



ANSIBLE



# Soutenance de Stage d'Initiation Professionnelle

Automatisation de la gestion d'infrastructure

Réalisé par:  
**Ayoub SAMI**

Examiné par:  
**Mme Hassania MESSOUD**

Encadré par:  
**Mr Ismail AIT ZAID**

Soutenue le 21 / 11 / 2023

# INTRODUCTION

## Les objectifs du stage SIP

- ✓ Découvrir les activités et le fonctionnement d'une entreprise;
- ✓ Compléter ses connaissances théoriques acquises au cours de sa formation par des travaux concrets et pratiques ;
- ✓ Développer le sens d'observation;
- ✓ Vivre une expérience sociotechnique enrichissante;

# PLAN DE LA PRÉSENTATION

## I. ORGANISME D'ACCUEIL

1. Présentation d'IT24
2. Les domaines de travail d'IT24
3. Les services offertes par IT24

## II. LE TRAVAIL RÉALISÉ

1. Spécification de besoin
2. La solution proposée
  - a. Ansible
  - b. Pourquoi Ansible?
3. La conception
4. La réalisation
5. La mise en œuvre

## III. CONCLUSION

## II. ORGANISME D'ACCUEIL

### 1. Présentation d'IT24

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).

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



**Logo IT24**

# I. ORGANISME D'ACCUEIL

## 2. Les domaines

Les principaux domaines de travail d'IT24:

- Réseaux informatiques
- Sécurité des réseaux
- Sécurité électronique

## 3. Les services

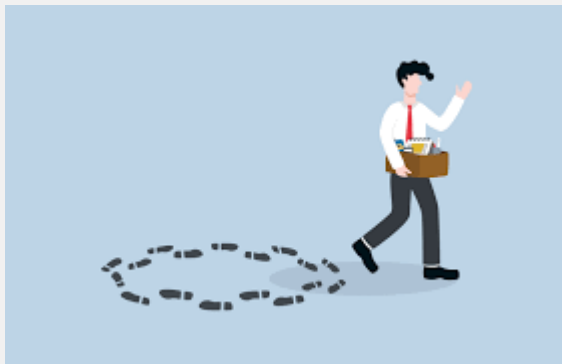
Parmi les services d'IT24:

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

## II. LE TRAVAIL RÉALISÉ

### 1. La problématique

- Chaque application était associée à un serveur unique
- Gestion des serveurs séparée
- Des tâches répétitives



Pour optimiser la gestion de l'infrastructure on doit procédé à une phase d'automatisation.

## 2. La solution

La solution proposée est l'utilisation d'un outil de gestion des configuration comme ANSIBLE qui est la tendance actuelle dans le domaine d'automatisation de gestion des configurations.

### a. Ansible

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.

Le projet Ansible est une communauté open source sponsorisée par Red Hat. C'est aussi un langage d'automatisation simple qui décrit parfaitement les environnements d'applications informatiques dans Ansible Playbooks.





ANSIBLE

## Les fonctions d'Ansible

Les logiciels Ansible Automation Platform de Red Hat assurent:

- ☐ L'automatisation du provisionnement
- ☐ L'automatisation de la configuration
- ☐ L'automatisation de la gestion
- ☐ L'automatisation du déploiement
- ☐ L'orchestration

Ansible exécute ces activités au moyen de playbooks, lesquels contiennent des instructions Ansible et un langage commun pour l'automatisation.

```
---  
- name: install and start apache  
  hosts: web  
  become: yes vars:  
    http_port: 80  
  
  tasks:  
    - name: httpd package is present  
      yum:  
        name: httpd state:  
        latest  
  
    - name: latest index.html file is present  
      copy:  
        src: files/index.html dest:  
        /var/www/html/  
  
    - name: httpd is started  
      service:  
        name: httpd  
        state: started
```





