



المدرسة الحسنية للأشغال العمومية
ÉCOLE HASSANIA DES TRAVAUX PUBLICS

RAPPORT DE STAGE

D'initiation professionnelle

*Automatisation de la gestion
d'infrastructure*

Rapport de Stage de D'initiation Professionnelle

[Pour l'Obtention du Titre]

Ingénieur d'Etat en informatique

Sujet : Automatisation de la gestion d'Infrastructure
D'entreprise

Année universitaire 2022/2023

Réalisé par : Ayoub SAMI

Encadré par : Ismail AIT ZAID

Remerciements :

Je tiens à remercier dans un premier temps, toute l'équipe pédagogique responsable de l'Ecole Hassania des Travaux Publics [EHTP], pour m'avoir apporté cette opportunité mémorable.

Je tiens à remercier tout particulièrement et à témoigner toute ma reconnaissance à **M. Ismail AIT ZAID** Ingénieur en Cloud & DevOps, pour l'expérience enrichissante et pleine d'intérêt qu'il m'a offert durant les sept semaines du stage.

Enfin, je remercie toutes personnes qui m'ont encouragé et qui ont participé pour que mon stage se déroule dans de bonnes conditions.

Avant-propos :

En tant qu'élève ingénieur à l'Ecole Hassania des Travaux Publics et vu l'importance de la pratique et des opérations techniques dans le domaine de l'ingénierie, j'ai eu l'occasion d'effectuer un stage d'initiation professionnelle en Administration des systèmes et DevOps. Ce stage qui s'étend sur sept semaines au sein de l'entreprise de services et conseil informatique **IT24** à Salé, ces sept semaines qui s'étend du 07 Juillet au 25 Août.

Ce stage d'initiation est une occasion pour l'élève ingénieur de découvrir les activités et le fonctionnement d'une entreprise et de compléter ses connaissances théoriques acquises au cours de sa formation par des travaux concrets et pratiques tout en lui permettant de vivre une expérience sociotechnique enrichissante.

Résumé :

Afin d'automatiser la gestion d'infrastructure (les ordinateurs, bases données et serveurs) de l'entreprise par l'administrateur, l'entreprise IT24 a défini ses besoins en termes d'exigences à satisfaire dans la future solution intégrée dans leur infrastructure informatique. Il s'agit de la mise en place d'une solution qui va permettre un accès externe au réseau local d'entreprise de manière sécurisée via une connexion SSH de type OpenSSH qui s'appuiera sur une machine Master et l'utilisation de la technologie ANSIBLE.

Nous verrons comment configurer l'ansible dans la machine Master, ainsi que la création des clés SSH-KEYS afin de permettre la connexion SSH à l'infrastructure de l'entreprise. Cet accès ne sera garanti qu'à la machine Master qui dispose d'une gestion complète sur toute l'infrastructure. La machine Master est une machine virtuelle créée par la distribution RHEL 7.9. En effet, l'administrateur n'a besoin que de la machine Master pour gérer toutes les ressources de l'entreprise.

La réalisation de cette étude nécessite d'une infrastructure virtuelle avec l'outil de virtualisation open source VirtualBox renfermant le serveur de l'entreprise et les machines contenant les solutions pour réaliser ce projet.

Mots clé : SSH, OpenSSH, ANSIBLE, machine Master, RHEL et SSH-KEYS.

Abstract:

In order to automate the management of the infrastructure (computers, databases and servers) of the company by the administrator, the company IT24 has defined its needs in terms of requirements to be met in the future solution integrated into their IT infrastructure. This is the implementation of a solution that will allow external access to the corporate local network in a secure way via an OpenSSH SSH connection that will rely on a Master machine and the use of ANSIBLE technology.

We will see how to configure the ansible in the Master machine, as well as the creation of SSH-KEY keys to allow the SSH connection to the enterprise infrastructure. This access will only be guaranteed to the Master machine which has a complete management over the entire infrastructure. The Master machine is a virtual machine created by the RHEL 7.9 distribution. Indeed, the administrator only needs the Master machine to manage all the resources of the company.

The realization of this study requires a virtual infrastructure with the open-source virtualization tool VirtualBox containing the company server and the machines containing the solutions to realize this project.

Keywords: SSH, OpenSSH, ANSIBLE, Master machine, RHEL and SSH-KEY.

Liste des acronymes :

Abréviation	Désignation
BSD	Berkeley Software Distribution
DNS	Domaine Name System
HTTP	HyperText Transfert Protocol
MPLS	Multi Protocol Label Switching
NOC	Network Operations Center
RHEL	Red Hat Enterprise Linux
SOC	System On a Chip
SSH	Secure SHell
SSHD	OpenSSH server
ToIP	Telephony over IP
VMs	Virtual Machines

Table des matières

CHAPITRE 1 : CONTEXT GENERALE DE PROJET	11
I. Introduction	11
II. Cadre général de travail	11
1. Présentation de l'organisme d'accueil	11
2. Présentation de projet	13
III. Conclusion.....	13
CHAPITRE 2 : SPECIFICATION DE BESOIN	15
I. Introduction	15
II. Etude de l'existant.....	15
1. Situation actuelle	15
2. Insuffisances.....	16
III. Analyse de besoin	16
1. Besoins fonctionnels	16
2. Besoins non fonctionnels	17
IV. Conclusion.....	17
CHAPITRE 3 : CONCEPTION	19
I. Introduction	19
II. Architecture de la solution proposée	19
III. Conclusion.....	20
CHAPITRE 4 : REALISATION	22
I. Introduction	22
II. Infrastructure virtuelle	22
1. La machine Master : nœud maitre	22
2. Machines virtuelles clientes	22
III. Configuration de l'Ansible dans la machine Master	23
1. Création de l'utilisateur ansible.....	24
2. La configuration de DNS	24
3. L'installation de l'ansible	24
4. La configuration de l'ansible.....	25
IV. Configuration des nœuds gérées	26
1. Création de l'utilisateur ansible.....	26
V. Accès Sécurisé Via SSH	26

1.	L'installation d'OpenSSH.....	27
2.	Configuration d'OpenSSH.....	28
VI.	Création des clés SSH-KEY.....	29
1.	Génération de la clé SSH.....	29
2.	Copie la clé vers les machines gérées.....	29
3.	Tester la connectivité.....	30
VII.	Conclusion.....	31
CHAPITRE 5 : MISE EN ŒUVRE.....		33
I.	Hébergement du site web d'entreprise sur le serveur web.....	33
1.	Introduction.....	33
2.	L'hébergement du site web d'entreprise.....	33
II.	Configuration de Load Balancer.....	37
1.	Introduction.....	37
2.	La configuration.....	37
3.	Conclusion.....	43
Conclusion générale.....		44
Bibliographie.....		45
Webographie.....		45
Annexes.....		46

Liste des tableaux :

Tableau 1 : Présentation IT24	11
Tableau 2 : Infrastructure d'entreprise	20
Tableau 3 : Définitions	50

Liste des figures :

Figure 1: Logo IT24	12
Figure 2 : L'architecture de la solution proposée	19
Figure 3 : Création d'utilisateur ansible	24
Figure 4 : Ajout d'user ansible au group Wheel	24
Figure 5 : La configurations du fichier /etc/hosts.....	24
Figure 6 : Installation de l'ansible	25
Figure 7 : le dossier tasks.....	25
Figure 8 : Configuration de fichier ansible.cfg.....	25
Figure 9 : Fichier d'inventaire.....	26
Figure 10 : Création d'utilisateur ansible	26
Figure 11 : Installation du package openssh-server	27
Figure 12 : installation de package openssh-clients.....	27
Figure 13 : Activer le service sshd sur le nœud maitre.....	28
Figure 14 : Vérifier l'activation du service sshd sur le Master.....	28
Figure 15 : Activer le service sshd sur le webserver.....	28
Figure 16 : Vérifier l'activation du service sshd sur le woker 1.....	28
Figure 17 : Création de clé ssh.....	29
Figure 18 : Copie la clé id_rsa.pub vers webserver	29
Figure 19 : Copie la clé id_rsa.pub vers worker 2.....	30
Figure 20 : Copie la clé id_rsa.pub vers worker 3.....	30
Figure 21 : La connectivité réussite	30
Figure 22 : Structure de site	33
Figure 23 : Page d'accueil.....	33
Figure 24 : Site.yml playbook	35
Figure 25 : Exécution de playbook site.yml	35
Figure 26 : Interface de logiciel FileZilla Client.....	36
Figure 27 : Copier les fichiers du site vers le serveur	37
Figure 28 : Le load balancing machines.....	37
Figure 29 : site héberger sur le serveur principal	38
Figure 30 : site héberger sur le serveur web 2.....	38
Figure 31 : site héberger sur le serveur web 3.....	38
Figure 32 : Fichier inventory	39
Figure 33 : Création des rôles.....	39

Figure 34 : webserver/tasks/main.yml	39
Figure 35: webserver/vars/main.yml	40
Figure 36 : webserver/handlers/main.yml.....	40
Figure 37 : loadbalancer/tasks/main.yml	40
Figure 38 : loadbalancer/vars/ main.yml	41
Figure 39 : loadbalancer/handlers/main.yml	41
Figure 40 : task.yml.....	41
Figure 41 : Exécution du task.yml	42
Figure 42 : Fonctionnement de loadbalancer.....	43
Figure 43 : Informations sur la machine master	46
Figure 44 : Le choix de langage de support.....	47
Figure 45: Attribution de l'adresse IP statique.....	47
Figure 46 : Sélection de Serveur avec GUI	47
Figure 47 : Activer la connectivité et donner le nom de la machine	47
Figure 48 : Création d'utilisateur et de mot de passe de super user root	48
Figure 49 : Configuration de serveur terminée.....	48
Figure 50 : Configuration du machine webserver – worker 1	48
Figure 51 : Le choix de langage de support.....	49
Figure 52 : Sélection d'installation minimale.....	49
Figure 53 : Attribution de l'adresse IP statique et activation de la connectivité.....	49
Figure 54 : Création d'utilisateur et de mot de passe de super user root	49

Introduction générale

Aujourd'hui, les entreprises informatiques telle qu'IT24 font face à un ensemble de défis complexes liés à la gestion de leur infrastructure et à l'optimisation de leurs ressources informatiques. L'ancien modèle où chaque application était associée à un serveur unique a créé des problèmes de gestion considérables. Les équipes informatiques consacraient plus de temps à résoudre les problèmes de serveurs et à gérer les ressources matérielles et logicielles qu'à innover et à répondre aux besoins en constante évolution de l'entreprise.

Dans ce contexte, le concept de l'automatisation est devenu la réponse à ces défis de sous-utilisation et de complexité de gestion. L'automatisation permet de rationaliser la gestion de l'infrastructure, d'optimiser l'utilisation des ressources et de libérer le potentiel d'innovation des équipes informatiques.

Notre projet, mené au sein de l'entreprise IT24, vise à mettre en place une solution d'automatisation complète pour répondre aux besoins spécifiques de l'entreprise. L'objectif central est de simplifier et d'optimiser la gestion de l'infrastructure informatique, de manière à garantir une utilisation efficiente des ressources et à renforcer la sécurité des opérations.

Pour aborder efficacement ce projet d'automatisation, une compréhension approfondie des besoins et des exigences est cruciale. C'est dans cette optique que nous avons structuré notre rapport de projet en plusieurs chapitres.

Dans le premier chapitre, nous présenterons une vue d'ensemble du projet, de l'entreprise IT24 et de son environnement. Le deuxième chapitre se concentrera sur la spécification des besoins, qui découlera d'une étude minutieuse de l'existant. Dans le troisième chapitre, nous entamant la conception nécessaire à la réalisation du projet.

Le quatrième chapitre abordera la conception de l'architecture de notre système d'automatisation, en tenant compte les besoins spécifiques d'IT24. Ensuite, dans le cinquième chapitre, nous détaillerons les étapes de mise en œuvre du projet.

Enfin, dans la conclusion générale, nous récapitulerons les accomplissements de ce projet et discuterons des perspectives futures pour l'automatisation au sein d'IT24.

CHAPTER 1

CHAPITRE 1 : CONTEXT GENEARELE DE PROJET

I. Introduction

Il est primordial pour tout projet, de le situer dans son environnement organisationnel et contextuel pour bien entamer les différentes phases de sa réalisation.

