

TP Déploiement WikiJS & AD via Azure

Table des matières

I. Contexte	2
II. Objectifs	2
III. Déploiement & configuration WikiJS	3
1. Création de la machine virtuelle :	3
2. Configuration de la machine virtuelle :	5
a. Mise en place de mémoire SWAP	5
b. Création d'un utilisateur de service (pour wikijs) :	6
3. Installation de WikiJS :	6
a. Installation & configuration de MariaDB :	7
b. Installation & configuration de l'application WikiJS :	8
c. Ouverture du port Azure & accès à l'interface web :	10
IV. Déploiement & Configuration d'un Active Directory sous Azure	12
1. Création de la machine virtuelle :	12
2. Déploiement de l'Active Directory :	13
a. Installation du rôle serveur :	13
b. Promotion en DC :	14
c. Configuration de l'AD :	16
V. Intégration de l'Active Directory à WIKIJS	17
1. Préparation de l'intégration côté Serveur AD :	17
2. Configuration côté Azure :	18
3. Configuration côté WIKIJS :	19
a. Ajout de la stratégie :	19
b. Configuration de la stratégie :	20
c. Test de connexion :	20
VI. Administration de WIKIJS	21
1. Création d'un groupe :	21
2. Sauvegarde de la base locale :	23
3. Activation de la 2FA à la connexion :	24

Compétences mobilisées :

- Recenser et identifier les ressources numériques
- Mettre en place et vérifier les niveaux d'habilitation associés à un service
- Gérer des sauvegardes
- Déployer un service
- Réaliser les tests d'intégration et d'acceptation d'un service

I. Contexte

Dans le cadre de ce TP, l'objectif est de mettre en place un outil documentaire permettant d'unifier, centraliser et sécuriser la gestion de la documentation technique d'une entreprise.

De plus, la mise en œuvre se fera via Microsoft Azure, où nous hébergerons & déploieront les machines.

II. Objectifs

- Déployer une infrastructure sur Azure (VM Linux et Windows).
- Installer une solution documentaire WikiJS avec base de données et services associés.
- Mettre en place une authentification centralisée via un Active Directory.

III. Déploiement & configuration WikiJS

1. Création de la machine virtuelle :

NB : Dans le cadre du statut étudiant, Azure nous donne accès à 100€ gratuitement, que je vais utiliser pour la mise en œuvre de ce TP. La procédure pour y accéder ne sera pas expliqué ici

Tout d'abord il nous faut, depuis l'interface Azure créer notre machine virtuelle :

Microsoft Azure

Rechercher dans les ressources, services et documents (G+)

Copilot

Accueil >

Créer une machine virtuelle

M'aider à créer une machine virtuelle optimisée pour la haute disponibilité M'aider à choisir la taille de machine virtuelle adaptée à ma charge de travail +1

M'aider à créer une machine virtuelle à faible coût M'aider à créer une machine virtuelle optimisée pour la haute disponibilité M'aider à choisir la taille de machine virtuelle adaptée à ma charge de travail

De base Disques Mise en réseau Administration Supervision Paramètres avancés Étiquettes Vérifier + créer

Créez une machine virtuelle qui exécute Linux ou Windows. Sélectionnez une image dans la Place de marché Azure ou utilisez une image personnalisée. Renseignez l'onglet De base et sélectionnez Vérifier + créer pour provisionner une machine virtuelle avec des paramètres par défaut, ou passez en revue chaque onglet pour une personnalisation complète. [En savoir plus](#)

Cet abonnement n'est peut-être pas éligible pour déployer des machines virtuelles de certaines tailles dans des régions spécifiques.

Détails du projet

Sélectionnez l'abonnement pour gérer les coûts et les ressources déployées. Utilisez les groupes de ressources comme les dossiers pour organiser et gérer toutes vos ressources.

Abonnement * Azure for Students

Groupe de ressources * (Nouveau) TPWIKI-JS [Créer nouveau](#)

Détails de l'instance

Nom de la machine virtuelle * WIKI-JS

Région * (Europe) Norway East [Déployer sur une zone étendue Azure](#)

Options de disponibilité Zone de disponibilité

Options de zone ☒ Zone autosélectionnée Choisissez jusqu'à 3 zones de disponibilité, une machine virtuelle par zone ☐ Zone sélectionnée par Azure (préversion) Laissez Azure attribuer la meilleure zone pour vos besoins

Zone de disponibilité * Zone 1 ☒ Vous pouvez désormais sélectionner plusieurs zones. La sélection de plusieurs zones crée une machine virtuelle par zone. [En savoir plus](#)

Type de sécurité Lancer des machines virtuelles approuvées [Configurer les fonctionnalités de sécurité](#)

Image * Rocky Linux 9 with LVM - x64 de 2e génération

< Précédent Suivant : Disques > Vérifier + créer

Envoyer des commentaires

Microsoft Azure

Rechercher dans les ressources, services et documents (G+)

Copilot

Accueil

Créer une machine virtuelle

M'aider à créer une machine virtuelle optimisée pour la haute disponibilité

M'aider à créer une machine virtuelle à faible coût

M'aider à choisir la taille de machine virtuelle adaptée à ma charge de travail

Taille *

Activer la mise en veille prolongée ☐

Compte d'administrateur

Type d'authentification ☒ Clé publique SSH ☐ Mot de passe

Nom d'utilisateur *

Source de la clé publique SSH

Type de clé SSH ☒ Format SSH RSA ☐ Format SSH Ed25519

Nom de la paire de clés *

< Précédent

Suivant : Disques >

Vérifier + créer

Envoyer des commentaires

Les paramètres suivants sont appliqués (*dans l'ordre*) :

- La machine sera dans le groupe de ressources : TPWIKI-JS
- La machine sera hébergée dans la région Norvège-Est
- Une version de Rocky Linux 9 sera installée automatiquement dessus
- La machine sera de taille « B1 » (*soit 1vCpu, 1go de RAM*)
- La connexion pour l'administration (SSH pour linux) se fera via clé publique.

Une fois les paramètres choisis nous pouvons faire « Vérifier + créer » afin de lancer la création. Une fois terminée nous aurons accès aux informations de la machine et notamment son IP Publique :

Bases		Vue JSON	
Groupe de ressources (déplacer)		Système d'exploitation	
TPWIKI-JS		Linux (rocky 9.6)	
Statut		Taille	
En cours d'exécution		Standard_B1s	
Emplacement		Adresse IP publique de la carte réseau principale	
Norway East (Zone 1)		20.251.154.158	
Abonnement (déplacer)		1 adresses IP publiques associées	
Azure for Students		Réseau/sous-réseau virtuel	
ID d'abonnement		vnet-norwayeast/snet-norwayeast-1	
8fe5dce3-60e3-4759-b9df-611556553071		Nom DNS	
Zone de disponibilité		Non configurée	
1		État d'intégrité	
		-	
		Heure de création	
		31/10/2025 13:24 UTC	
Étiquettes (modifier)			
Ajouter des étiquettes			

Avec cette IP, dans un terminal, nous pouvons nous y connecter en SSH (via clé dans mon cas) :

```
scoquel@WIKI-JS:~
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations
s

PS C:\Users\mumuel> ssh scoquel@20.251.154.158 -i "C:\Users\mumuel\Downloads\WIKIJS_key.pem"
The authenticity of host '20.251.154.158 (20.251.154.158)' can't be established.
ED25519 key fingerprint is SHA256:gYd+wX64RbVRvb90w8sal0bWBUSB3xu2kdBiw3rGZaI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '20.251.154.158' (ED25519) to the list of known hosts.
[scoquel@WIKI-JS ~]$
```

2. Configuration de la machine virtuelle :

Maintenant que nous sommes connectés en SSH, il va falloir réaliser les premières configurations dans le but de déployer WIKIJS.

a. Mise en place de mémoire SWAP

La machine était un modèle « B1 » soit les plus petits formats de machines, il est recommandé d'y ajouter de la mémoire dite SWAP, qui est du stockage que nous allons affecter à notre système pour qu'il utilise comme mémoire RAM.

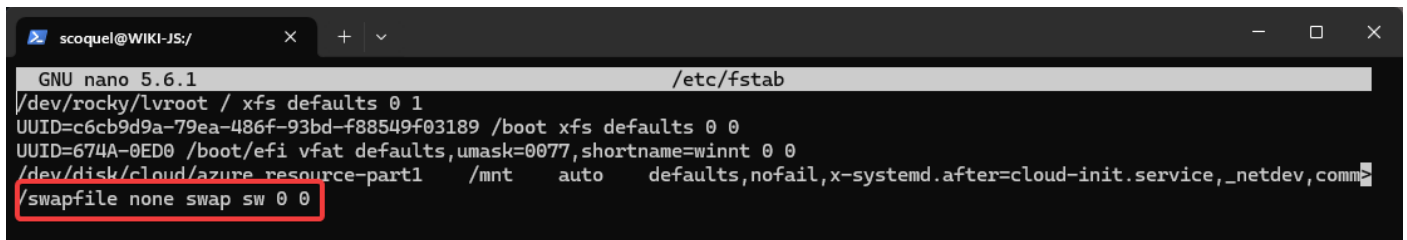
Pour ce faire nous allons créer un fichier à la racine, appelé swapfile :

