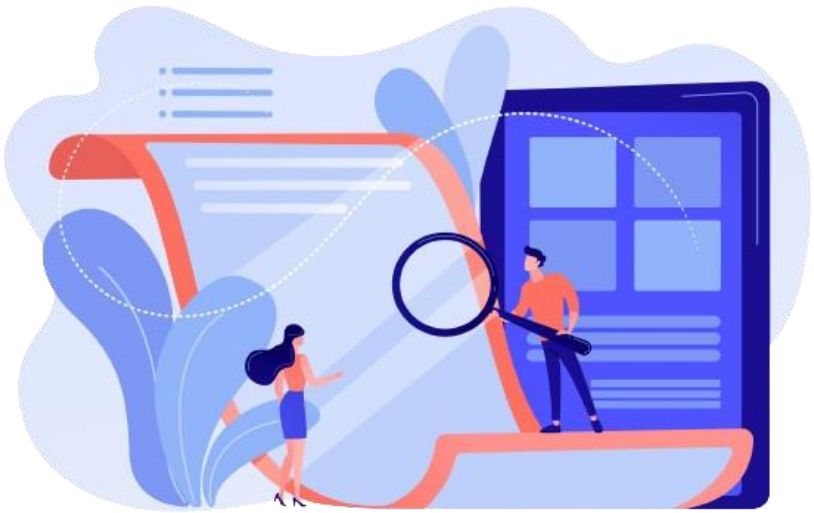
BTS Services Informatiques aux Organisations

Option Solutions d'Infrastructure, Systèmes et Réseaux
(S.I.S.R)



Documentation Technique

Épreuve E5 - Situation 1

WAÏ-LUNE Nathan

Session 2024

Historique des modifications

Par	Date	Description
WAÏ-LUNE Nathan	06/06/2024	Première Version du document

Table des matières

Historique des modifications 1

Table des matières 2

Mode Opératoire Active Directory / Contrôleur de domaine 3

- **Installation du service AD DS** 3
- **Promotion du serveur en DC** 9
- **Conclusion** 13

Mode Opératoire DNS 14

- **Rappel** 14
- **Installation du service DNS** 15
- **Création d'une zone directe** 20
- **Création d'un nouvel enregistrement** 25
- **Création d'une zone inversée** 26
- **Création d'un nouvel enregistrement PTR** 30
- **Conclusion** 32

Mode Opératoire Serveur Web sous Debian12 33

- **Installation du serveur Apache** 33
- **Installation du serveur MariaDB** 36
- **Installation de PHP** 38
- **Conclusion** 39

Mode Opératoire Virtual Host 40

- **Importation fichier SQL** 47
- **Conclusion** 49

Mode Opératoire Active Directory / Contrôleur de domaine

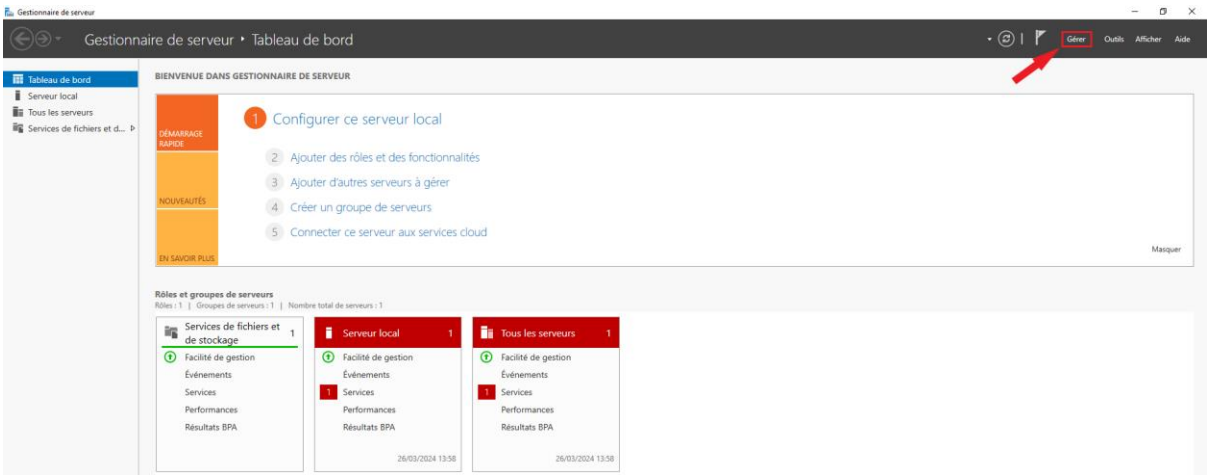
Ce mode opératoire explique comment créer un domaine sur Windows Server 2019. Pour ce faire, nous installerons le rôle "Active Directory Domain Service".

Les services de domaine Active Directory (AD DS) stockent des informations à propos des objets sur le réseau et rendent ces informations disponibles pour les utilisateurs et les administrateurs du réseau. Les services AD DS utilisent les contrôleurs de domaine pour donner aux utilisateurs du réseau un accès aux ressources autorisées n'importe où sur le réseau via un processus d'ouverture de session unique.

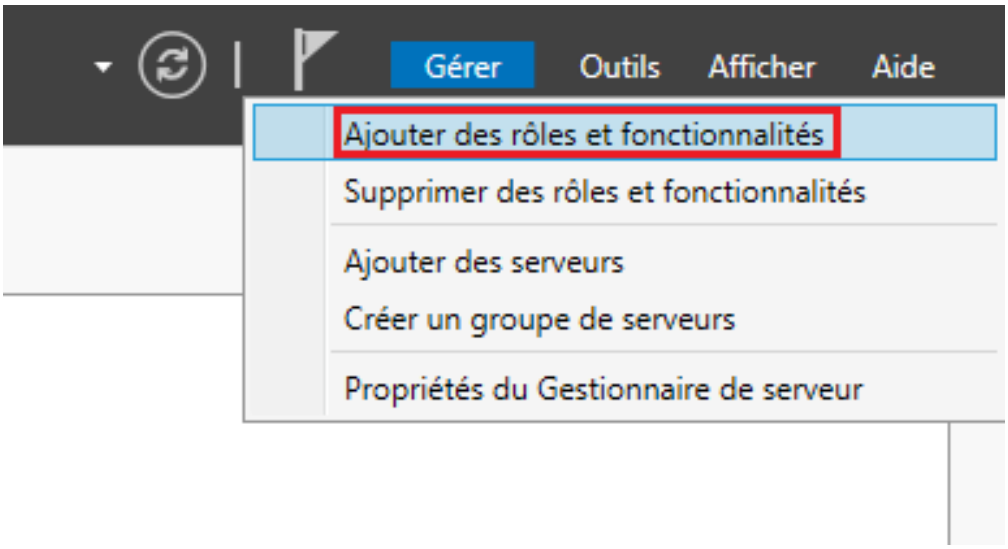
AD DS aide les administrateurs à gérer de manière sécurisée ces informations et facilite la collaboration entre les utilisateurs d'un même domaine.

Installation du service AD DS

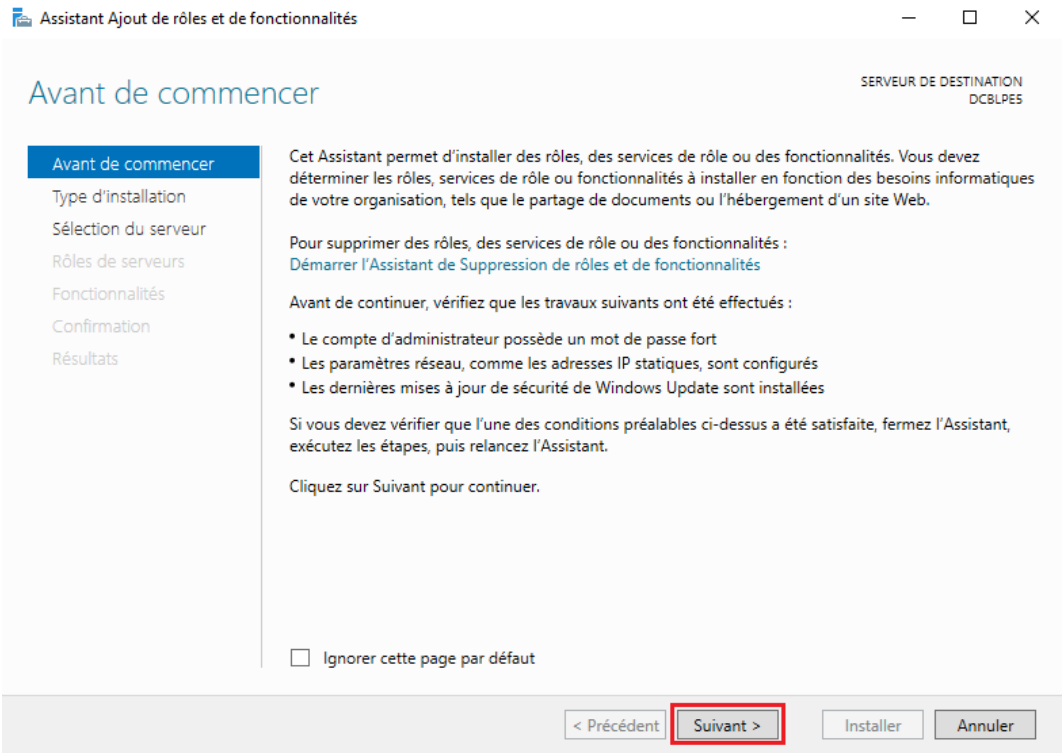
Après avoir installé Windows Server 2019, il vous faudra ouvrir le gestionnaire de serveur (s'il ne s'est pas ouvert automatiquement), cliquer sur Gérer :



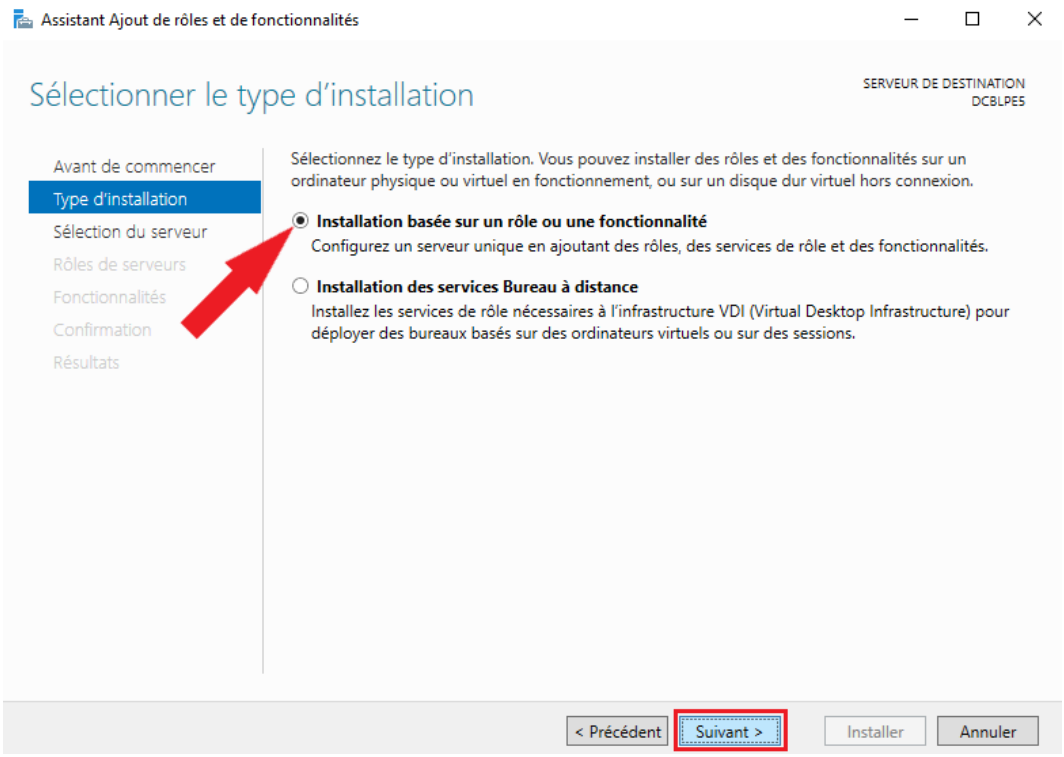
Puis « Ajouter des rôles et fonctionnalités » :