Ainsi, j'aborderai au cours de ce chapitre le contexte général du projet en commençant par une présentation de cabinet de services et conseil informatiques **IT24**, une définition de la problématique ainsi que les objectifs et les motivations de ce travail, ensuite une proposition d'une solution qui répond aux besoins exprimés, en décrivant la méthodologie adoptée pour mettre au point cette solution.

II. Cadre général de travail

1. Présentation de l'organisme d'accueil

1.1. IT24.MA

IT24 est une entreprise marocaine spécialisée dans l'intégration de solutions informatiques, de réseaux d'entreprise, de sécurité des réseaux (cybersécurité) et de sécurité électronique (vidéosurveillance, contrôle d'accès, biométrie).

TABLEAU 1 : PRESENTATION IT24

Adresse	N° 44, Rue Jbel Siroua Hay Ennacer KOM
Code postal	11130
Téléphone	+212 651-651591
E-mail	ismail@it24.ma
Forme juridique	Société à Responsabilité Limitée à Associé Unique
Effectif	1
Secteur d'activité	Solution et service informatique et gestion commercial
Ville	Salé
Pays	Maroc



FIGURE 1: LOGO IT24

1.2. Métier d'entreprise IT24.MA

Les services d'IT24 s'articulent autour des trois principaux domaines suivants :

- ❖ RESEAUX INFORMATIQUES : INTRANET, Plateformes Internet, IP/MPLS, Collaboration (Visioconférence, ToIP), WIFI
- ❖ SECURITE DES RESEAUX : Cyber sécurité, Cyber Défense, NOC, SOC.
- ❖ SECURITE ELECTRONIQUE : vidéosurveillance IP, Contrôle d'accès, Biométrie, Bâtiments intelligents, Analytique, ...

Les principaux services offerts par IT24.MA sont les suivants :

- Maintenance informatique
- Installation serveurs
- Support informatique et maintenance régulière
- Sauvegarde et récupération de données
- Interventions sur site
- Développement informatique (Site web, Application web, ...)
- Sécurité des données
- Contrôle d'accès
- etc.

2. Présentation de projet

L'idée du projet est de mettre en place une plateforme permettant à l'administrateur d'infrastructure de l'entreprise cliente de gérer d'automatiser ses tâches, via ANSIBLE.

Notre projet d'automatisation chez IT24 révolutionne la gestion de l'infrastructure informatique. En utilisant des outils d'automatisation comme Ansible, nous simplifions les opérations, renforçons la sécurité et libérons le potentiel d'innovation.

Cette initiative, axée sur la virtualisation et la centralisation des ressources, optimise l'efficacité et réduit les coûts. Nous présentons ici notre démarche, les étapes clés de mise en œuvre et les avantages significatifs de cette automatisation pour IT24.

III. Conclusion

Une fois la présentation du sujet et du cadre du travail a été mis en évidence, nous passons au choix technique les plus appropriées qui permettront de satisfaire toutes les contraintes. C'est l'objet du chapitre suivant.

CHAPTER 2

CHAPITRE 2 : SPECIFICATION DE BESOIN

I. Introduction

Une étape essentielle de tout projet consiste à effectuer une étude préalable. Dans ce chapitre, nous procéderons à une analyse détaillée des besoins qui ont motivé notre projet d'automatisation de la gestion de l'infrastructure pour IT24. Comprendre ces besoins est essentiel pour déterminer les exigences clés du projet et garantir son alignement sur les objectifs de l'entreprise.

Cette étude consiste à examiner le système auquel nous voulons apporter la solution afin de déceler les défaillances et les insuffisances auxquelles il doit remédier. En effet, la réalisation de ce présent travail a nécessité une étude approfondie sur certaines notions qui touchent non seulement au cadre général du projet, mais aussi à sa réalisation.

Pour bien assimiler ces différentes notions, cette section sera consacrée pour présenter brièvement le diagnostic de l'existant, l'étude des besoins fonctionnels et non fonctionnels ainsi qu'une analyse détaillée de la matrice des règles à suivre pour réussir la réalisation de ce projet.

II. Etude de l'existant

Il est essentiel de disposer d'informations précises sur l'infrastructure informatique et les problèmes qui ont une incidence sur le fonctionnement du réseau.

En effet, ces informations affectent une grande partie des décisions à prendre dans le choix de la solution et de son déploiement.

1. Situation actuelle

IT24 a pour vocation d'intégrer de nouvelles technologies et des standards Internet au sein de son système d'information. Dans ce cadre, nous allons décrire la situation actuelle d'IT14 qui souhaite automatiser et centraliser la gestion de son infrastructure.

Dans ce qui suit, on va détailler les composants du système d'information existant de l'entreprise IT24 :

1.1. Serveurs web

L'entreprise IT24 dispose des serveurs web auxquels est hébergé le site web d'entreprise. Ces serveurs sont virtuels avec un système d'exploitation Red Hat Entreprise Linux 7.9. Le serveur web RHEL 7.9 est un élément clé de l'infrastructure informatique d'IT24.

1.2. Serveurs de la production

Le serveur de production, qui est une machine virtuelle exécutant le système d'exploitation RHEL 7.9 (Red Hat Enterprise Linux 7.9), occupe une place centrale dans l'infrastructure d'IT24.

1.3. Bases données

Les bases de données sont essentielles pour stocker, gérer et accéder efficacement aux données de l'entreprise. Une base de données peut contenir une variété d'informations cruciales, telles que des données sur les clients, des transactions financières, des informations sur les produits, des enregistrements d'activités, et bien plus encore.

En outre, l'automatisation de certaines tâches liées à la base de données peut améliorer l'efficacité opérationnelle. Par exemple, l'utilisation de scripts ou d'outils d'automatisation peut simplifier la sauvegarde, la restauration et la maintenance de la base de données.

2. Insuffisances

Parmi les insuffisances, on peut citer des problèmes de sécurité potentiels, tels que des vulnérabilités dans les systèmes, qui pourraient exposer des données sensibles à des menaces. De plus, la gestion des performances peut poser problème, notamment si les systèmes ne répondent pas aux besoins de l'entreprise en matière de vitesse et d'efficacité. Enfin, la gestion globale des systèmes, y compris la maintenance, les sauvegardes et les mises à jour, peut être un défi. Pour aborder ces insuffisances, IT24 peut envisager des solutions d'automatisation et d'amélioration de la sécurité, des mises à jour technologiques, une gestion proactive des performances et une planification stratégique pour l'avenir de ses systèmes informatiques.

III. Analyse de besoin

1. Besoins fonctionnels

- **Simplification de la Gestion**

IT24 fait face à une complexité croissante de la gestion de son infrastructure informatique. Les opérations manuelles répétitives, telles que les déploiements logiciels, la configuration des serveurs et la gestion des correctifs, entraînent une utilisation inefficace des ressources et une charge de travail excessive pour l'équipe informatique.

- **Optimisation des Ressources**

IT24 cherche à optimiser l'utilisation de ses ressources informatiques existantes. La consolidation des serveurs, la rationalisation des processus et la réduction des coûts opérationnels sont des objectifs importants pour l'entreprise.

- **Cohérence des Opérations**

L'entreprise souhaite garantir la cohérence et la fiabilité de ses opérations informatiques. Les variations dans les configurations, les erreurs humaines et les incohérences opérationnelles doivent être minimisées.

2. Besoins non fonctionnels

- **Sécurité et Fiabilité**

La sécurité des données et des opérations est une priorité essentielle pour IT24. La nécessité de renforcer la sécurité des accès, de surveiller en temps réel les menaces potentielles et de garantir la conformité aux normes de sécurité devient de plus en plus critique.

- **Flexibilité et Évolutivité**

IT24 prévoit une croissance continue de ses activités. Par conséquent, elle recherche une solution qui offre une flexibilité et une évolutivité pour s'adapter aux besoins futurs sans perturber les opérations existantes.

IV. Conclusion

En conclusion, IT24 a besoin d'une solution d'automatisation de gestion de l'infrastructure qui simplifie les opérations, renforce la sécurité, optimise les ressources, assure la cohérence opérationnelle, offre une flexibilité et une évolutivité, tout en répondant aux normes de sécurité. Ce chapitre établit une base solide pour la conception et la mise en œuvre de notre projet d'automatisation. Dans les chapitres suivants, nous explorerons en détail la manière dont ces besoins sont adressés grâce à notre solution.

CHAPTER 3

CHAPITRE 3 : CONCEPTION

I. Introduction

Pour réussir la réalisation d'un projet, cela nécessite l'adoption d'une bonne méthode de conception. La conception est le processus qui consiste à concevoir la future solution sur des bases solides afin d'entamer sûrement l'étape de réalisation.

Nous abordons dans ce chapitre une phase très importante dans un projet, la conception, dans laquelle on décrira l'architecture de l'infrastructure informatique de l'entreprise.

II. Architecture de la solution proposée

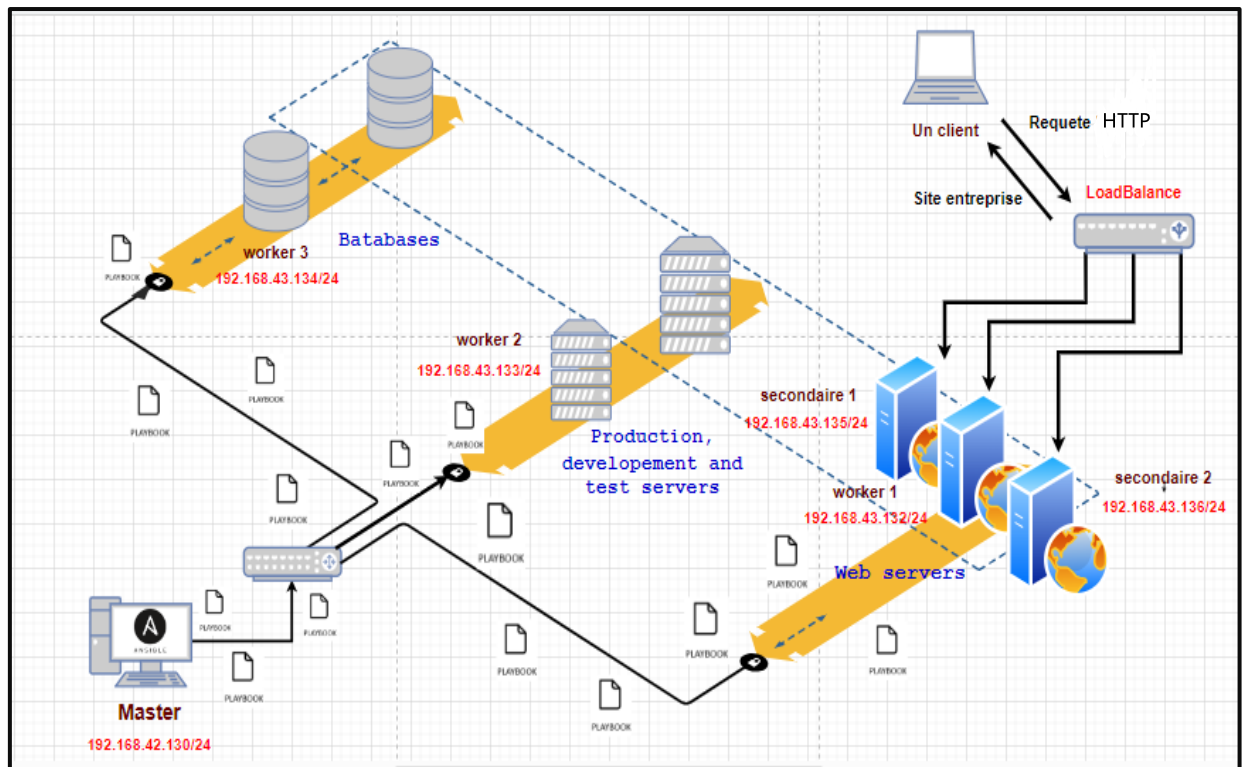


FIGURE 2 : L'ARCHITECTURE DE LA SOLUTION PROPOSEE

Notre solution se base sur l'outil d'automatisation Ansible. En effet, Ansible est un outil d'automatisation de gestion de configuration - d'automatisation informatique - Open Source qui automatise le provisionnement, la gestion des configurations, le déploiement des applications, l'orchestration et bien d'autres processus informatiques.