- Auquel nous allons allouer 1 Gb via la commande : **sudo fallocation -l 1G /swapfile**
- Nous allons donner les droits de lecture & écriture via la commande : **sudo chmod 600 /swapfile**

```
scoquel@WIKI-JS:/
[scoquel@WIKI-JS /]$ sudo fallocation -l 1G /swapfile
[scoquel@WIKI-JS /]$ sudo chmod 600 /swapfile
[scoquel@WIKI-JS /]$ ls -lh /
total 1.1G
drwxr-xr-x. 2 root root 6 Nov 3 2024 afs
lrwxrwxrwx. 1 root root 7 Nov 3 2024 bin -> usr/bin
drwxr-xr-x. 5 root root 4.0K Oct 31 13:25 boot
drwxr-xr-x. 20 root root 3.1K Oct 31 13:24 dev
drwxr-xr-x. 89 root root 8.0K Oct 31 13:25 etc
drwxrwxr-x. 3 root root 19 May 31 04:33 grub2
drwxr-xr-x. 3 root root 21 Oct 31 13:25 home
lrwxrwxrwx. 1 root root 7 Nov 3 2024 lib -> usr/lib
lrwxrwxrwx. 1 root root 9 Nov 3 2024 lib64 -> usr/lib64
drwxr-xr-x. 2 root root 6 Nov 3 2024 media
drwxr-xr-x. 3 root root 4.0K Oct 31 13:25 mnt
drwxr-xr-x. 2 root root 6 Nov 3 2024 opt
dr-xr-xr-x. 174 root root 0 Oct 31 13:24 proc
dr-xr-x---. 3 root root 103 Oct 31 13:24 root
drwxr-xr-x. 32 root root 1.1K Oct 31 13:25 run
lrwxrwxrwx. 1 root root 8 Nov 3 2024 sbin -> usr/sbin
drwxr-xr-x. 2 root root 6 Nov 3 2024 srv
-rw-----. 1 root root 1.0G Oct 31 13:33 swapfile
dr-xr-xr-x. 13 root root 0 Oct 31 13:24 sys
```

Pour finir il faudra enregistrer le fichier que nous venons de créer en tant que fichier de SWAP avec les commandes suivantes :

- Initialiser le fichier /swapfile comme du swap utilisable par le système : **sudo mkswap /swapfile**
- Activer le fichier swap immédiatement : **sudo swapon /swapfile**
- Ajouter à fstab une ligne qui permettra d'activer ce fichier de swap automatiquement au démarrage : **echo '/swapfile none swap sw 0 0' | sudo tee -a /etc/fstab**



```

GNU nano 5.6.1 /etc/fstab
/dev/rocky/lvroot / xfs defaults 0 1
UUID=c6cb9d9a-79ea-486f-93bd-f88549f03189 /boot xfs defaults 0 0
UUID=674A-0ED0 /boot/efi vfat defaults,umask=0077,shortname=winnt 0 0
/dev/disk/cloud/azure_resource-part1 /mnt auto defaults,nofail,x-systemd.after=cloud-init.service,_netdev,comm
/swapfile none swap sw 0 0

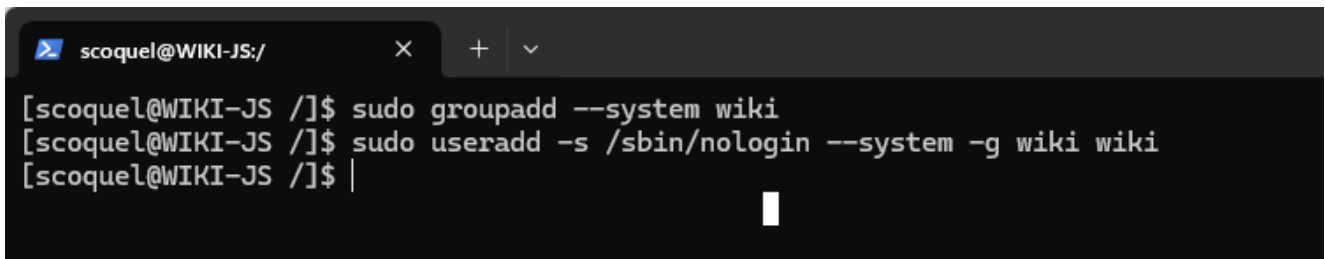
```

b. Création d'un utilisateur de service (pour wikijs) :

Pour assurer un minimum de sécurité, nous n'allons pas faire exécuter le service wikijs que nous nous apprêtons à déployer en root, par conséquent nous allons passer par un compte de service, que nous allons appeler wiki, avec son groupe associé.

Pour réaliser cela, il faut utiliser les commandes :

- **Sudo groupadd --system wiki (qui va créer un groupe système wiki)**
- **Sudo useradd -s /sbin/nologin --system -g wiki wiki (qui créé un user systeme wiki sans terminal et l'ajoute au groupe wiki)**



```

[scoquel@WIKI-JS /]$ sudo groupadd --system wiki
[scoquel@WIKI-JS /]$ sudo useradd -s /sbin/nologin --system -g wiki wiki
[scoquel@WIKI-JS /]$

```

3. Installation de WikiJS :

Maintenant que notre serveur est préconfiguré, nous allons pouvoir télécharger et installer WikiJS avant de le déployer.

Avant de commencer nous allons installer les dépendances nécessaires qui sont les suivantes :

Git, vim, wget, unzip, socat, mariadb-server, nodejs, nginx, redis (et epel-release avant afin d'étendre nos sources pour les paquets)

La commande pour installer les paquets : **sudo yum install**

```
scoquel@WIKI-JS:/
[scoquel@WIKI-JS /]$ sudo yum install e
[scoquel@WIKI-JS /]$ sudo yum install epel-release
Last metadata expiration check: 0:03:25 ago on Fri 31 Oct 2025 01:36:59 PM UTC.
Dependencies resolved.
=====
Package                                Architecture  Version      Repository    Size
=====
Installing:
epel-release                           noarch        9-7.el9      extras        19 k
=====
```

```
scoquel@WIKI-JS:/
[scoquel@WIKI-JS /]$ sudo yum install -y git vim wget unzip socat mariadb-server nodejs nginx redis
```

a. Installation & configuration de MariaDB :

Pour commencer nous allons installer et configurer la base de données nécessaire pour wikiJS, dans notre cas nous allons utiliser mariadb.

Pour l'installation de la base nous allons lancer la BDD via systemctl et lancer le binaire d'installation avec la commande : **sudo mysql_secure_installation**

```
scoquel@WIKI-JS:/srv/wiki
[scoquel@WIKI-JS wiki]$ sudo systemctl start mariadb
[scoquel@WIKI-JS wiki]$ sudo mysql_secure_installation
```

Nous serons ensuite invités à choisir différentes options comme désactiver le compte anonyme, changer le mot de passe root de la base...

```
scoquel@WIKI-JS:/srv/wiki
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] Y
Enabled successfully!
Reloading privilege tables..
... Success!

You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] n
... skipping.

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.
```

Une fois fini, la base est bien installée et nous pouvons nous y connecter pour y réaliser la création de l'utilisateur et de la base de WikiJS via la commande **mysql**

Les premières commandes à réaliser sont :

- Créer l'utilisateur wiki (dans mariadb)
- Créer la base de données pour que wikijs y crée ses tables...
- Donner les droits à l'utilisateur que nous venons de créer pour qu'il puisse interagir avec

```
scoquel@WIKI-JS:/srv/wiki
[scoquel@WIKI-JS wiki]$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 14
Server version: 10.5.27-MariaDB MariaDB Server

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)]> CREATE USER 'wiki'@'localhost' IDENTIFIED BY ' ';
Query OK, 0 rows affected (0.003 sec)

MariaDB [(none)]> CREATE DATABASE dbwiki;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON dbwiki.* TO 'wiki'@'localhost';
Query OK, 0 rows affected (0.014 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]>
```

Mot de
passe à
mettre ici

```
scoquel@WIKI-JS:/srv/wiki
MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| dbwiki   |
| information_schema |
| mysql    |
| performance_schema |
+-----+
4 rows in set (0.000 sec)

MariaDB [(none)]> |
```

b. Installation & configuration de l'application WikiJS :

La base est maintenant configurée et lancée, nous pouvons nous tourner vers l'installation de l'application en elle-même.

Dans un premier temps, nous allons télécharger & extraire l'archive dans le fichier /srv

Pour télécharger, nous allons utiliser la commande : **wget**

Pour l'extraire, nous allons utiliser la commande : **tar**

```
scoquel@WIKI-JS:~$ cd /srv
[scoquel@WIKI-JS ~]$ cd /
[scoquel@WIKI-JS ~]$ ls
afs  boot  etc  home  lib64  mnt  proc  run  srv  sys  usr
bin  dev  grub2  lib  media  opt  root  sbin  swapfile  tmp  var
[scoquel@WIKI-JS ~]$ cd /srv
[scoquel@WIKI-JS ~]$ sudo mkdir wiki
[scoquel@WIKI-JS ~]$ sudo wget https://github.com/Requarks/wiki/releases/latest/download/wiki-js.tar.gz
```

```
scoquel@WIKI-JS:~$ cd /srv/wiki
[scoquel@WIKI-JS ~]$ sudo tar -xvf wiki-js.tar.gz -C ./wiki
[scoquel@WIKI-JS ~]$ cd wiki
[scoquel@WIKI-JS ~]$ ls
assets  config.sample.yml  data  LICENSE  node_modules  package.json  server
[scoquel@WIKI-JS ~]$
```

Maintenant que l'application est téléchargée sur le serveur, il nous faut copier le fichier de configuration yml (qui est en sample actuellement) et donner les droits à notre utilisateur wiki sur le dossier afin qu'il puisse y accéder et la lancer :

- Pour lui donner les droits il faut faire : **sudo chown -R wiki : wiki /srv/wiki/ & sudo chmod 755 /srv/wiki/**
- Pour copier le fichier de configuration (une fois dans /srv/wiki) : **sudo cp config.sample.yml config.yml**