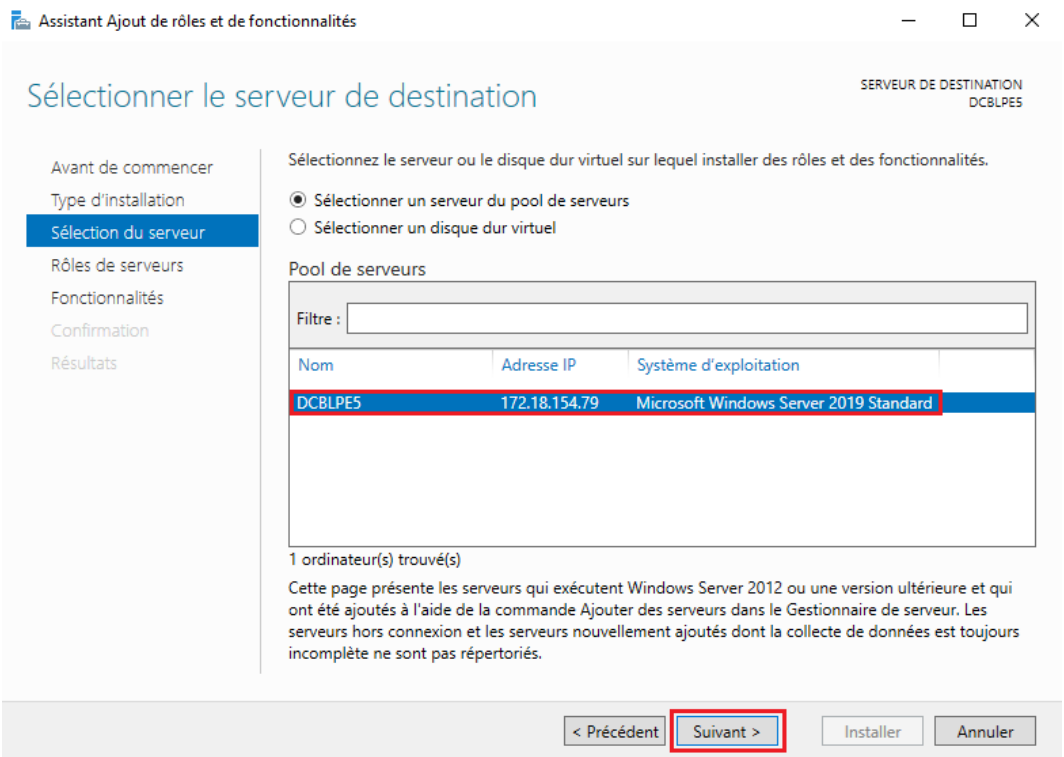
Dans l'assistant d'Ajout de rôles et de fonctionnalités, lisez attentivement les informations qui vous sont présentées et cliquez sur suivant :



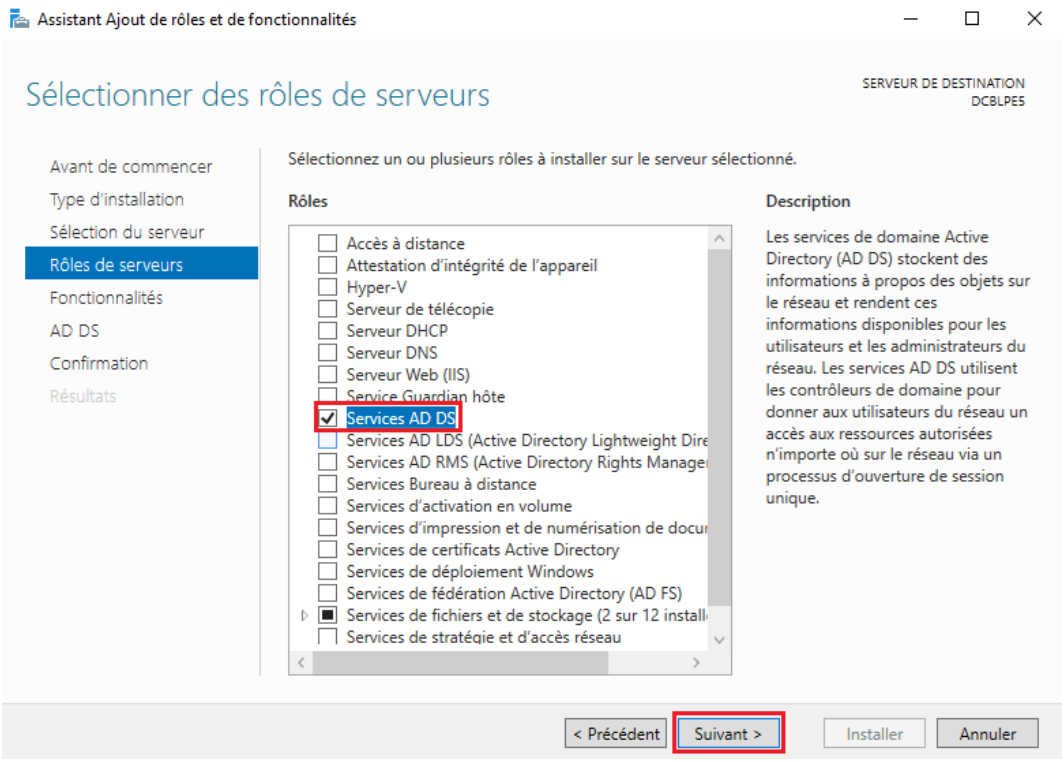
Sélectionner le type d'installation de votre choix, dans cette situation, on choisira une Installation basée sur un rôle ou une fonctionnalité. Ensuite, cliquez sur suivant :



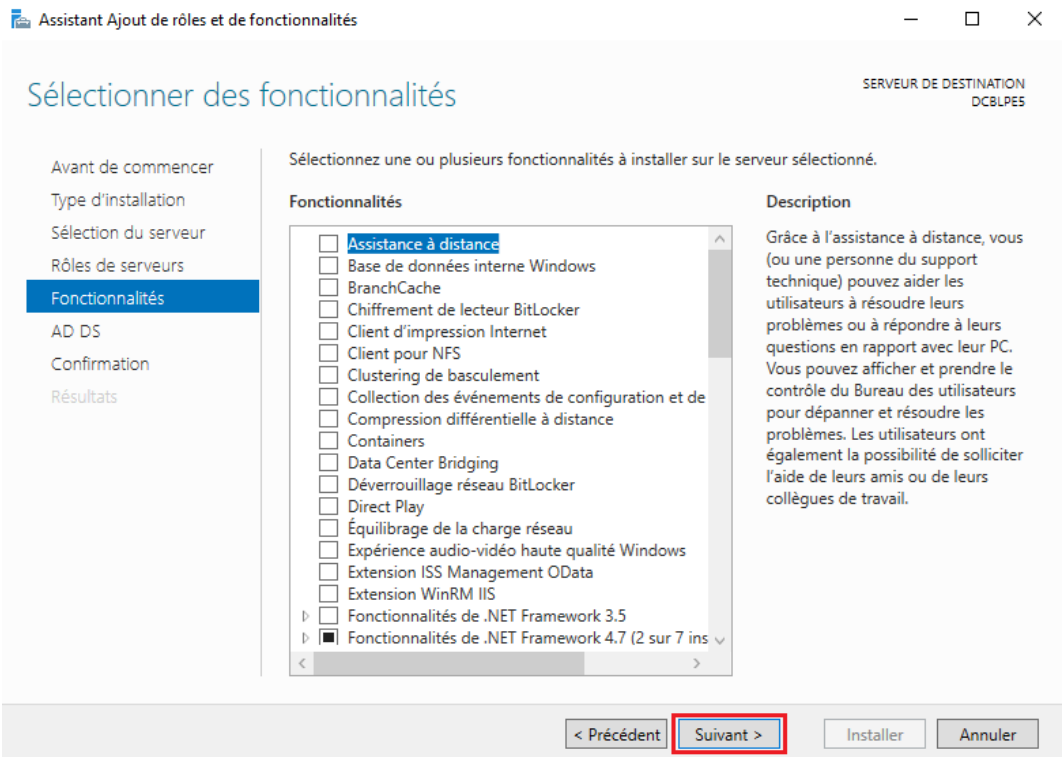
On sélectionne le serveur de destination et on clique sur suivant :



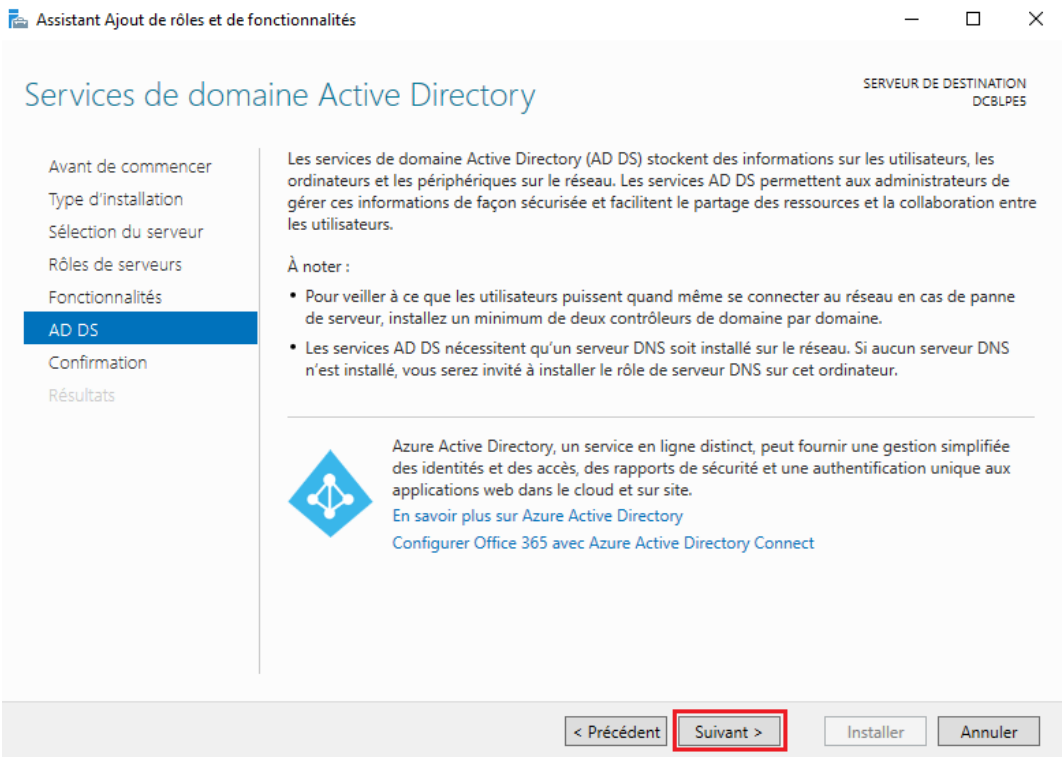
On sélectionne ensuite le ou les rôles qu'on souhaite installer, en l'occurrence, on souhaite installer toute la pile AD DS et on clique sur suivant :



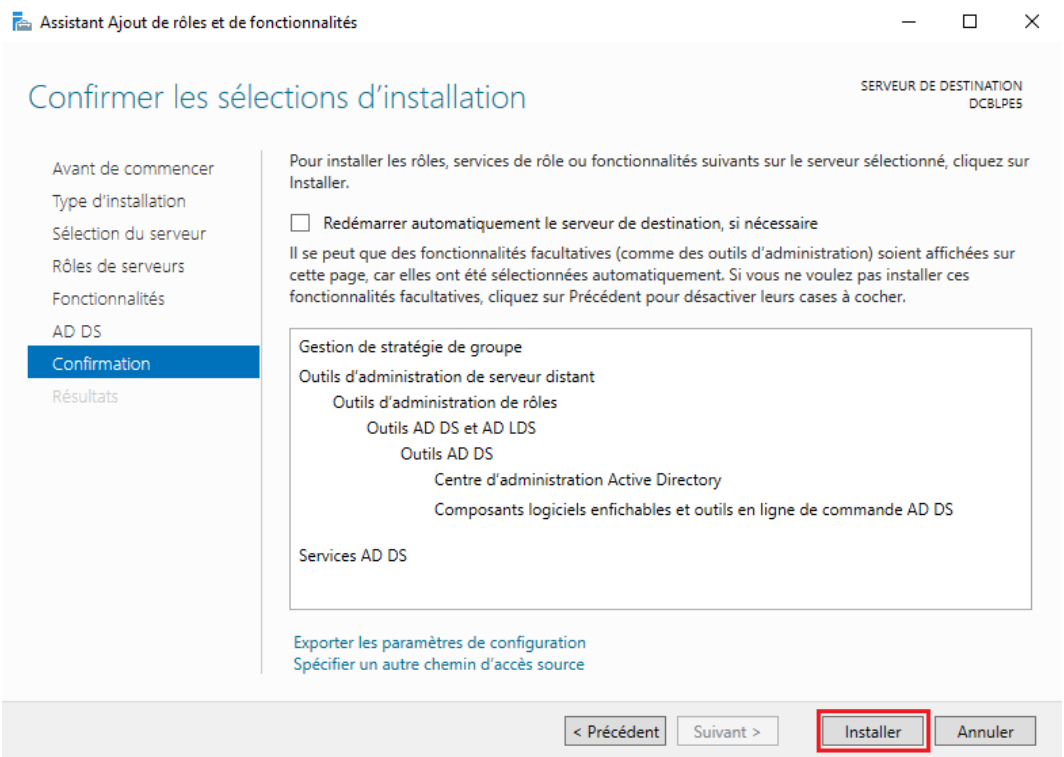
Vous pouvez sélectionner les fonctionnalités de votre choix, dans notre situation aucune fonctionnalités a été ajouter, et cliquez sur suivant :



