

# Étude de cas Ansible

## **Mise au contexte :**

**GreenOps** est une société coopérative et participative (SCOP) spécialisée dans le développement d'une plateforme numérique permettant de référencer et de cartographier

les cavistes commercialisant du vin entièrement naturel. Ce modèle d'entreprise met l'accent sur la collaboration interne, le partage des connaissances et l'implication des salariés dans les décisions stratégiques.

## **Votre rôle au sein de GreenOps**

Vous occupez le poste d'ingénieur DevOps dans cette structure. Plusieurs experts DevOps travaillent déjà dans l'entreprise et échangent régulièrement des bonnes pratiques, des méthodes de travail et des outils. Cependant, ces informations circulent encore de manière informelle, ce qui peut nuire à la transmission du savoir et à la cohérence des pratiques internes.

En tant que nouveau membre de l'équipe, vous avez été désigné collégialement pour mettre en place un wiki qui servira de plateforme centralisée de documentation. Cet outil devra permettre à tous les collaborateurs de contribuer, de partager leurs connaissances et de structurer l'information de manière durable et collaborative.

Une recherche rapide vous apprend qu'un wiki est un système de gestion de contenu facilitant l'écriture collective, la mise à jour continue et l'organisation structurée des informations. Vous tenez donc votre première mission : créer un wiki opérationnel et adapté aux besoins de l'entreprise.

## **Lancement du projet**

Rapidement, plusieurs questions se posent :

- Quel logiciel de wiki faut-il choisir ?
- Comment procéder à son installation ?
- Comment déployer automatiquement cet outil et avec quelle technologie ?

Pour répondre à ces problématiques, la première étape consiste à définir des critères pertinents en tenant compte du contexte et des exigences. Après analyse, vous optez pour MediaWiki, une solution libre, extrêmement robuste, modulaire et largement éprouvée. Ce choix s'aligne parfaitement avec l'esprit de l'entreprise et son environnement technique.

## **Étude de l'installation de MediaWiki**

Pour comprendre la procédure d'installation, vous consultez la documentation officielle de MediaWiki ([https://www.mediawiki.org/wiki/Manual:Installing\\_MediaWiki/fr](https://www.mediawiki.org/wiki/Manual:Installing_MediaWiki/fr)). Vous identifiez les étapes essentielles suivantes :

1. Installer un serveur web capable de fournir les pages au navigateur.
2. Installer PHP pour permettre l'exécution de MediaWiki.

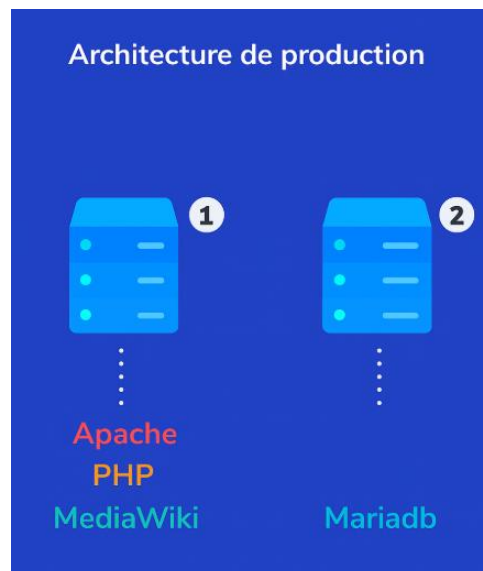
3. Mettre en place une base de données pour stocker le contenu du wiki.
4. Télécharger les sources de MediaWiki et les déployer sur le serveur web.
5. Configurer le serveur web pour accéder à MediaWiki via une URL dédiée.
6. Finaliser la configuration grâce au script d'installation fourni.

## Élaboration de l'architecture technique

Une fois l'installation comprise, vous concevez l'architecture technique de l'application. Pour assurer modularité, évolutivité et bonnes pratiques DevOps, vous décidez de séparer les services :

- Un serveur Linux (par exemple Debian) hébergeant Apache, PHP et MediaWiki.
- Un second serveur Linux hébergeant la base de données MariaDB.

Même si un seul serveur aurait pu suffire, cette séparation prépare l'infrastructure à une montée en charge future. En cas d'augmentation du trafic, il serait très simple – par exemple avec Ansible – d'ajouter de nouveaux serveurs web pour absorber la charge. Cette approche s'inscrit pleinement dans une stratégie d'architecture scalable et modulaire.



## Préparation de l'automatisation

L'installation de MediaWiki pourrait être faite manuellement, mais en tant qu'ingénieur DevOps, vous privilégiez naturellement l'automatisation. Vous appliquez le principe de l'Infrastructure as Code (IaC), qui consiste à décrire l'infrastructure et les actions d'installation via des fichiers de configuration plutôt qu'en effectuant des opérations manuelles.

Les bénéfices sont nombreux :

- installation reproductible sans erreurs,
- gains de temps pour les futurs déploiements,
- possibilité de partager les scripts avec le reste de l'équipe.

## Choix de l'outil d'automatisation

Il vous faut alors choisir un outil de gestion de configuration. Plusieurs solutions bien connues existent, notamment Chef, Puppet et Ansible. Le choix dépend de plusieurs critères :

- technologies et outils déjà utilisés ou non dans l'entreprise,
- compétences internes : Puppet et Chef utilisent Ruby, Ansible s'appuie sur Python,
- besoins principaux : conformité, automatisation du déploiement, performance, simplicité ou sécurité.

Dans votre cas, aucun outil n'est encore en place, et Ansible s'impose comme la solution la plus simple à adopter. Vous maîtrisez déjà Python, et Ansible n'impose pas d'agent à déployer sur les serveurs. Vous envisagez même qu'il puisse être utilisé à terme pour administrer l'ensemble de l'infrastructure de l'entreprise.

## Ce que vous allez automatiser

L'équipe système se charge de préparer l'infrastructure de base : installation de Linux sur les deux serveurs et configuration réseau. De votre côté, vous recevez un accès root direct à chacun des serveurs.

Vous devrez alors automatiser à l'aide d'Ansible les étapes suivantes :

1. Installer Apache sur le serveur web.
2. Installer PHP sur ce même serveur.
3. Installer MariaDB sur le serveur de base de données.
4. Télécharger les fichiers sources de MediaWiki et les placer sur le serveur Apache.
5. Configurer Apache pour rendre le wiki accessible via l'adresse <http://http1/mediawiki>.
6. Finaliser la mise en service de MediaWiki à l'aide du script d'installation.

**L'objectif est clair :** rendre possible le déploiement complet de MediaWiki en une seule exécution automatisée, fiable et réutilisable.

### Livrables :

Une documentation détaillée de la procédure ou la méthodologie utilisée pour mettre en place l'automatisation de l'installation de Wikimedia (incluant Apache, PHP, MariaDB et les 2 serveurs).

Lister l'ensemble des erreurs et des problèmes rencontrés pour pouvoir les éviter dans un processus futur de déploiement.

---

# Mise en place automatisée de MediaWiki avec Ansible : procédure détaillée et analyse des problèmes de déploiement

## 1. Introduction

Dans le cadre de ce projet, il faut mettre en place un wiki interne pour l'entreprise GreenOps afin de centraliser la documentation et les bonnes pratiques DevOps. L'outil choisi est MediaWiki, car c'est un logiciel libre, utilisé par Wikipedia, robuste et bien documenté. L'objectif principal est d'automatiser l'installation de MediaWiki et de tous les composants nécessaires (Apache, PHP, MariaDB) sur deux serveurs Debian 12 en utilisant Ansible, afin d'appliquer le principe d'Infrastructure as Code et de rendre le déploiement reproductible.

## Architecture technique

Pour respecter les bonnes pratiques et préparer une montée en charge future, l'architecture est séparée en deux serveurs :

### Serveur web (http1)

Système : Debian 12

Rôle : héberger Apache, PHP et les fichiers de MediaWiki.

Accès : via l'URL <http://http1/mediawiki>.

### Serveur base de données (db1)

Système : Debian 12

Rôle : héberger MariaDB et la base de données MediaWiki.

### Machine de contrôle Ansible

Système : Debian

Rôle : exécuter Ansible pour configurer automatiquement les deux serveurs.

Cette séparation permet d'isoler la base de données du serveur web, d'améliorer la sécurité et de faciliter l'ajout de nouveaux serveurs web si le trafic augmente.

## 2. Préparer l'environnement Ansible

Étapes à faire une fois, sur ta machine de contrôle (où tu lances Ansible) :

Installer Ansible (sur Debian :

```
apt install ansible
```

```

root@ansible:~/mediawiki-ansible# nano inventory.ini
root@ansible:~/mediawiki-ansible# apt install ansible
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  ansible-core ieee-data python3-babel-localedata python3-anyio
  python3-argcomplete python3-babel python3-click python3-colorama
  python3-distutils python3-dnspython python3-h11 python3-h2 python3-hpack
  python3-httpcore python3-httpx python3-hyperframe python3-jinja2
  python3-jmespath python3-kerberos python3-lib2to3 python3-libcloud
  python3-lockfile python3-markdown-it python3-mdurl python3-netaddr
  python3-ntlm-auth python3-packaging python3-passlib python3-pygments
  python3-requests-kerberos python3-requests-ntlm python3-requests-toolbelt
  python3-resolvelib python3-rfc3986 python3-rich python3-selinux
  python3-simplejson python3-sniffio python3-tz python3-winrm
  python3-xmltodict python3-yaml
Paquets suggérés :
  cowsay sshpass python3-trio python3-aiouic python3-jinja2-doc
  python-lockfile-doc ipython3 python-netaddr-docs python-pygments-doc
  ttf-bitstream-vera
Les NOUVEAUX paquets suivants seront installés :
  ansible ansible-core ieee-data python-babel-localedata python3-anyio
  python3-argcomplete python3-babel python3-click python3-colorama
  python3-distutils python3-dnspython python3-h11 python3-h2 python3-hpack
  python3-httpcore python3-httpx python3-hyperframe python3-jinja2
  python3-jmespath python3-kerberos python3-lib2to3 python3-libcloud
  python3-lockfile python3-markdown-it python3-mdurl python3-netaddr
  python3-ntlm-auth python3-packaging python3-passlib python3-pygments
  python3-requests-kerberos python3-requests-ntlm python3-requests-toolbelt
  python3-resolvelib python3-rfc3986 python3-rich python3-selinux
  python3-simplejson python3-sniffio python3-tz python3-winrm
  python3-xmltodict python3-yaml
0 mis à jour, 43 nouvellement installés, 0 à enlever et 0 non mis à jour.