TABLEAU 2 : INFRASTRUCTURE D'ENTREPRISE

Name	IP Adresse	Hostname	Fonction
Worker 1	192.168.43.132	worker1	Web serveur
Worker 2	192.168.43.133	worker2	Serveur de production
Worker 3	192.168.43.134	worker3	Base données
Web server 2	192.168 .43.135	webserver2	Serveur web secondaire
Web server 3	192.168 .43.136	webserver3	Serveur web secondaire

Grâce à Ansible on peut se connecter aux différents serveurs du système qu'on souhaite automatiser et on lance des programmes, nommés playbooks, chargés d'exécuter des instructions qui devraient sinon être exécutées manuellement. Ces programmes utilisent des modules Ansible conçus pour répondre aux attentes spécifiques du point de terminaison en matière de connectivité, d'interface et de commandes. Ansible exécute ensuite ces modules (via une connexion SSH standard par défaut). Par conséquent, toute notre infrastructure est gérée grâce au machine master.

Un playbook Ansible est un modèle de tâches d'automatisation, qui sont des opérations informatiques complexes exécutées sans intervention humaine. Les playbooks Ansible sont écrits au format YAML lisible par l'homme et exécutés sur un ensemble, un groupe ou une classification d'hôtes, qui forment ensemble un inventaire.

III. Conclusion

Dans ce chapitre, nous avons présenté l'architecture du futur réseau d'information de l'entreprise IT24 après avoir intégrée une solution fiable d'automatisation grâce à l'outil Ansible et via une connexion sécurisée OpenSSH. Une fois achevée, la phase de conception, déterminés, les tâches à réaliser, il ne reste plus qu'à mettre en place cette conception.

CHAPTER 4

CHAPITRE 4 : REALISATION

I. Introduction

Le projet vise à mettre en place une solution d'automatisation de gestion d'infrastructure au sein de l'entreprise cliente, offrant un accès sécurisé via SSH à l'infrastructure. La solution s'appuie sur la technologie Ansible et implique la création d'une infrastructure virtuelle avec VirtualBox, la configuration d'Ansible sur une machine Master, et l'utilisation de clés SSH-KEY pour l'authentification sécurisée.

II. Infrastructure virtuelle

1. La machine Master : nœud maitre

La machine Master est une machine virtuelle de type RHEL 7.9. La Machine Master est au cœur du projet, orchestrant l'automatisation et la gestion de l'infrastructure. Ce chapitre explore sa configuration et son rôle central dans l'efficacité opérationnelle.

➤ Configuration logicielle :

- Système d'exploitation : Red Hat Enterprise Linux 7.9 Server avec GUI.
- Service : nœud maitre (Serveur d'infrastructure).
- Adresse IP: 192.168.42.130/24

➤ Installation (Voir Annexe 1).

2. Machines virtuelles clientes

Dans cette partie, on va décrire la création et la configuration des machines virtuelles sur le serveur Master en utilisant le client ansible.

2.1. VM1 : worker1 (webserver)

Son rôle c'est un serveur web dans l'infrastructure de l'entreprise.

➤ Configuration logicielle :

- Système d'exploitation : Red Hat Enterprise Linux 7.9 version minimale.

- Service: Serveur web (Web Server)
- Adresse IP: 192 .168.43.132/24

Dans cette section, on va suivre la procédure de création de la machine virtuelle représentant le serveur web de l'entreprise.

➤ **Installation (Voir Annexe 1).**

2.2. VM2 : worker 2

Son rôle c'est un serveur de production dans l'infrastructure de l'entreprise.

➤ **Configuration logicielle :**

- Système d'exploitation : Red Hat Enterprise Linux 7.9 version minimale.
- Service : Serveur de production
- Adresse IP : 192 .168.43.133/24

➤ **Installation (Voir Annexe 1).**

2.3. VM3 : worker 3

Son rôle c'est une base de données dans l'infrastructure de l'entreprise.

➤ **Configuration logicielle :**

- Système d'exploitation : Red Hat Enterprise Linux 7.9 version minimale.
- Service : Base de données (Data base)
- Adresse IP : 192 .168.43.134/24

Dans cette section, on va suivre la procédure de création de la machine virtuelle représentant la base de données de l'entreprise.

➤ **Installation (Voir Annexe 1).**

III. Configuration de l'Ansible dans la machine Master

Dans cette partie on va installer et configurer l'Ansible dans le nœud de contrôle nommé master (Host Name : Master).

1. Création de l'utilisateur ansible.

Après l'installation de la machine Master on crée un nouvel utilisateur nommé « **ansible** », auquel nous donnerons les droits de **sudo** ; tel que nous l'ajoutons group « **wheel** ».

```
[root@master ~]# hostname  
master  
[root@master ~]# sudo useradd ansible  
useradd: user 'ansible' already exists  
[root@master ~]# █
```

FIGURE 3 : CREATION D'UTILISATEUR ANSIBLE

```
[root@master ~]#  
[root@master ~]#  
[root@master ~]# sudo usermod -aG wheel ansible  
[root@master ~]# █
```

FIGURE 4 : AJOUT D'USER ANSIBLE AU GROUP WHEEL

2. La configuration de DNS

Pour faciliter le travail on réalise le DNS local, pour nous puissions utiliser les hostnames au lieu des adresses IP. Pour se faire nous configurons le fichier **/etc/hosts** comme suivant :

```
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4  
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6  
192.168.43.132 worker1 worker1.localdomain localhost7 localhost7.localdomain7  
192.168.43.133 worker2 worker2.localdomain localhost8 localhost8.localdomain8  
192.168.43.134 worker3 worker3.localdomain localhost9 localhost9.localdomain9
```

FIGURE 5 : LA CONFIGURATIONS DU FICHIER /ETC/HOSTS

3. L'installation de l'ansible

Pour installer l'ansible on se connecte par l'utilisateur ansible puis on commence l'installation :

```
[root@master ~]# sudo su - ansible
Last login: Sat Aug 5 22:49:57 +01 2023 from desktop-th03937 on pts/1
[ansible@master ~]$ sudo yum install ansible
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
rhel-7-server-ansible-2.9-rpms | 4.0 kB 00:00:00
rhel-7-server-rpms | 3.5 kB 00:00:00
(1/2): rhel-7-server-rpms/7Server/x86_64/updateinfo | 4.3 MB 00:00:06
(2/2): rhel-7-server-rpms/7Server/x86_64/primary_db | 95 MB 00:01:04
Package ansible-2.9.27-1.el7ae.noarch already installed and latest version
Nothing to do
```

FIGURE 6 : INSTALLATION DE L'ANSIBLE

4. La configuration de l'ansible

Le fichier de configuration Ansible par défaut se trouve sous `/etc/ansible/ansible.cfg`.
Mais nous allons créer les fichiers suivants :

- `/home/ansible/tasks/ansible.cfg` : le fichier de configuration.
- `/home/ansible/tasks/inventory` : le fichier d'inventaire pour les machines (hosts).

```
[ansible@master ~]$ cd tasks
[ansible@master tasks]$ ll
total 8
-rw-rw-r--. 1 ansible ansible 221 Jul 28 19:15 ansible.cfg
-rw-rw-r--. 1 ansible ansible 58 Jul 28 19:21 inventory
[ansible@master tasks]$
```

FIGURE 7 : LE DOSSIER TASKS

3.1. Configuration du fichier `ansible.cfg`

Le fichier `ansible.cfg` permet aux utilisateurs de voir tous les paramètres de configuration disponibles, leurs valeurs par défaut, comment les définir et d'où vient leur valeur actuelle.

```
[defaults]
inventory = /home/ansible/tasks/inventory
roles_path = /etc/ansible/roles:/home/ansible/tasks/roles
remote_port = 22
remote_user = ansible
module_name = command
forks = 5
[privilege_escalation]
become = false
~
```

FIGURE 8 : CONFIGURATION DE FICHIER ANSIBLE.CFG

3.2. Configuration du fichier d'inventaire

Un inventaire est une source de données connue d'avance sur les cibles de gestion Ansible organisée en groupes. Les tâches sont exécutées pour des hôtes ou des groupes d'hôtes dans un inventaire défini.

```
[webservers]
worker1

[prod]
worker2

[database]
worker3
```

FIGURE 9 : FICHIER D'INVENTAIRE

IV. Configuration des nœuds gérées

1. Création de l'utilisateur ansible.

Pour toutes les machines clientes nous allons créer un utilisateur nommé ansible avec le même mot de passe pour faciliter les tâches. Et de même comme le cas de nœud maître nous allons attribuer les droits de **sudo** à l'utilisateur ansible, en l'ajoutant au groupe wheel.

V. Accès Sécurisé Via SSH

Dans cette partie nous allons configurer l'accès SSH. Tel que l'accès SSH sera configuré pour être uniquement autorisé depuis la machine Master vers les autres machines de l'infrastructure.

```
[root@localhost ~]# sudo adduser ansible
[root@localhost ~]# passwd ansible
Changing password for user ansible.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@localhost ~]#
```

FIGURE 10 : CREATION D'UTILISATEUR ANSIBLE

De cette manière on crée l'utilisateur Ansible dans toutes les machines à savoir worker 1, 2 et 3. Ainsi que dans le nœud maître.

1. L'installation d'OpenSSH.

OpenSSH est un ensemble d'outils informatiques libres permettant des communications sécurisées sur un réseau informatique en utilisant le protocole SSH.

1.1. L'installation de openssh-server dans la machine Master

Nous allons installer le package openssh-server dans le nœud maitre :

```
[ansible@master ~]$ sudo yum install openssh-server
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
Resolving Dependencies
--> Running transaction check
--> Package openssh-server.x86_64 0:7.4p1-22.el7_9 will be updated
--> Package openssh-server.x86_64 0:7.4p1-23.el7_9 will be an update
--> Processing Dependency: openssh = 7.4p1-23.el7_9 for package: openssh-server-7.4p1-23.el7_9.x86_64
--> Running transaction check
--> Package openssh.x86_64 0:7.4p1-22.el7_9 will be updated
--> Processing Dependency: openssh = 7.4p1-22.el7_9 for package: openssh-clients-7.4p1-22.el7_9.x86_64
--> Package openssh.x86_64 0:7.4p1-23.el7_9 will be an update
--> Running transaction check
--> Package openssh-clients.x86_64 0:7.4p1-22.el7_9 will be updated
--> Package openssh-clients.x86_64 0:7.4p1-23.el7_9 will be an update
--> Finished Dependency Resolution

Dependencies Resolved
```

FIGURE 11 : INSTALLATION DU PACKAGE OPENSSH-SERVER

1.2. L'installation d'openssh-client dans les travailleurs

Dans cette partie nous allons installer le package openssh-clients dans toutes les machines gérées.

```
[ansible@worker3 ~]$ sudo yum install openssh-clients
Loaded plugins: product-id, search-disabled-repos, subscription-manager
Resolving Dependencies
--> Running transaction check
--> Package openssh-clients.x86_64 0:7.4p1-22.el7_9 will be updated
--> Package openssh-clients.x86_64 0:7.4p1-23.el7_9 will be an update
--> Processing Dependency: openssh = 7.4p1-23.el7_9 for package: openssh-clients-7.4p1-23.el7_9.x86_64
--> Running transaction check
--> Package openssh.x86_64 0:7.4p1-22.el7_9 will be updated
--> Processing Dependency: openssh = 7.4p1-22.el7_9 for package: openssh-server-7.4p1-22.el7_9.x86_64
--> Package openssh.x86_64 0:7.4p1-23.el7_9 will be an update
--> Running transaction check
--> Package openssh-server.x86_64 0:7.4p1-22.el7_9 will be updated
--> Package openssh-server.x86_64 0:7.4p1-23.el7_9 will be an update
--> Finished Dependency Resolution

Dependencies Resolved
```

FIGURE 12 : INSTALLATION DE PACKAGE OPENSSH-CLIENTS

De même pour les autres machines travailleurs worker 2 et worker 3. Nous installons le package openssh-clients.