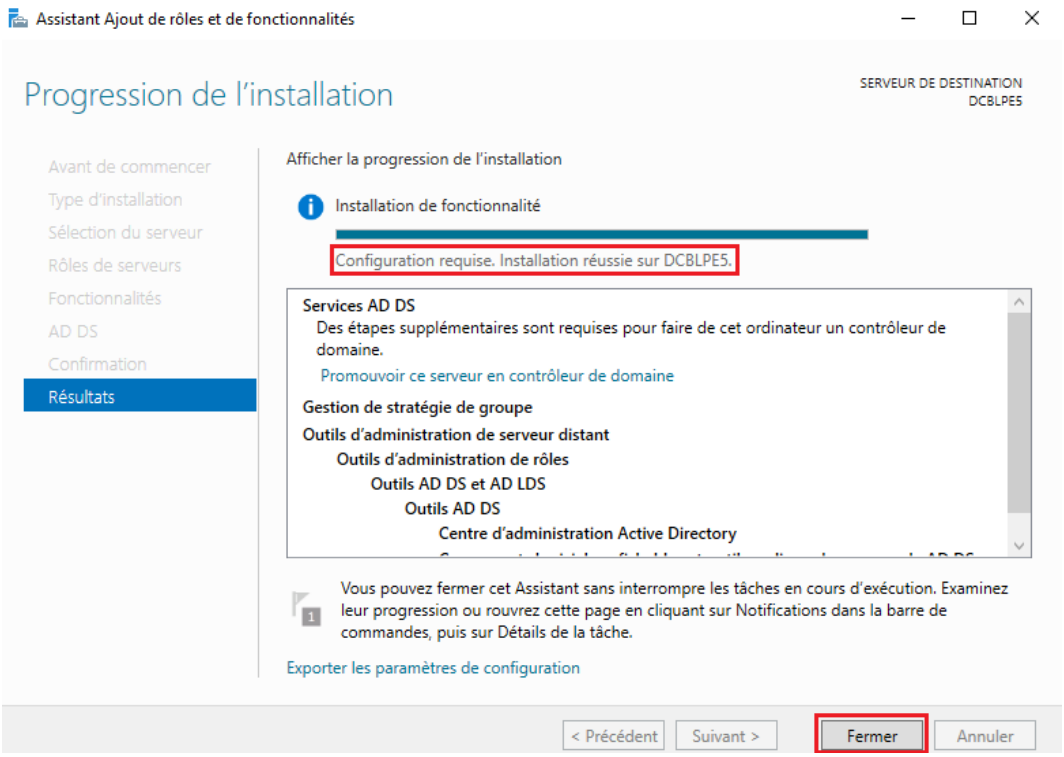
Prenez connaissance de la description du rôle AD DS et cliquez sur suivant :



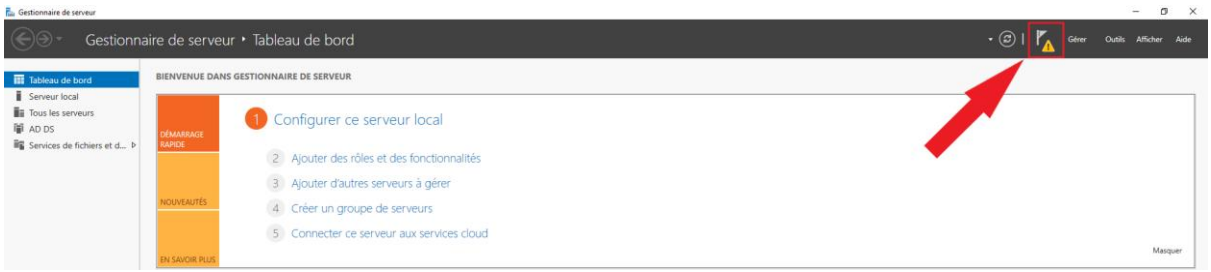
Assurez-vous d'avoir bien sélectionné les éléments à installer et cliquez sur Installer :



Lorsque l'installation est terminée et réussie, vous pouvez fermer l'assistant d'ajout de rôles et de fonctionnalités en cliquant sur fermer, car comme vous pourrez le voir, une configuration est requise :

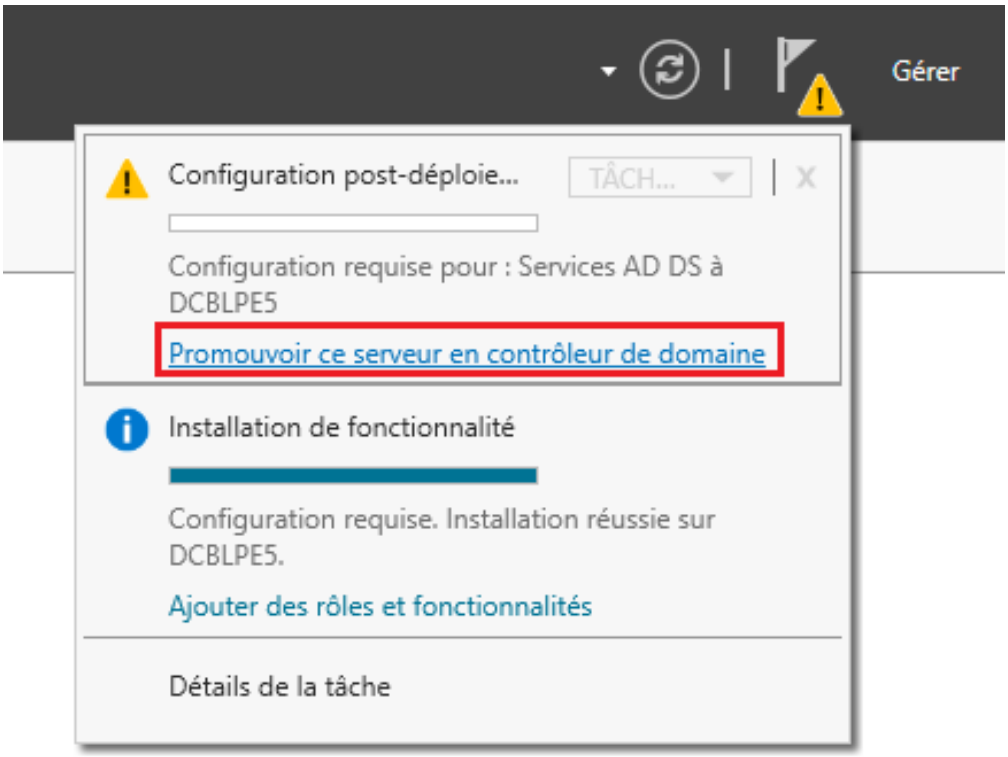


En haut à droite, vous remarquerez un triangle jaune à côté d'un drapeau, cliquez sur ce dernier :

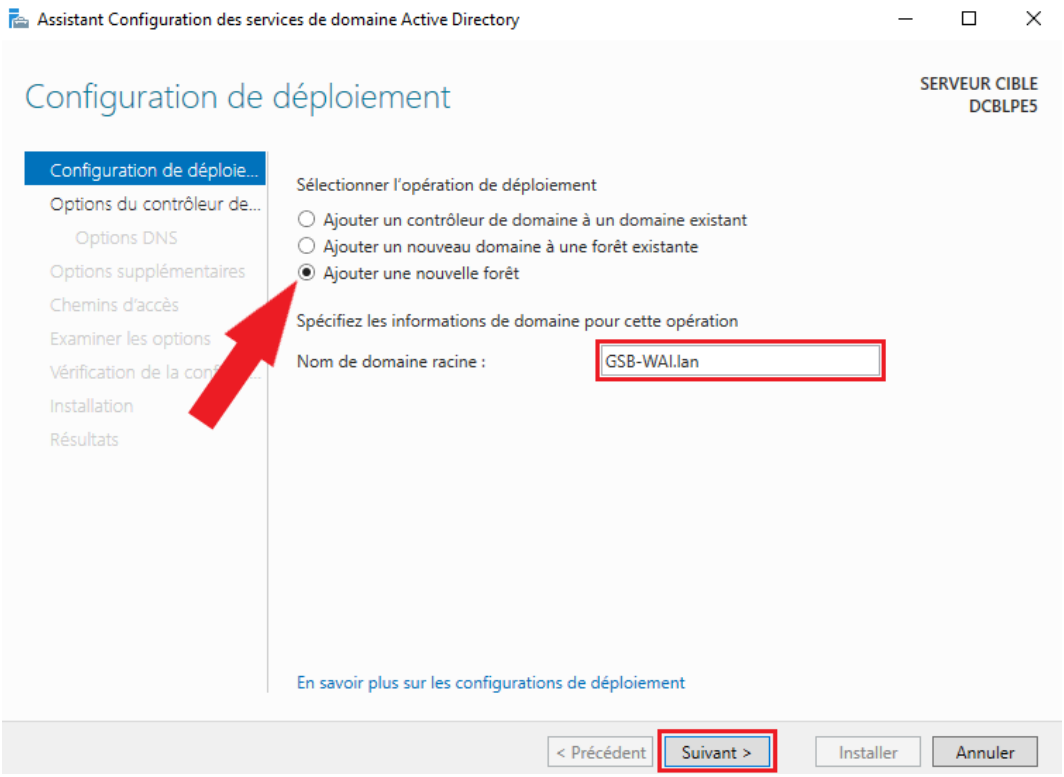


Promotion du serveur en DC

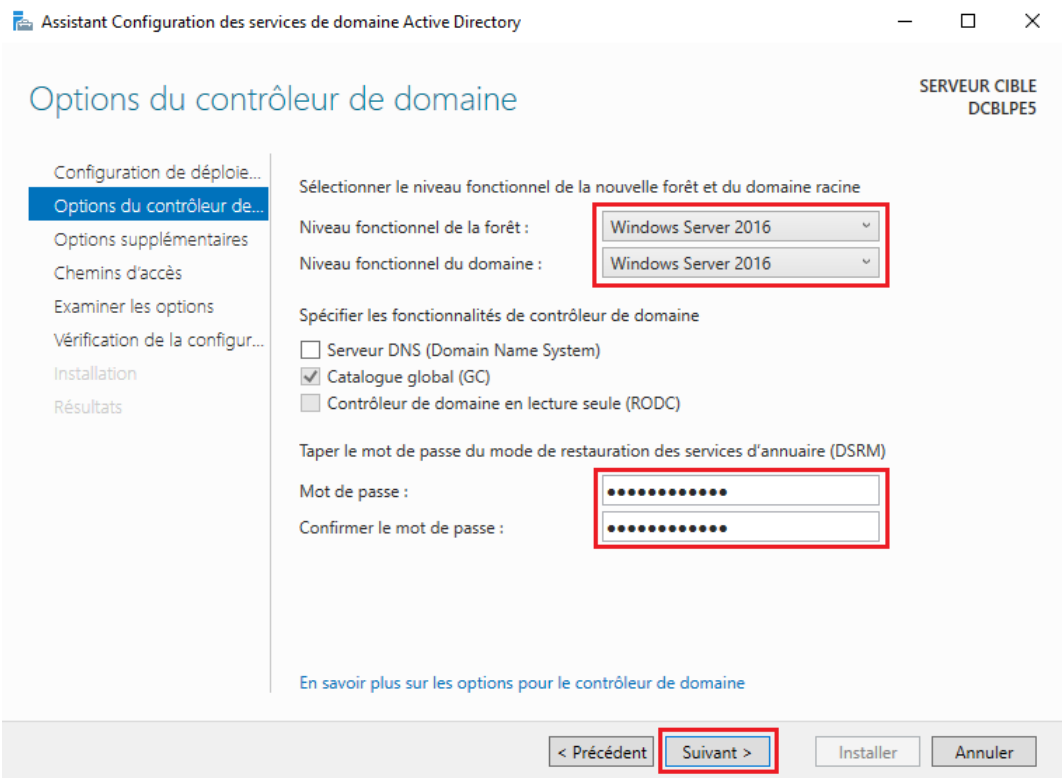
Vous pourrez voir une configuration post déploiement qui consiste à promouvoir votre serveur en contrôleur de domaine, cliquez sur promouvoir ce serveur en contrôleur de domaine :