Une fois ces commandes effectuées nous pouvons modifier le fichier config que nous avons copié et y rentrer les identifiants de notre DB :

```
GNU nano 5.6.1 config.yml Modified
port: 3000

# -----
# Database
# -----
# Supported Database Engines:
# - postgres = PostgreSQL 9.5 or later
# - mysql = MySQL 8.0 or later (5.7.8 partially supported, refer to docs)
# - mariadb = MariaDB 10.2.7 or later
# - mssql = MS SQL Server 2012 or later
# - sqlite = SQLite 3.9 or later

db:
  type: mariadb

# PostgreSQL / MySQL / MariaDB / MS SQL Server only:
host: localhost
port: 3306
user: wiki
pass: mot de passe
db: dbwiki
ssl: false
```

Il faut aussi, comme nous sommes sous rocky linux qui est équipé de selinux, autoriser la connexion au port par défaut de l'interface wikijs à savoir le port 3000 :

```
scoquel@WIKI-JS:srv/wiki$ sudo semanage port -a -t http_port_t -p tcp 3000
[scoquel@WIKI-JS wiki]$ sudo setsebool -P httpd_can_network_connect 1
[scoquel@WIKI-JS wiki]$
```

Pour terminer il nous suffit de lancer l'application via la commande : **sudo node server**

Nous serons invités à nous connecter sur l'interface web afin de finir l'installation.

c. Ouverture du port Azure & accès à l'interface web :

La configuration étant terminée côté serveur et le service lancé il faut nous rendre dans la console azure afin d'ouvrir le port pour y accéder via navigateur. Il faut aller dans les paramètres réseaux de notre machine virtuelle et y créer une nouvelle règle de port entrant :

Ajouter une règle de sécurité de trafic e...
WIKI-JS-nsg

Source
Any

Plages de ports sources *
*

Destination
Any

Service
Custom

Plages de ports de destination *
3000

Protocole
☒ Any
☐ TCP
☐ UDP
☐ ICMPv4
☐ ICMPv6

Action
☒ Autoriser
☐ Refuser

Priorité *
320

Nom *
acces_wikijs

Description

Ajouter Annuler Envoyer des commentaires

✓  Groupe de sécurité réseau **WIKI-JS-nsg** (attaché à l'interface réseau: **wiki-js953_z1**)
Impacte 0 sous-réseaux, 1 interfaces réseau

Rechercher des règles

Source == tout Destination == tout Protocole == tout Action == tout

Priorité ↑	Nom	Port	Protocole
Règles des ports d'entrée (5)			
300	 SSH	22	TCP
310	Acces_web_wikijs	3000	Tous
65000	AllowVnetInBound ⓘ	Tous	Tous
65001	AllowAzureLoadBalancerInBound ⓘ	Tous	Tous
65500	DenyAllInBound ⓘ	Tous	Tous
Règles de port de sortie (3)			

Nous pouvons désormais accéder à l'interface web en contactant l'ip publique de notre serveur sur le port 3000. Nous serons accueillis par la page de première installation du service où il faudra créer un premier compte administrateur :



 You are about to install Wiki.js 2.5.308.

ADMINISTRATOR ACCOUNT

Administrator Email

The email address of the administrator account.

Password

At least 8 characters long. 19 / 255

Confirm Password

Verify your password again. 19 / 255

SITE URL

Site URL

Full URL to your wiki, without the trailing slash (e.g. https://wiki.example.com). This should be the public facing URL, not the internal one if using a reverse-proxy.

TELEMETRY

☒ Allow Telemetry
Help Wiki.js developers improve this app with anonymized telemetry.
[Learn more](#)

✓ INSTALL



Welcome to your wiki!

Let's get started and create the home page.

+ CREATE HOME PAGE

ADMINISTRATION

IV. Déploiement & Configuration d'un Active Directory sous Azure

L'application WikiJS étant bien installée, nous allons y intégrer un Active Directory afin de permettre à des utilisateurs de s'authentifier directement via leur compte AD ce qui est une pratique intéressante en entreprise.

1. Création de la machine virtuelle :

Comme pour la première machine, il faut créer via le menu de azure une nouvelle VM. Étant donné que nous allons y héberger un Windows serveur elle sera un peu différente :

- La taille de la machine sera une B2 et non une B1s (pour avoir 2vpu & 4go de RAM)
- L'OS à choisir sera un Windows Serveur 2025 (En pay as you go)

Microsoft Azure

Accueil >

Créer une machine virtuelle

M'aider à créer une machine virtuelle à faible coût M'aider à choisir la taille de la machine virtuelle

M'aider à créer une machine virtuelle à faible coût M'aider à créer une machine virtuelle optimisée pour la haute disponibilité

Détails du projet

Sélectionnez l'abonnement pour gérer les coûts et les ressources déployées. Utilisez les groupes de ressources comme les dossiers pour organiser et gérer toutes vos ressources.

Abonnement * Azure for Students

Groupe de ressources * TPWNO-15

Détails de l'instance

Nom de la machine virtuelle * AD

Région * Europe Nord-Est

Options de disponibilité * Zone de disponibilité

Options de zone * Zone auto-sélectionnée

Zone de disponibilité * Zone 1

Type de sécurité * Lancer des machines virtuelles approuvées

Image * Windows Server 2025 Datacenter: Azure Edition - x64 de 2e génération

Architecture de machine virtuelle * x64

Précédent Suivant : Disques Voirifier > créer

Microsoft Azure

Accueil >

Créer une machine virtuelle

M'aider à créer une machine virtuelle à faible coût M'aider à choisir la taille de la machine virtuelle

M'aider à créer une machine virtuelle à faible coût M'aider à créer une machine virtuelle optimisée pour la haute disponibilité

Var toutes les images Configurer la génération de machine virtuelle

Architecture de machine virtuelle * x64

Exécuter avec la remise Azure Spot * Non

Taille * Standard_B2s - 2 processeurs virtuels, 4 Go de mémoire (44,38 \$/mois)

Activer la mise en veille prolongée * Non

Compte d'administrateur

Nom d'utilisateur * scoquel

Mot de passe * *****

Confirmer le mot de passe * *****

Règles des ports d'entrée

Sélectionnez les ports réseau de machine virtuelle accessibles publiquement à partir d'Internet. Vous pouvez spécifier un accès réseau plus limité ou granulaire sous l'onglet Mise en réseau.

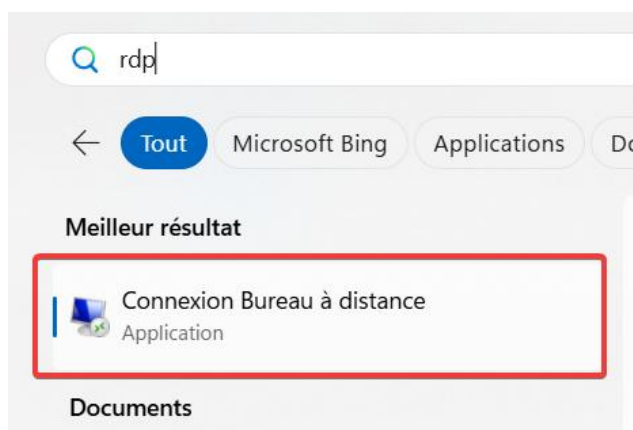
Ports d'entrée publics * Autoriser les ports sélectionnés

Sélectionner des ports d'entrée * RDP (3389)

Cela permet à toutes les adresses IP d'accéder à votre machine virtuelle. Ceci est recommandé uniquement pour les tests. Utilisez les contrôles avancés de l'onglet Mise en réseau pour créer des règles afin de limiter le trafic entrant sur les adresses IP connues.

Précédent Suivant : Disques Voirifier > créer

Une fois la machine créée et lancée nous pouvons y accéder via **la connexion de bureau à distance (RDP)** :