2. Configuration d'OpenSSH

La configuration d'OpenSSH consiste à l'activation de service sshd.

```
[ansible@master ~]$ service sshd start
Redirecting to /bin/systemctl start sshd.service
[ansible@master ~]$
```

FIGURE 13 : ACTIVER LE SERVICE SSHD SUR LE NŒUD MAITRE

```
[ansible@master ~]$ service sshd status
Redirecting to /bin/systemctl status sshd.service
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; vendor preset: enable
   d)
   Active: active (running) since Thu 2023-08-24 11:49:21 +01; 1h 5min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 4638 (sshd)
      Tasks: 1
   CGroup: /system.slice/ssh.service
           └─4638 /usr/sbin/sshd -D

Aug 24 11:49:21 master systemd[1]: Stopped OpenSSH server daemon.
Aug 24 11:49:21 master systemd[1]: Starting OpenSSH server daemon...
Aug 24 11:49:21 master sshd[4638]: Server listening on 0.0.0.0 port 22.
Aug 24 11:49:21 master sshd[4638]: Server listening on :: port 22.
Aug 24 11:49:21 master systemd[1]: Started OpenSSH server daemon.
[ansible@master ~]$
```

FIGURE 14 : VERIFIER L'ACTIVATION DU SERVICE SSHD SUR LE MASTER

```
[ansible@worker1 ~]$ service sshd start
Redirecting to /bin/systemctl start sshd.service
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to manage system services or units.
Authenticating as: root
Password:
==== AUTHENTICATION COMPLETE ====
```

FIGURE 15 : ACTIVER LE SERVICE SSHD SUR LE WEBSERVER

```
[ansible@worker1 ~]$ service sshd status
Redirecting to /bin/systemctl status sshd.service
■ sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; vendor preset: enable
   d)
   Active: active (running) since Thu 2023-08-24 12:58:22 +01; 4min 47s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 1339 (sshd)
      CGroup: /system.slice/ssh.service
           └─1339 /usr/sbin/sshd -D
[ansible@worker1 ~]$
```

FIGURE 16 : VERIFIER L'ACTIVATION DU SERVICE SSHD SUR LE WOKER 1

De même pour les machines worker 2 et worker 3. Nous activons le package sshd. Puis nous vérifions s'activation.

VI. Création des clés SSH-KEY

1. Génération de la clé SSH

Sur la machine Master nous générons la clé SSH grâce au commande **ssh-keygen** comme suit :

```
[ansible@master home]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ansible/.ssh/id_rsa):
/home/ansible/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ansible/.ssh/id_rsa.
Your public key has been saved in /home/ansible/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:ehqNqfMGVSL0AtvT5DAy0H6E3oa3wbFmxR7JJ+KtrfDg ansible@master
The key's randomart image is:
+---[RSA 2048]---+
|.o+oo+...|
|o==*O o|
|o.=oXo+|
|+ @.=|
|* + .S|
|o .|=|
|. += o|
|E.o+|
|.oBo|
+-----[SHA256]-----+
[ansible@master home]$
```

FIGURE 17 : CREATION DE CLE SSH

2. Copie la clé vers les machines gérées

Maintenant nous allons copier la clé générée précédemment vers les machines à gérées à savoir worker1, worker 2 et worker 3.

❖ Copie la clé vers la machine webserver

```
[ansible@master home]$ ssh-copy-id ansible@192.168.43.132
/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ansible/.ssh/id_rsa.pub"
/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
ansible@192.168.43.132's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'ansible@192.168.43.132'"
and check to make sure that only the key(s) you wanted were added.
[ansible@master home]$
```

FIGURE 18 : COPIE LA CLE ID_RSA.PUB VERS WEBSERVER

❖ Copie la clé vers la machine worker 2

```
[ansible@master home]$ ssh-copy-id ansible@192.168.43.133
/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ansible/.ssh/id_rsa.pub"
/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
ansible@192.168.43.133's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'ansible@192.168.43.133'"
and check to make sure that only the key(s) you wanted were added.

[ansible@master home]$ █
```

FIGURE 19 : COPIE LA CLE ID_RSA.PUB VERS WORKER 2

❖ Copie la clé vers la machine worker 3

```
[ansible@master home]$ ssh-copy-id ansible@192.168.43.134
/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ansible/.ssh/id_rsa.pub"
/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
ansible@192.168.43.134's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'ansible@192.168.43.134'"
and check to make sure that only the key(s) you wanted were added.

[ansible@master home]$ █
```

FIGURE 20 : COPIE LA CLE ID_RSA.PUB VERS WORKER 3

3. Tester la connectivité

Après que nous avons copier la clé ssh dans toutes les machines travailleuses, nous allons tester l'accès à ses machines à partir de la machine master :

```
[ansible@master home]$ ansible all -m ping -i /home/ansible/tasks/inventory
worker2 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
worker3 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
worker1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
[ansible@master home]$ █
```

FIGURE 21 : LA CONNECTIVITE REUSSITE

Nous remarquons que toutes les machines sont accessibles à partir du nœud maître à savoir la machine master. Autrement dit nous avons le contrôle total et complet sur toutes les machines sans nécessité d'accès direct physique sur chaque machine.

VII. Conclusion

Grâce à ce chapitre nous arrivons à manager les machines travailleurs grâce au machine master qui joue le rôle d'orchestre de notre infrastructure.

En conclusion, la réalisation de notre projet d'automatisation a été un succès. La solution déployée répond aux besoins d'IT24.MA en simplifiant la gestion de l'infrastructure, en renforçant la sécurité et en optimisant les ressources. Les résultats obtenus ouvrent la voie à une efficacité opérationnelle accrue et à une plus grande agilité informatique pour l'entreprise. Dans le chapitre suivant, nous exploitons la solution mis en œuvre pour héberger le site web statique d'entreprise.

CHAPTER 5

CHAPITRE 5 : MISE EN ŒUVRE

I. Hébergement du site web d'entreprise sur le serveur web

1. Introduction

Pour héberger le site web d'entreprise **IT24** sur le serveur web worker 1, nous allons écrire le playbook que permettra d'installer les packages nécessaires.

2. L'hébergement du site web d'entreprise

2.1. La création de site web

Nous sommes demandés de créer un site internet statique pour l'entreprise IT24, ce site doit présenter à ses visiteurs les domaines de travail de l'entreprise ainsi que ses services.

Pour créer ce site nous avons utilisé le logiciel Visual Studio code pour coder ce site. Ainsi que nous avons utilisé le model MVC et Bootstrap pour rendre de site responsif.

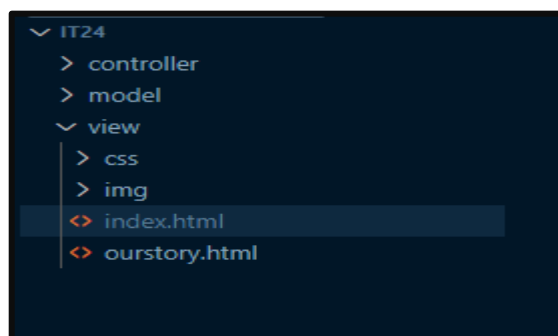


FIGURE 22 : STRUCTURE DE SITE

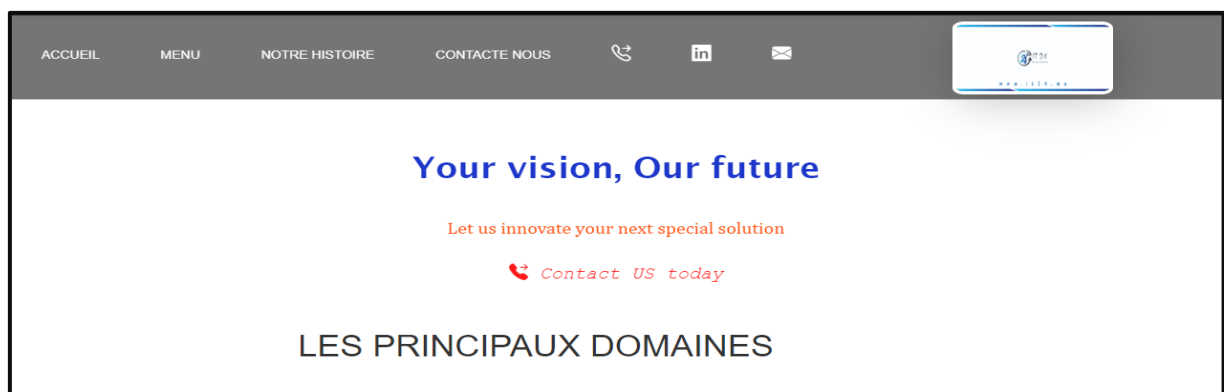


FIGURE 23 : PAGE D'ACCUEIL

2.2. Installation de HTTPD

Nous allons écrire le fichier YAML qui nous permet d'installer les packages nécessaires pour l'hébergement du site web d'entreprise IT24 sur son serveur web.

Dans notre cas nous avons un serveur virtuel avec un système d'exploitation linux RHEL 7.9. Puis on a choisi Apache tant que logiciel de web serveur (le package HTTPD). Ensuite, allons copier les fichiers de site vers le serveur.

Voici le script YAML pour déployer le serveur web dans le web server :

```
---
- name: Deploy IT24 Website on worker 1 webserver
  hosts: webservers
  become: yes # Use sudo to execute commands as root

  tasks:
    - name: Update System Packages
      yum:
        name: '*'
        state: latest
      become: yes

    - name: Install Apache
      yum:
        name: httpd
        state: present
      become: yes

    - name: Start Apache Service
      service:
        name: httpd
        state: started
        enabled: yes
      become: yes
```

```

- name: Allow HTTP and HTTPS Through Firewall
  firewallld:
    service: "{{ item }}"
    permanent: yes
    state: enabled
  with_items:
    - http
    - https
  become: yes

- name: Create Website Directory
  file:
    path: /var/www/html/it24
    state: directory
  become: yes

- name: Set Apache User and Group Ownership
  file:
    path: /var/www/html/it24
    owner: apache
    group: apache
    recurse: yes
  become: yes
```

FIGURE 24 : SITE.YML PLAYBOOK

```
[ansible@master tasks]$ gedit site.yml
[ansible@master tasks]$ ansible-playbook site.yml

PLAY [Deploy Website on RHEL 7.9 Web Server] *****

TASK [Gathering Facts] *****
ok: [worker1]

TASK [Update System Packages] *****
ok: [worker1]

TASK [Install Apache] *****
ok: [worker1]

TASK [Start Apache Service] *****
ok: [worker1]

TASK [Allow HTTP and HTTPS Through Firewall] *****
ok: [worker1] => (item=http)
ok: [worker1] => (item=https)

TASK [Create Website Directory] *****
ok: [worker1]

TASK [Set Apache User and Group Ownership] *****
changed: [worker1]

TASK [Test Apache Configuration] *****
changed: [worker1]

TASK [Restart Apache Service] *****
changed: [worker1]

PLAY RECAP *****
worker1 : ok=9    changed=3    unreachable=0    failed=0    skipped=0
0 rescued=0    ignored=0

[ansible@master tasks]$
```

FIGURE 25 : EXECUTION DE PLAYBOOK SITE.YML

Ce playbook nous a permis d'installer le package **HTTPD**, puis on l'active.

Pour copier les fichiers de site vers la machine master puis vers le serveur nous avons suivi les étapes suivantes :

- D'abord nous avons installé le logiciel FileZilla Client, puis se connecté au machine Master et copie le projet Vs code de notre site web de notre machine personnelle Windows 10 vers la machine virtuelle Master.