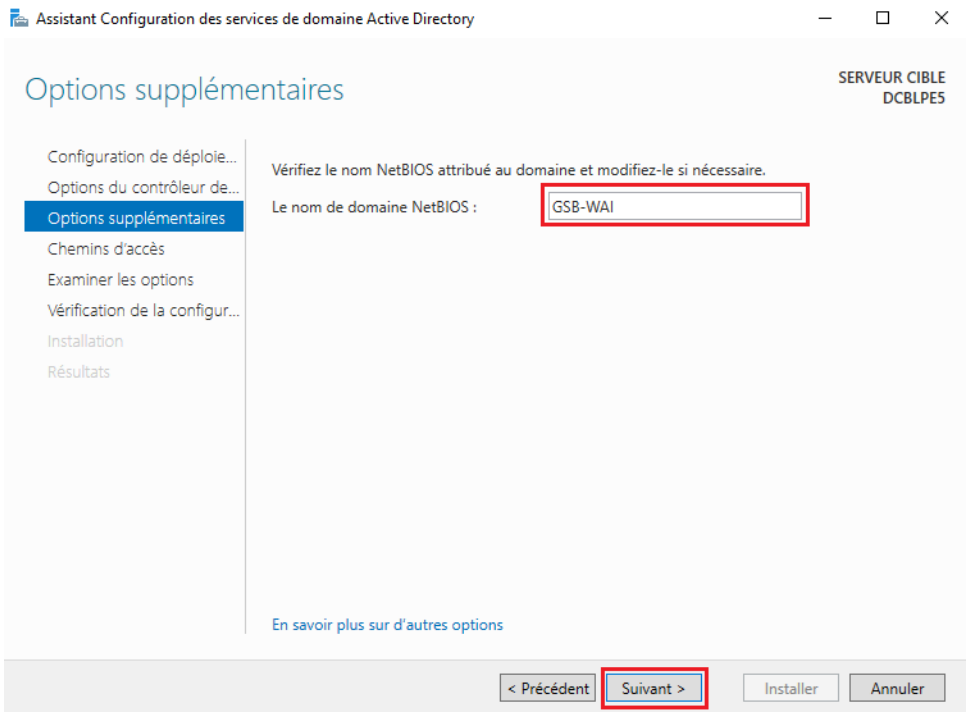
L'assistant de configuration des services de domaine Active Directory se lance et vous demande de choisir une configuration de déploiement. Dans notre situation, n'ayant pas de forêt ou de domaine déjà existant, nous allons créer une nouvelle forêt :



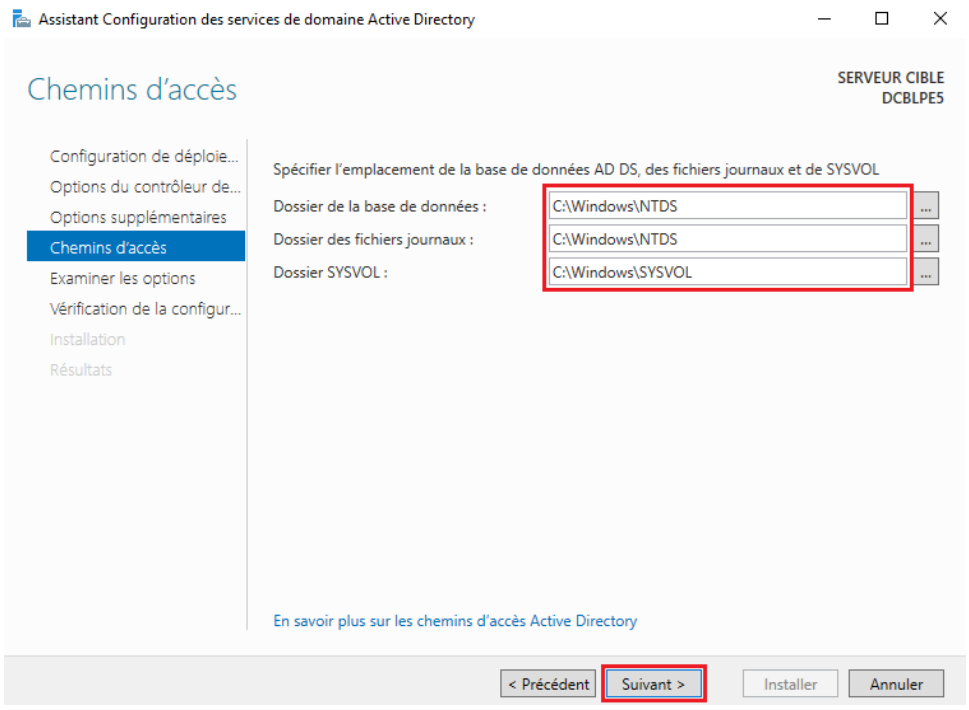
Il vous faudra ensuite choisir les différentes options du contrôleur de domaine :



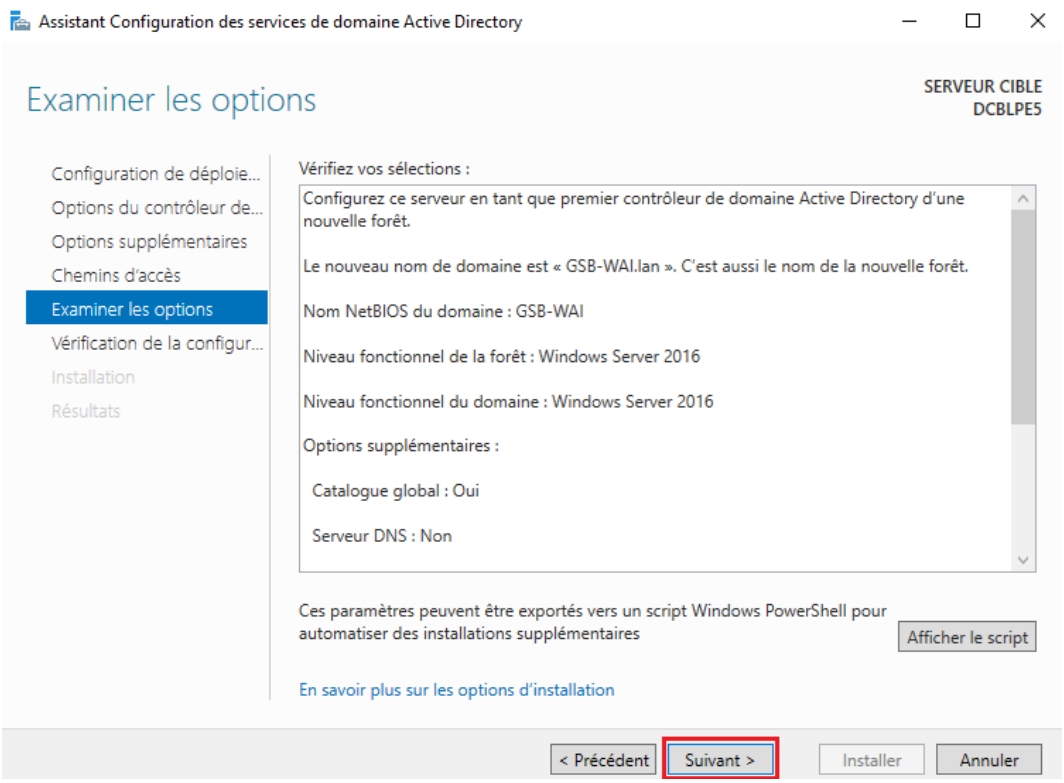
Vérifiez votre nom de domaine :



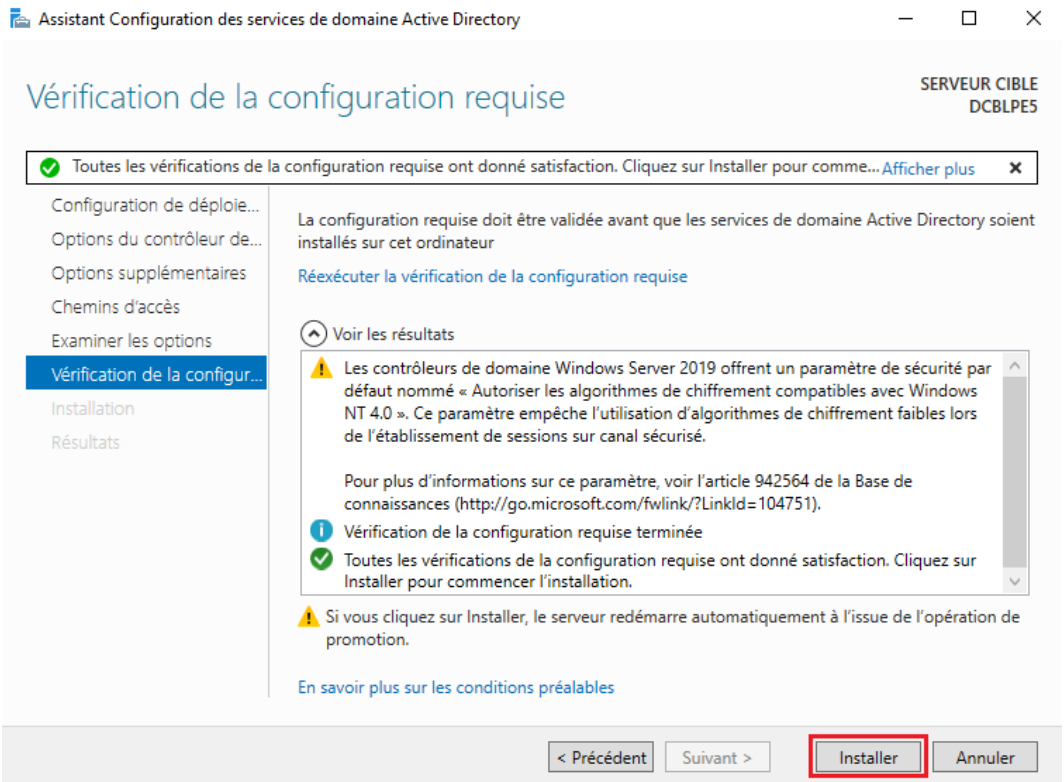
Vous devrez spécifier l'emplacement de la base de données AD DS, des fichiers journaux et de SYSVOL. Dans notre situation, nous laisserons les chemins par défaut :



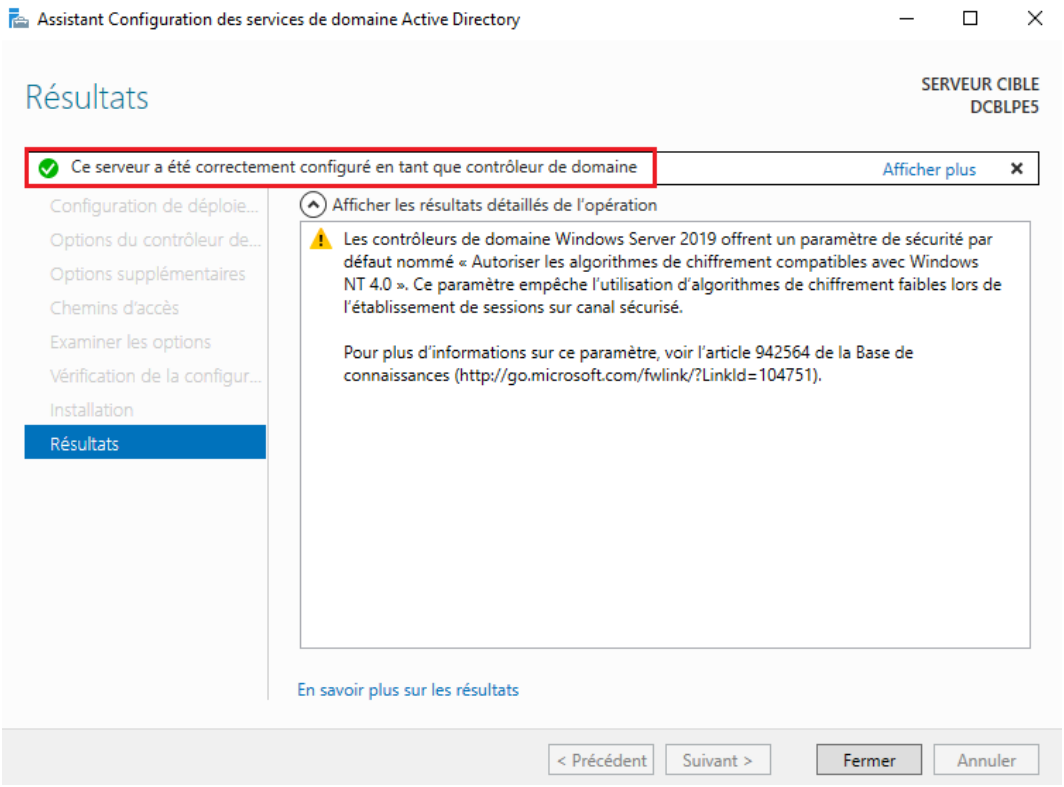
Vous devrez confirmer vos sélections :



L'assistant de configuration des services de domaine Active Directory vérifie ensuite la conformité de la configuration et vous pourrez installer la configuration :



Le serveur va redémarrer automatiquement afin d'appliquer les modifications :



Conclusion

En résumé, la mise en place du contrôleur de domaine Active Directory sous Windows Server a été réalisée avec succès. Cette configuration permet une gestion centralisée des utilisateurs, des groupes et des ressources, ainsi que l'implémentation d'options avancées telles que l'authentification unique et la politique de sécurité centralisée. Active Directory renforce la sécurité des environnement Windows, simplifie leurs administrations et les rends plus adaptable à nos besoins informatiques en constante évolution.

