collecting cffi>=1.4.1 (from pynacl>=1.0.1->paramiko->ansible==2.7.10)

```

Figure 5: Installation de ansible dans notre environnement virtuel

Nous allons utiliser cette commande **sudo ansible --version** pour vérifier la version de ansible.

### Vérification de notre installation

```

user-ansible@nodemanager: ~
(ansible2.7.10) user-ansible@nodemanager:~$ ls ansible2.7.10/bin/ansible* -l
-rwxr-xr-x 1 user-ansible user-ansible 5870 janv. 23 17:17 ansible2.7.10/bin/ansible
-rwxr-xr-x 1 user-ansible user-ansible 5870 janv. 23 17:17 ansible2.7.10/bin/ansible-config
-rwxr-xr-x 1 user-ansible user-ansible 11709 janv. 23 17:17 ansible2.7.10/bin/ansible-connection
-rwxr-xr-x 1 user-ansible user-ansible 5870 janv. 23 17:17 ansible2.7.10/bin/ansible-console
-rwxr-xr-x 1 user-ansible user-ansible 5870 janv. 23 17:17 ansible2.7.10/bin/ansible-doc
-rwxr-xr-x 1 user-ansible user-ansible 5870 janv. 23 17:17 ansible2.7.10/bin/ansible-galaxy
-rwxr-xr-x 1 user-ansible user-ansible 5870 janv. 23 17:17 ansible2.7.10/bin/ansible-inventory
-rwxr-xr-x 1 user-ansible user-ansible 5870 janv. 23 17:17 ansible2.7.10/bin/ansible-playbook
-rwxr-xr-x 1 user-ansible user-ansible 5870 janv. 23 17:17 ansible2.7.10/bin/ansible-pull
-rwxr-xr-x 1 user-ansible user-ansible 5870 janv. 23 17:17 ansible2.7.10/bin/ansible-vault
(ansible2.7.10) user-ansible@nodemanager:~$

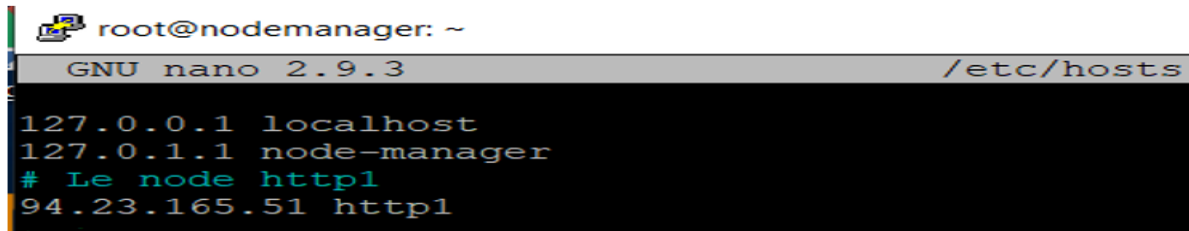
```

Figure 6: Vérification de notre installation

### Se connecter en root sur le node manager

Pour nous connecter en root au node manager nous allons utiliser cette commande **sudo su** et nous entrons notre mot de passe puis nous aurons l'interface suivante **root@nodemanager: ~\$**

### Enregistrement du node1 sur le node manager dans le fichier hosts

A terminal window showing the nano text editor editing the /etc/hosts file. The prompt is root@nodemanager: ~. The file content is: 127.0.0.1 localhost, 127.0.1.1 node-manager, # Le node http1, 94.23.165.51 http1.