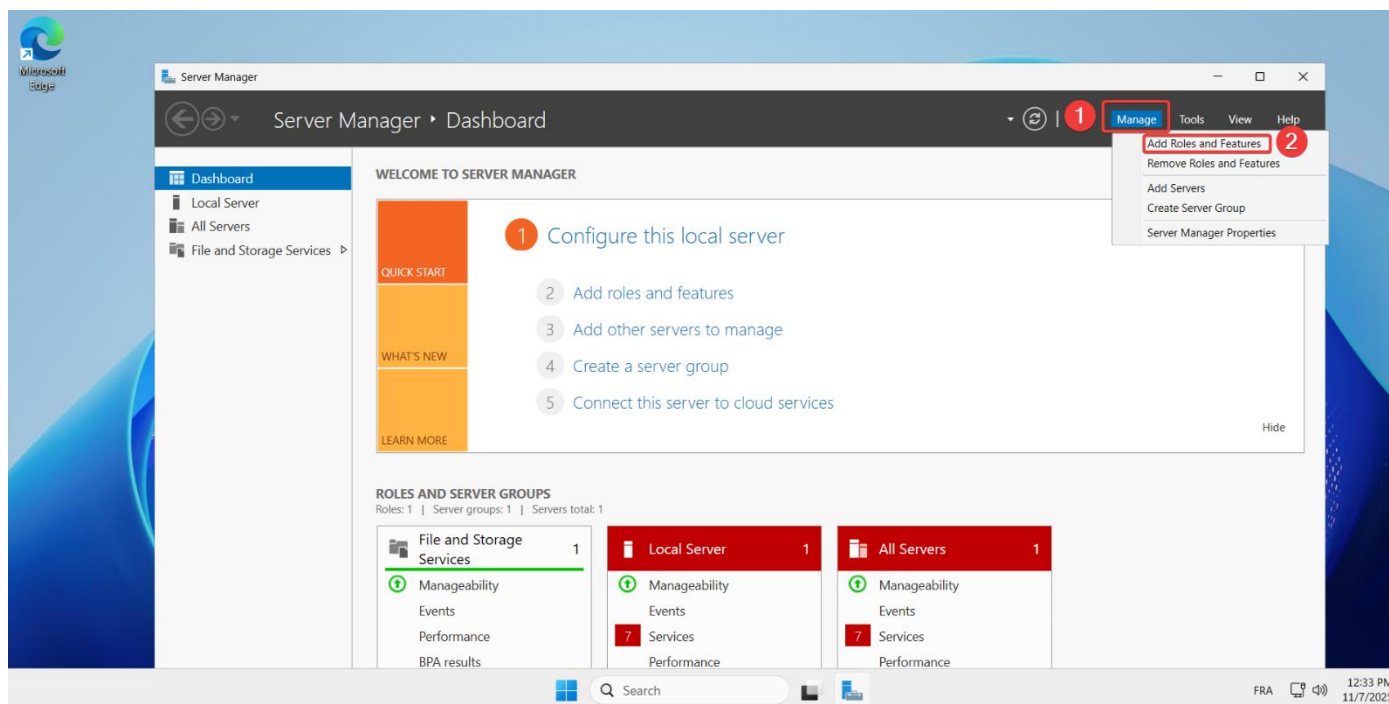


Il faudra rentrer l'ip publique du serveur, et l'identifiant/mot de passe renseignés lors de la création.

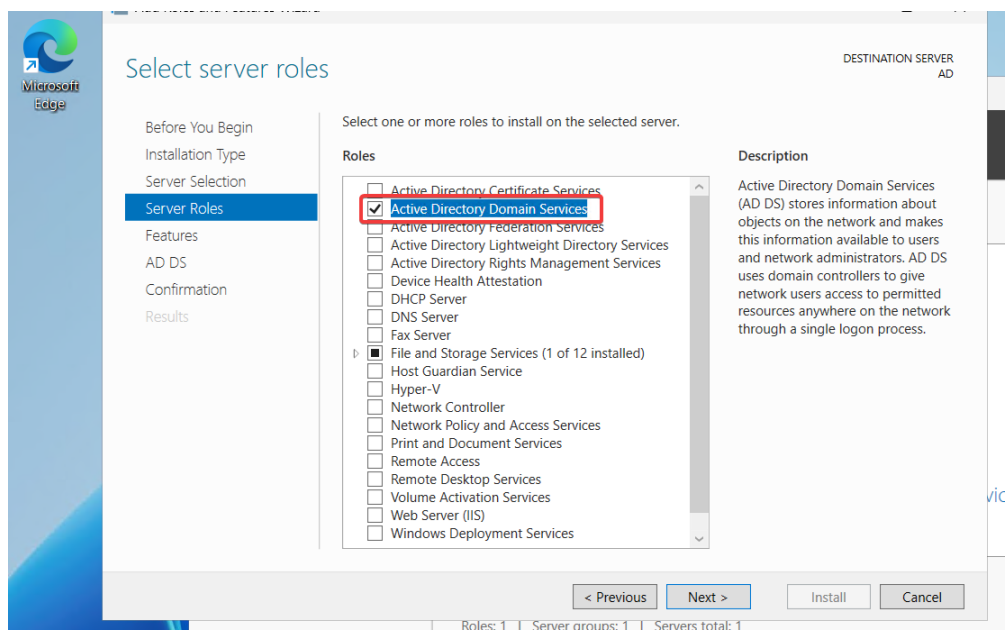
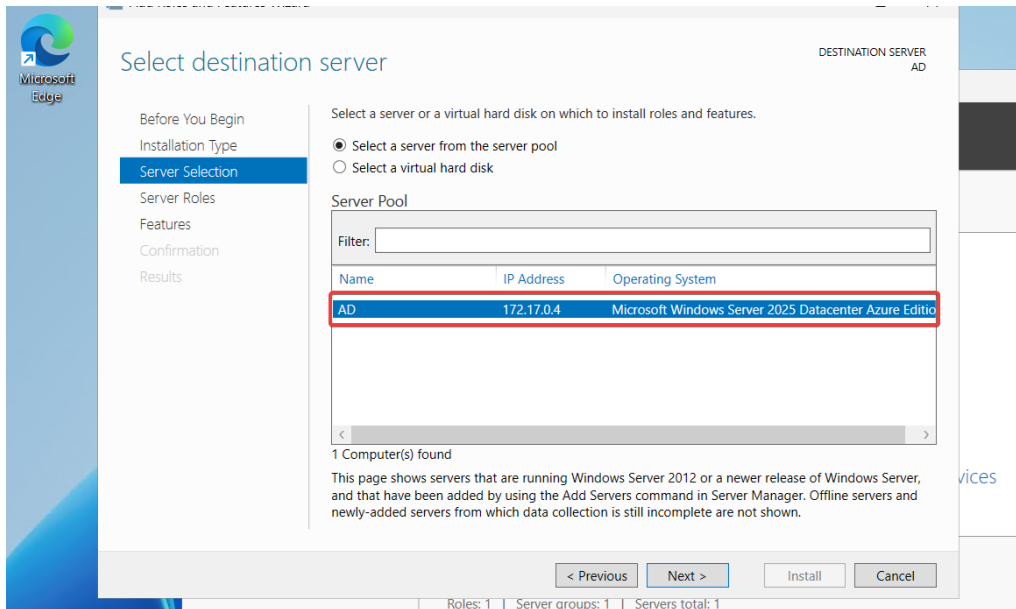
2. Déploiement de l'Active Directory :

a. Installation du rôle serveur :

Pour déployer l'active directory sur le serveur, il faut d'abord lui installer le rôle ADDS via le menu **Gérer (Manage) → Ajouter des rôles et fonctionnalités (Add roles and features)** dans le Gestionnaire de serveur :



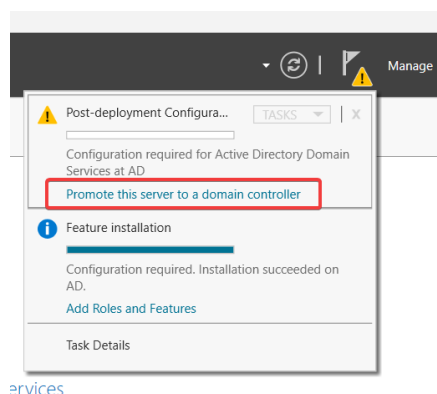
Pour finir il suffit de choisir notre serveur et de choisir dans le menu suivant **Active Directory Domain Services** :



Il ne reste plus qu'à suivre les menus et arriver à **install**

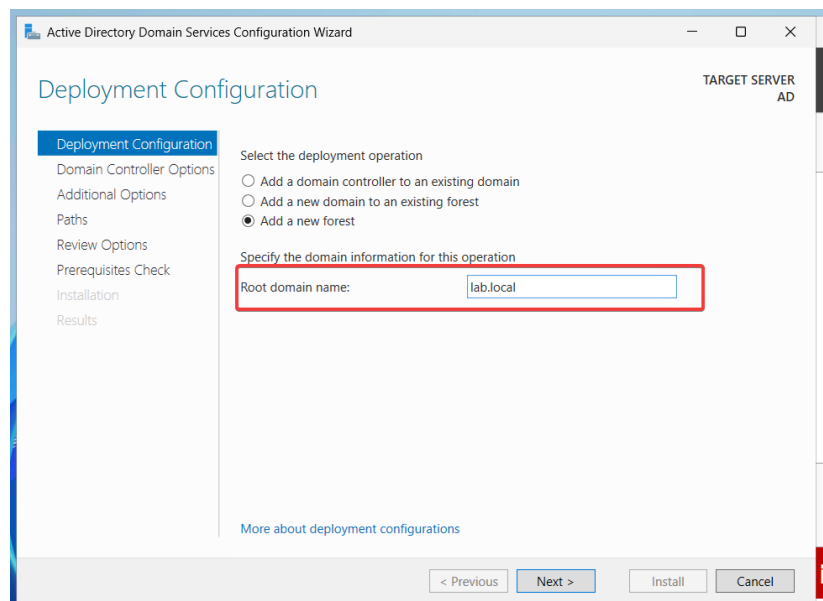
b. Promotion en DC :

Une fois que le rôle à fini d'être installé sur le serveur, il ne reste plus qu'à le promouvoir en tant que contrôleur de domaine (DC), un message dans le gestionnaire de serveur va nous le proposer :

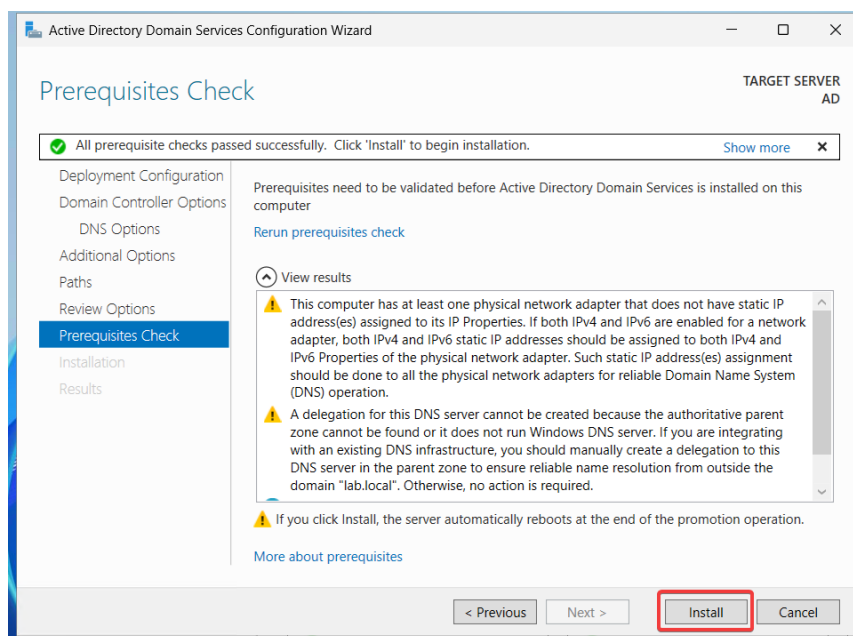


Il faut suivre les instructions et indiquer les paramètres demandés à savoir :