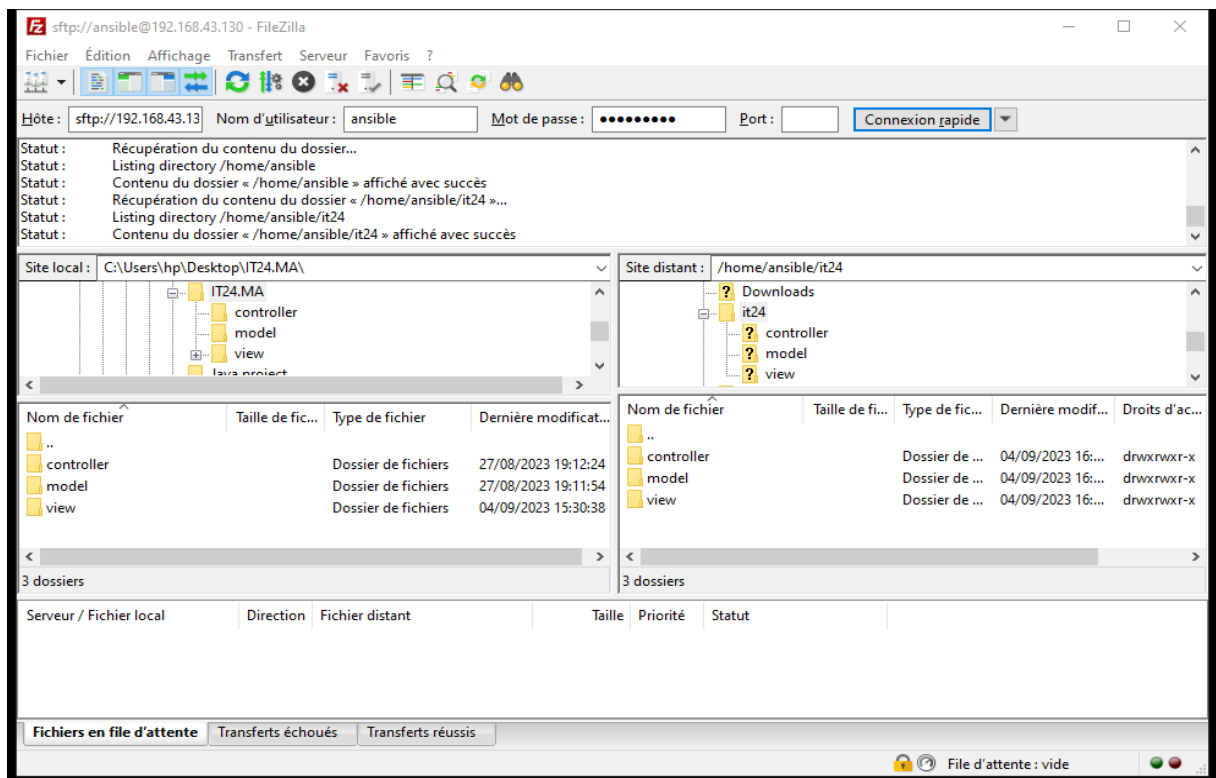


FIGURE 26 : INTERFACE DE LOGICIEL FILEZILLA CLIENT

- Puis on passe au machine master pour copier les fichiers de site vers le serveur web, à savoir dans notre cas est le worker 1, grâce au ssh key qu'on a créée précédemment (Rappel : l'accès au serveur est donné uniquement au machine Master).

```
[ansible@master ~]$ scp -r /home/ansible/it24 root@192.168.43.132:/var/www/html/it24.ma

root@192.168.43.132's password:
controle.php          100%   9      8.3KB/s   00:00
index.html            100% 13KB    8.4MB/s   00:00
ourstory.html         100% 2339    2.0MB/s   00:00
style.css             100% 6359    5.0MB/s   00:00
Développement informatique.png 100% 8375    4.8MB/s   00:00
Développement informatique.jpg 100% 8410    5.4MB/s   00:00
Installation serveurs 1.jpg    100% 12KB    7.1MB/s   00:00
Installation serveurs.jpg    100% 9841    673.9KB/s 00:00
Interventions sur site.jpg    100% 7546    920.2KB/s 00:00
it24 logo.jpg          100% 722KB   10.3MB/s   00:00
logo.png              100% 5731    2.7MB/s   00:00
Maintenance informatique.jpg   100% 9371    2.9MB/s   00:00
Sauvegarde et récupération de données.jpg 100% 7092    3.5MB/s   00:00
sauvegarde-1.jpg       100% 63KB    3.7MB/s   00:00
Support informatique et maintenance régulière.jpg 100% 16KB    1.3MB/s   00:00
Sécurité des données.png    100% 3501    863.6KB/s 00:00
INTRANET.jpg          100% 9463    806.8KB/s 00:00
Intranet1.jpg          100% 10KB    2.6MB/s   00:00
IP MPLS.jpg            100% 7734    3.5MB/s   00:00
Plateformes Internet.jpg    100% 9365    855.3KB/s 00:00
Visioconférence.jpg    100% 12KB    3.2MB/s   00:00
Cyber Défense.jpg      100% 7814    2.9MB/s   00:00
```

FIGURE 27 : COPIER LES FICHIERS DU SITE VERS LE SERVEUR

II. Configuration de Load Balancer

1. Introduction

Pour réduire le temps de réponse de notre site suite aux requêtes des utilisateurs. Nous allons mettre en place le load balancing, ou répartition de charge, qui est une technologie conçue pour distribuer la charge de travail entre différents serveurs.

2. La configuration

Pour configurer le load balancing nous avons créé deux serveurs web secondaires avec adresses IP respectivement **192.168.43.135/24** et **192.168.43.136/24**. Ainsi que le serveur Load Balancer, son adresse IP **192.168.43.140/24**, qui s'occupera de rôle du load balancer.

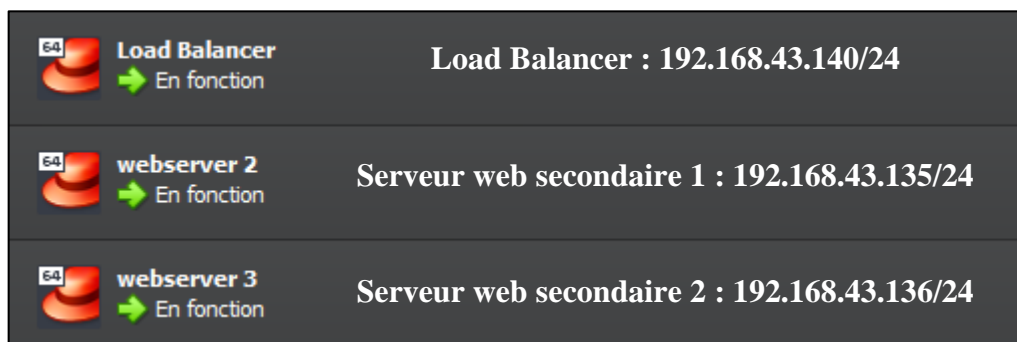


FIGURE 28 : LE LOAD BALANCING MACHINES

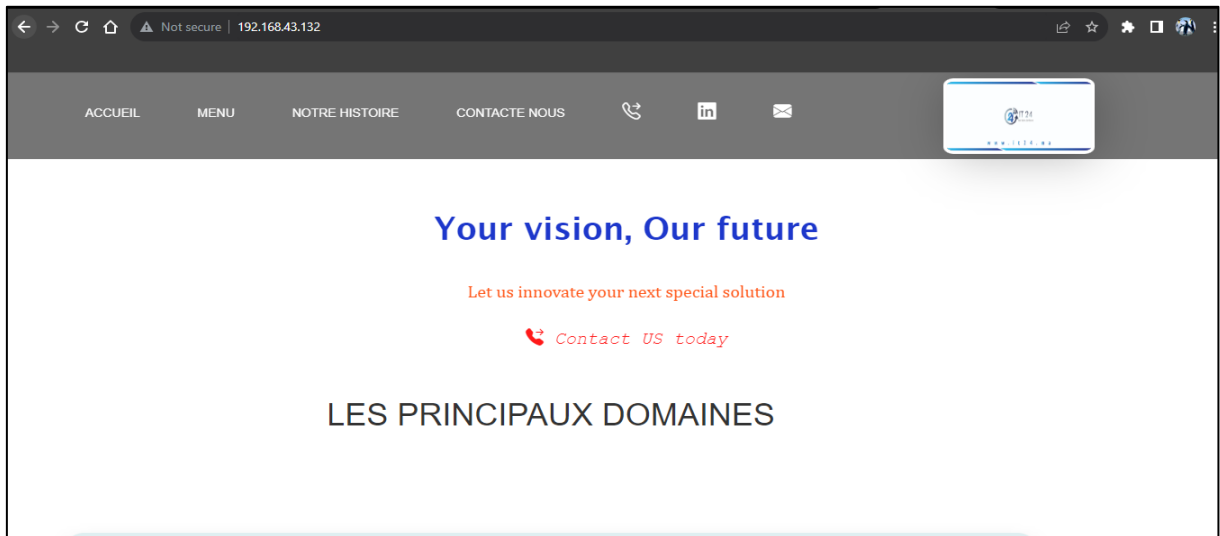


FIGURE 29 : SITE HEBERGER SUR LE SERVEUR PRINCIPAL

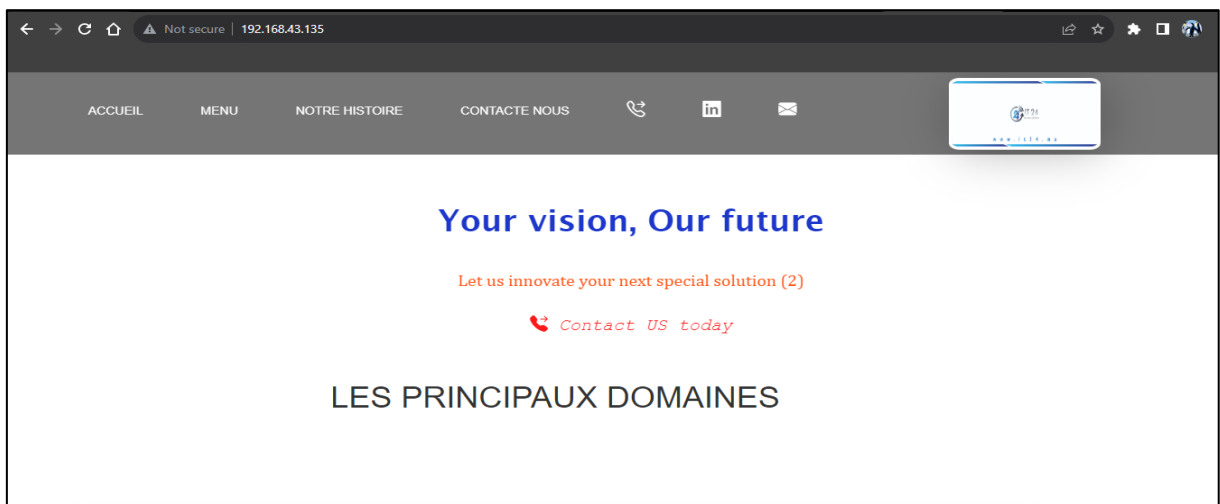


FIGURE 30 : SITE HEBERGER SUR LE SERVEUR WEB 2

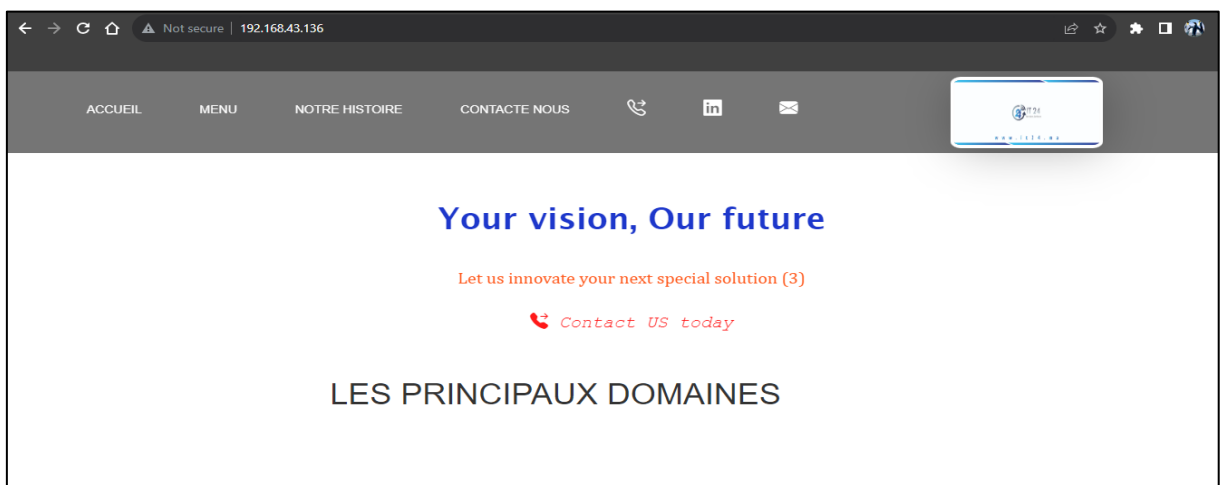


FIGURE 31 : SITE HEBERGER SUR LE SERVEUR WEB 3

❖ La mise à jour de configuration d'Ansible

On ajoute les deux serveurs, ainsi que le load balancer au fichier inventory :

```
[ansible@master tasks]$ gedit inventory
[ansible@master tasks]$ cat inventory

[webservers]
worker1
webserver2
webserver3

[loadBalancer]
loadBalancer

[prod]
worker2

[database]
worker3
[ansible@master tasks]$
```

FIGURE 32 : FICHER INVENTORY

❖ Création des rôles ansible

On va créer les rôles Ansible pour configurer LoadBalancer et Apache Web Server. Les rôles vous permettent de charger automatiquement les vars_files, les tâches, les gestionnaires et autres artefacts Ansible associés en fonction d'une structure de fichiers connue.