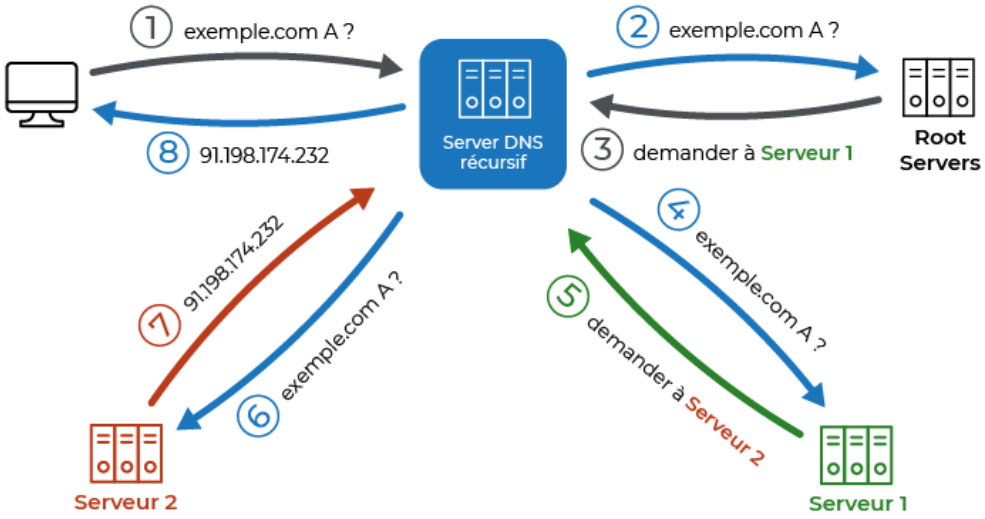
Mode Opératoire DNS

Ce mode opératoire explique comment installer et configurer le rôle serveur Domain Name System sur un serveur Windows Server 2019.

Rappel

Lorsque vous entrez une adresse dans votre navigateur web, celui-ci envoie une demande pour obtenir l'adresse IP du serveur web associé à cette adresse. Par exemple, si vous visitez *www.exemple.com*, votre navigateur demande à votre serveur DNS local l'adresse IP du serveur nommé *www* dans la zone DNS *exemple.com*. En envoyant une requête de type *A* au serveur DNS configuré sur votre poste, il recherche cette information. Si votre serveur DNS ne la possède pas, il se tourne vers un autre serveur DNS, souvent un des serveurs racines, qui gèrent les enregistrements de la zone *"."*.

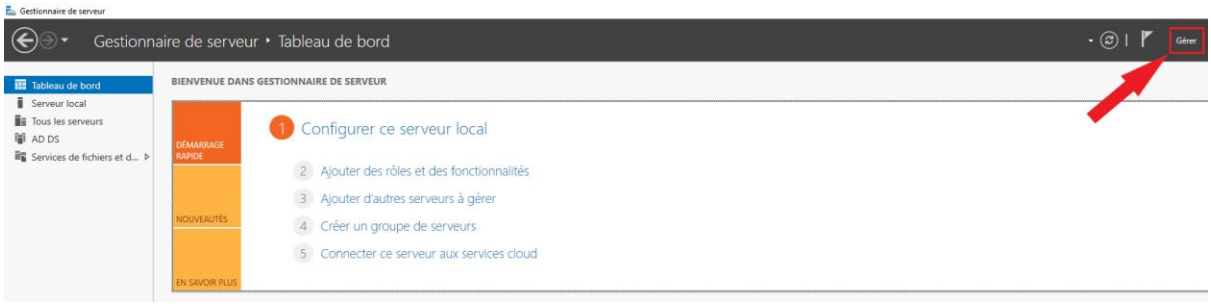
Dans cette zone, on trouve les serveurs des zones *".com"*, *".fr"*, et ainsi de suite pour toutes les extensions de noms de domaine. Votre serveur interroge alors le serveur de nom de la zone *".com"* pour trouver le serveur DNS de la zone *".com"*. Ce processus se répète jusqu'à ce que la zone *exemple.com* soit trouvée, renvoyant finalement l'enregistrement *A* correspondant au champ *www* de sa zone, ce qui devrait vous donner l'adresse IP 91.198.174.232.



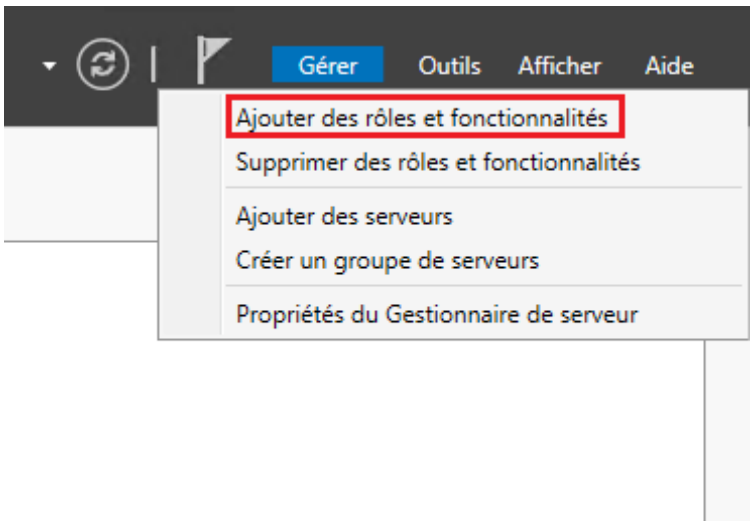
Source : https://openclassrooms.com/fr/

Installation du service DNS

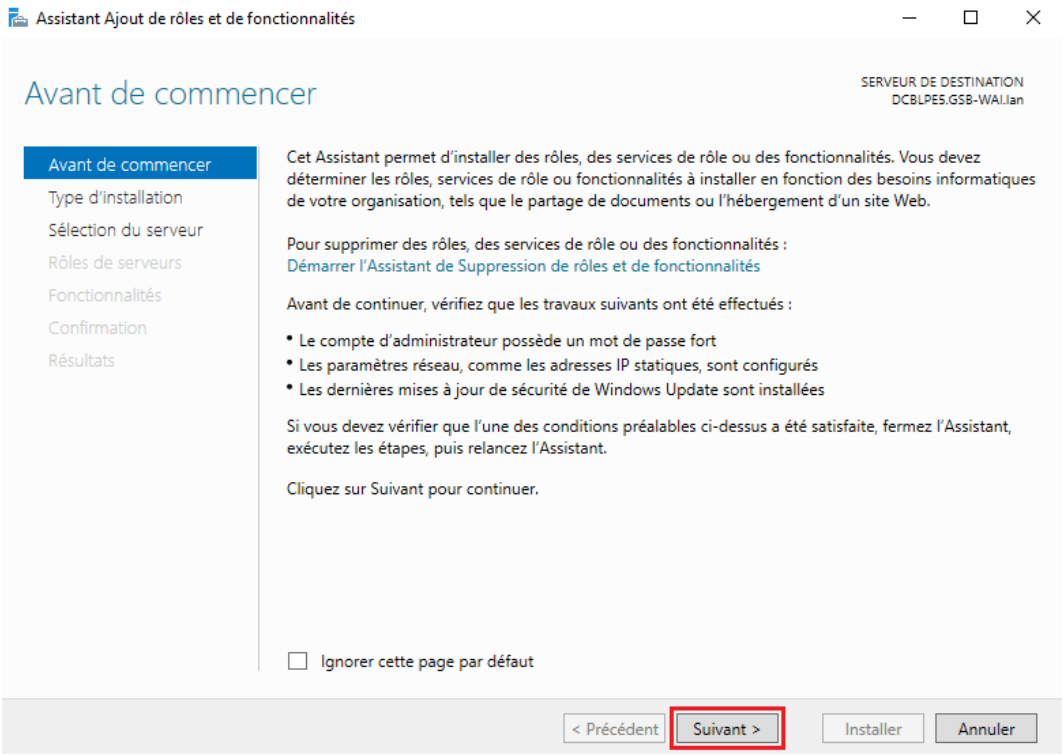
Après avoir installé Windows Server 2019, il vous faudra ouvrir le gestionnaire de serveur (s'il ne s'est pas ouvert automatiquement), cliquer sur Gérer :



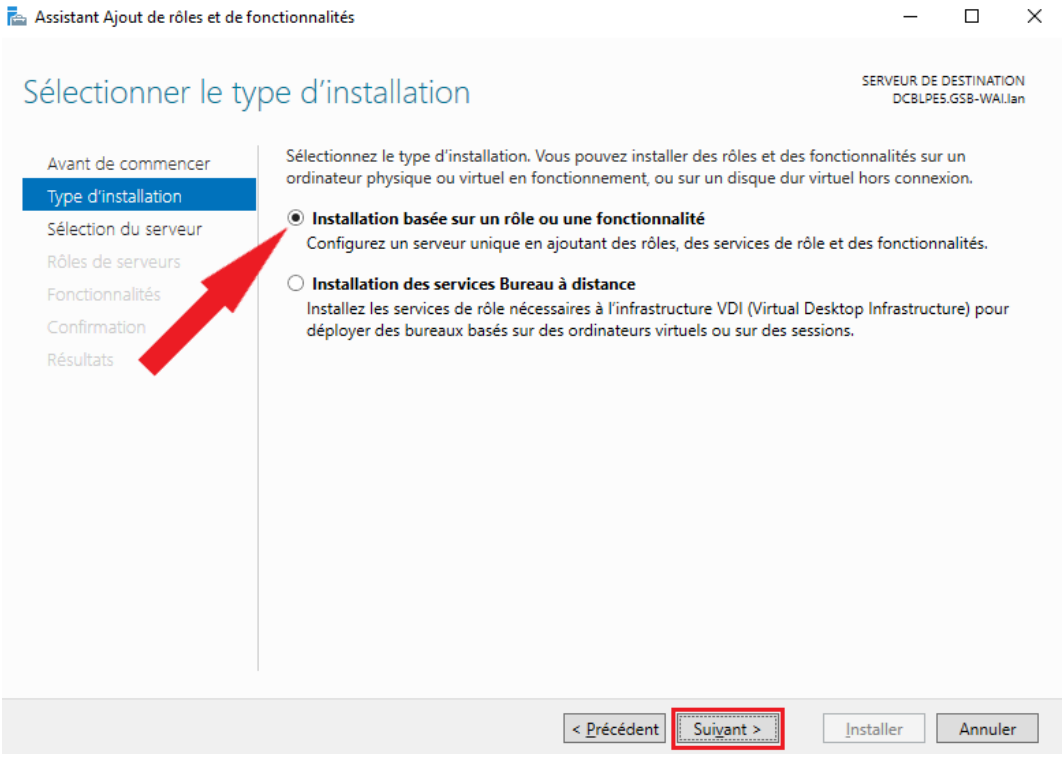
Puis ajouter des rôles et fonctionnalités :



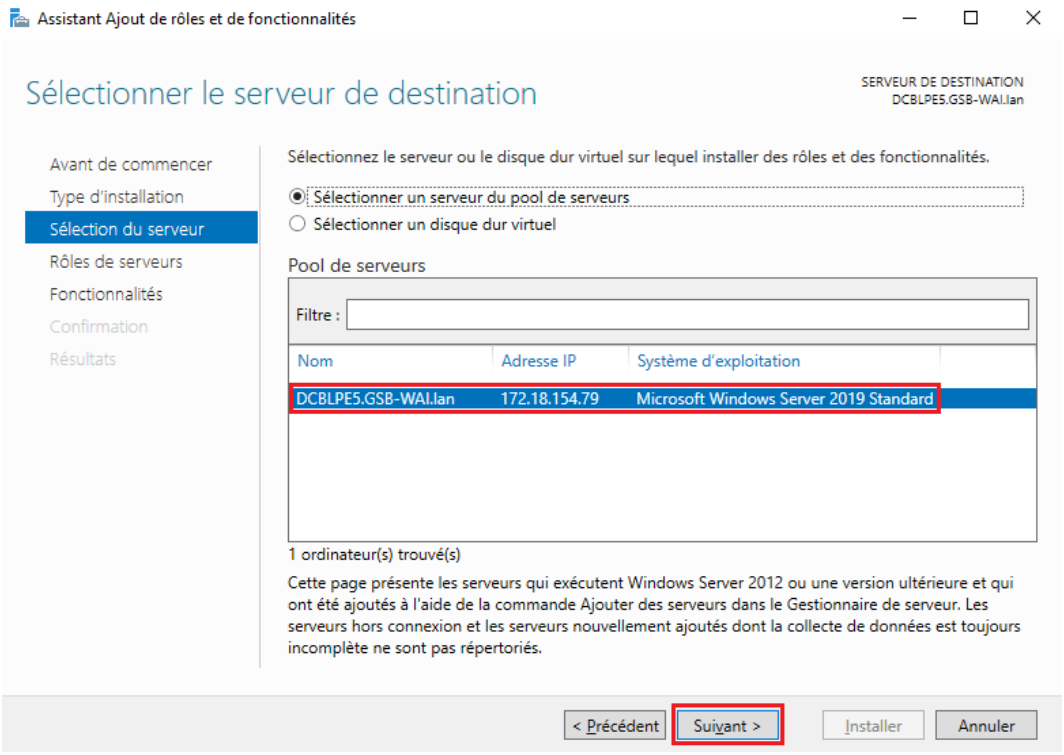
Dans l'assistant d'Ajout de rôles et de fonctionnalités, lisez attentivement les informations qui vous sont présentées et cliquez sur suivant :