```
root@nodemanager: ~  
GNU nano 2.9.3 /etc/hosts  
127.0.0.1 localhost  
127.0.1.1 node-manager  
# Le node http1  
94.23.165.51 http1
```

Figure 7: Enregistrement du node1 dans le fichier hosts

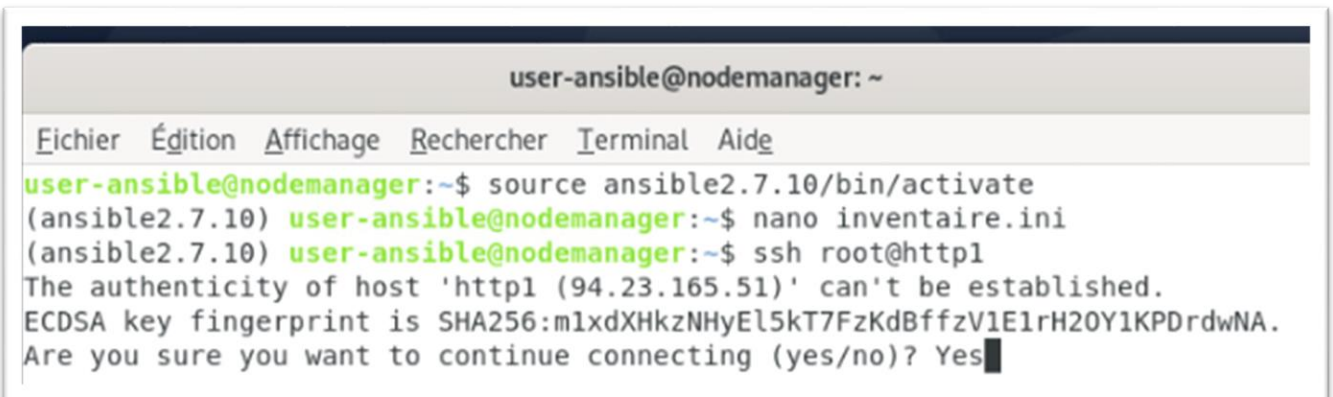
### Création du fichier inventaire ansible.

Le fichier inventaire ansible, qui possédant la liste de node est indispensable pour le fonctionnement de Ansible.

Ainsi pour enregistrer notre node dans ce fichier, il important pour nous de nous placer dans notre environnement virtuel. Donc nous ferons appel à cette commande `su - user-ansible` pour nous placer dans l'environnement virtuel. Et en l'activant nous allons éditer via le fichier inventaire.ini afin de mettre le nom de notre node http1.

Nous allons utiliser cette commande `ssh root@http1` pour vérifier si la communication passe sur les différents nodes. Nous allons nous connecter en ssh pour enregistrer la clé ou la fingerprint sur le node manager.

Sur le node1 ou http1

A terminal window showing the user-ansible@nodemanager: ~ prompt. The user runs 'source ansible2.7.10/bin/activate', then 'nano inventaire.ini', and finally 'ssh root@http1'. The ssh command prompts for the authenticity of host 'http1 (94.23.165.51)' and the user responds 'Yes'.

```
user-ansible@nodemanager: ~  
Fichier Édition Affichage Recherche Terminal Aide  
user-ansible@nodemanager:~$ source ansible2.7.10/bin/activate  
(ansible2.7.10) user-ansible@nodemanager:~$ nano inventaire.ini  
(ansible2.7.10) user-ansible@nodemanager:~$ ssh root@http1  
The authenticity of host 'http1 (94.23.165.51)' can't be established.  
ECDSA key fingerprint is SHA256:m1xdXHkzNHyl5kT7FzKdBffzV1E1rH20Y1KPDrdwNA.  
Are you sure you want to continue connecting (yes/no)? Yes
```

Figure 8: Vérification de la communication

Cette figure montre que la connexion est établie

```
Eichier  Édition  Affichage  Recherche  Terminal  Aide
user-ansible@nodemanager:~$ source ansible2.7.10/bin/activate
(ansible2.7.10) user-ansible@nodemanager:~$ nano inventaire.ini
(ansible2.7.10) user-ansible@nodemanager:~$ ssh root@http1
The authenticity of host 'http1 (94.23.165.51)' can't be established.
ECDSA key fingerprint is SHA256:m1xdXHkzNHyeEl5kT7FzKdBffzV1E1rH20Y1KPDrdwNA.
Are you sure you want to continue connecting (yes/no)? Yes
Warning: Permanently added 'http1,94.23.165.51' (ECDSA) to the list of known hos
ts.
root@http1's password:
Linux http1 4.19.0-18-amd64 #1 SMP Debian 4.19.208-1 (2021-09-29) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jan 25 00:34:48 2022 from 154.66.172.97
root@http1:~#
```

Figure 9 : L'établissement de la connexion au niveau du http1

Nous allons lancer un ping sur le node1 avec ansible dans notre environnement virtuel tout en faisant appel à la commande ansible en mode ad-hoc pour nous faciliter les tâches.