```

## 1. Création du répertoire Ansible et inventaire

la création du répertoire **/root/mediawiki-ansible** et du fichier `inventory.ini` sur le serveur Ansible, ce qui prépare l'environnement pour automatiser le déploiement de MediaWiki

```

root@ansible:~# mkdir -p /root/mediawiki-ansible
root@ansible:~# cd /root/mediawiki-ansible
root@ansible:~/mediawiki-ansible#
root@ansible:~/mediawiki-ansible# nano inventory.ini

```

**Le fichier `inventory.ini`** définit deux groupes d'hôtes, `[web]` et `[db]`, correspondant respectivement aux serveurs `srv-web-wiki` et `srv-db-wiki`, avec les paramètres SSH (utilisateur, mot de passe et options) nécessaires pour permettre à Ansible de s'y connecter automatiquement.

`[web]`

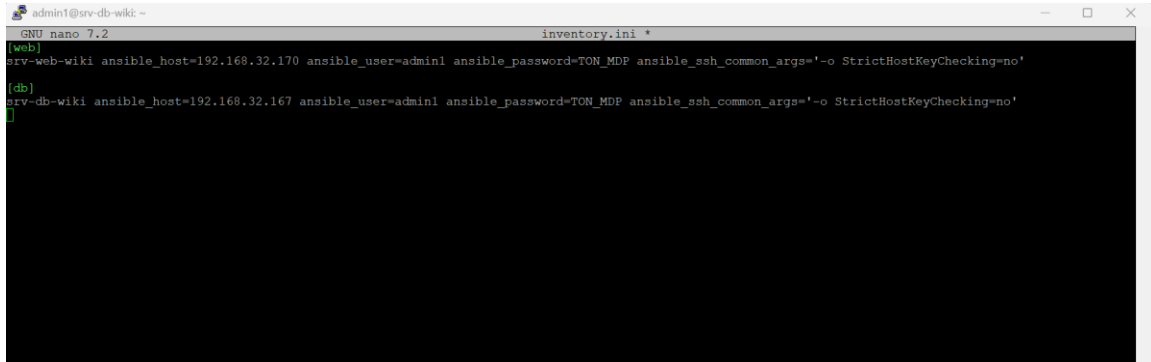
```

srv-web-wiki ansible_host=192.168.32.170 ansible_user=admin1
ansible_password=TON_MDP ansible_ssh_common_args='-o
StrictHostKeyChecking=no'

```

[db]

```
srv-db-wiki ansible_host=192.168.32.167 ansible_user=admin1
ansible_password=TON_MDP ansible_ssh_common_args='-o
StrictHostKeyChecking=no'
```



```
admin1@srv-db-wiki: ~
GNU nano 7.2 inventory.ini *
[web]
srv-web-wiki ansible_host=192.168.32.170 ansible_user=admin1 ansible_password=TON_MDP ansible_ssh_common_args='-o StrictHostKeyChecking=no'

[db]
srv-db-wiki ansible_host=192.168.32.167 ansible_user=admin1 ansible_password=TON_MDP ansible_ssh_common_args='-o StrictHostKeyChecking=no'


```

## Tester la connexion :

### Installation de sshpass :

L'installation du paquet `sshpass` après un `apt update`, afin de permettre à Ansible d'utiliser l'authentification par mot de passe sur les hôtes distants lorsque les clés SSH ne sont pas configurées.

```
root@ansible:~/mediawiki-ansible# apt update
Atteint :1 http://security.debian.org/debian-security bookworm-security InRelease
Atteint :2 http://mirror.csclub.uwaterloo.ca/debian bookworm InRelease
Atteint :3 http://mirror.csclub.uwaterloo.ca/debian bookworm-updates InRelease
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Tous les paquets sont à jour.
root@ansible:~/mediawiki-ansible# apt install -y sshpass
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les NOUVEAUX paquets suivants seront installés :
  sshpass
0 mis à jour, 1 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 13,0 ko dans les archives.
Après cette opération, 38,9 ko d'espace disque supplémentaires seront utilisés.
Réception de :1 http://mirror.csclub.uwaterloo.ca/debian bookworm/main amd64 sshpass amd64 1.09-1+b1 [13,0 kB]
13,0 ko réceptionnés en 0s (81,5 ko/s)
Sélection du paquet sshpass précédemment désélectionné.
(Lecture de la base de données... 178774 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../sshpass_1.09-1+b1_amd64.deb ...
Dépaquetage de sshpass (1.09-1+b1) ...
Paramétrage de sshpass (1.09-1+b1) ...
Traitement des actions différées (« triggers ») pour man-db (2.11.2-2) ...
```



## Test Ansible avec le module ping

`ansible all -i inventory.ini -m ping`

La commande `ansible all -i inventory.ini -m ping`, qui répond pong pour `srv-web-wiki` et `srv-db-wiki`, confirmant que le serveur Ansible peut se connecter

```

# Traitement des actions différées (« triggers ») pour man-db (2.11.2-2) ...
root@ansible:~/mediawiki-ansible# ansible all -i inventory.ini -m ping
srv-web-wiki | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
srv-db-wiki | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
root@ansible:~/mediawiki-ansible#
```

## 2. Playbook mediawiki.yml (MariaDB + Apache/PHP + MediaWiki)

Toujours dans `/root/mediawiki-ansible` :

`bash`

`nano playbook_mediawiki.yml`

---

---

**# Playbook unique pour MariaDB + Apache/PHP + MediaWiki**

**- name: Préparation du serveur de base de données (MariaDB)**

**hosts: db**

**become: yes**

**tasks:**

**- name: Installer MariaDB**

**apt:**

**name: mariadb-server**

**state: present**

**update\_cache: yes**

**- name: S'assurer que MariaDB est démarré et activé**  
**service:**

**name: mariadb**

**state: started**

**enabled: yes**

**- name: Installer les modules Python pour MariaDB**

**apt:**

**name:**

**- python3-pymysql**

**state: present**

**- name: Créer la base de données MediaWiki**

**community.mysql.mysql\_db:**

**name: mediawiki**

**state: present**

**login\_user: wikiuser**

**login\_password: "123456\*"**

**- name: Créer l'utilisateur dédié pour MediaWiki**

**community.mysql.mysql\_user:**

**name: wikiuser**

**password: "123456\*"**

**host: "%" "**

**priv: "mediawiki.\*:ALL"**

**state: present**

**login\_password: "123456\*"**

**- name: Préparation du serveur web (Apache/PHP/MediaWiki)**

**hosts: web**

**become: yes**

**tasks:**

**- name: Installer Apache et PHP**

**apt:**

**name:**

**- apache2**

**- php**

**- php-mysql**

**- php-xml**

**- php-intl**

**- php-mbstring**

**- php-apcu**

**- php-gd**

**state: present**

**update\_cache: yes**

**- name: S'assurer qu'Apache est démarré et activé**

**service:**

**name: apache2**  
**state: started**  
**enabled: yes**

**- name: Télécharger MediaWiki**

**get\_url:**  
**url: https://releases.wikimedia.org/mediawiki/1.41/mediawiki-1.41.1.tar.gz**  
**dest: /tmp/mediawiki.tar.gz**

**- name: Extraire MediaWiki**

**unarchive:**  
**src: /tmp/mediawiki.tar.gz**  
**dest: /var/www/**  
**remote\_src: yes**

**- name: Renommer le dossier MediaWiki**

**command: mv /var/www/mediawiki-1.41.1 /var/www/mediawiki**  
**args:**  
**creates: /var/www/mediawiki**

**- name: Créer le vhost pour MediaWiki**

**copy:**  
**dest: /etc/apache2/sites-available/mediawiki.conf**  
**content: |**  
**<VirtualHost \*:80>**  
**ServerName srv-web-wiki**  
**DocumentRoot /var/www**  
**Alias /mediawiki /var/www/mediawiki**  
  
**<Directory /var/www/mediawiki>**  
**Options FollowSymLinks**  
**AllowOverride All**  
**Require all granted**  
**</Directory>**  
**</VirtualHost>**

**- name: Activer le site MediaWiki**

**command: a2ensite mediawiki.conf**

**- name: Activer mod\_rewrite**

**command: a2enmod rewrite**

**- name: Recharger Apache**

**service:**  
**name: apache2**  
**state: restarted**

---

```
admin1@srv-db-wiki: ~
GNU nano 7.2                                playbook mediawiki.yml *
# Playbook unique pour MariaDB + Apache/PHP + MediaWiki
- name: Préparation du serveur de base de données (MariaDB)
  hosts: db
  become: yes

  tasks:
    - name: Installer MariaDB
      apt:
        name: mariadb-server
        state: present
        update_cache: yes

    - name: S'assurer que MariaDB est démarré et activé
      service:
        name: mariadb
        state: started
        enabled: yes

    - name: Installer les modules Python pour MariaDB
      apt:
        name:
          - python3-pymysql
        state: present

    - name: Créer la base de données MediaWiki
      community.mysql.mysql_db:
        name: mediawiki
        state: present
```