# How Ansible Automation Works

Module code is executed locally on the control node

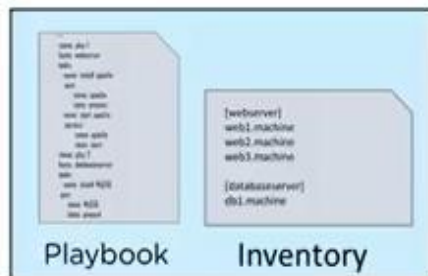


Network Devices /  
API Endpoints

Module code is copied to the managed node, executed, then removed



Linux / Windows  
Hosts



Local Machine

SSH



Node

SSH



Node

SSH



Node

## b. Pourquoi Ansible



### **SIMPLE**

Automatisation lisible par l'homme  
Aucune compétence particulière de codage requise  
Tâches exécutées dans l'ordre  
Utilisable par chaque équipe  
Soyez productif rapidement



### **POWERFUL**

Déploiement d'applications  
Gestion de la configuration  
Orchestration du flux de travail  
Automatisation du réseau

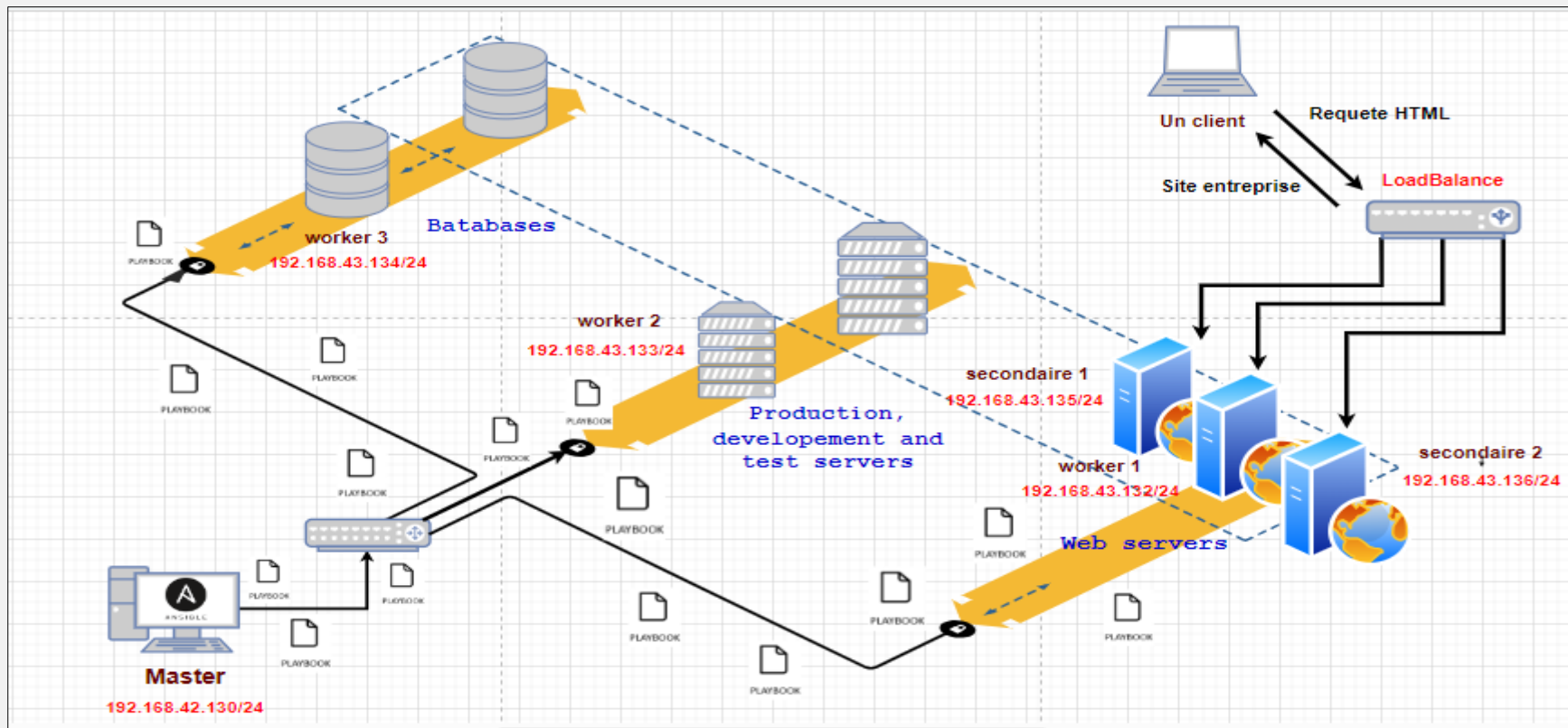


### **AGENTLESS**

Architecture sans agent  
Utilise OpenSSH et WinRM  
Aucun agent à exploiter ou à mettre à jour  
Plus efficace et plus sécurisé

### 3. La conception

L'infrastructure de future système informatique d'IT24 après l'intégration de la solution proposée:

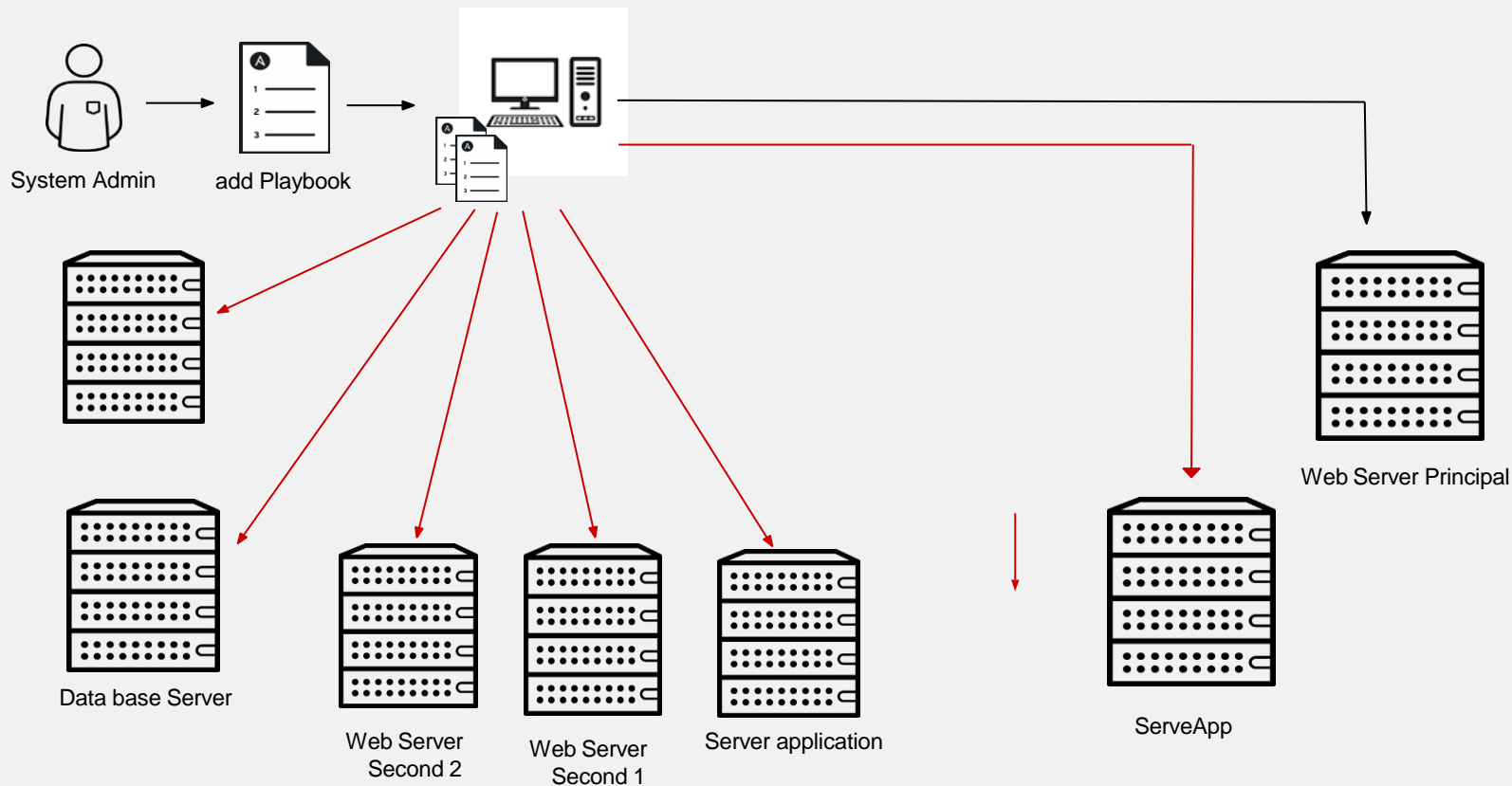


# 1. La réalisation

## a. L'infrastructure virtuelle

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

# Vue d'ensemble d'infrastructure



## b. La configuration d'Ansible

### – Fichier ansible.cfg

```
[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
~
~
```



## – 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
webserver2
webserver3

[LoadBalancer]
loadBalancer

[prod]
worker2

[database]
worker3
```

```
[webservers]
192.168.43.132
192.168.43.135
192.168.43.136

[LoadBalancer]
192.168.43.140

[prod]
192.168.43.133

[database]
192.168.43.134
```

## – Génération de la clé ssh

Les clés SSH servent à authentifier les utilisateurs sans utiliser de noms d'utilisateur ni de mots de passe. Au lieu de cela, les utilisateurs disposent d'une paire de clés SSH de confiance qui les authentifie comme la personne qu'ils disent être.

```
[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:ehqNqfMGVSl0AtvT5DAy0H6E3oa3wbFmxR7J+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]$
```

## – Copie la clé ssh vers les nœuds gérées

Pour établir la connexion SSH par clés SSH, il faut que notre clé publique soit présente dans le fichier `~/.ssh/authorized_keys` d'un utilisateur sur le serveur distant.

```
[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]$ █
```

## – Test d'accessibilité

Maintenant nous avons le contrôle total et complet sur toutes les machines sans nécessité d'accès direct physique sur chaque machine.

```
[ansible@master ~]$ ansible all -m ping -i /home/ansible/tasks/inventory
192.168.133.139 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
192.168.133.135 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
192.168.133.138 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
192.168.133.134 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
192.168.133.132 | SUCCESS => {
  "ansible_facts": {
```

## 4. La mise en ouvre

### PLAYBOOK : Déploiement du site

```
---  
- name: Deploy IT24 Website on worker 1 webserver  
  hosts: webserver  
  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
  firewalld:
    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
```

I

```
- name: Test Apache Configuration
  command: apachectl configtest
  register: apache_config_test
  become: yes
  ignore_errors: yes

- name: Restart Apache Service
  service:
    name: httpd
    state: restarted
  become: yes
  when: apache_config_test.rc == 0
```

Ce playbook permet de déployer le site web d'IT24 sur ses serveurs web.

## – Exécution de playbook de déploiement de site d'IT24

```
[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]