```
(ansible2.7.10) user-ansible@nodemanager:~$ ansible -i inventaire.ini -m ping ht
tp1 --user root --ask-pass
/home/user-ansible/ansible2.7.10/local/lib/python2.7/site-packages/ansible/parsi
ng/vault/__init__.py:41: CryptographyDeprecationWarning: Python 2 is no longer s
upported by the Python core team. Support for it is now deprecated in cryptograp
hy, and will be removed in the next release.
  from cryptography.exceptions import InvalidSignature
SSH password:
http1 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
(ansible2.7.10) user-ansible@nodemanager:~$
```

Figure 10: Lancement de la commande ping en mode ah-doc

Vérification de l'installation de python sur le http1.

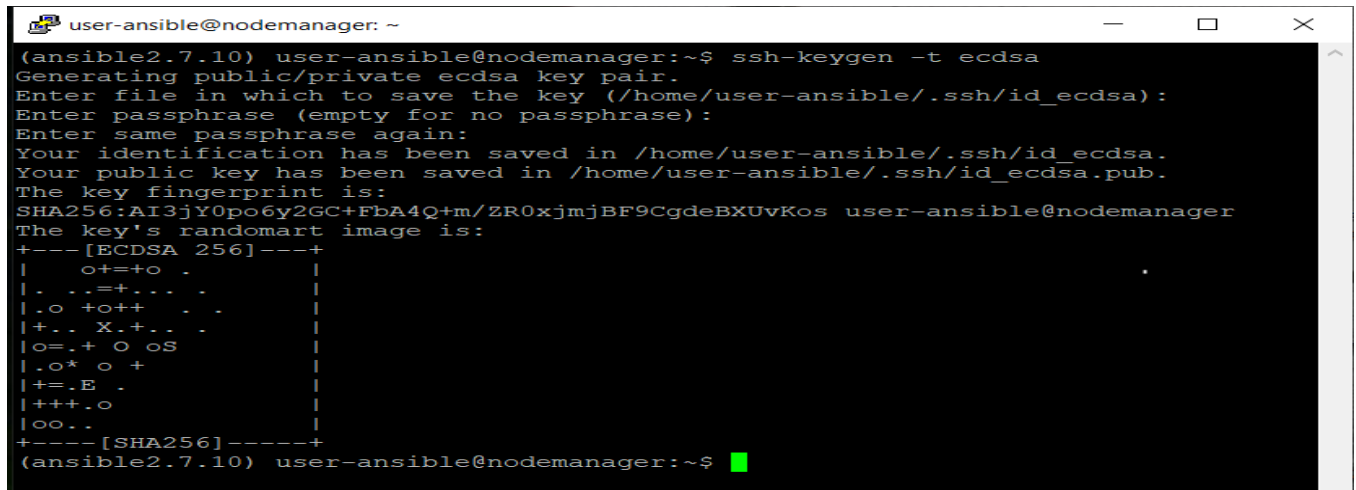
```
(ansible2.7.10) user-ansible@nodemanager:~$ ansible localhost -i inventaire.ini
-m debug -a "msg={{ 'passforce' | password_hash('sha512', 'sceretsalt') }}"
/home/user-ansible/ansible2.7.10/local/lib/python2.7/site-packages/ansible/parsi
ng/vault/__init__.py:41: CryptographyDeprecationWarning: Python 2 is no longer s
upported by the Python core team. Support for it is now deprecated in cryptograp
hy, and will be removed in the next release.
  from cryptography.exceptions import InvalidSignature
localhost | SUCCESS => {
  "msg": "$6$sceretsalt$tBcfGEgifQpQZsg5CIGZ79XC55h5vHy7UWrys7cAF37KNCQQbm7iCv
y58M1LQLaS2fLF6ZjqDVHhVrkMdRi0f0"
}
(ansible2.7.10) user-ansible@nodemanager:~$
```

Figure 11 : Vérification de l'installation de python sur notre node

La création d'un utilisateur user-ansible sur notre node http1. Après avoir créé notre *user-ansible* nous allons lui accorder le droit sudo.

### La création des clés ssh

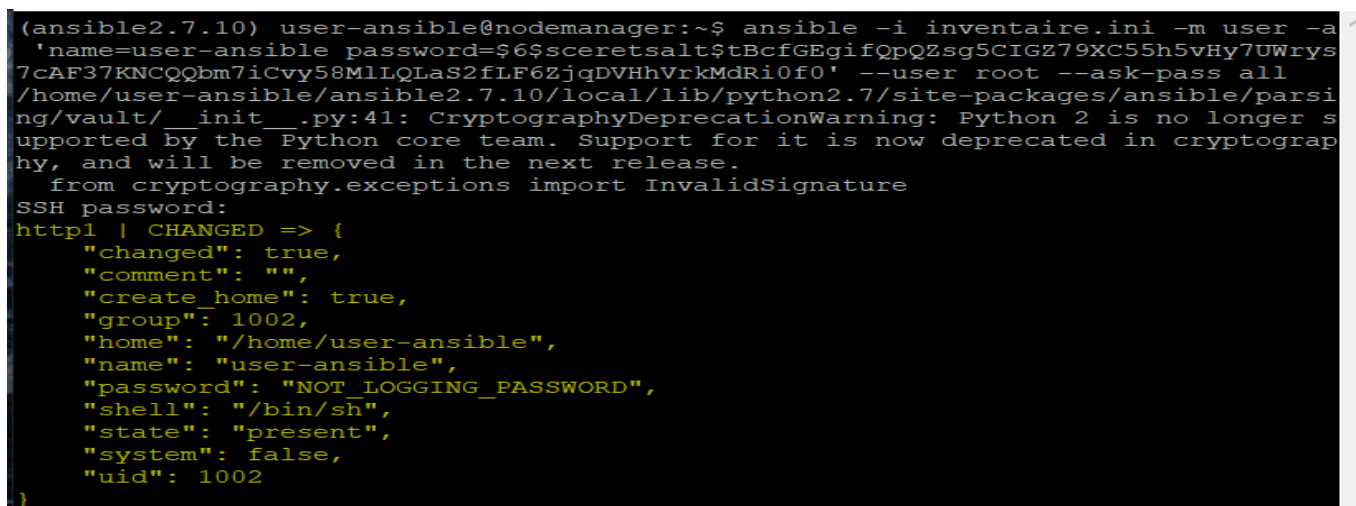
La connexion en ssh préconise l'utilisation d'une paire de clés au lieu d'user d'un mot de passe. Cette manière de procéder assure un niveau de d'authentification plus efficace. Nous allons générer une paire de clé de type ecdsa pour notre utilisateur en utilisant la commande *su -.* *user-ansible* pour nous connecter.