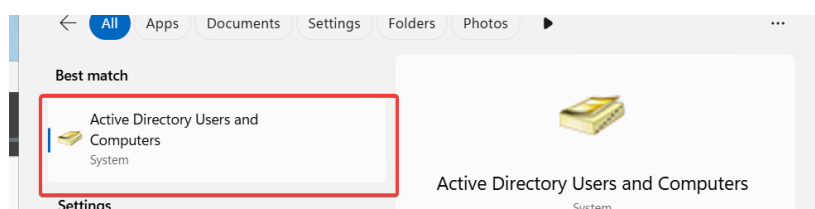
- Le nom de la nouvelle forêt (ici nous allons l'appeler lab.local)
- Le mot de passe DSRM (pour passer en mode récupération si besoin)



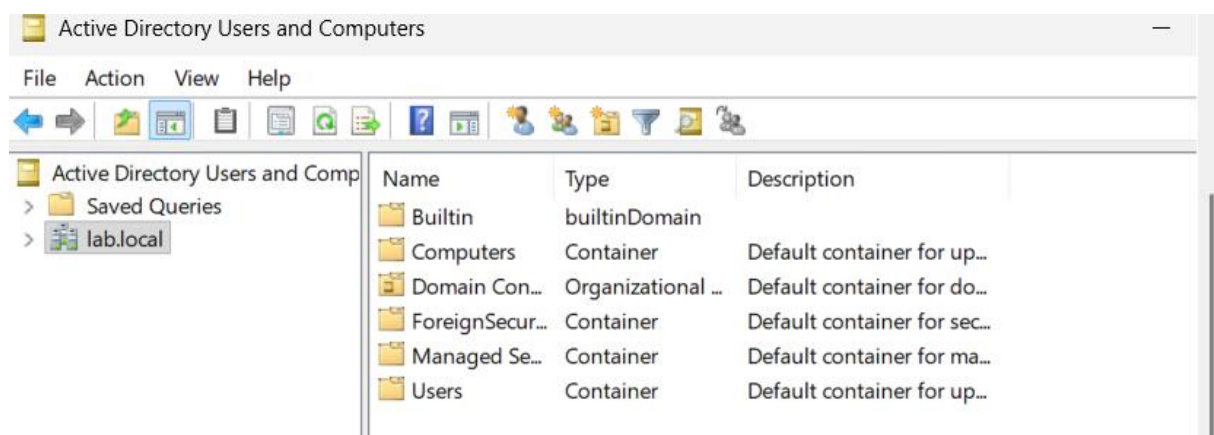
Nous pouvons passer le reste des rubriques car nous n'allons pas changer les paramètres. Une fois les prérequis validés, l'installation est proposée :



À la fin de l'installation, le serveur va se redémarrer automatiquement et le domaine sera créé, pour vérifier nous pouvons ouvrir le menu **Utilisateurs et Ordinateurs de l'AD** :



Et y vérifier que le domaine est bien présent avec les Unités d'Organisation par défaut :

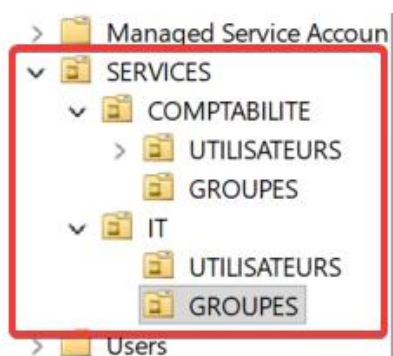


c. Configuration de l'AD :

Avant d'intégrer notre AD à l'application WIKIJS, il faut le configurer en ajoutant les OU et utilisateurs selon les instructions fournies au préalable :

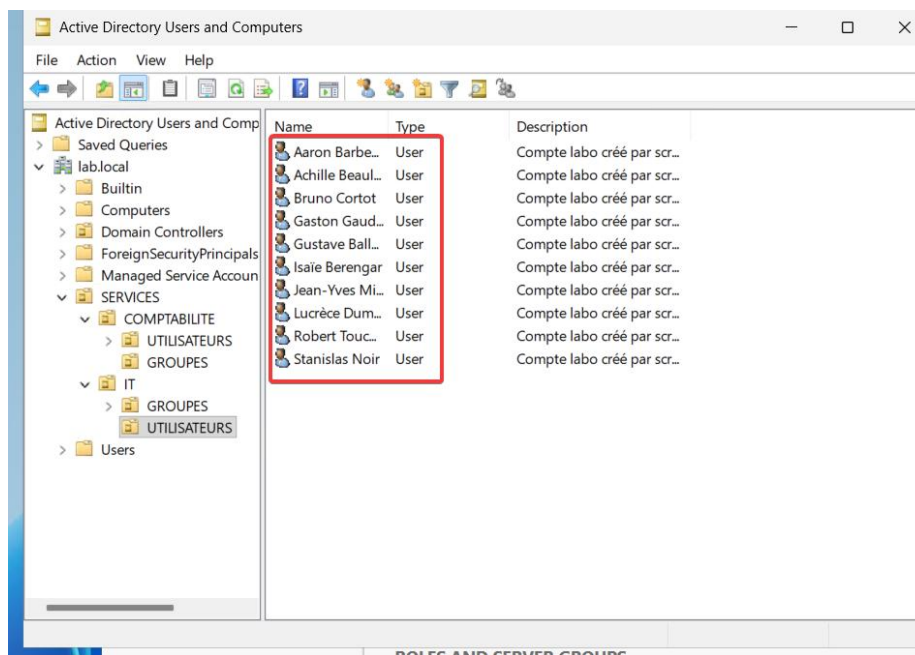
- L'arborescence suivante :
 - **SERVICES → COMPTABILITÉ**
 - ↳ IT
 - Dans chacun des services : **Une OU GROUPES & une OU UTILISATEURS**
- Les utilisateurs suivants (Dans le service IT) :
 - **Stanislas Noir**
 - **Lucrèce Dumont**
 - **Gaston Gaudin**
 - **Aaron Barbeau**
 - **Gustave Ballesdens**
 - **Jean-Yves Micheaux**
 - **Robert Touchard**
 - **Achille Beaulieu**
 - **Isaïe Berengar**
 - **Bruno Cortot**

Création des OU (via **Nouveau → Unité d'Organisation**) :



Création des Utilisateurs (via **Nouveau →Utilisateur**) via un script powershell :

```
PS C:\Users\scoquel> C:\Users\scoquel\Documents\create User AD.ps1
```



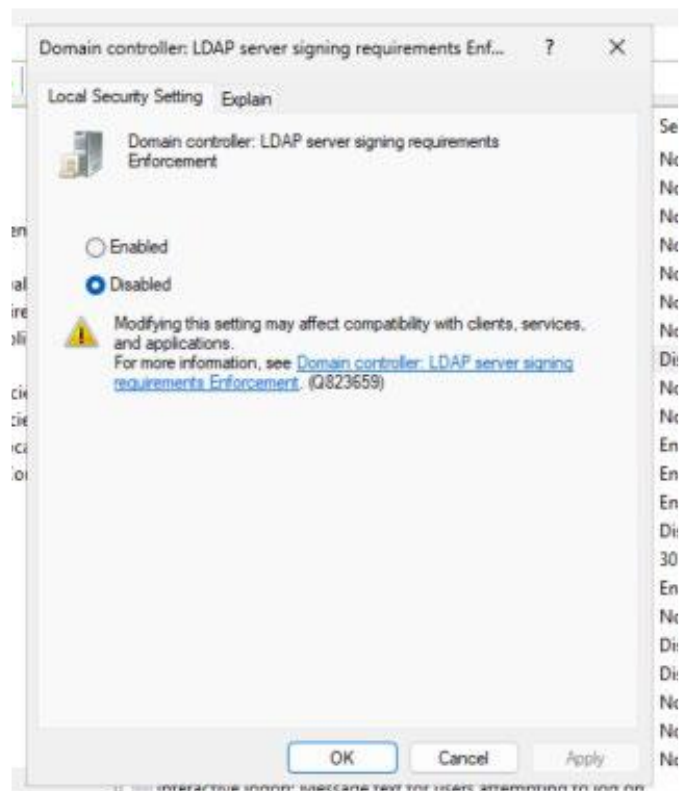
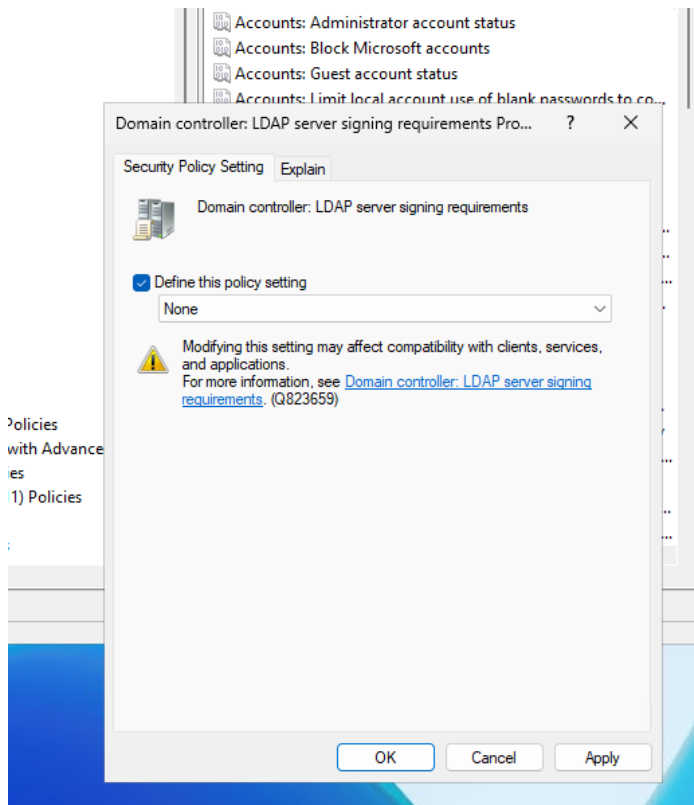
V. Intégration de l'Active Directory à WIKIJS

Les deux serveurs étant prêts nous pouvons les lier ensemble via une intégration Active Directory (LDAP) sur WIKIJS afin de permettre à nos utilisateurs de se connecter avec juste leur identifiants Active directory.