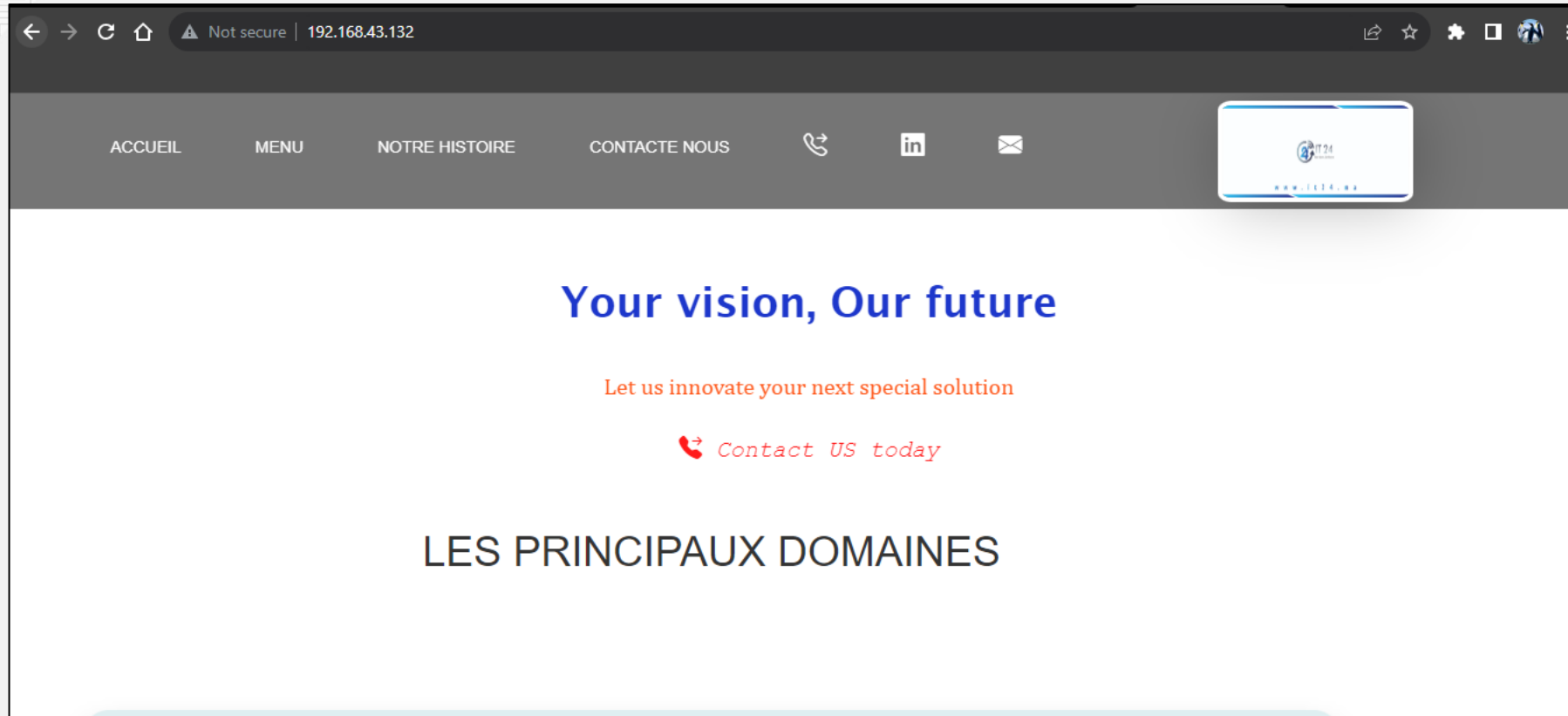
TASK [Allow HTTP and HTTPS Through Firewall] *****
ok: [worker1] => (item=http)
ok: [worker1] => (item=https)