```
user-ansible@nodemanager: ~  
(ansible2.7.10) user-ansible@nodemanager:~$ ssh-keygen -t ecdsa  
Generating public/private ecdsa key pair.  
Enter file in which to save the key (/home/user-ansible/.ssh/id_ecdsa):  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/user-ansible/.ssh/id_ecdsa.  
Your public key has been saved in /home/user-ansible/.ssh/id_ecdsa.pub.  
The key fingerprint is:  
SHA256:AI3jY0po6y2GC+FbA4Q+m/ZR0xjmjBF9CgdeBXUvKos user-ansible@nodemanager  
The key's randomart image is:  
+---[ECDSA 256]---+  
|  o+=+o . |  
|. .+=+... |  
|.o +o++ . |  
|+.. X.+.. |  
|o=+. O oS |  
|.o* o + |  
|+=.E . |  
|+++.o |  
|oo.. |  
+---[SHA256]-----+  
(ansible2.7.10) user-ansible@nodemanager:~$
```

Figure 12 : Création des clés ssh

Cette figure ci-dessous montre l'ajout de la clé publique de l'utilisateur user-ansible sur notre node http1.



```
(ansible2.7.10) user-ansible@nodemanager:~$ ansible -i inventaire.ini -m user -a  
'name=user-ansible password=$6$sceretsalt$tBcfGEgifQpQZsg5CIGZ79XC55h5vHy7UWrys  
7cAF37KNCQqbm7iCvy58MLQLaS2fLF6ZjqDVHhVrkMdRi0f0' --user root --ask-pass all  
/home/user-ansible/ansible2.7.10/local/lib/python2.7/site-packages/ansible/parsi  
ng/vault/__init__.py:41: CryptographyDeprecationWarning: Python 2 is no longer s  
upported by the Python core team. Support for it is now deprecated in cryptograp  
hy, and will be removed in the next release.  
from cryptography.exceptions import InvalidSignature  
SSH password:  
http1 | CHANGED => {  
  "changed": true,  
  "comment": "",  
  "create_home": true,  
  "group": 1002,  
  "home": "/home/user-ansible",  
  "name": "user-ansible",  
  "password": "NOT_LOGGING_PASSWORD",  
  "shell": "/bin/sh",  
  "state": "present",  
  "system": false,  
  "uid": 1002  
}
```

Figure 13 : L'ajout de la clé publique de l'utilisateur user-ansible

## 5.3 Organisation des opérations d'automatisation avec les rôles ansible



Dans cette partie nous allons créer des rôles pour les différentes installations et configurations.

Nous allons créer une arborescence de fichiers de configurations qui nous permettra de dérouler les phases de déploiement de MediaWiki et d'écrire les scripts de Ansible dans ces fichiers de configuration. Par défaut ces rôles sont créés dans le répertoire courant. Pour cela nous allons débiter par créer le répertoire "roles" dans notre espace de travail qui contiendra tous les rôles que nous allons créer. Nous avons créé les rôles à l'aide de cette commande **mkdir roles** et nous utilisons cette commande **cd roles** pour nous placer dans le répertoire.

Pour créer de manière automatique le rôle ansible-galaxy, nous procéderons comme suit en utilisant cette commande : **ansible-galaxy init apache**.

Nous allons utiliser la commande **tree apache** pour afficher l'arborescence des fichiers qui donne le résultat suivant.

```
apache/
├── defaults
│   └── main.yml
├── files
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── README.md
├── tasks
│   └── main.yml
├── templates
├── tests
│   ├── inventory
│   └── test.yml
├── vars
│   └── main.yml
```

Figure 14 : Afficher l'arborescence des fichiers

Un rôle pour installer Apache

Pour installer Apache et PHP nous devons nous placer dans le répertoire "roles" tout en réalisant une mise à jour pour se rassurer que les modifications aient été prises en compte. Cependant nous ne garderons que les répertoires nécessaires tels que **handlers**, **meta** et **tasks**.

Nous utiliserons cette commande pour supprimer les répertoires inutiles **cd apache &&rm-r files meta templates vars defaults tests README.md** qui nous donnera ce résultat.

```
roles/apache/
├── handlers
│   └── main.yml
├── tasks
│   └── main.yml
```

Figure 15 : Création d'un rôle pour installation d'Apache

Un rôle pour installer Maria DB

Pour créer un rôle mariadb, il nous faut installer des paquets sur Linux (Debian) et démarrer un service. Pour cela voici les commandes à utiliser **`mkdir -p mariadb/tasks/`** et **`touch mariadb/tasks/main.yml`**. Nous rappelons que le fichier **`main.yml`** comporte les tâches à exécuter pour l'installation de MariaDB. Cette commande donne l'arborescence suivant :

```
|— mariadb
|   |— tasks
|       |— main.yml
```

Figure 16 : Création d'un rôle pour installation de MariaDB

Un rôle pour configurer Apache pour MediaWiki

Nous créerons un répertoire MediaWiki avec cette commande **`mkdir mediawiki`** qui nous donne le résultat suivant.

```
../roles/
|— apache
|   |— handlers
|       |— main.yml
|   |— meta
|       |— main.yml
|   |— tasks
|       |— main.yml
|— mariadb
|   |— tasks
|       |— main.yml
|— mediawiki
```

Figure 17 : Création d'un rôle pour configurer Apache pour MediaWiki

Il nous faut maintenant créer le rôle commun qui contiendra des variables partagées entre les rôles **`confdb`** et **`confapache`**. Cette commande **`mkdir -p mediawiki/commun/defaults/`** et **`touch mediawiki/commun/defaults/main.yml`** permet de créer cette arborescence.