### 3. Finalisation MediaWiki (script d'installation)

La sécurisation du compte root de MariaDB sur srv-db-wiki. Après connexion au client mariadb, le mot de passe du compte root@localhost est défini avec la commande ALTER USER, puis les droits sont rechargés avec FLUSH PRIVILEGES, ce qui protège l'accès à la base de données avant l'installation de MediaWiki.

```
root@srv-db-wiki:~# mariadb -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 33
Server version: 10.11.14-MariaDB-0+deb12u2 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> ALTER USER 'root'@'localhost' IDENTIFIED BY '123456*'
-> ;
**Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> EXIT;
Bye
root@srv-db-wiki:~#
```

### 4. Vérifier MediaWiki et Apache sur srv-web-wiki

La configuration du VirtualHost Apache pour MediaWiki sur srv-web-wiki :

le répertoire /var/www/mediawiki qui contient les fichiers de l'application, ainsi que le fichier mediawiki.conf dans /etc/apache2/sites-available. Le VirtualHost définit le DocumentRoot vers /var/www et un alias /mediawiki pointant vers /var/www/mediawiki, avec un bloc <Directory> qui autorise l'accès au contenu du wiki.

```
root@srv-web-wiki:~# ls -ld /var/www/mediawiki
drwxr-xr-x 14 root root 4096 22 déc 19:12 /var/www/mediawiki
root@srv-web-wiki:~# ls /var/www
html mediawiki
root@srv-web-wiki:~# ls /etc/apache2/sites-available
000-default.conf default-ssl.conf mediawiki.conf
root@srv-web-wiki:~# ls /etc/apache2/sites-enabled
000-default.conf mediawiki.conf
root@srv-web-wiki:~# cat /etc/apache2/sites-available/mediawiki.conf
<VirtualHost *:80>
    ServerName srv-web-wiki
    DocumentRoot /var/www
    Alias /mediawiki /var/www/mediawiki

    <Directory /var/www/mediawiki>
        Options FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

## La vérification et l'état du service Apache2 sur srv-web-wiki

Avant de lancer l'installation de MediaWiki, la commande `apachectl configtest` confirme que la configuration d'Apache est valide, puis `systemctl status apache2` montre que le service est actif (active (running)), activé au démarrage (enabled) et qu'il écoute correctement les requêtes HTTP.

```
root@srv-web-wiki:~# apachectl configtest
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress
this message
Syntax OK
root@srv-web-wiki:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-12-22 19:12:10 EST; 4min 45s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 13079 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 13085 (apache2)
      Tasks: 6 (limit: 2257)
     Memory: 15.9M
        CPU: 99ms
    CGroup: /system.slice/apache2.service
            └─13085 /usr/sbin/apache2 -k start
              └─13087 /usr/sbin/apache2 -k start
                └─13088 /usr/sbin/apache2 -k start
                  └─13089 /usr/sbin/apache2 -k start
                    └─13090 /usr/sbin/apache2 -k start
                      └─13091 /usr/sbin/apache2 -k start

déc 22 19:12:10 srv-web-wiki systemd[1]: Starting apache2.service - The Apache HTTP Server...
déc 22 19:12:10 srv-web-wiki apachectl[13084]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message
déc 22 19:12:10 srv-web-wiki systemd[1]: Started apache2.service - The Apache HTTP Server.
```

## 5. L'exécution complète du playbook Ansible

l'exécution du playbook `playbook_mediawiki.yml` depuis le serveur Ansible. Dans un premier temps, les tâches de préparation du serveur de base de données (installation et démarrage de

MariaDB, création de la base mediawiki et de l'utilisateur dédié) puis du serveur web (installation d'Apache/PHP et téléchargement de MediaWiki) s'enchaînent avec le statut ok ou changed.

```
/
root@ansible:~/mediawiki-ansible# nano playbook_mediawiki.yml
root@ansible:~/mediawiki-ansible# ansible-playbook -i inventory.ini playbook_mediawiki.yml

PLAY [Préparation du serveur de base de données (MariaDB)] *****

TASK [Gathering Facts] *****
ok: [srv-db-wiki]

TASK [Installer MariaDB] *****
ok: [srv-db-wiki]

TASK [S'assurer que MariaDB est démarré et activé] *****
ok: [srv-db-wiki]

TASK [Installer les modules Python pour MariaDB] *****
ok: [srv-db-wiki]

TASK [Créer la base de données MediaWiki] *****
ok: [srv-db-wiki]

TASK [Créer l'utilisateur dédié pour MediaWiki] *****
ok: [srv-db-wiki]

PLAY [Préparation du serveur web (Apache/PHP/MediaWiki)] *****

TASK [Gathering Facts] *****
ok: [srv-web-wiki]

TASK [Installer Apache et PHP] *****
ok: [srv-web-wiki]

TASK [S'assurer qu'Apache est démarré et activé] *****
ok: [srv-web-wiki]

TASK [Télécharger MediaWiki] *****
ok: [srv-web-wiki]
```

```

TASK [S'assurer qu'Apache est démarré et activé] *****
ok: [srv-web-wiki]

TASK [Télécharger MediaWiki] *****
ok: [srv-web-wiki]

TASK [Extraire MediaWiki] *****
changed: [srv-web-wiki]

TASK [Renommer le dossier MediaWiki] *****
ok: [srv-web-wiki]

TASK [Créer le vhost pour MediaWiki] *****
ok: [srv-web-wiki]

TASK [Activer le site MediaWiki] *****
changed: [srv-web-wiki]

TASK [Activer mod_rewrite] *****
changed: [srv-web-wiki]

TASK [Recharger Apache] *****
changed: [srv-web-wiki]

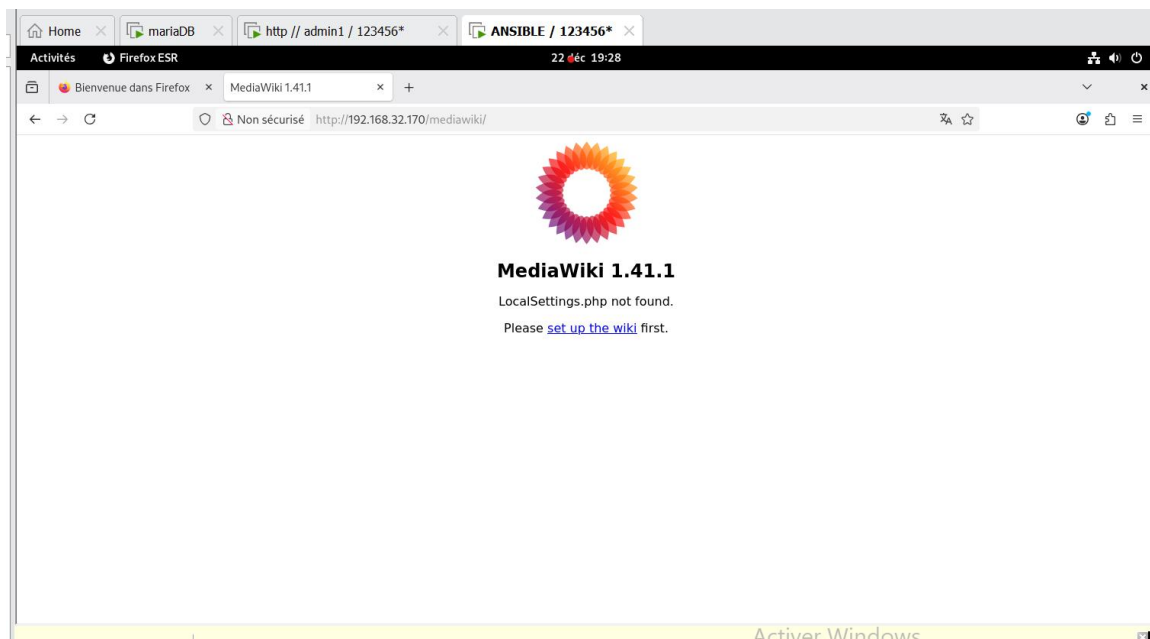
PLAY RECAP *****
srv-db-wiki      : ok=6    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
srv-web-wiki     : ok=10   changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

root@ansible:~/mediawiki-ansible# ls ~/Téléchargements/LocalSettings.php
ls: impossible d'accéder à '/root/Téléchargements/LocalSettings.php': Aucun fichier ou dossier de ce type
root@ansible:~/mediawiki-ansible#