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

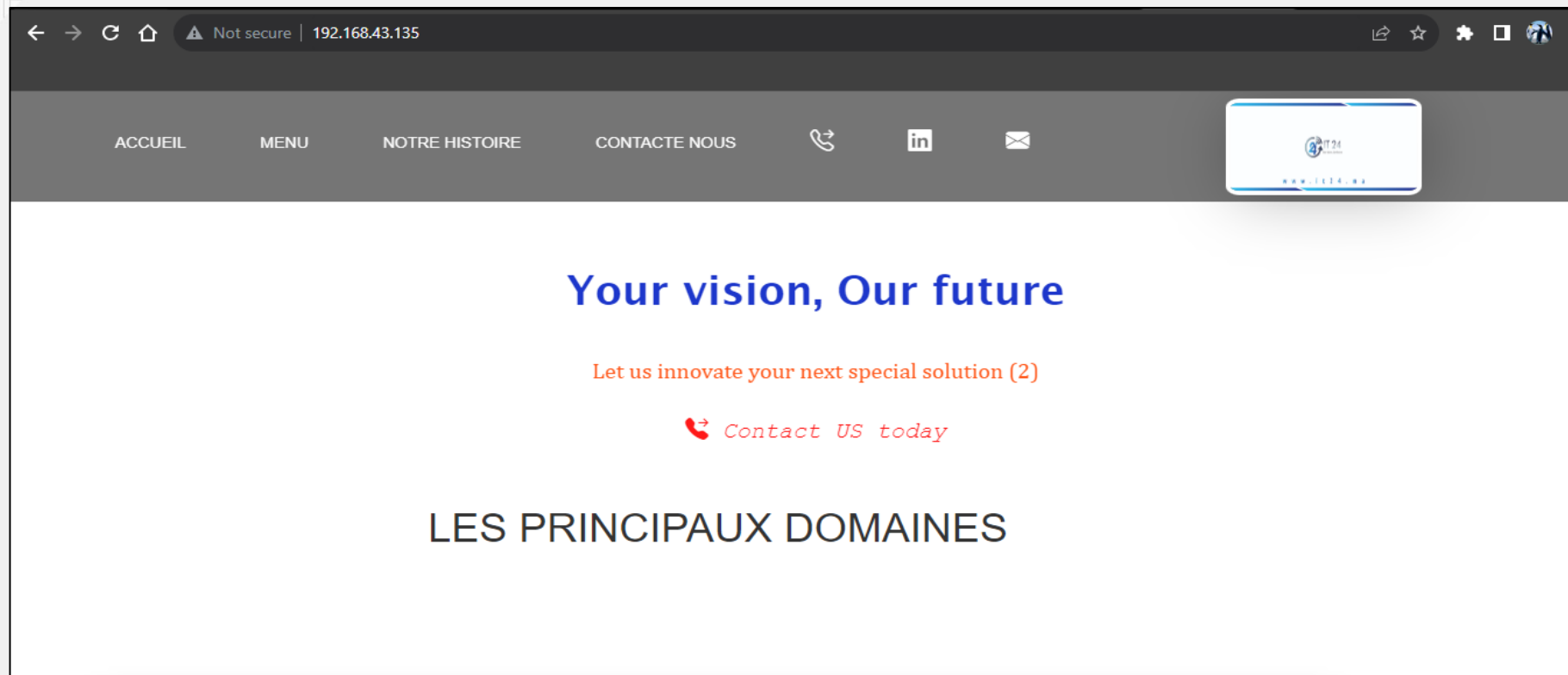
TASK [Set Apache User and Group Ownership] *****
changed: [worker1]
```



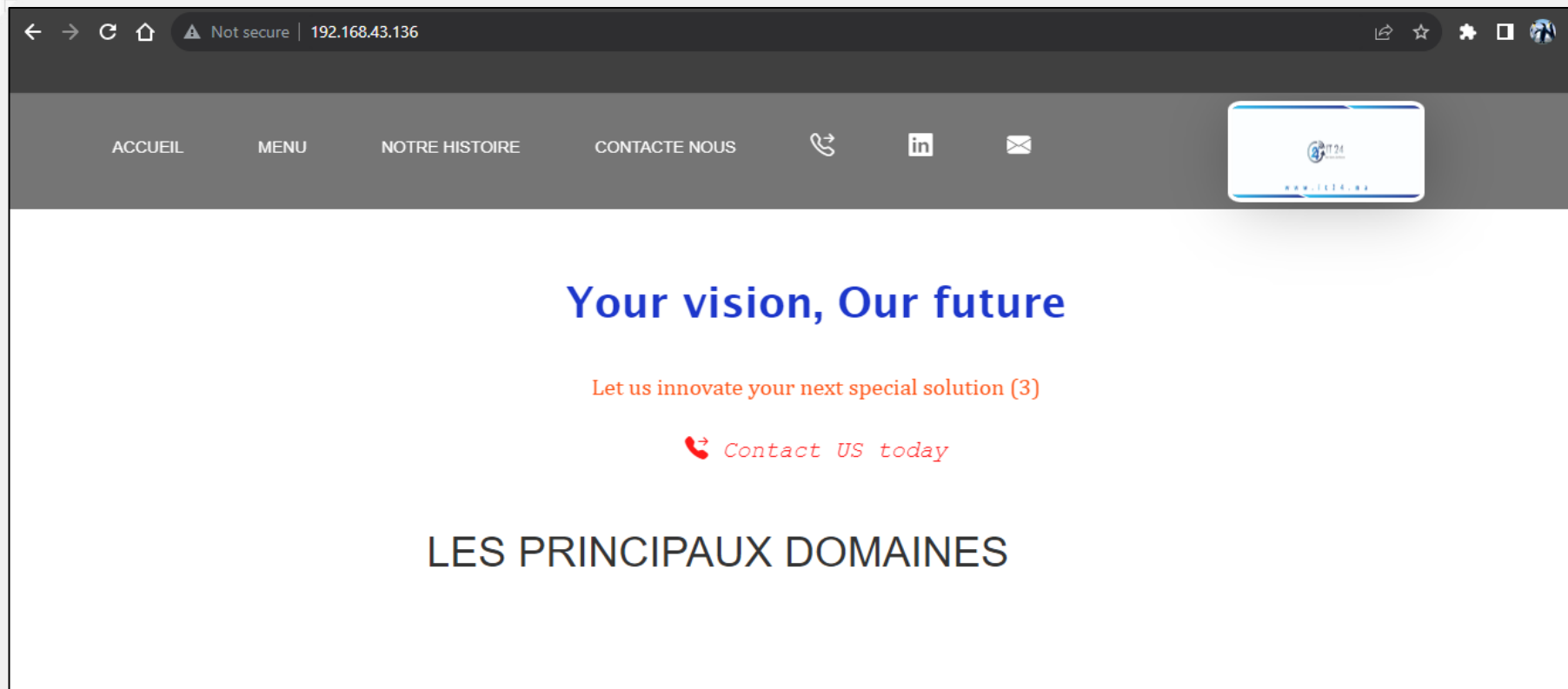
# Serveur web principal



# Serveur web secondaire 1



## Serveur web secondaire 2



## PLAYBOOK : Configuration du Load balancing

Le load balancing, ou répartition de charge, est une technologie conçue pour distribuer la charge de travail entre différents serveurs ou applications.