```
|— mediawiki
|   |— commun
|       |— defaults
|           |— main.yml
```

Figure 18 : Création d'un rôle commun contenant **`confdb`** et **`confapache`**

Nous allons créer également le rôle **`confdb`** pour enfin créer une base de données et de lui attribuer les droits. Cette commande **`mkdir -p mediawiki/confdb/meta mediawiki/confdb/tasks`** nous permet de créer ce rôle avec notamment les répertoires **`meta`** et **`tasks`**. Nous créerons les



fichiers **main.yml** qui contiendront les différentes actions. Les commandes **touch mediawiki/confdb/tasks/main.yml** et **touch mediawiki/confdb/meta/main.yml** permettent d'avoir cette arborescence

```
confdb
├── meta
│   └── main.yml
└── tasks
    └── main.yml
```

Figure 19 : Création d'un rôle confdb pour la base de données

Nous créerons un rôle confapache afin de partager les variables globales avec le rôle confdb. Alors il est nécessaire d'ajouter une dépendance avec le rôle commun. Il nous incombe de créer le répertoire d'installation de MediaWiki, de télécharger les fichiers de MediaWiki sur le site officiel, de lancer le script d'installation et mettre à jour les bases de données. Nous avons besoins des répertoires **meta** et **tasks** avec cette commande **~/roles/mediawiki\$ mkdir -p confapache/meta confapache/tasks** et nous avons les fichiers **main.yml** qui comporteront les actions. Ils seront créés à l'aide de cette commande **touch confapache/tasks/main.yml confapache/meta/main.yml** qui donne les arborescences suivantes.

```
├── confapache
│   ├── meta
│   │   └── main.yml
│   └── tasks
│       └── main.yml
```

Figure 20 : Création d'un rôle confapache pour le partage des variables globales

Nous éditerons maintenant le fichier inventaire.ini en utilisant la commande suivante : **nano inventaire.ini** pour éditer les deux groupes **db** et **apache** au niveau de notre node.