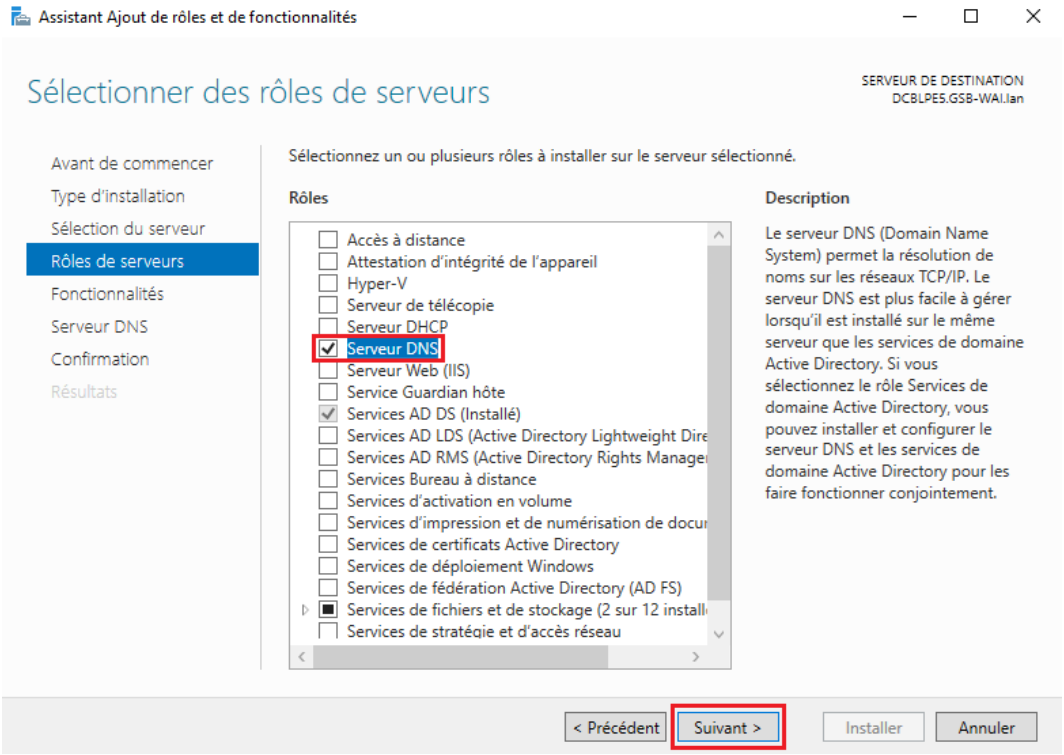
Sélectionner le type d'installation de votre choix, dans cette situation, on choisira une Installation basée sur un rôle ou une fonctionnalité. Ensuite, cliquez sur suivant :



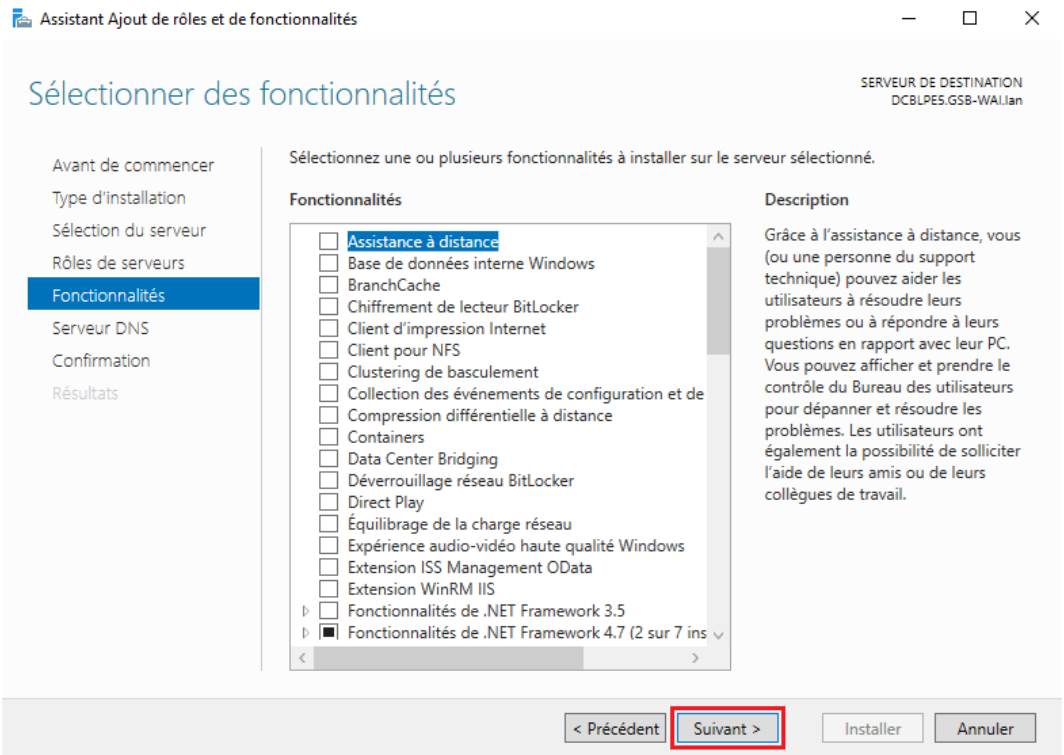
On sélectionne le serveur de destination et on clique sur suivant :



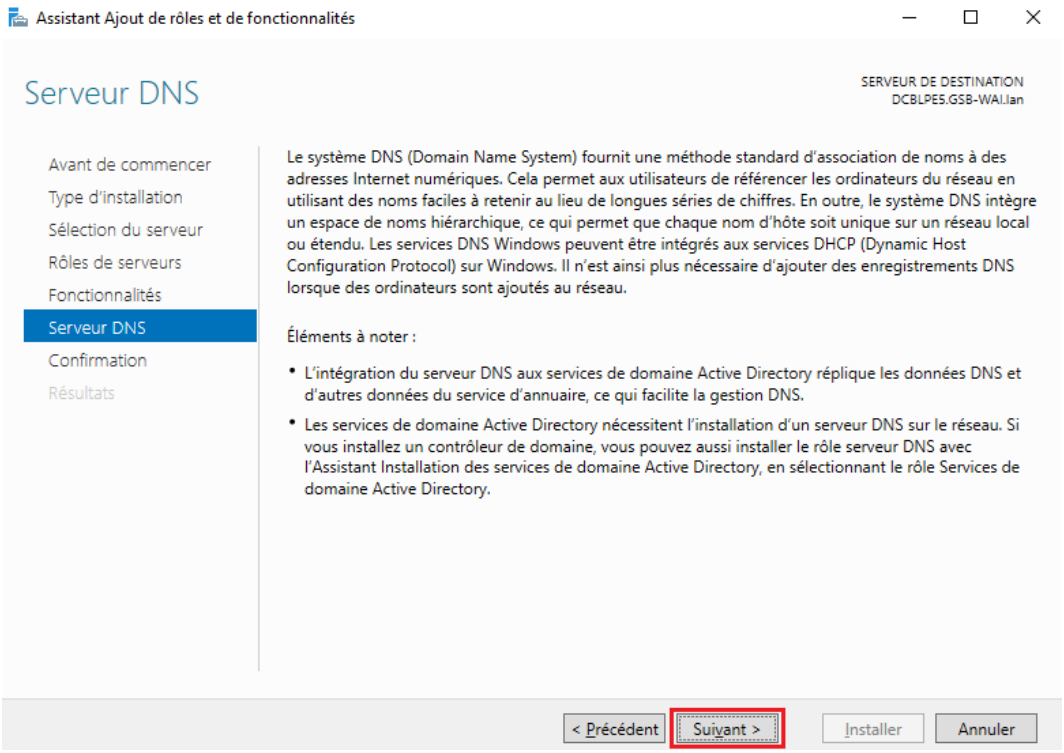
On sélectionne ensuite le ou les rôles qu'on souhaite installer, en l'occurrence, on souhaite installer toute la pile DNS et on clique sur suivant :



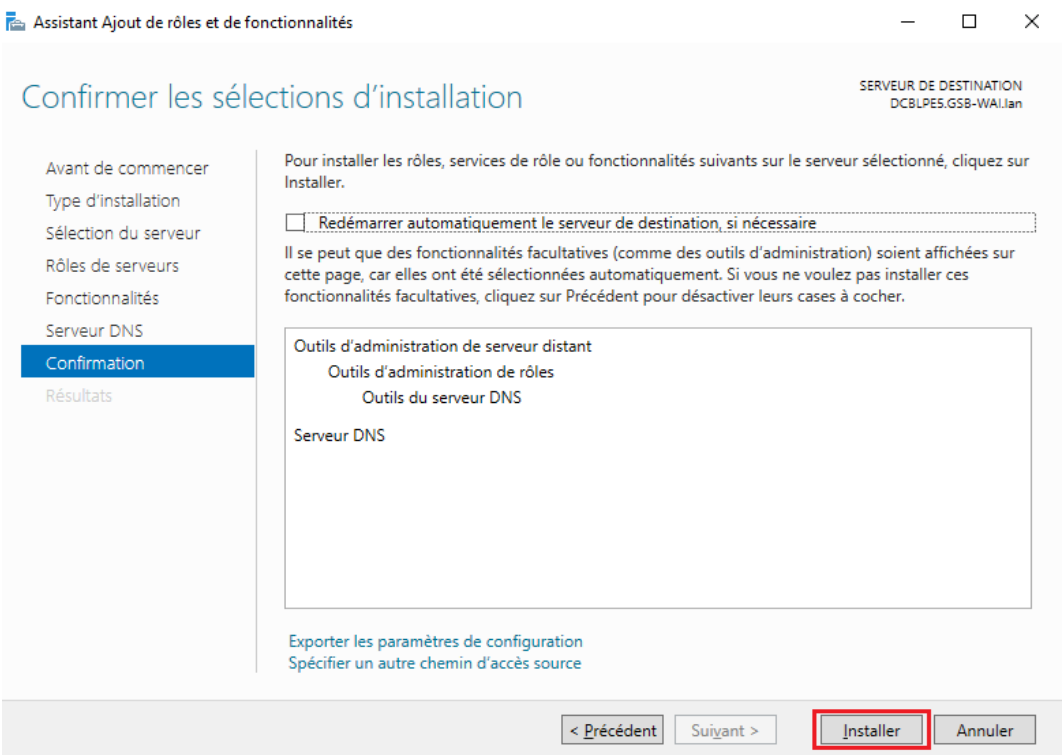
Vous pouvez sélectionner les fonctionnalités de votre choix, dans notre situation aucune fonctionnalités a été ajouté, et cliquez sur suivant :



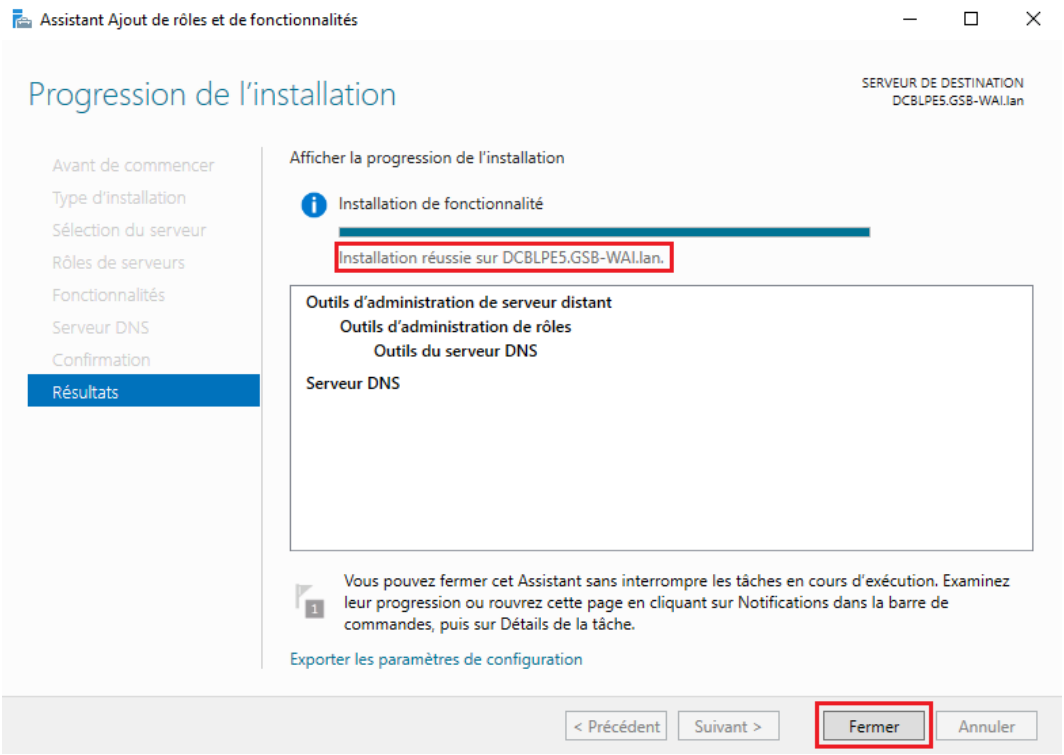
Prenez connaissance de la description du rôle Serveur DNS et cliquez sur suivant :