```
[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]$
```

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

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

- hosts: loadBalancer
  roles:
    - loadbalancer
```

## – Exécution de playbook de configuration du load balancing

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

PLAY [webserver] *****

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

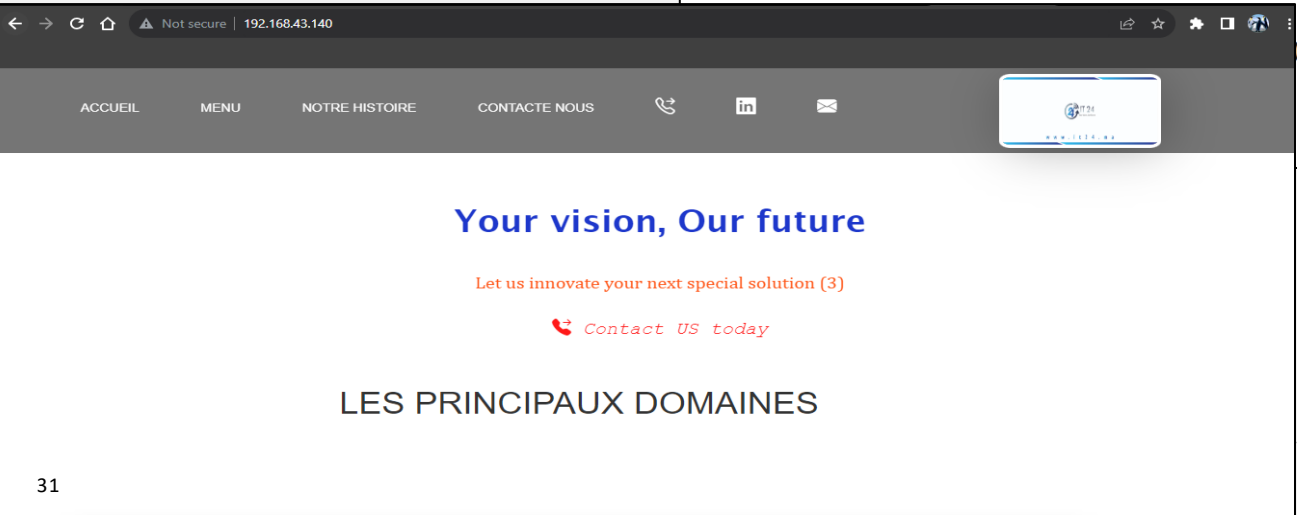
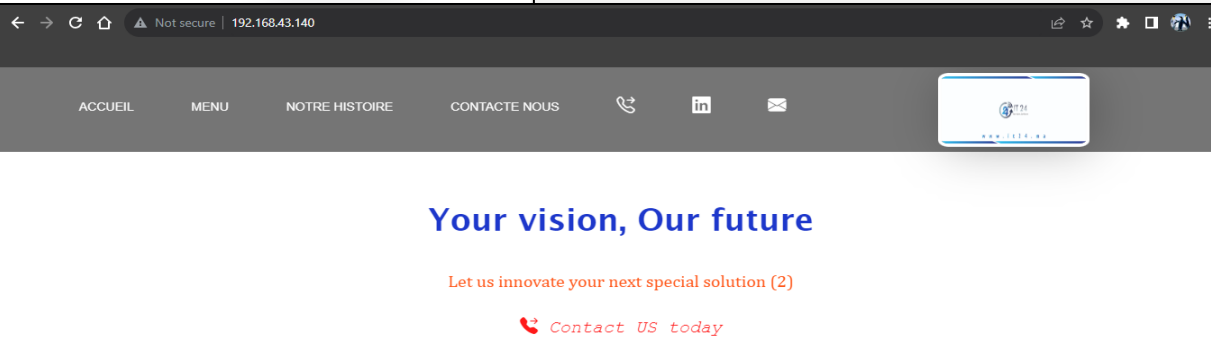
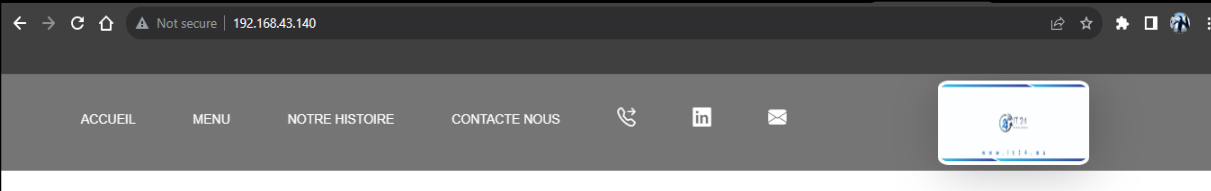
TASK [webserver : Install Httpd software] *****
ok: [worker1]
ok: [webserver3]
ok: [webserver2]

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]
```



**Voir la réalisation**



### III. CONCLUSION

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. Il nous a permis aussi d'approfondir et d'appliquer nos connaissances théoriques dans le domaine d'administration des systèmes informatiques. Enfin il été aussi une occasion d'apprendre l'utilisation la distribution Red Hat.



**MERCI POUR VOTRE ATTENTION**



# Soutenance de Stage d'Initiation Professionnelle

Automatisation de la gestion d'infrastructure

Réalisé par:  
**Ayoub SAMI**

Examiné par:  
**Mme Hassania MESSOUD**

Encadré par:  
**Mr Ismail AIT ZAID**

Soutenue le 21 / 11 / 2023

# HOW ANSIBLE WORKS