```
GNU nano 3.2                                inventaire.ini
[apache]
[db]
http1
```

#### 5.4 Observer l'exécution des opérations et coordonner les actions

Nous allons écrire des scripts d'automatisation en achevant les fichiers **main.yml** contenus dans chaque rôle. Les rôles nous garantiront l'exécution des tâches et de terminer les actions. Dans notre arborescence des rôles précédent nous avons eu à créer 5 rôles qui correspondent à 9 fichiers YAML représentant les rôles tels qu'apache, mariadb, commun, confapache et confdb.

Il est nécessaire pour nous de créer les fichiers YAML pour les 5 rôles.

Pour le rôle **apache**

**Tasks/main.yml** est ce fichier qui contient les actions pour l'installation d'Apache, un fichier de configuration est sollicité pour installer PHP et une notification pour redémarrer Apache.

Pour redémarrer le service Apache il nous faut un fichier **handler/main.yml** possédant les actions.

Pour le rôle **mariadb**

Pour installer MariaDB il faut les actions contenues dans le fichier **defauts/main.yml**.

Pour le rôle **commun** de MediaWiki

Nous allons utiliser le fichier **defauts/main.yml** contenant les variables d'installation qui seront exploitées les rôles ci-dessous.

Le rôle confapache de MediaWiki

Le fichier **meta/main.yml** possède la dépendance avec le rôle commun.

Le fichier **tasks/main.yml** comporte les actions pour la configuration de MariaDB pour MediaWiki.

- Construction de nos fichiers YAML

Le fichier de configuration YAML est ce fichier que nous retrouvons dans les rôles qui peut être une liste de variables ou une liste de tâches.

Pour construire un fichier de configuration YAML nous devons respecter la syntaxe suivante :

Il débute toujours par 3 tirets (---)

Nous avons également les différentes tâches qui se succèdent tout en commençant par 1 tiret (-) et le nom de la tâche. Chacune de ces tâches utilise un module avec ses options qui sont décalées à la suivante par 2 espaces.

- Création d'un fichier YAML pour le rôle Apache.

Pour l'installation d'Apache et de PHP nous allons utiliser la commande **nano roles/apache/tasks/main.yml**.

***A compléter par la capture***

Nous utiliserons également cette commande **nano roles/apache/tasks/php-install.yml** afin d'ajouter le fichier de configuration pour installation spécifique de PHP pour séparer l'installation de PHP de l'installation d'Apache. ***A compléter par la capture***

Pour démarrer notre service, il nous est nécessaire de créer un script handler afin de relancer le service Apache. Pour cela nous allons utiliser la commande **cat roles/apache/handlers/main.yml**

## *A compléter par la capture*

- Création d'un fichier YAML pour le rôle mariadb

### **5.5 Contrôle de l'exécution des opérations et coordonner plusieurs actions en paramétrant les rôles.**

D'abord, nous allons utiliser les playbooks Ansible pour assembler toutes ces opérations et automatiser le déploiement de MediaWiki. Ensuite, nous jouerons ces playbooks avec la commande `ansible-playbook` et enfin de déployer automatiquement MediaWiki.

- Vérification du niveau de sécurité sur le node

Il est nécessaire de vérifier que le firewalling et le AppArmor ne sont pas activés sur le node avant de lancer les configurations. Car ces mesures de sécurité peuvent bloquer l'installation de MediaWiki.

Nous avons ici quelques commandes à utiliser pour empêcher ces blocages :