Assurez-vous d'avoir bien sélectionné les éléments à installer et cliquez sur Installer :

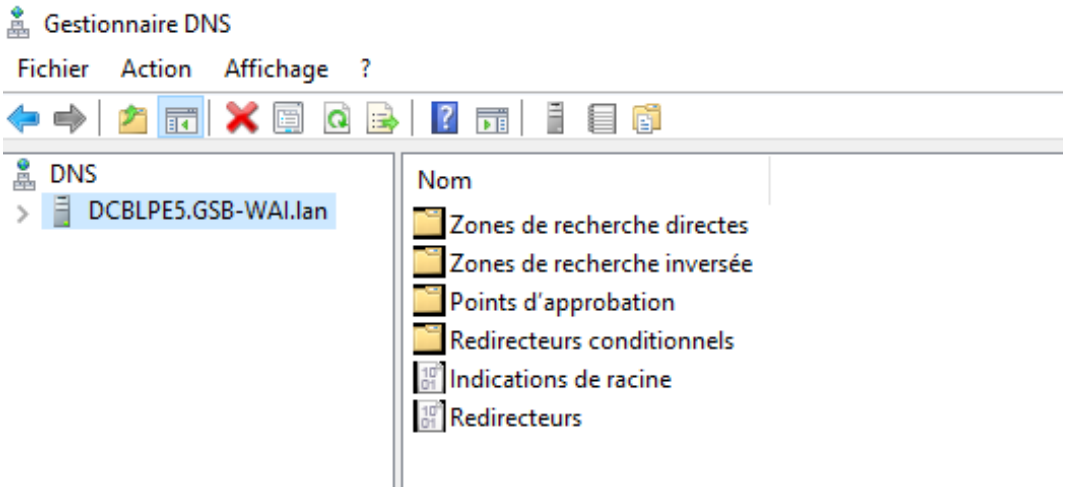


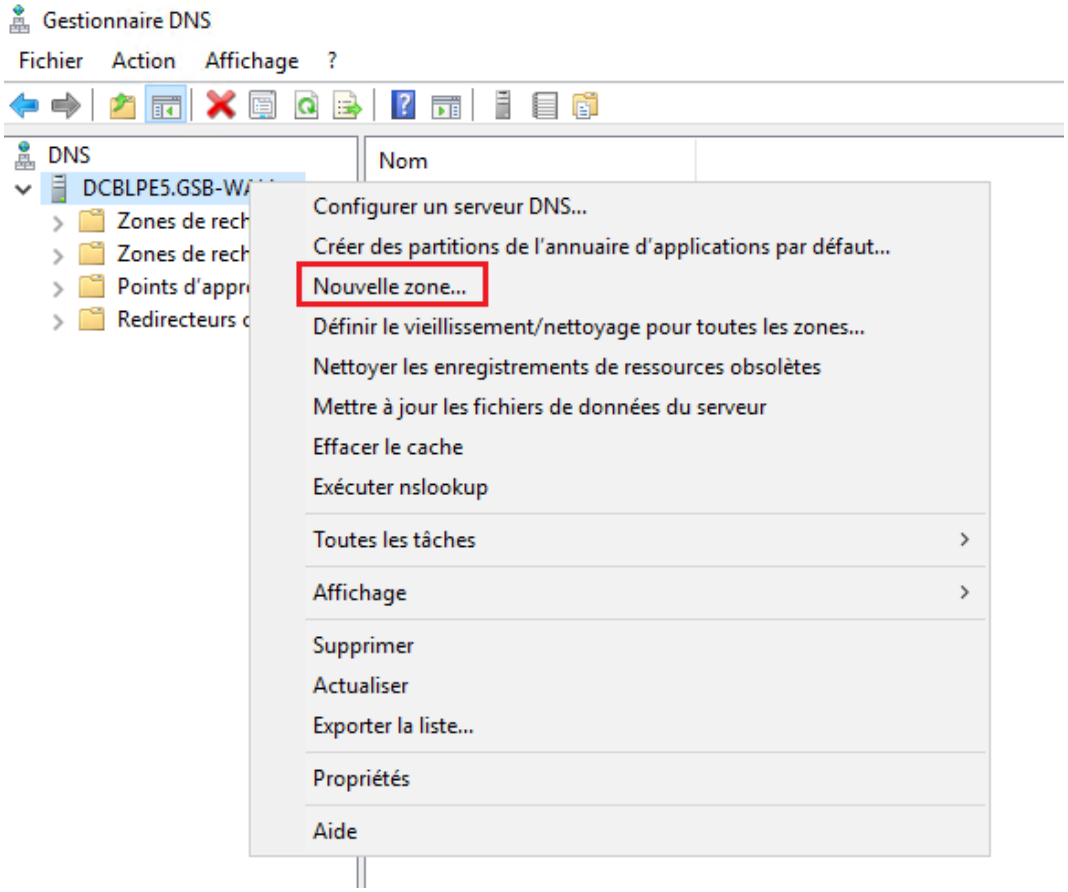
Lorsque l'installation est terminée et réussie, vous pouvez fermer l'assistant d'ajout de rôles et de fonctionnalités en cliquant sur fermer :



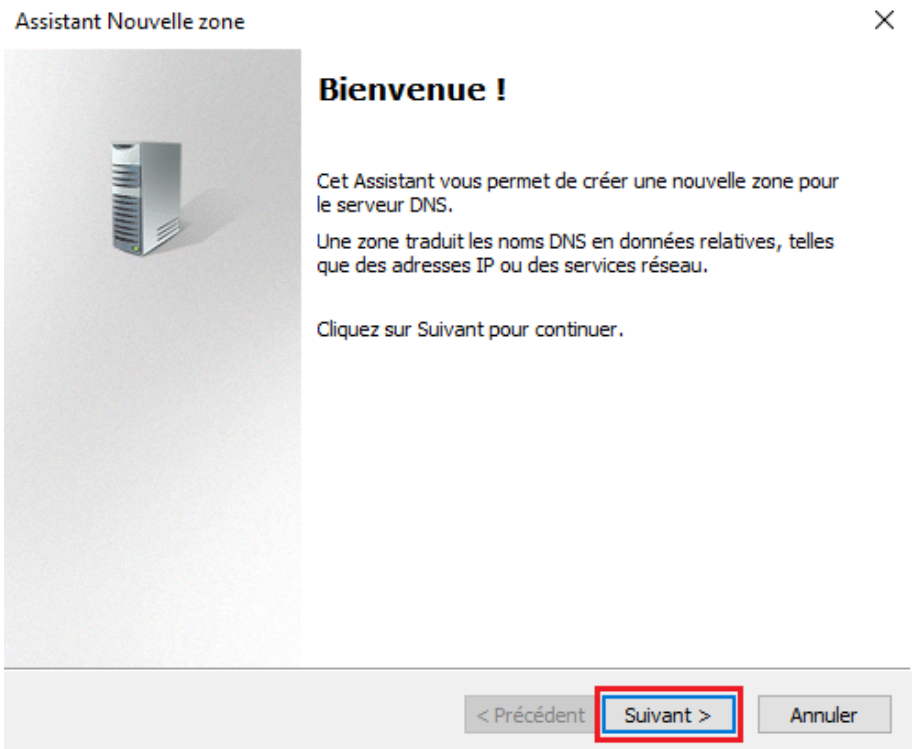
Création d'une zone directe

Maintenant, entrons dans la configuration de ce service. Pour ce faire, nous utiliserons la console spécifique à l'administration du rôle DNS, qui nous permettra de créer les diverses zones requises pour son fonctionnement. La première étape consiste à créer une **zone directe**, permettant l'association d'un nom à une adresse IP. Sur le nom du serveur, faites un clic droit, puis cliquez sur « Nouvelle zone » :

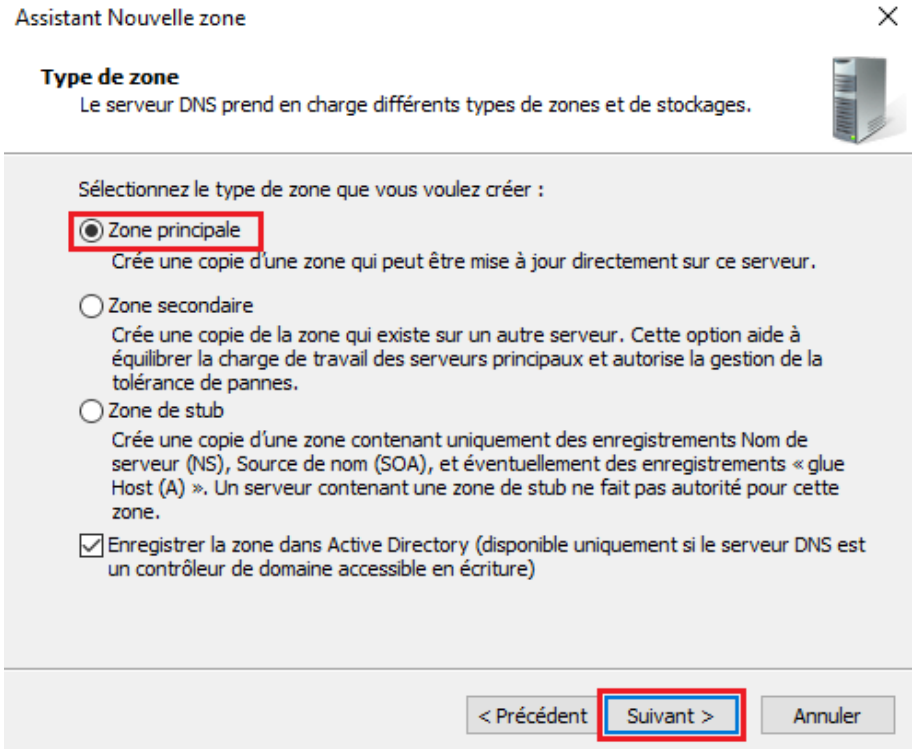




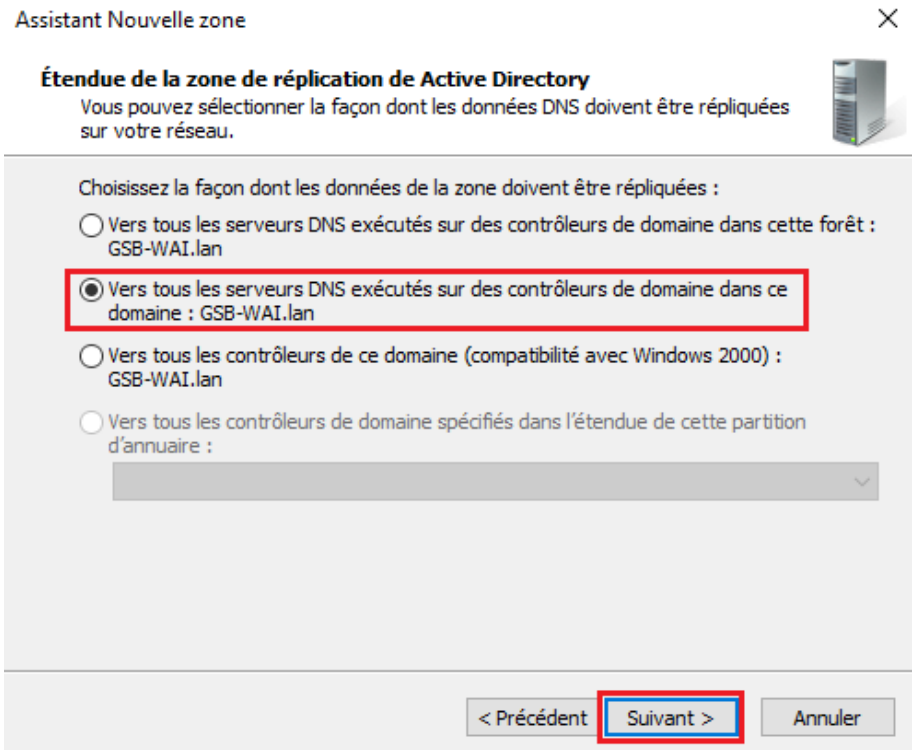
L'assistant de création de nouvelle zone s'exécute, il permet comme son nom l'indique de créer une nouvelle zone pour le serveur DNS :