On crée deux rôles Ansible, un pour web server et un second pour loadbalancer comme suit :

```
[ansible@master tasks]$ ansible-galaxy init webservers
- Role webservers was created successfully
```

```
[ansible@master tasks]$ ansible-galaxy init loadbalancer
```

FIGURE 33 : CREATION DES ROLES

Maintenant, après avoir créé les rôles, on accède aux rôles du serveur Web :

- Création de la tâche pour configurer le serveur Web

```
[ansible@master tasks]$ gedit webserver/tasks/main.yml
[ansible@master tasks]$ cat webserver/tasks/main.yml
---
# tasks file for webserver
- name: Install Httpd software
  package:
    name: "{{ package_name }}"
    state: present
- name: copy website
  template:
    src: /home/ansible/it24
    dest: "{{ doc_root }}"
- name: Httpd service start
  service:
    name: "{{ service_name }}"
    state: started
[ansible@master tasks]$
```

FIGURE 34 : WEBSERVER/TASKS/MAIN.YML

- Configuration des gestionnaires dans le répertoire des gestionnaires

```
[ansible@master tasks]$ gedit webserver/vars/main.yml
[ansible@master tasks]$ cat webserver/vars/main.yml
---
# vars file for webserver
package_name: "httpd"
service_name: "httpd"
doc_root: "/var/www/html/"
[ansible@master tasks]$ █
```

FIGURE 35: WEBSERVER/VARS/MAIN.YML

- Définition des variables utilisées dans le fichier main.yml des rôles de serveur

```
[ansible@master tasks]$ gedit webserver/handlers/main.yml
[ansible@master tasks]$ cat webserver/handlers/main.yml
---
# handlers file for webserver
- name: Restart httpd
  service:
    name: "{{ service_name }}"
    state: restarted
[ansible@master tasks]$ █
```

FIGURE 36: WEBSERVER/HANDLERS/MAIN.YML

On accède maintenant aux rôles du load balancer.

- Fichier loadbalancer/tasks/main.yml

```
[ansible@master tasks]$ cat loadbalancer/tasks/main.yml
---
# tasks file for loadbalancer
- name: Install haproxy software
  package:
    name: "{{ package_name }}"
    state: present

- name: Start haproxy service
  service:
    name: "{{ service_name }}"
    state: started
    enabled: yes
[ansible@master tasks]$
```

FIGURE 37: LOADBALANCER/TASKS/MAIN.YML

- Fichier loadbalancer /vars/main.yml

```
[ansible@master tasks]$ cat loadbalancer/vars/main.yml
---
# vars file for loadbalancer
package_name: haproxy
service_name: haproxy
[ansible@master tasks]$
```

FIGURE 38 : LOADBALANCER/VARS/MAIN.YML

- Fichier loadbalancer /handlers/main.yml

```
[ansible@master tasks]$ cat loadbalancer/handlers/main.yml
---
# handlers file for loadbalancer
- name: restart haproxy
  service:
    name: "{{ service_name }}"
    state: restarted
[ansible@master tasks]$
```

FIGURE 39 : LOADBALANCER/HANDLERS/MAIN.YML

❖ Exécution des taches

On crée un playbook **task.yml** qui combinera ces deux rôles ensemble et pourra exécuter toutes ces tâches ensemble.

```
---
- hosts: webservers
  roles:
    - webserver

- hosts: loadBalancer
  roles:
    - loadbalancer
```

FIGURE 40 : TASK.YML

```
[ansible@master tasks]$ ansible-playbook task.yml

PLAY [webserver] *****

TASK [Gathering Facts] *****
ok: [webserver3]
ok: [worker1]
ok: [webserver2]

TASK [webserver : Install Httpd software] *****
ok: [worker1]
ok: [webserver3]
ok: [webserver2]

TASK [webserver : Httpd service start] *****
ok: [worker1]
ok: [webserver3]
ok: [webserver2]

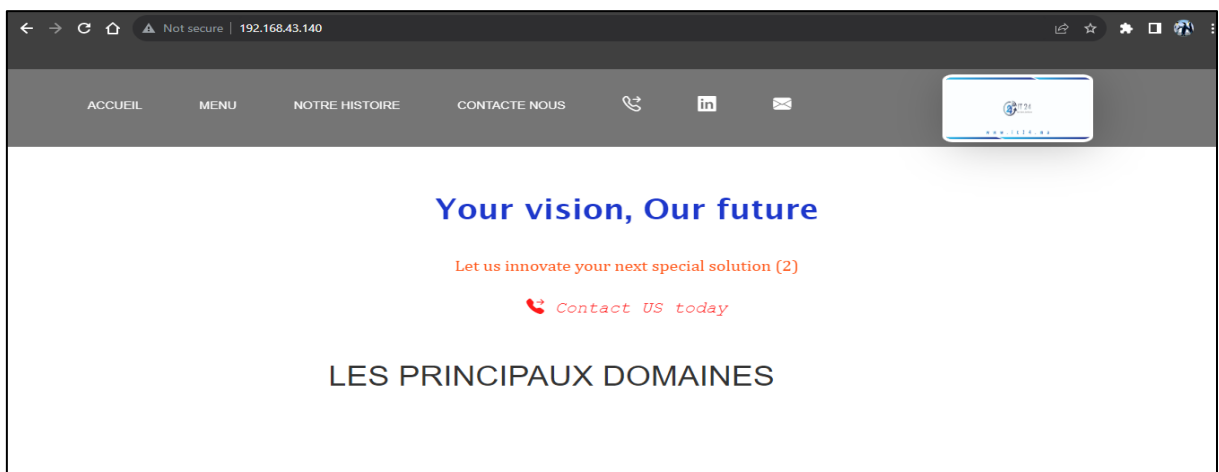
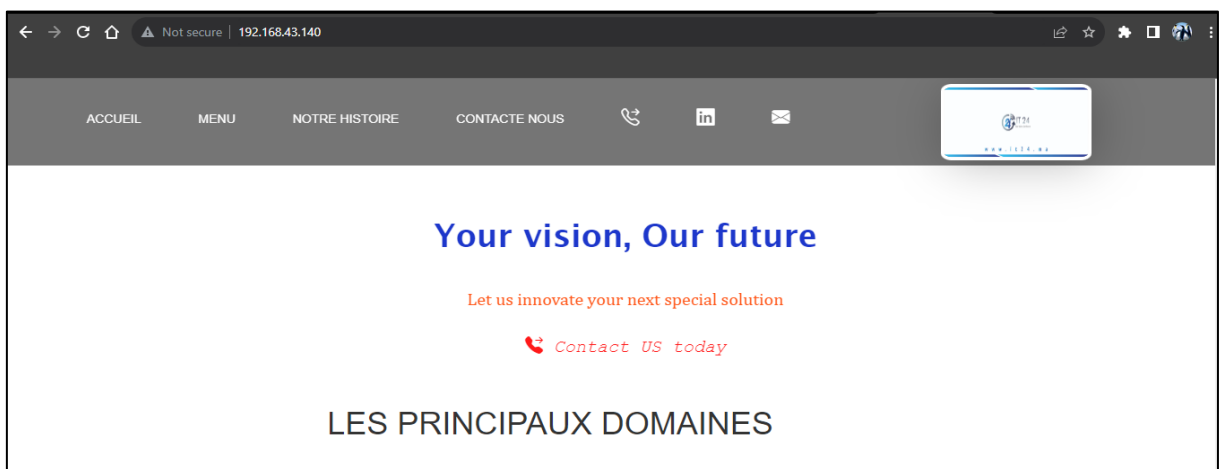
PLAY [LoadBalancer] *****

TASK [Gathering Facts] *****
ok: [loadBalancer]

TASK [loadbalancer : Install haproxy software] *****
ok: [loadBalancer]
```

FIGURE 41 : EXECUTION DU TASK.YML

Par consequence nous avons fini la configuration du load balancing, on teste le fonctionnement grace à un navigateur en tapant l'adresse IP de load balancer:



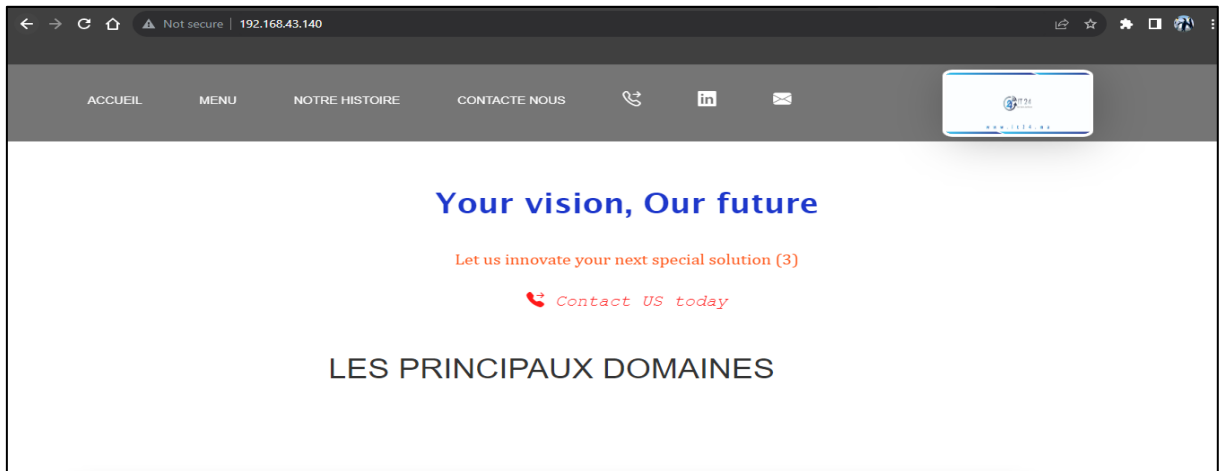


FIGURE 42 : FONCTIONNEMENT DE LOANDBALANCER

Ici, nous pouvons voir que l'IP change, ce qui signifie que LoadBalancer est configuré avec succès.

3. Conclusion

Nous avons configuré avec succès le LoadBalancer.

III. Conclusion

Au cours de ce chapitre, nous avons exploité notre infrastructure automatisée pour gérer efficacement diverses tâches. Tout d'abord, nous avons réussi à héberger le site web de l'entreprise IT24 sur ses propres serveurs web. De plus, nous avons mis en place le load balancing pour optimiser la gestion du trafic des visiteurs sur le site web de l'entreprise.

Conclusion générale

Ce présent projet est élaboré dans le cadre d'un stage d'initiation professionnelle intégré dans la formation d'ingénieur à l'école Hassania des Travaux Publics, effectué au sein de cabinet de conseils et services informatique IT24.MA. Et a pour but final l'obtention du diplôme d'ingénieur d'Etat en informatique EHTP.

Ce projet intitulé « Automatisation de gestion d'infrastructure d'entreprise IT24.MA » se place dans le cadre d'automatisation informatique pour couvrir un besoin interne pour l'entreprise IT24.MA. Ce besoin est de permet un accès au réseau local d'entreprise de manière sécurisée via une connexion SSH de type OpenSSH, de centraliser ainsi qu'automatiser la gestion des ressources.

La mise en place de ce projet a été une occasion pour mieux comprendre le concept de la virtualisation et comment l'implémenter d'une part, et d'autre part apprendre comment configurer l'outil d'automatisation Ansible et aussi d'apprendre comment héberger un site web sur un serveur local. Pour répondre à des besoins spécifiques. Il nous a permis aussi d'approfondir et d'appliquer nos connaissances théoriques dans le domaine de virtualisation et d'administration des systèmes informatiques.