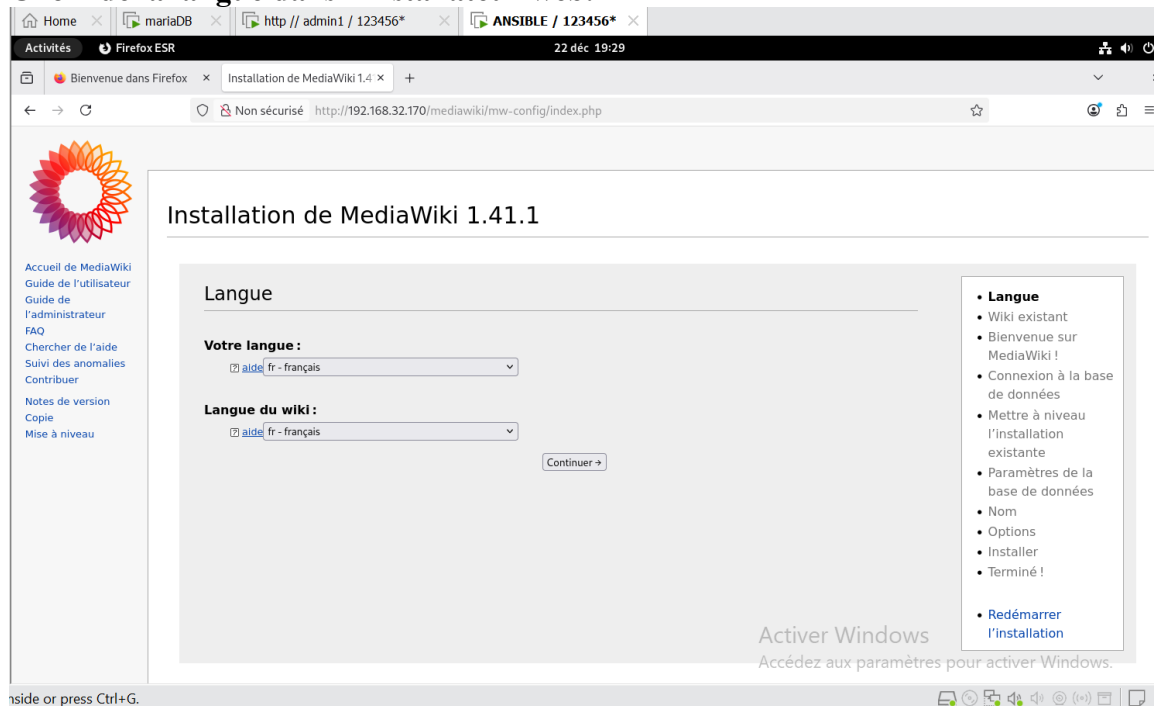
```

## 6. Déploiement et configuration de MediaWiki via l'interface web

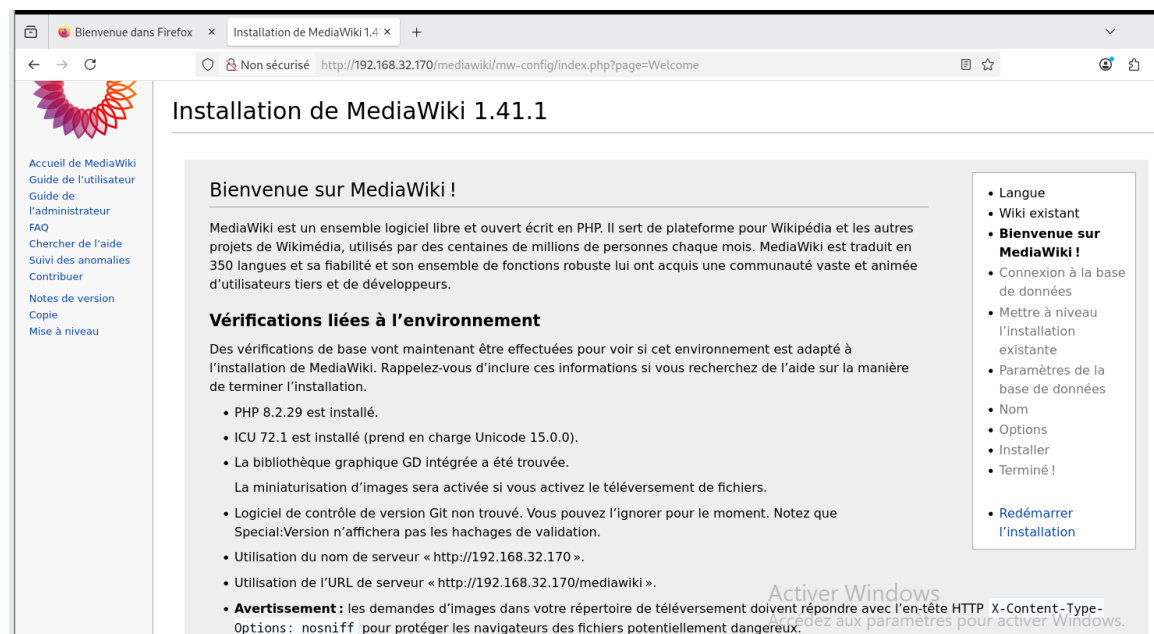
Affichage initial de MediaWiki après déploiement Ansible (LocalSettings.php introuvable).



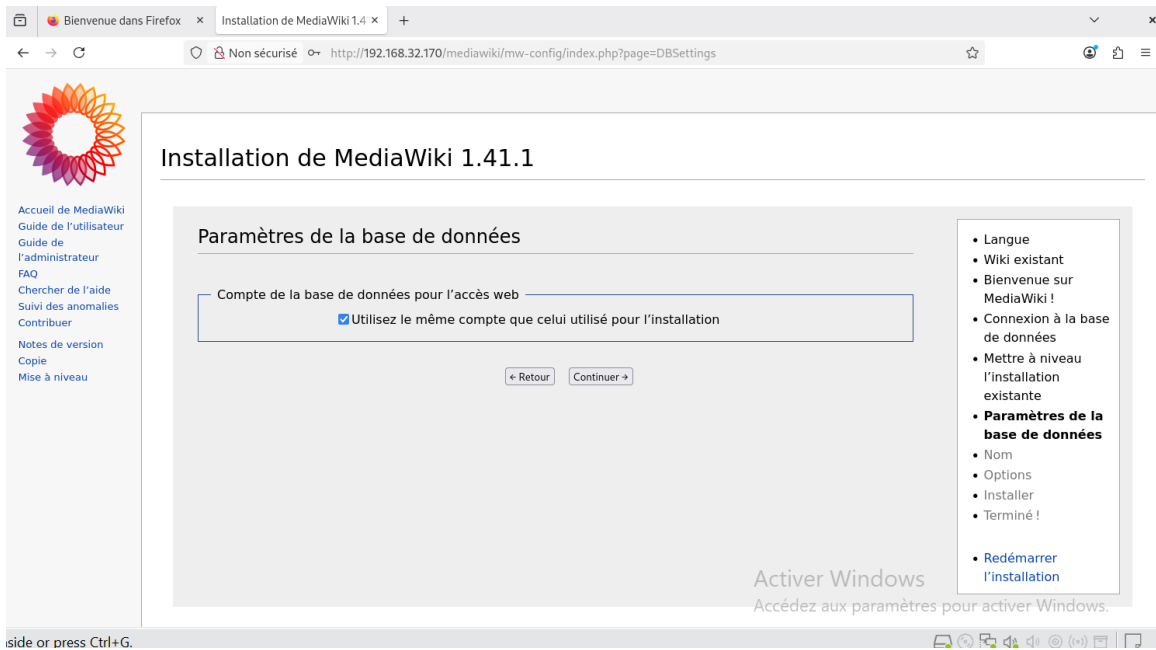
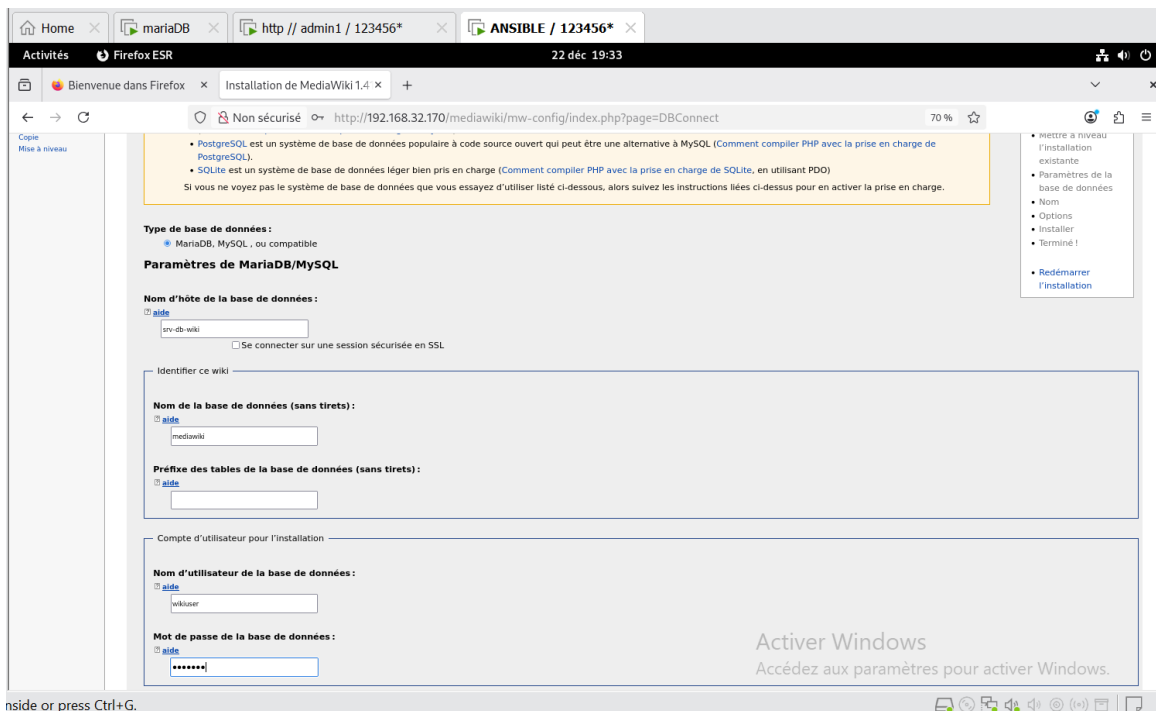
## Choix de la langue dans l'installateur web.

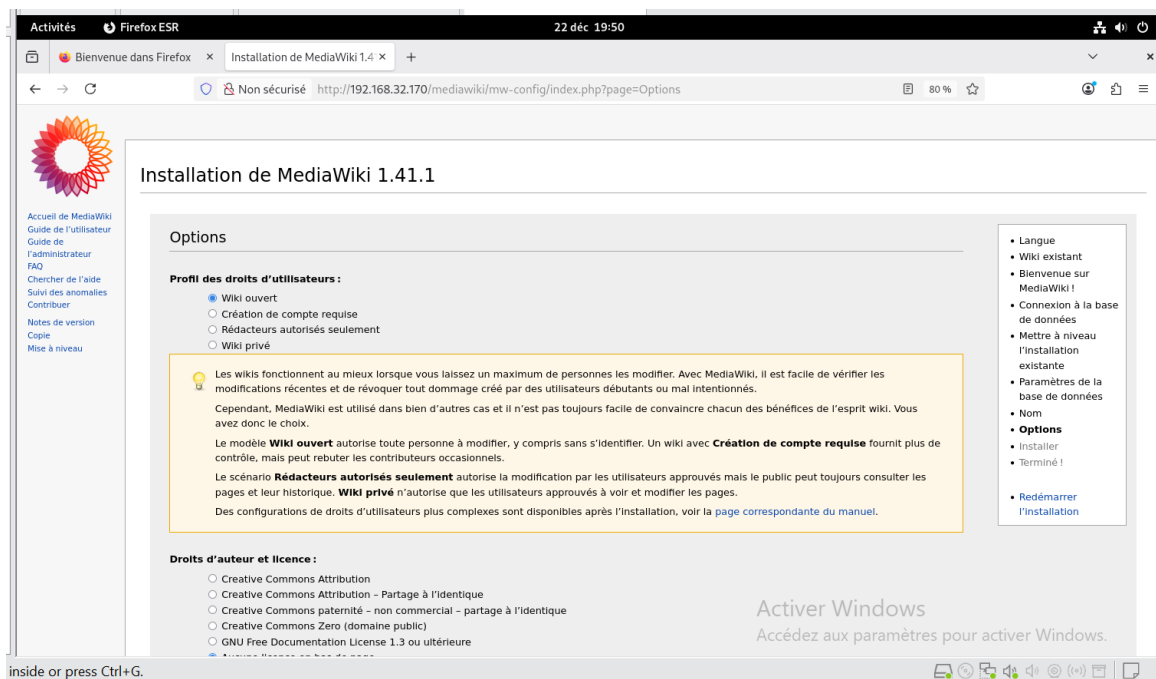
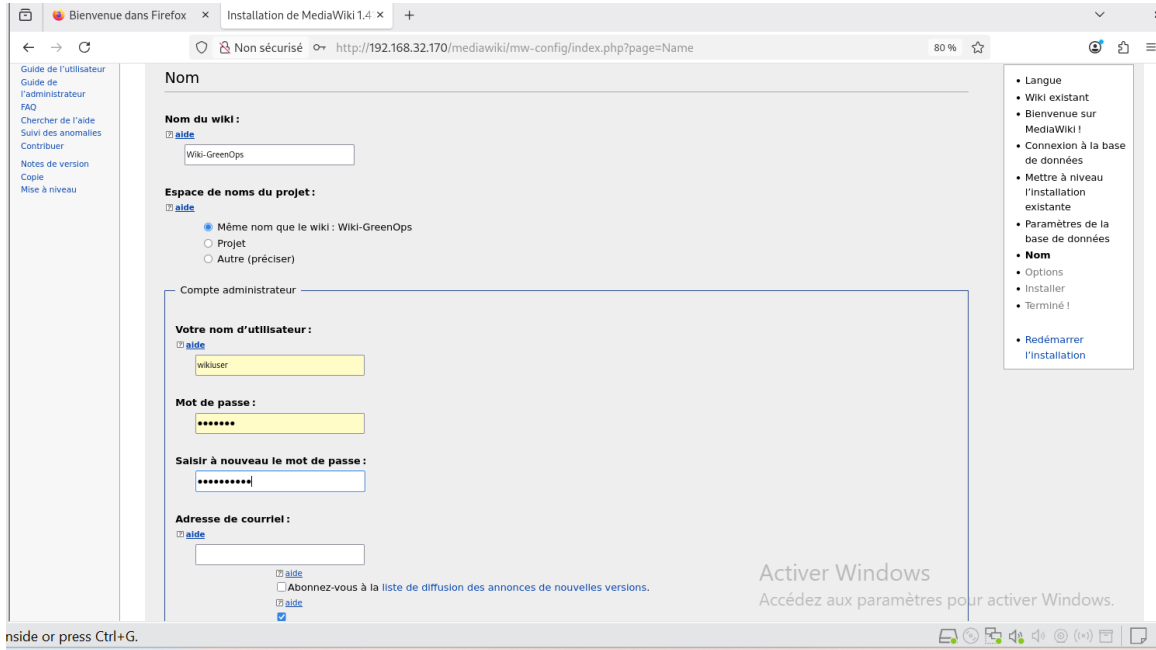


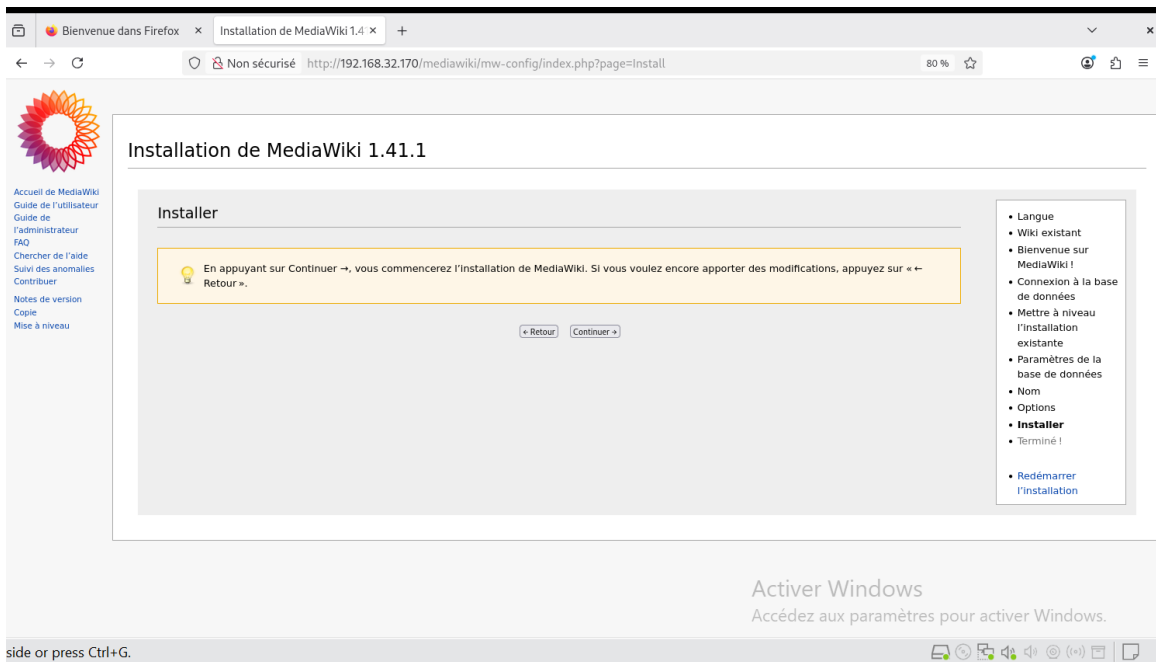
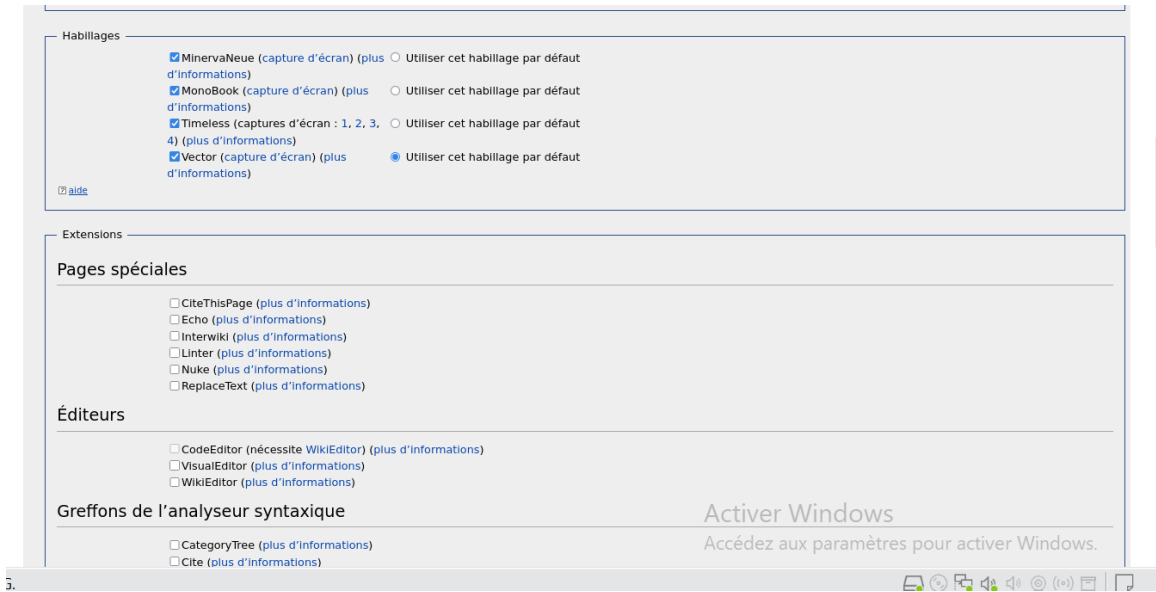