1. Préparation de l'intégration côté Serveur AD :

Nous devons au préalable désactiver certaines sécurité (**dans le cadre du lab uniquement, à ne pas reproduire en production**) afin de permettre au serveur d'accepter nos connexions malgré qu'elles ne soient pas sécurisées.

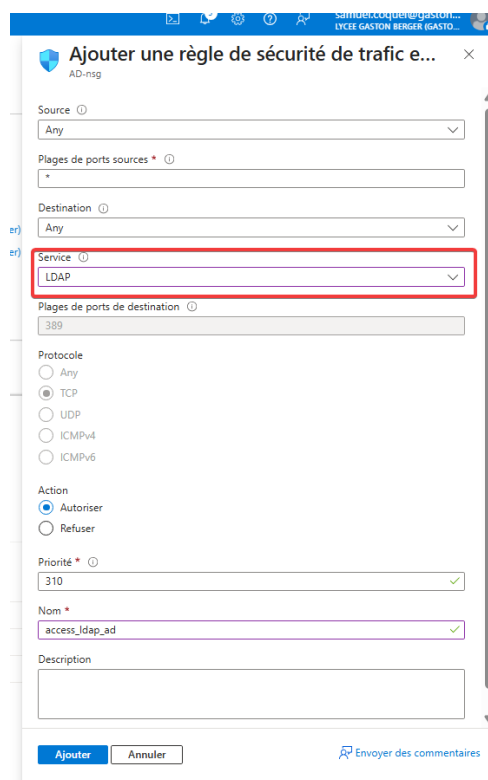
Il faut, sur le serveur Windows nous rendre dans le menu **Gestion des stratégies de groupes (GPO)** et éditer la stratégie globale du domaine afin de désactiver les deux options suivantes :



Ces deux options maintenant désactivées nous assurent que le serveur va accepter les connexions LDAP même nous sécurisées.

2. Configuration côté Azure :

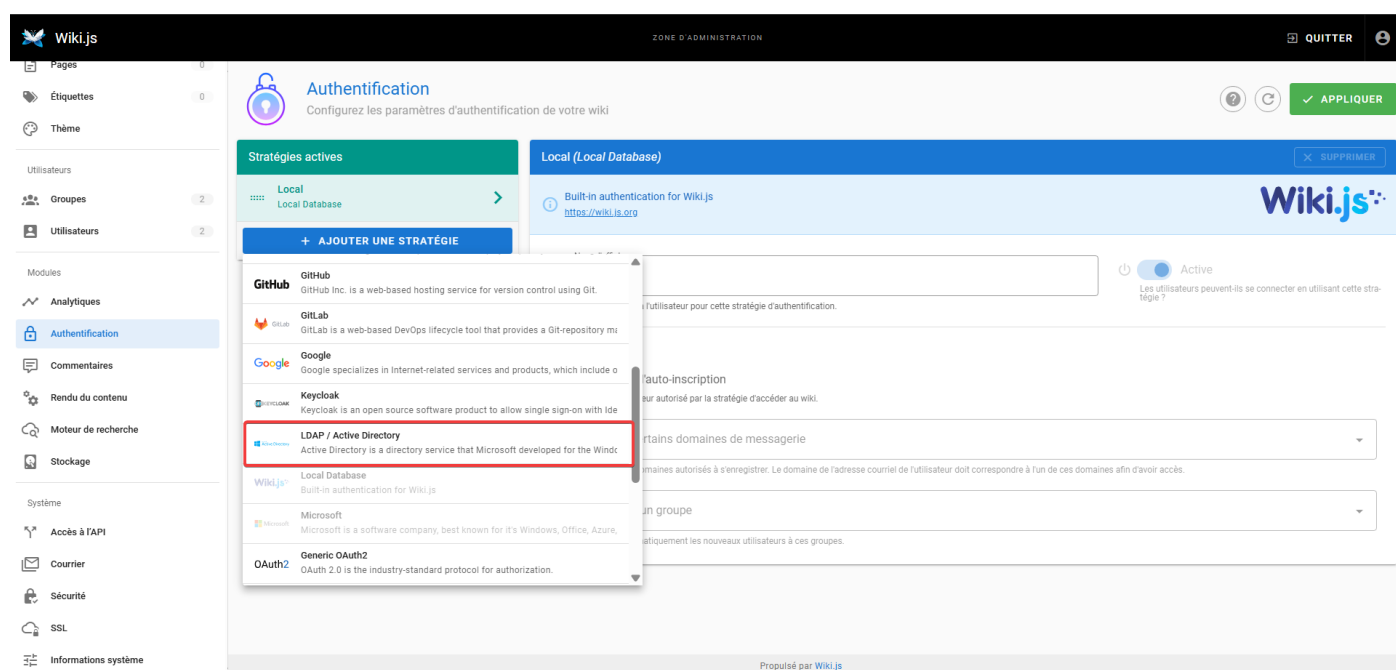
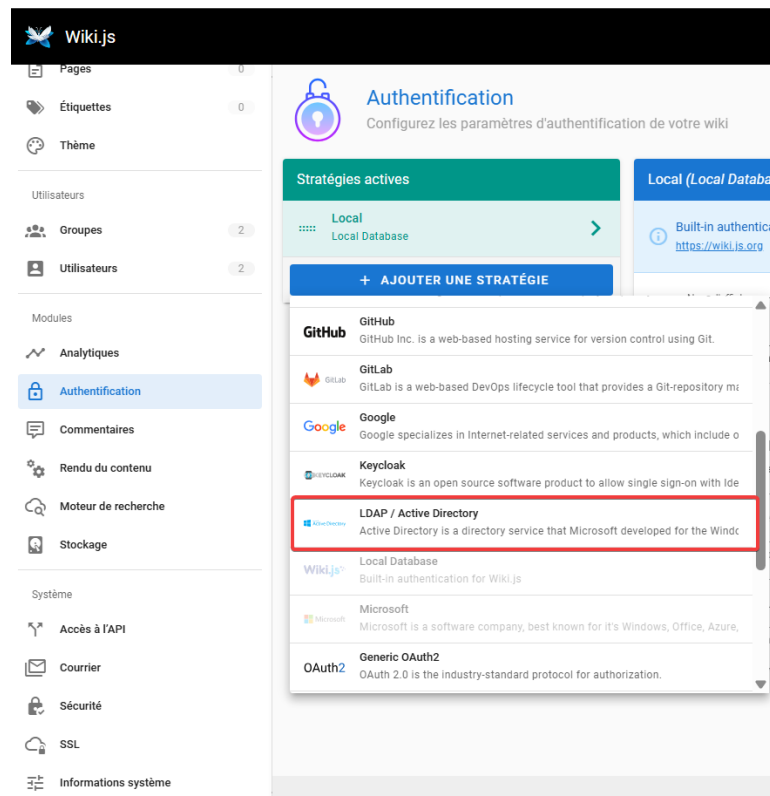
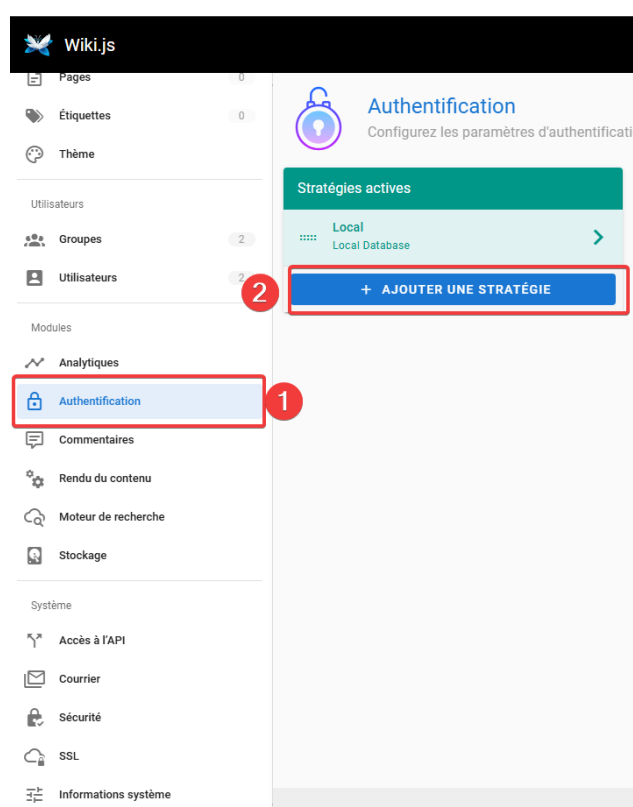
Il faut maintenant ouvrir le port LDAP sur l'interface Azure de notre AD pour permettre à WIKI.JS de s'y connecter par la suite :