Finalement, ce projet était une bonne occasion enrichissante et fructueuse pour s'intégrer dans le domaine professionnel et de découvrir les problèmes qui peuvent survenir tout au long de la réalisation d'un projet. Il m'a permis encore de mettre en valeur les connaissances acquises en classe ce qui m'apporte beaucoup d'enthousiasme et de motivation.

Bibliographie

Nous n'avons trouvé aucune source dans le document actif.

Webographie

https://docs.ansible.com/ansible/latest/getting_started/index.html

<https://www.redhat.com/fr/technologies/management/ansible/what-is-ansible>

<https://www.redhat.com/sysadmin/configuring-ansible>

<https://www.redhat.com/sysadmin/deploying-static-website-ansible>

Annexes

Annexe 1 : Installation et configuration

■ Serveur Master

Pour installer la machine master, nous utiliserons l'image **ISO** de système d'exploitation RHEL 7.9.

Tout d'abord on démarre sur l'image ISO d'installation dans VirtualBox on suit les étapes suivantes :

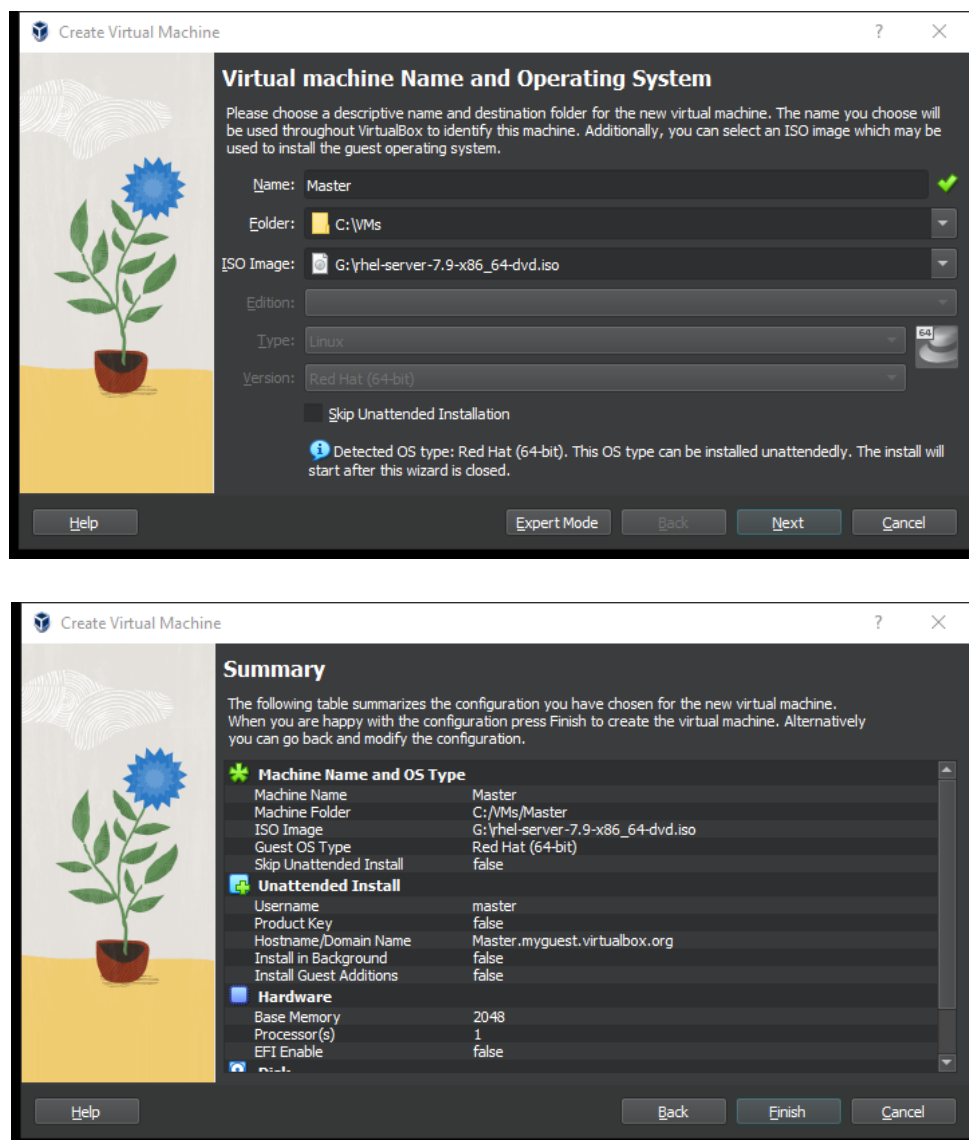


FIGURE 43 : INFORMATIONS SUR LA MACHINE MASTER

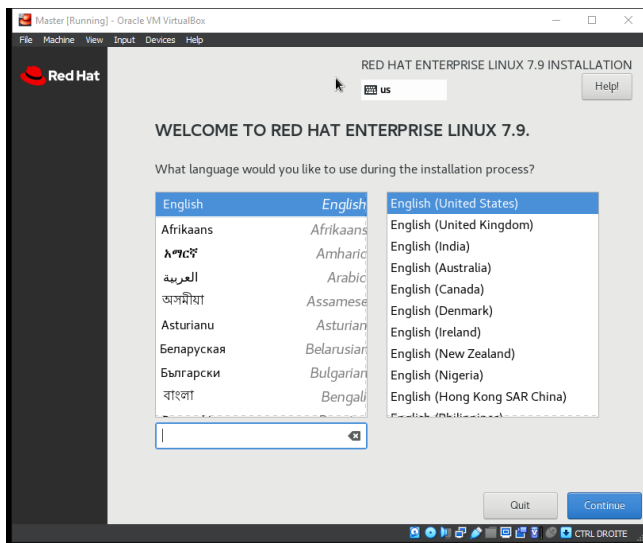


FIGURE 44 : LE CHOIX DE LANGAGE DE SUPPORT

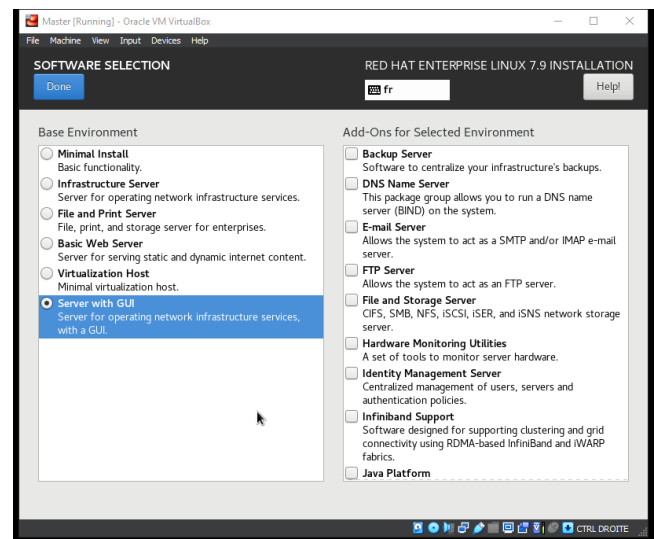


FIGURE 46 : SELECTION DE SERVEUR AVEC GUI

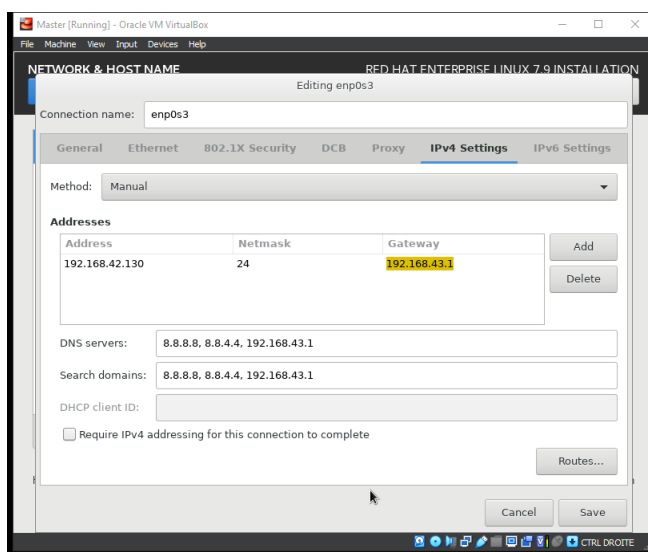


FIGURE 45: ATTRIBUTION DE L'ADRESSE IP
STATIQUE

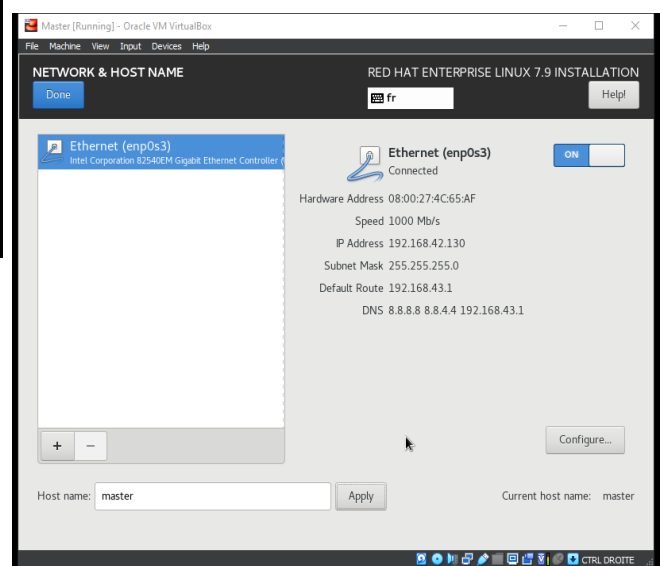


FIGURE 47 : ACTIVER LA CONNECTIVITE ET
DONNER LE NOM DE LA MACHINE

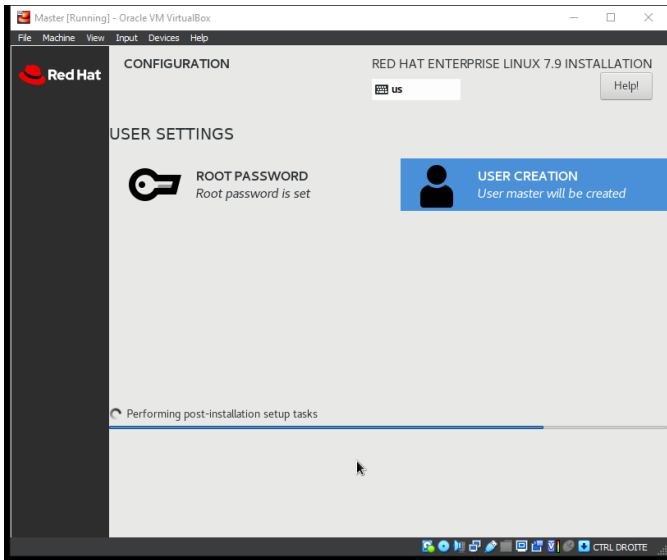


FIGURE 49 : CREATION D'UTILISATEUR ET DE MOT DE PASSE DE SUPER USER ROOT

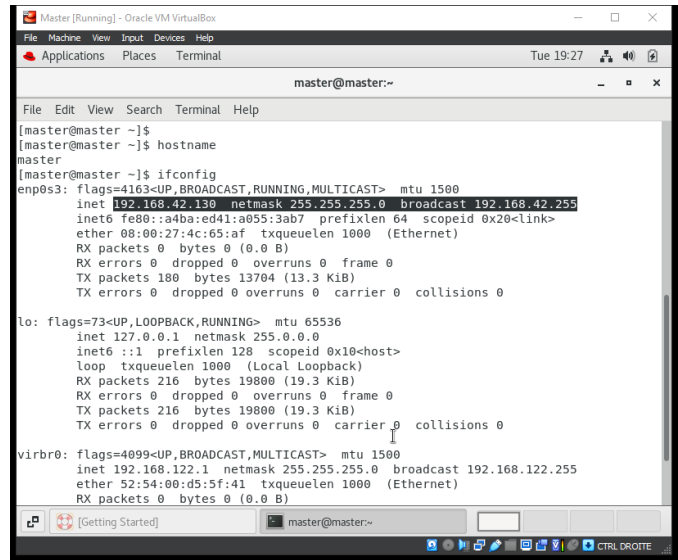


FIGURE 48 : CONFIGURATION DE SERVEUR TERMINEE

■ Webserver, Worker 2 et Worker 3

Pour installer la machine webserver (respectivement worker 2 et 3), nous utiliserons l'image **ISO** de système d'exploitation RHEL 7.9.