## Configuration de la base de données MariaDB.



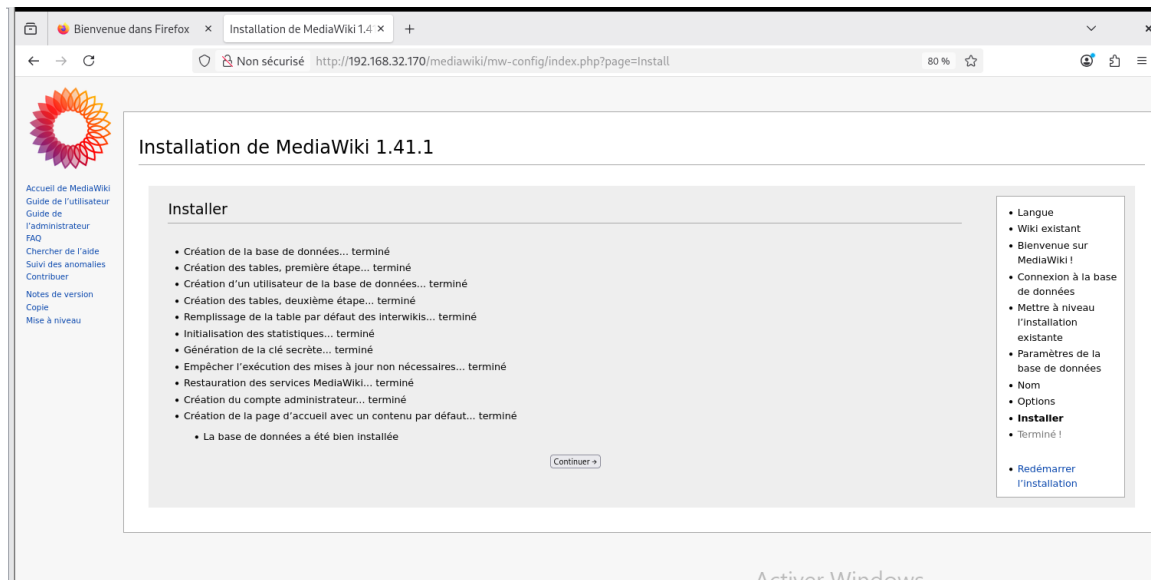






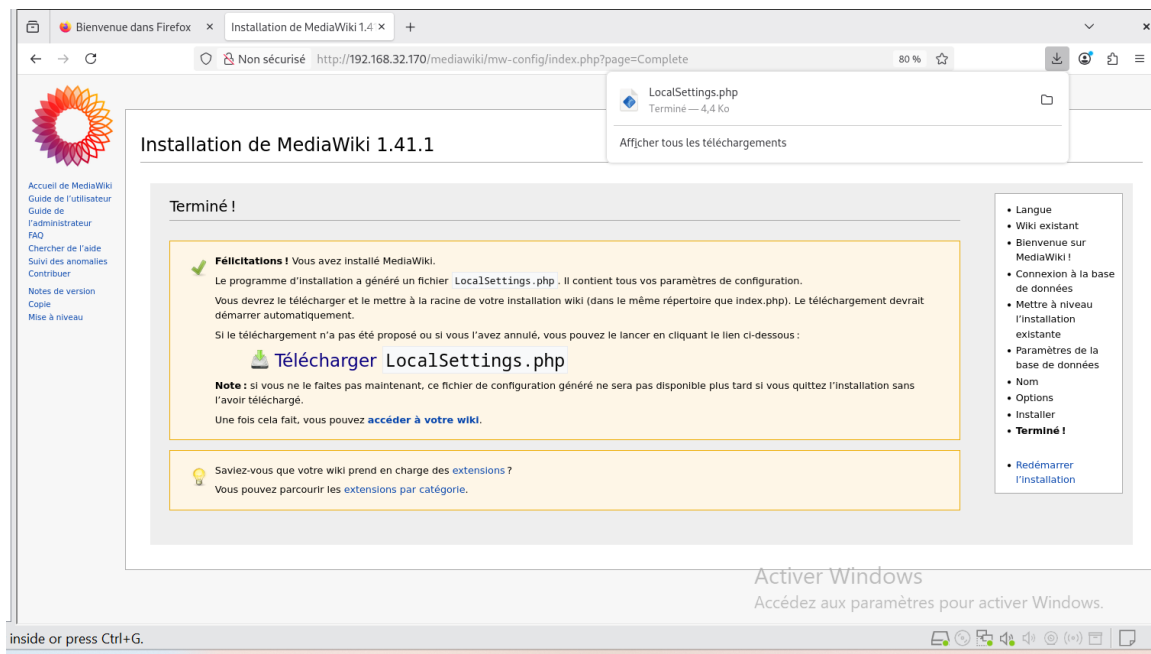


## Étapes d'installation terminées.

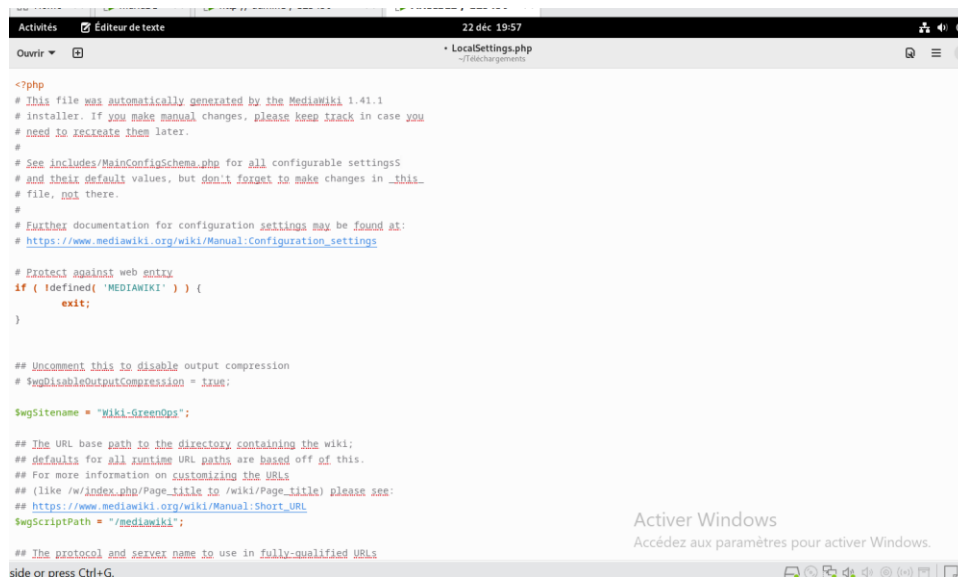


## Intégration du fichier LocalSettings.php sur le serveur web

La dernière étape de l'installateur MediaWiki, qui génère le fichier LocalSettings.php et le propose au téléchargement. Ce fichier contient toute la configuration du wiki (nom du site, base de données, langue, habillages, etc.).



C'est le fichier `LocalSettings.php` avant son transfert vers le serveur `srv-web-wiki`.