3. Configuration côté WIKI.JS :

a. Ajout de la stratégie :

Pour finir l'intégration nous devons l'ajouter côté applicatif, pour ce faire il faut se rendre dans le **panneau d'administration**→**Modules**→**Authentification** Et y ajouter une nouvelle stratégie Microsoft Active Directory :



b. Configuration de la stratégie :

La stratégie maintenant bien installée, nous pouvons la configurer en y ajoutant les informations de notre serveur, les identifiants de connexion :

The screenshot shows the Wiki.js configuration page for the 'LDAP / Active Directory' strategy. The left sidebar contains navigation links: Pages, Étiquettes, Thème, Utilisateurs, Groupes, Utilisateurs, Modules, Authentification (highlighted), Commentaires, Rendu du contenu, Moteur de recherche, Stockage, Système, Accès à l'API, Courrier, Sécurité, SSL, and Informations système. The main content area is titled 'Active Directory is a directory service that Microsoft developed for the Windows domain networks.' and includes a Microsoft logo. Below this, the strategy is named 'LDAP / Active Directory' and is marked as 'Active'. The configuration section, titled 'CONFIGURATION DE LA STRATÉGIE', contains several fields: 'LDAP URL' (ldap://20.251.144.108:389), 'Admin Bind DN' (cn='scoquel',CN=Users,DC=lab,DC=local), 'Admin Bind Credentials' (Mot de passe du compte adm indiqué au dessus), 'Search Base' (OU=UTILISATEURS,OU=IT,OU=SERVICES,DC=lab,DC=local), and 'Search Filter' ((sAMAccountName={{username}})). A 'Use TLS' toggle is at the bottom.

Wiki.js

QUITTER

Pages 0

Étiquettes 0

Thème

Utilisateurs

Groupes 2

Utilisateurs 2

Modules

Authentification

Commentaires

Rendu du contenu

Moteur de recherche

Stockage

Système

Accès à l'API

Courrier

Sécurité

SSL

Informations système

Active Directory is a directory service that Microsoft developed for the Windows domain networks.
<https://www.microsoft.com/windowsserver>

Microsoft Active Directory

Nom d'affichage

T LDAP / Active Directory

Le titre présenté à l'utilisateur pour cette stratégie d'authentification.

Active

Les utilisateurs peuvent-ils se connecter en utilisant cette stratégie ?

CONFIGURATION DE LA STRATÉGIE

LDAP URL

ldap://20.251.144.108:389

(e.g. ldap://serverhost:389 or ldaps://serverhost:636)

Admin Bind DN

cn='scoquel',CN=Users,DC=lab,DC=local

The distinguished name (dn) of the account used for binding.

Admin Bind Credentials

Mot de passe du compte adm indiqué au dessus

The password of the account used above for binding.

Search Base

OU=UTILISATEURS,OU=IT,OU=SERVICES,DC=lab,DC=local

The base DN from which to search for users.

Search Filter

((sAMAccountName={{username}}))

The query to use to match usernames. {{username}} must be present and will be interpolated with the user provided username when performing the LDAP search.

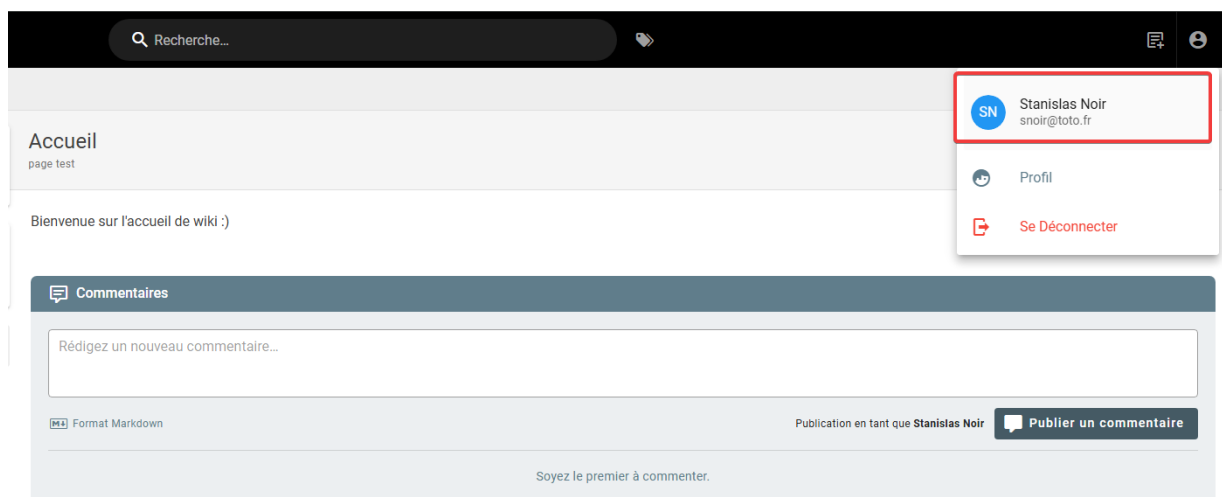
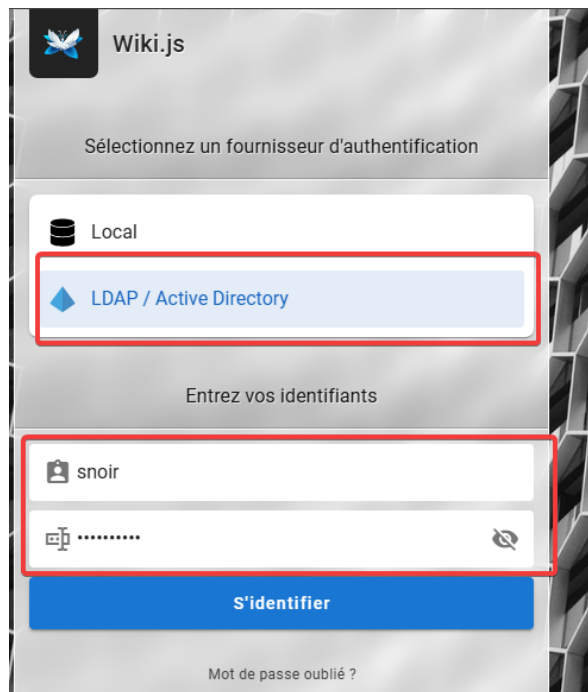
Use TLS

Les paramètres indiquent :

- L'adresse du serveur ldap (ici l'ip publique azure mais à ne pas faire en production)
- Le compte Admin de domaine à utiliser pour permettre à wikijs d'avoir accès à l'active directory
- L'OU dans laquelle se baser pour trouver les utilisateurs
- La propriété AD sur laquelle se baser pour les identifiants (ici l'identifiant de connexion AD)

c. Test de connexion :

Maintenant que la configuration est réalisée côté Application et que les deux serveurs sont en ligne, essayons de s'y connecter avec un compte provenant de l'AD :



VI. Administration de WIKI.JS

Maintenant que le logiciel est fonctionnel nous pouvons nous atteler à son administration applicative. En effet maintenant qu'un utilisateur se connecte avec son compte AD, un compte dans WikiJS est créé et nous pouvons lui assigner des droits particuliers, un groupe avec des droits etc...

1. Création d'un groupe :

Pour commencer nous allons créer un groupe Utilisateurs AD qui nous servira de point d'entrée pour nos utilisateurs via le menu d'administration→**Utilisateurs**→**Groupes** :

Wiki.js

QUITTER

Tableau de bord

Site

Général

Paramètres régionaux

Navigation

Pages 0

Étiquettes 0

Thème

Utilisateurs

Groupe has been created successfully.

manage groups and their permissions

+ NEW GROUP

ID	Name	Users	Created
1	Administrators	1	31/10/2025
2	Guests	1	31/10/2025
3	utilisateurs AD	0	Aujourd'hui à 08:24

Utilisateurs

Groupe 2

Utilisateurs 2

1

Ensuite pour automatiser l'ajout des utilisateurs AD à ce groupe, il faut l'ajouter dans les options de la stratégie :

INSCRIPTION

☒ Autoriser l'auto-inscription
Permettre à tout utilisateur autorisé par la stratégie d'accéder au wiki.

☐ Limiter à certains domaines de messagerie

Une liste des domaines autorisés à s'enregistrer. Le domaine de l'adresse courriel de l'utilisateur doit correspondre à l'un de ces domaines afin d'avoir accès.

Associer à un groupe

utilisateurs AD

Attribuer automatiquement les nouveaux utilisateurs à ces groupes.