Tout d'abord on démarre sur l'image ISO d'installation dans VirtualBox on suit les étapes suivantes :

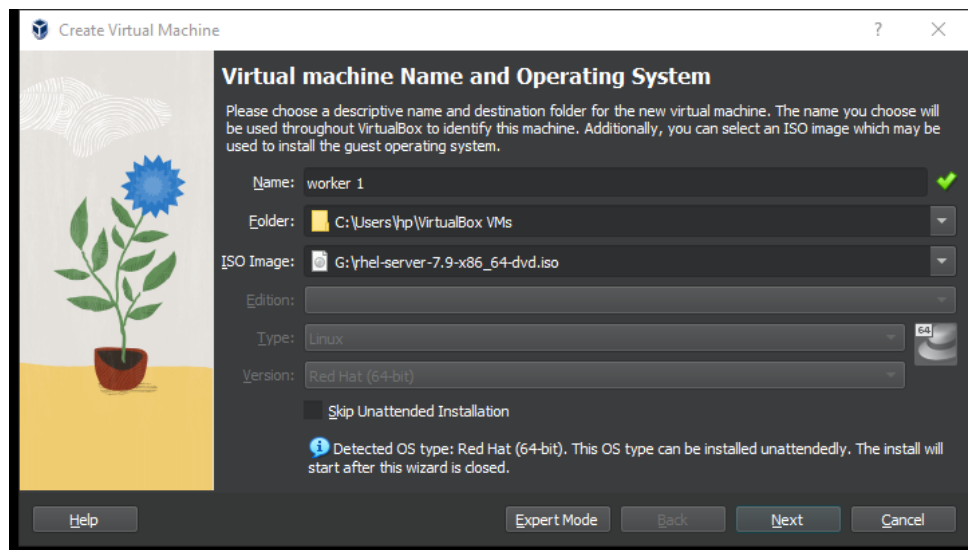


FIGURE 50 : CONFIGURATION DU MACHINE WEBSERVER – WORKER 1

(RESP WORKER 2 ET 3)

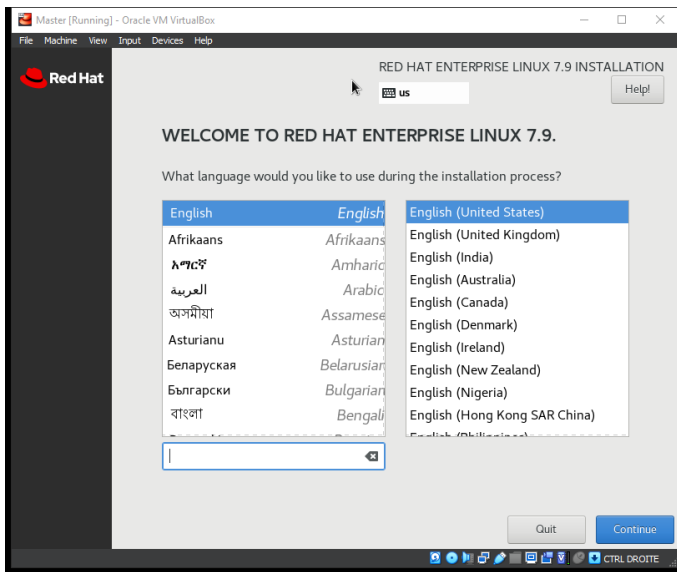


FIGURE 51 : LE CHOIX DE LANGAGE DE SUPPORT

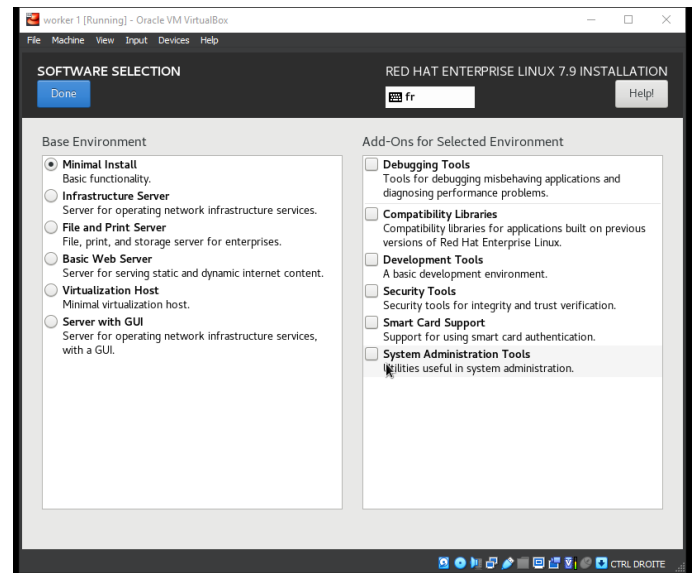


FIGURE 52 : SELECTION D'INSTALLATION MINIMALE

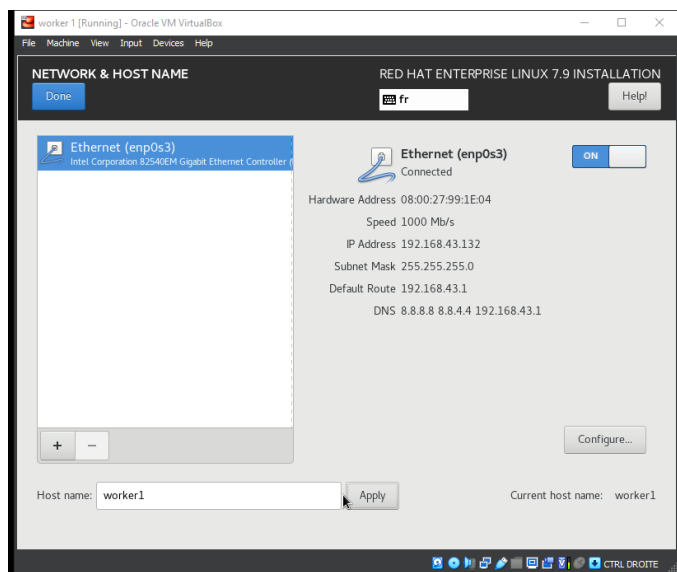


FIGURE 53 : ATTRIBUTION DE L'ADRESSE IP STATIQUE ET ACTIVATION DE LA CONNECTIVITE

Remarque :

Les mêmes étapes sont suivies pour installer et configurer les autres machines clientes à savoir worker 2 et worker 3.

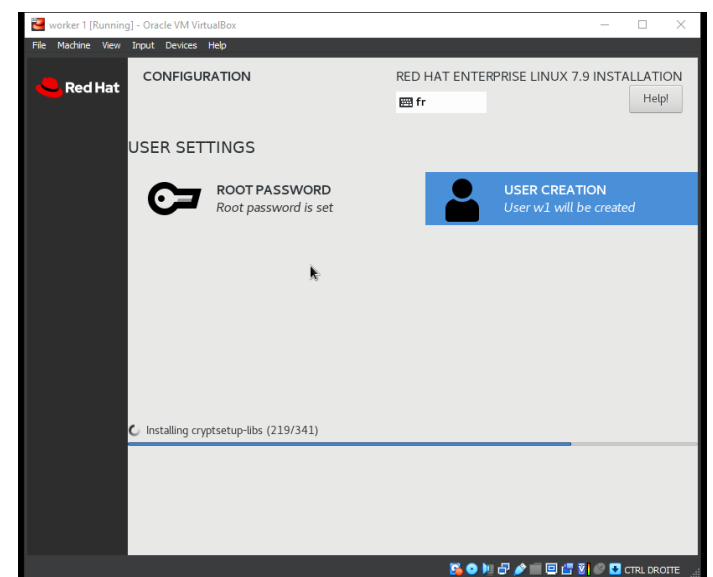


FIGURE 54 : CREATION D'UTILISATEUR ET DE MOT DE PASSE DE SUPER USER ROOT

Annexe 2 : Définitions

TABLEAU 3 : DEFINITIONS

Technologie / Outil / technique	Définition
SSH	Secure Shell, un protocole de communication sécurisé utilisé pour les connexions réseau cryptées.
RHEL	Red Hat Enterprise Linux, une distribution Linux. Red Hat Enterprise Linux est la plateforme Linux d'entreprise leader sur le marché, certifiée pour des centaines de clouds et des milliers de fournisseurs de matériel et de logiciels. Accessible via une souscription, Red Hat Enterprise Linux convient à des cas d'utilisation spécifiques tels que l'edge computing ou les charges de travail SAP.
OpenSSH	Une implémentation open source du protocole SSH. OpenSSH est un ensemble d'outils informatiques libres permettant des communications sécurisées sur un réseau informatique en utilisant le protocole SSH.
SSH-KEYS	Paires de clés SSH comprenant une clé privée et une clé publique pour une authentification sécurisée.
Machine Master	La machine virtuelle RHEL 7.9 désignée comme point central pour la gestion de l'infrastructure.
YAML	<p>YAML est un langage de sérialisation des données lisible par l'utilisateur qui est souvent utilisé pour coder des fichiers de configuration. Pour certains, YAML est l'acronyme de Yet Another Markup Language, pour d'autres, c'est l'acronyme récursif de YAML Ain't Markup Language (YAML n'est pas un langage de balisage), ce qui souligne que le langage YAML s'utilise pour représenter des données plutôt que des documents.</p> <p>YAML est un langage de programmation fréquemment utilisé, car il est conçu pour être parfaitement lisible et compréhensible. Il peut également être utilisé en association avec d'autres langages de programmation. En raison de sa flexibilité et de son accessibilité, YAML est utilisé par l'outil d'automatisation Ansible pour créer des processus d'automatisation, sous la forme de playbooks Ansible.</p>
Wheel	Le groupe wheel est un groupe d'utilisateurs spécial utilisé sur certains systèmes Unix, principalement les systèmes BSD, pour contrôler l'accès à la commande su ou sudo, qui permet à un utilisateur de se faire passer pour un autre utilisateur (généralement le super utilisateur).

Playbook	Un playbook Ansible est un modèle de tâches d'automatisation, qui sont des opérations informatiques complexes exécutées sans intervention humaine ou presque. Les playbooks Ansible sont écrites en langage YAML et exécutés sur un ensemble, un groupe ou une classification d'hôtes, qui forment ensemble un inventaire.
Hébergement d'un site web	L'hébergement Web est la technologie qui permet aux utilisateurs de saisir un nom de domaine dans leur navigateur afin d'afficher le contenu du site Internet correspondant. Dans un premier temps, le créateur du site Internet doit enregistrer un nom de domaine puis placer les contenus sur le serveur d'un prestataire d'hébergement Web. Le système de nom de domaine (DNS) permet d'établir la liaison entre le nom de domaine et le serveur. Dans ce cadre, des serveurs de nom indiquent sous quelle adresse IP un domaine précis peut être trouvé. Si un utilisateur consulte une URL, le nom de domaine est résolu en une adresse IP et l'adresse correspondante est recherchée sur Internet. En cas de succès, le site Internet est affiché à l'utilisateur.
Apache	<p>Apache est un logiciel de serveur web gratuit et open-source qui alimente environ 46% des sites web à travers le monde. Le nom officiel est Serveur Apache HTTP et il est maintenu et développé par Apache Software Foundation.</p> <p>Un serveur Apache est un logiciel de serveur web open-source extrêmement populaire et largement utilisé dans le domaine du développement web. Il constitue la pierre angulaire des suites de développement web LAMP (Linux, Apache, MySQL, PHP), MAMP (Mac, Apache, MySQL, PHP), WAMP (Windows, Apache, MySQL, PHP) et XAMPP (Apache, MySQL, PHP, Perl).</p> <p>Il permet aux propriétaires de sites web de servir du contenu sur le web – d'où le nom « serveur web » -. C'est l'un des serveurs web les plus anciens et les plus fiables avec une première version sortie il y a plus de 20 ans, en 1995.</p>