```
<?php
# This file was automatically generated by the MediaWiki 1.41.1
# installer. If you make manual changes, please keep track in case you
# need to recreate them later.
#
# See includes/MainConfigSchema.php for all configurable settings
# and their default values, but don't forget to make changes in this
# file, not there.
#
# Further documentation for configuration settings may be found at:
# https://www.mediawiki.org/wiki/Manual:Configuration_settings

# Protect against web entry
if ( !defined( 'MEDIAWIKI' ) ) {
    exit;
}

## Uncomment this to disable output compression
# $wgDisableOutputCompression = true;

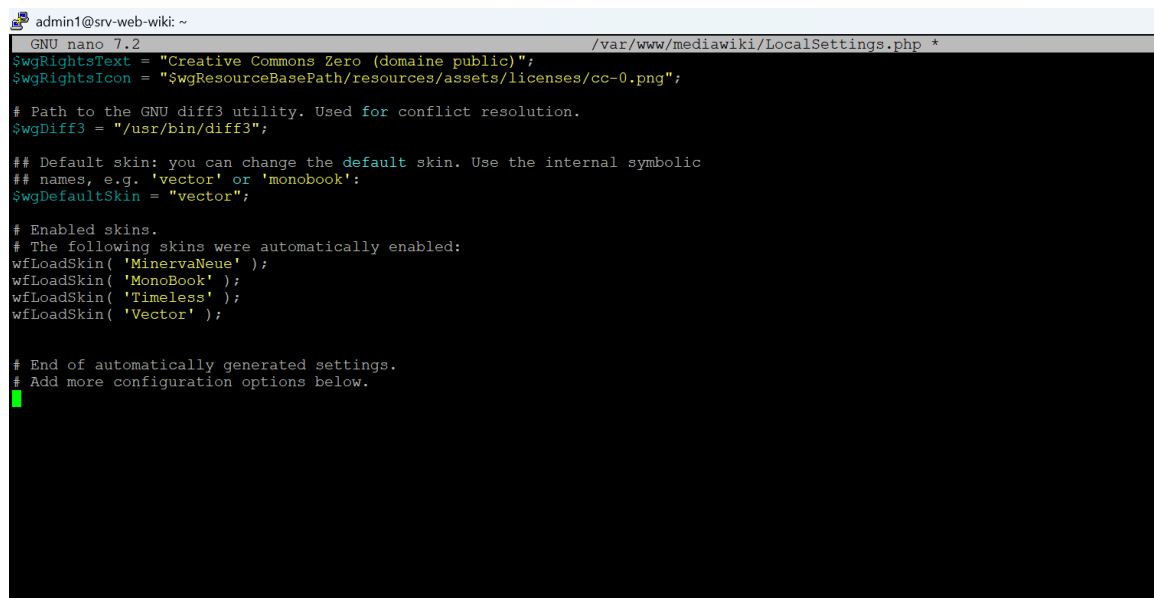
$wgSiteName = "Wiki-GreenOps";

## The URL base path to the directory containing the wiki;
## defaults for all runtime URL paths are based off of this.
## For more information on customizing the URLs
## (like /w/index.php/Page_title to /wiki/Page_title) please see:
## https://www.mediawiki.org/wiki/Manual:Short_URL
$wgScriptPath = "/mediawiki";