Maintenant, un utilisateur venant de l'AD sera automatiquement envoyé dans le groupe utilisateurs AD par défaut avec les droits associés :

Wiki.js

Profil

Mes informations personnelles

Mes informations

Nom complet
Benoit Test

Employement

Titre du poste

Authentification

Fournisseur

LDAP / Active Directory

Préférences

Parcourir les préférences

Format de date
Par défaut

Apparence
Par défaut

Groupe

utilisateurs AD

Activité

Ajouter à
vendredi 7 novembre 2025 09:08

Contrôle de la session de
vendredi 21 novembre 2025 08:17

Contrôle de la session de
vendredi 21 novembre 2025 08:17

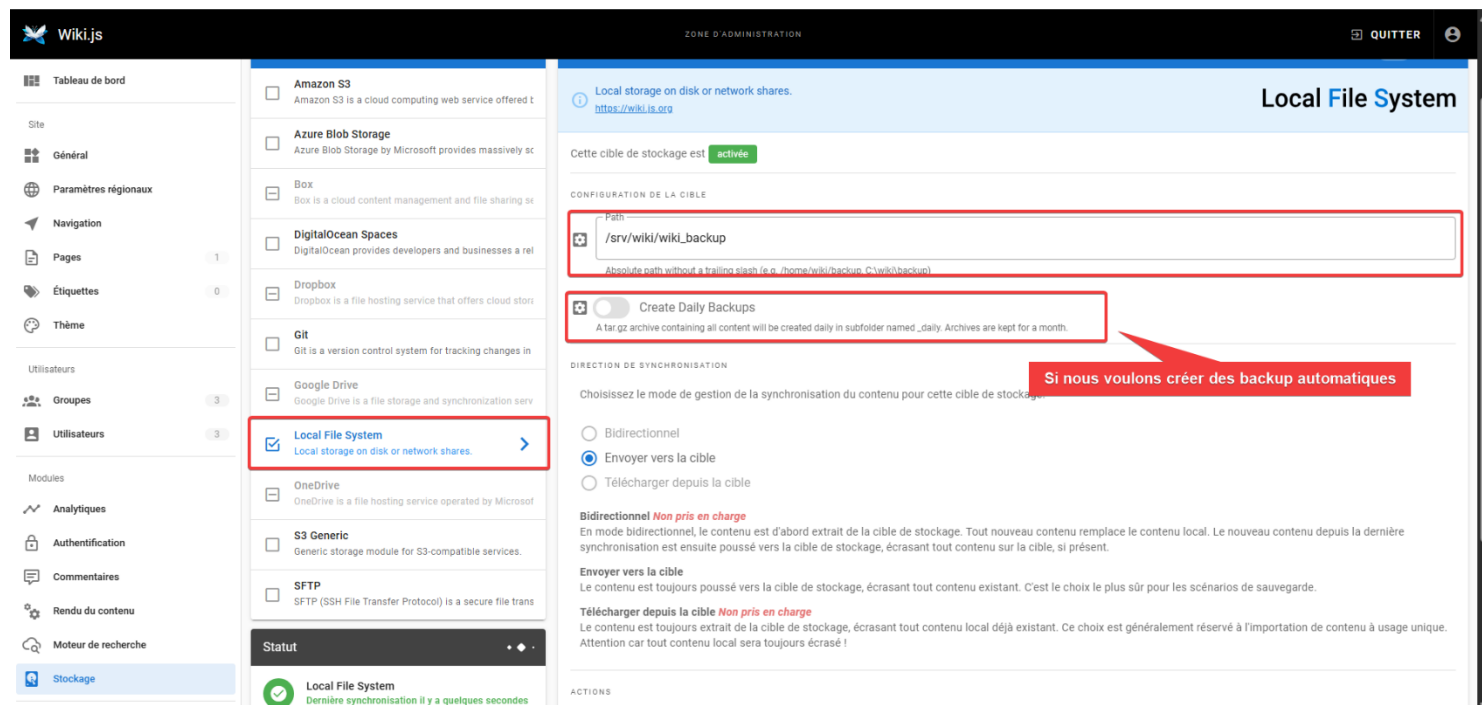
Pages visitées
0

Commentaires publiés
0

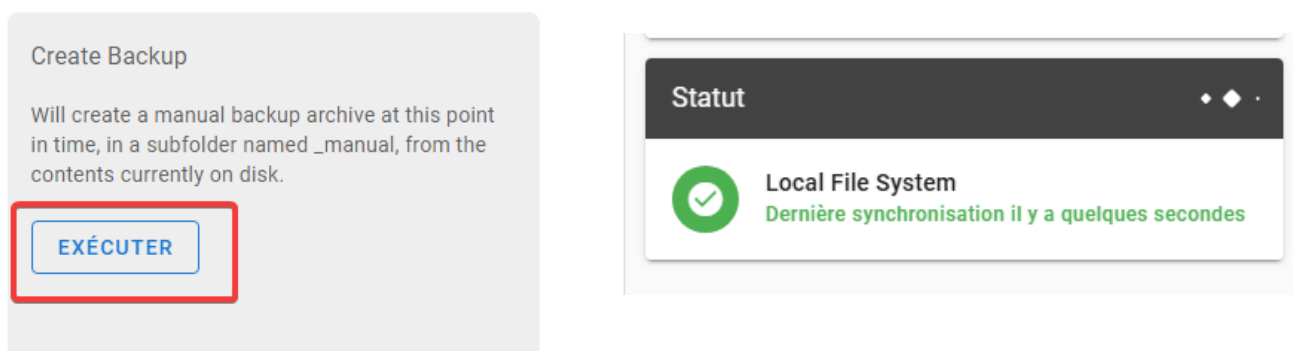
2. Sauvegarde de la base locale :

Un des éléments que nous pouvons aussi paramétrer est la sauvegarde de la base locale, qui nous permettra de créer des backups et de restaurer la base en cas de soucis. Cela est situé dans le Menu d'administration → **Modules** → **Stockage**

Il faut ensuite paramétrer le module en fonction de nos besoins, dans notre cas nous allons utiliser le stockage local avec les paramètres suivants :



Pour tester nous pouvons exécuter l'action manuellement via les actions en bas de la page et avoir le retour dans statut :



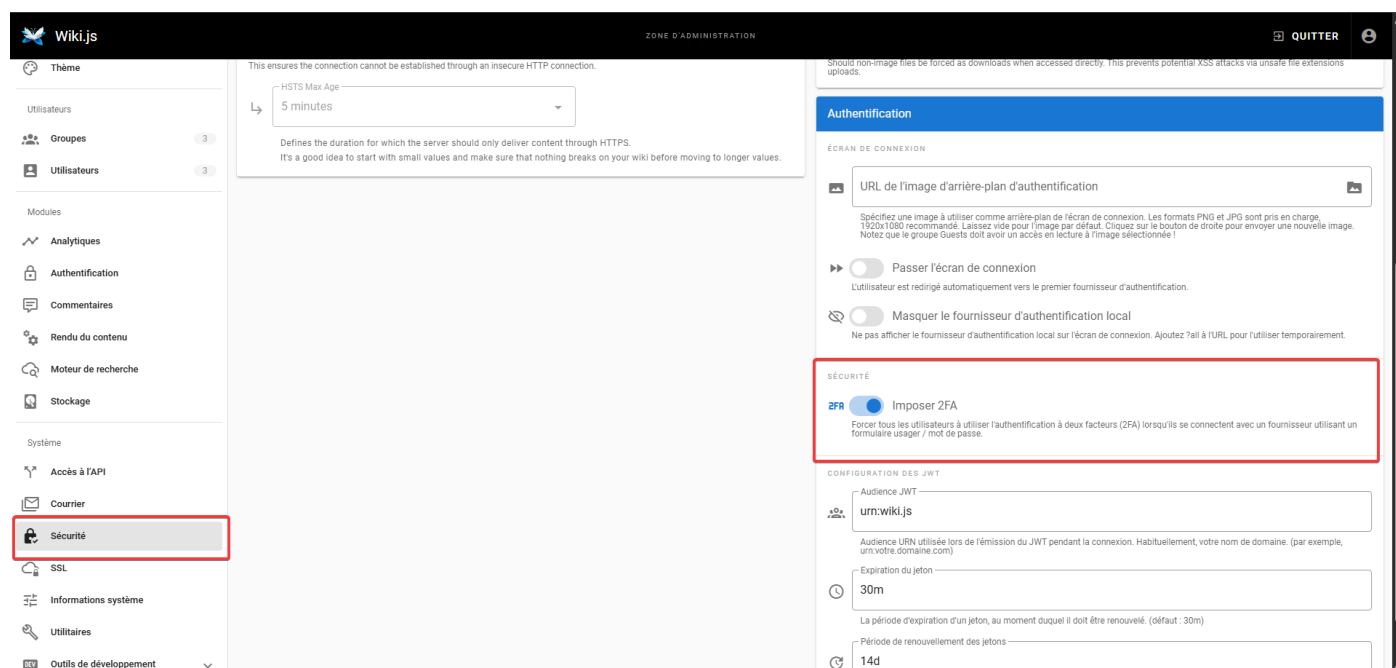
Le backup à bien été créé sur le serveur aussi :

```
scoquel@WIKI-JS:~$ cd /srv/wiki
[scoquel@WIKI-JS wiki]$ ls
assets  config.sample.yml  config.yml  data  LICENSE  node_modules  package.json  server  wiki_backup
[scoquel@WIKI-JS wiki]$ cd wiki_backup/
[scoquel@WIKI-JS wiki_backup]$ ls
_manual
[scoquel@WIKI-JS wiki_backup]$ cd _manual/
[scoquel@WIKI-JS _manual]$ ls
wiki-20251121-133811.tar.gz
[scoquel@WIKI-JS _manual]$
```

3. Activation de la 2FA à la connexion :

Une dernière chose que nous pouvons faire est d'activer le 2FA à la connexion afin de renforcer la sécurité et empêcher des comptes de se faire compromettre facilement en ajoutant un facteur d'authentification.

Pour la forcer, nous allons nous rendre dans **Système→Sécurité** et activer l'option « Imposer 2FA » :



Maintenant si l'on se connecte avec un utilisateur, nous avons un pop-up nous indiquant d'activer la 2FA via l'application de notre choix (google authenticator, microsoft...) :