**`systemctl stop apparmor.service`** qui nous permet de désactiver temporairement AppArmor.

**`systemctl stop firewall.service`** qui nous permet de désactiver temporairement le firewalling.

- Activation de la résolution de noms sur le node

Etant donné qu'il n'y a pas de gestionnaire de noms DNS, il nous est nécessaire d'enregistrer le nom du node et les adresses IP afin qu'il puisse résoudre le nom de la machine utilisé dans les configurations. Dans le fichier **`/etc/hosts`** nous allons ajouter l'adresse **IP 94.23.165.51** et le nom **http1** sur notre node.

- Création des playbooks pour exécuter les rôles

Le playbook, écrit en Yaml, est un fichier de configuration qui possède une série de jeux d'instructions. Il aide à décrire une méthode de configuration ou de déploiement tout en architecturant les actions nécessaires. Pour cela, nous allons créer un playbook pour les différentes installations.

Nous nous connectons sur notre nodemanager en nous plaçons dans notre environnement virtuel afin de créer le playbook pour installer Apache. Nous allons utiliser cette commande ***nano install-apache.yml*** pour éditer le contenu du fichier. Nous allons lancer le playbook pour installer Apache sur http1 à l'aide de cette commande ***ansible-playbook -i inventaire.ini --user user-ansible --become --ask-become-pass install-apache.yml***. Alors nous constaterons que le callback affiche la liste des tâches pour chaque action définie dans le rôle apache avec leur changement. Nous pouvons nous connecter sur le node http1 pour vérifier qu'Apache et PHP sont bien installés en utilisant cette commande ***ssh user-ansible@http1 et php --version***. ***A compléter par la capture***

- Création d'un playbook pour installer MariaDB

Nous allons utiliser cette commande ***nano install-mariadb.yml*** qui nous permettra d'installer MariaDB qui consistera à lancer le rôle de mariadb. Et voici le contenu de ce playbook pour l'installation de MariaDB.

Maintenant pour exécuter le playbook avec les options de connections ***install-mariadb.yml*** nous utiliserons la commande ***ansible-playbook***. Pour voir le résultat de notre installation nous saisissons simplement cette commande ***sudo systemctl status mariadb*** tout en nous connectant au node http1. ***A compléter par la capture***

- Enfin nous créerons un playbook pour la configuration de MediaWiki

Cette configuration de MediaWiki a pour but de démarrer les rôles ***confapache*** et ***confdb*** tout en créant un playbook ***install-mediawiki.yml***. Notre playbook contient deux tâches et une pour chaque rôle comme le montre la capture d'écran suivante. ***A compléter par la capture***

Nous pouvons lancer alors le playbook de configuration de MediaWiki en utilisant cette commande ***ansible-playbook -i inventaire.ini --user user-ansible --become --ask-become-pass --ask-vault-pass install-mediawiki.yml***. Voilà le résultat. ***A compléter par la capture***

## Installation de MediaWiki

L'installation de MediaWiki nécessite plusieurs phases.

## CONCLUSION GENERALE

Ce présent travail a traité sur l'automatisation du déploiement des solutions sur le cloud en prenant la solution mediawiki comme exemple. Dans ce projet, nous étions amenés à optimiser le problème de déploiement de nos logiciels et d'autres services à travers l'automatisation grâce à l'outil Ansible. Il s'agissait de mettre en place une solution pour palier à ses problèmes.

Les sources Ansible sont disponibles sur GitHub : <https://github.com/ansible/ansible>

Le site officiel renferme une documentation claire et détaillée : <https://docs.ansible.com/ansible/latest/index.html>