## The protocol and server name to use in fully-qualified URLs
side or press Ctrl+G.
```

**Sur la machine serveur web : coller le contenu de LocalSettings.php dans `/var/www/mediawiki`**

**le fichier `/var/www/mediawiki/LocalSettings.php` édité directement sur `srv-web-wiki`, confirmant que le contenu a bien été copié sur le serveur web et que les habillages souhaités (Vector, MinervaNeue, Timeless, MonoBook) sont activés.**



```
GNU nano 7.2 /var/www/mediawiki/LocalSettings.php *
$wgRightsText = "Creative Commons Zero (domaine public)";
$wgRightsIcon = "$wgResourceBasePath/resources/assets/licenses/cc-0.png";

# Path to the GNU diff3 utility. Used for conflict resolution.
$wgDiff3 = "/usr/bin/diff3";

## Default skin: you can change the default skin. Use the internal symbolic
## names, e.g. 'vector' or 'monobook':
$wgDefaultSkin = "vector";

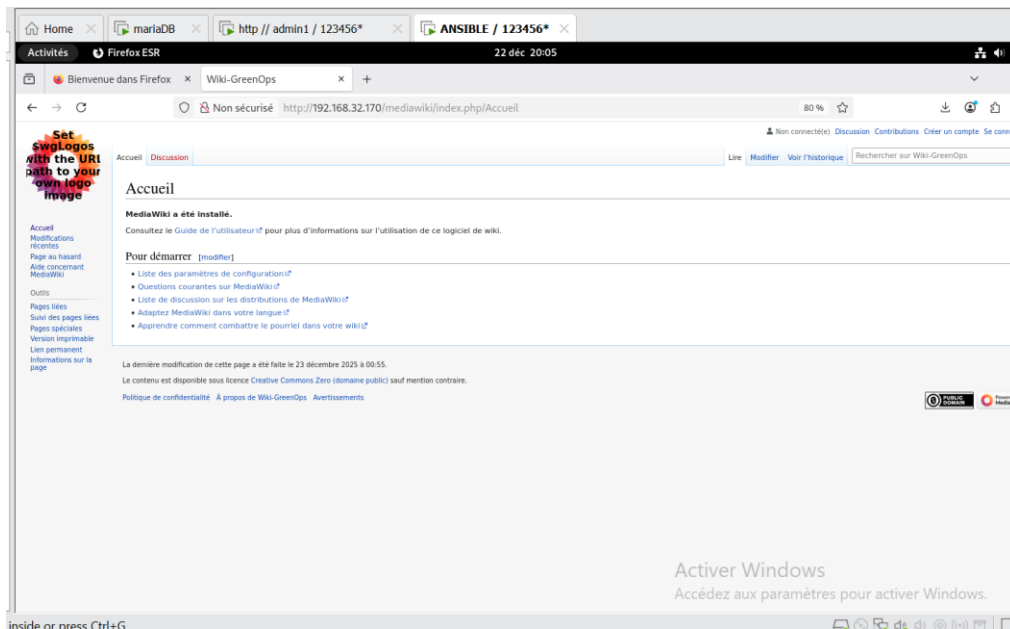
# Enabled skins.
# The following skins were automatically enabled:
wfLoadSkin( 'MinervaNeue' );
wfLoadSkin( 'MonoBook' );
wfLoadSkin( 'Timeless' );
wfLoadSkin( 'Vector' );

# End of automatically generated settings.
# Add more configuration options below.
```

## 7. Utilisation initiale de Wiki-GreenOps (création de pages et de comptes)

### Accueil du wiki

Cette capture montre la page d'accueil par défaut de Wiki-GreenOps après l'installation, confirmant que MediaWiki est fonctionnel et que l'instance est accessible via l'URL <http://192.168.32.170/mediawiki>.



## Création du compte LEO-admin1

Ces captures présentent le formulaire de création d'un compte utilisateur nommé LEO-admin1, puis le message de bienvenue confirmant la création du compte et l'authentification réussie sur Wiki-GreenOps.

Page spéciale

Créer un compte

Nom d'utilisateur  
LEO-admin1

Mot de passe  
••••••••

Il est recommandé d'utiliser un mot de passe unique que vous n'utilisez sur aucun autre site web.

Confirmez le mot de passe  
••••••••

Adresse de courriel (facultative)  
Entrez votre adresse de courriel

Nom réel (facultatif)  
Nom réel (facultatif)

Le vrai nom est facultatif. S'il est fourni, il sera utilisé pour vous attribuer vos contributions.

Créez votre compte

Wiki-GreenOps est réalisé par des gens comme vous.

0 modification

0 page

0 contributeur récent

Accueil  
Modifications récentes  
Page au hasard  
Aide concernant MediaWiki

Outils  
Pages spéciales  
Version imprimable

Politique de confidentialité À propos de Wiki-GreenOps Avertissements

Activer Windows  
Accédez aux paramètres pour activer Windows.

side or press Ctrl+G.

Page spéciale

Bienvenue, LEO-admin1 !

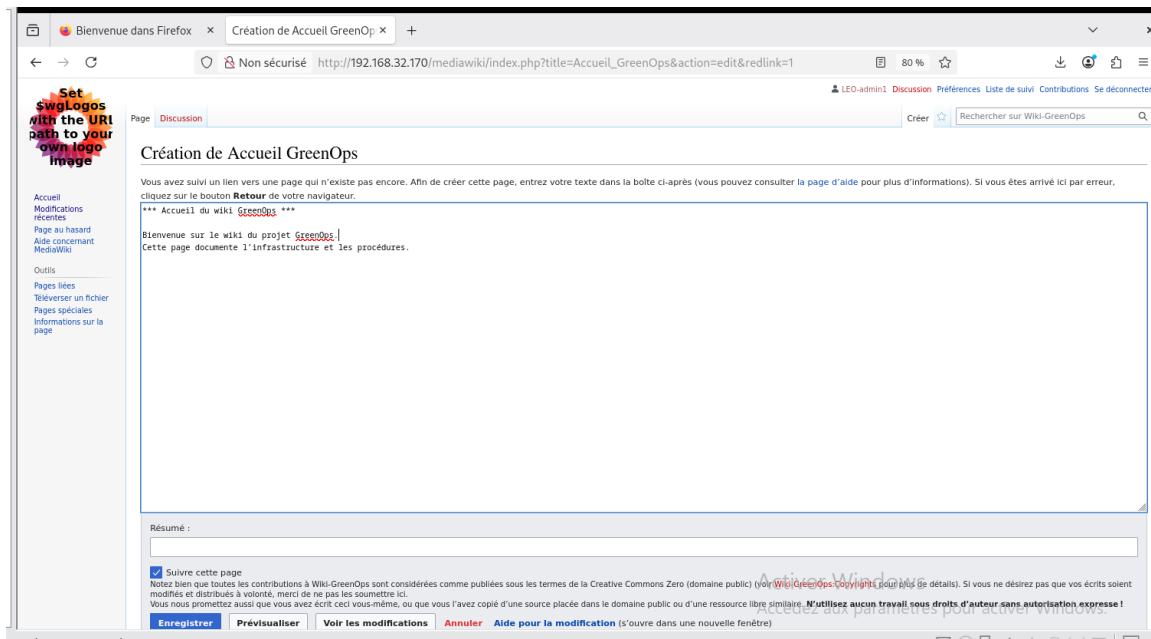
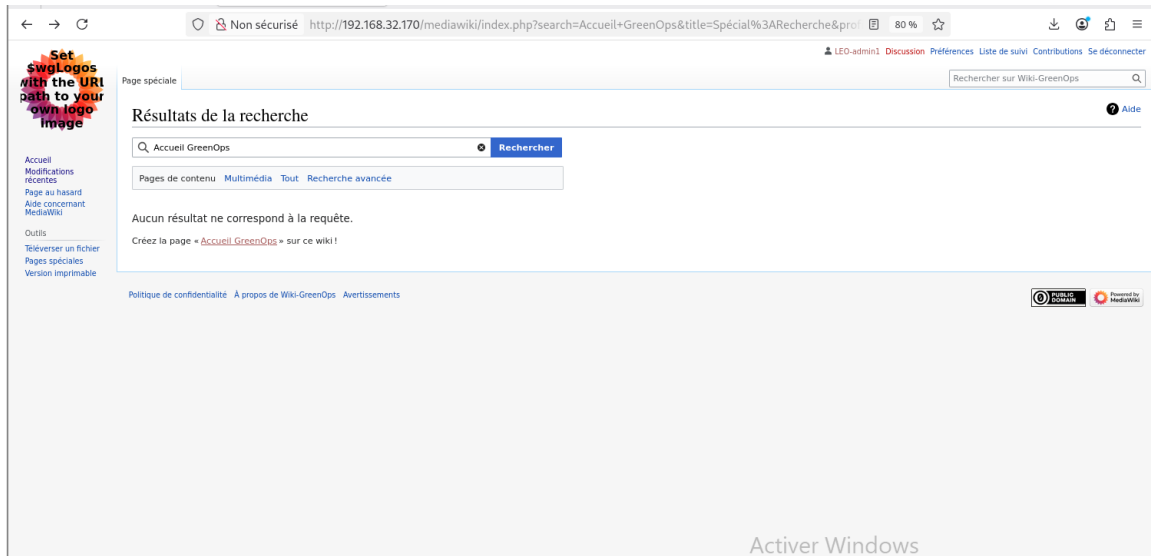
Votre compte a été créé. Vous pouvez modifier vos préférences pour Wiki-GreenOps si vous le souhaitez.  
Revenir à la page Accueil.

Politique de confidentialité À propos de Wiki-GreenOps Avertissements

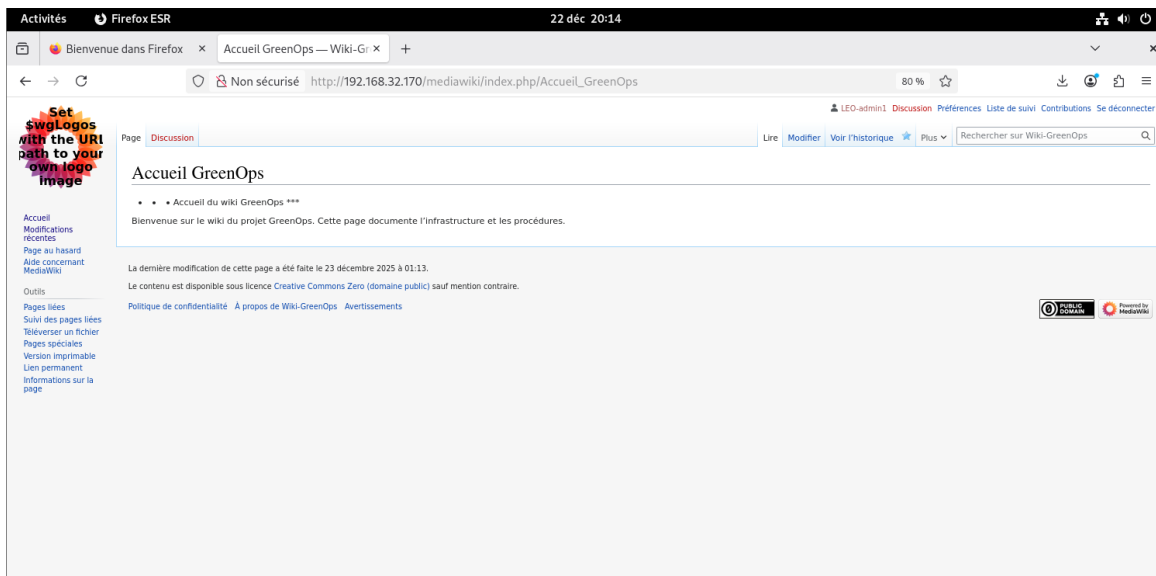
Activer Windows  
Accédez aux paramètres pour activer Windows.

side or press Ctrl+G.

## Création de la page Accueil GreenOps

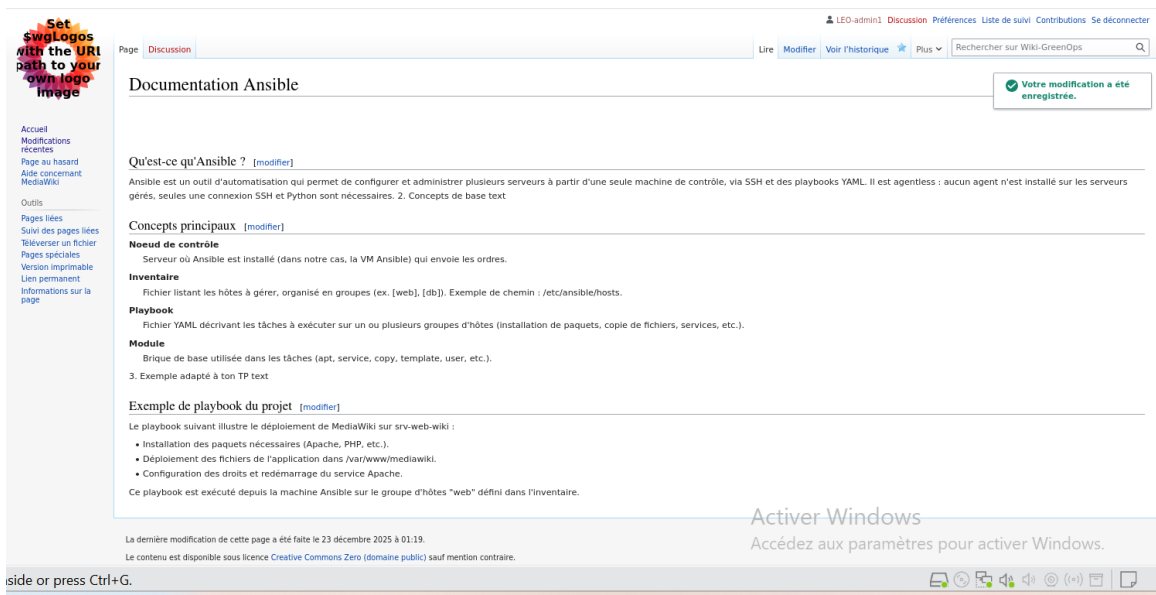




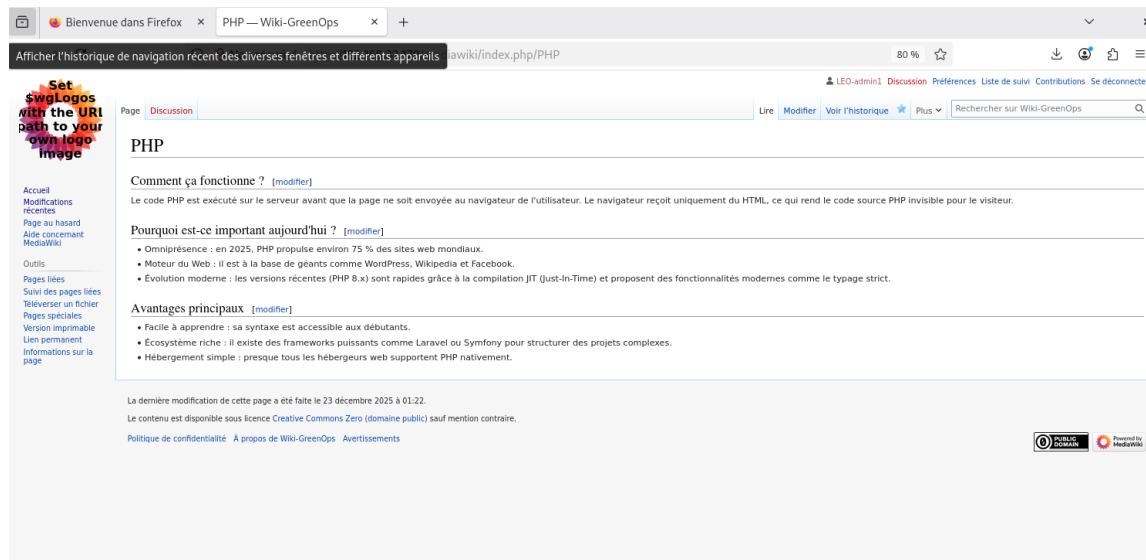


## Création des pages de documentation technique dans wiki-GreenOps

### Documentation Ansible



## PHP — Wiki-GreenOps



## 8. Difficultés rencontrées lors du déploiement de MediaWiki

- ☐ **Services MariaDB et Apache non démarrés ou non activés au démarrage**  
À plusieurs reprises, les services MariaDB ou Apache n'étaient pas actifs ou ne se lançaient pas automatiquement au boot, bloquant l'accès à la base de données ou au site web. Une vérification avec `systemctl status` ainsi que l'activation via `systemctl enable -now` ont été nécessaires.
- ☐ **Absence du fichier LocalSettings.php après le déploiement initial**  
Après l'exécution du playbook Ansible, MediaWiki affichait le message *"LocalSettings.php not found"*. Ce comportement est normal tant que l'installateur web n'est pas complété. Le fichier a été généré via l'interface web, puis copié manuellement dans le répertoire `/var/www/mediawiki`.
- ☐ **Erreurs de chemin et de permissions sur LocalSettings.php**  
Le fichier `LocalSettings.php` était initialement téléchargé dans le dossier

*Téléchargements* de la machine cliente avec des permissions locales. Il a fallu le transférer vers le serveur `srv-web-wiki`, le placer au bon emplacement et ajuster les droits d'accès afin qu'Apache puisse le lire correctement.

#### □ **Ajustements de la configuration Apache (vhost et modules)**

La configuration du virtual host MediaWiki, l'activation du site et du module `mod_rewrite` ont nécessité plusieurs tests de configuration à l'aide de `apachectl configtest` et des redémarrages du service. Une erreur dans le fichier `.conf` empêchait initialement Apache de redémarrer correctement.

#### □ **Erreurs mineures dans les playbooks Ansible**

Certains playbooks ont dû être exécutés plusieurs fois en raison de fautes de frappe, de noms de paquets incorrects, de chemins erronés ou de handlers manquants. Ces problèmes ont été corrigés en ajustant la syntaxe YAML et en vérifiant l'idempotence des playbooks.

#### □ **Problèmes de configuration applicative dans MediaWiki**

Lors de la configuration via l'interface web (nom du wiki, connexion à MariaDB, création du compte administrateur), des valeurs incorrectes pouvaient bloquer l'installation. Ces erreurs ont été résolues en vérifiant attentivement les paramètres saisis (adresse du serveur SQL, nom de la base de données, utilisateur et mot de passe).

## **9. Conclusion**

Ce livrable présente la mise en œuvre complète d'une plateforme MediaWiki déployée de façon automatisée sur deux serveurs Debian distincts (serveur web et serveur de base de données), en s'appuyant sur des scripts d'orchestration Ansible. L'ensemble des étapes techniques est documenté : préparation des systèmes, installation et sécurisation de MariaDB, déploiement d'Apache et de PHP, téléchargement et configuration de MediaWiki, création du vhost dédié, puis génération et intégration du fichier `LocalSettings.php`.

Les captures et commentaires montrent que le déploiement est reproductible (playbook Ansible idempotent), fonctionnel (accès au wiki, pages créées, compte administrateur opérationnel) et conforme au cahier des charges demandé.