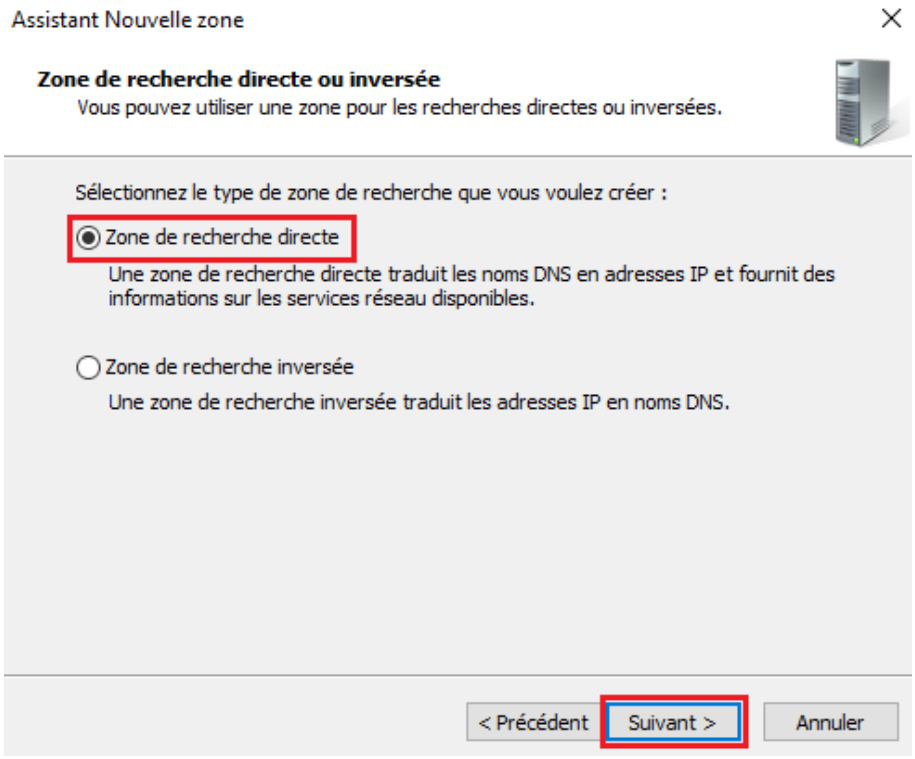
On vient sélectionner le type de zone à créer, en l'occurrence dans notre situation, nous mettrons en place une zone principale :



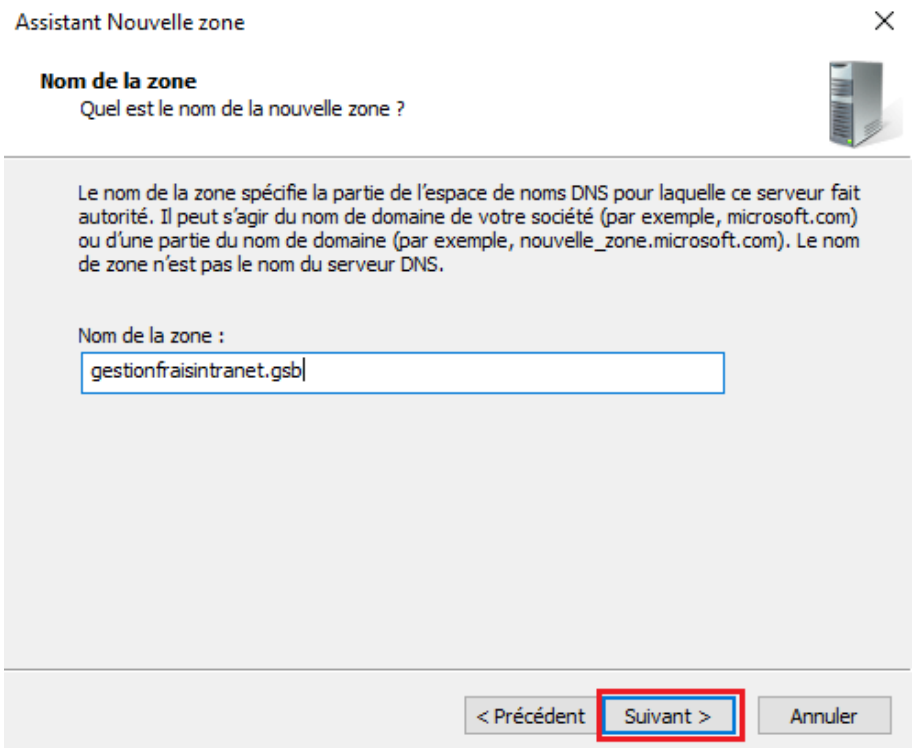
Il faudra ensuite définir la zone de réplication de la zone DNS sur le réseau. Pour notre situation, nous allons définir l'étendue vers tous les serveurs DNS exécutés sur des contrôleurs de domaine dans notre domaine :



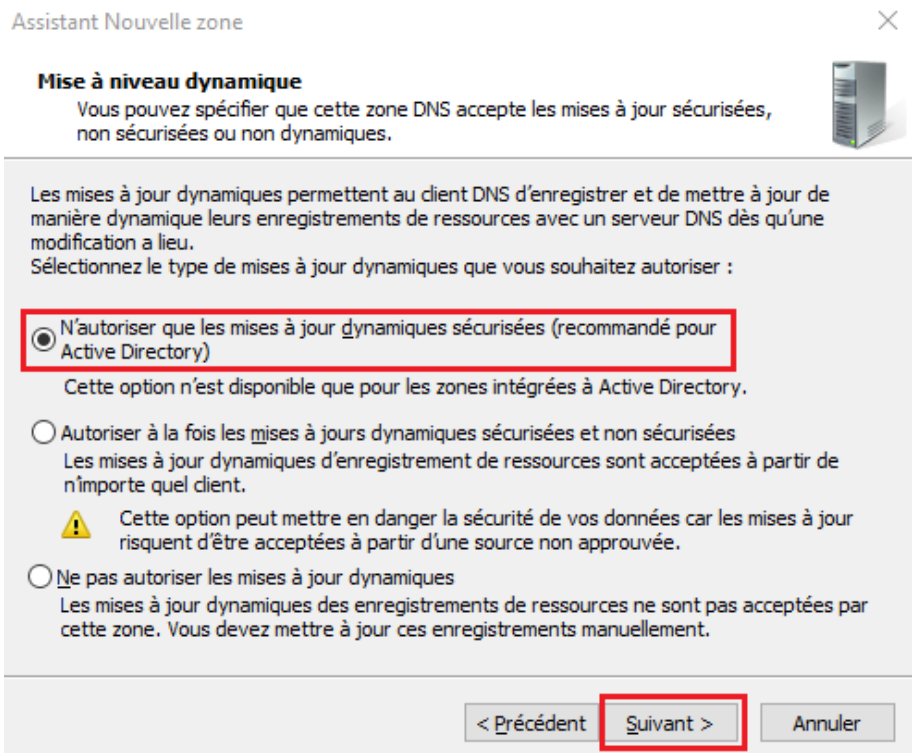
On sélectionne ensuite si on veut créer une zone directe ou une zone inversée :



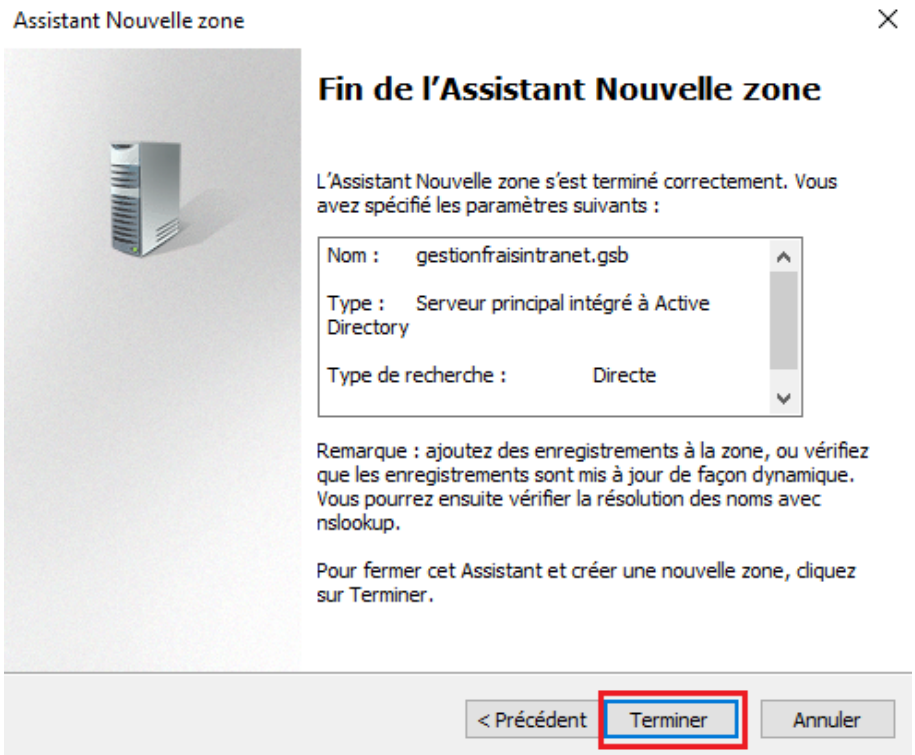
On définit ensuite le nom de cette nouvelle zone :



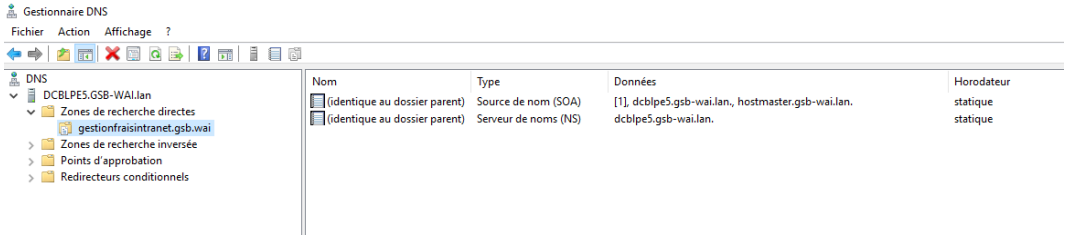
Et on choisit la politique de mise à niveau (dynamique) :



Une console récapitulative nous informe que la zone a été créer et configurer correctement, on clique sur terminer :

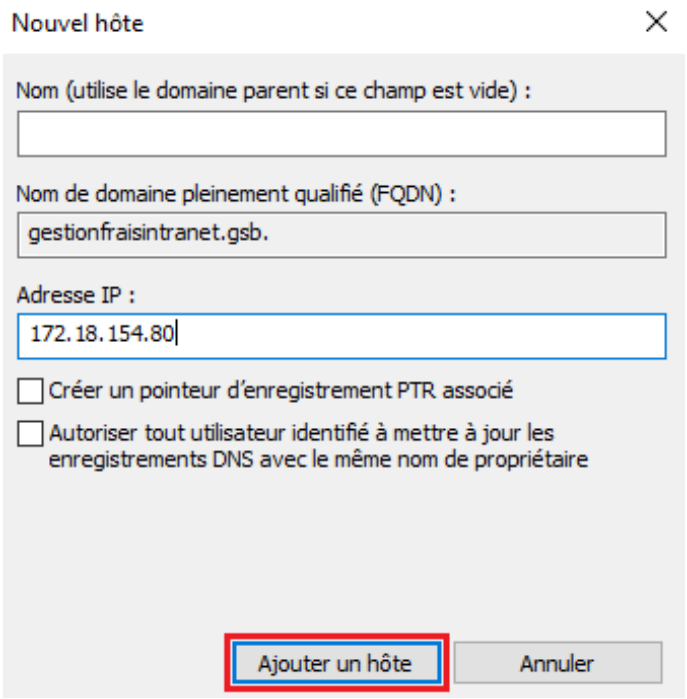


On pourra retrouver la nouvelle zone fraichement créer dans l'onglet Zones de recherche directes :

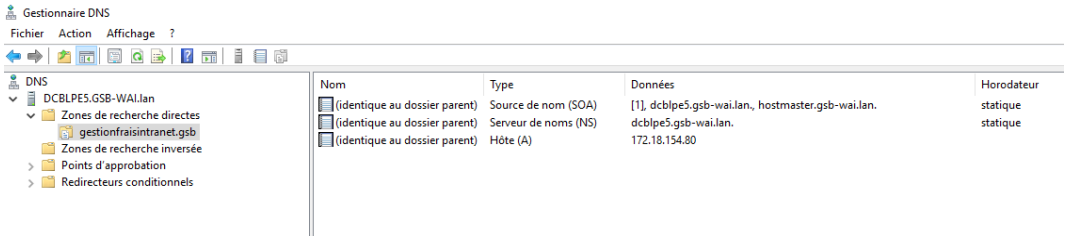


Création d'un nouvel enregistrement

Dans notre zone, se trouvent uniquement deux enregistrements, qui identifient le serveur faisant autorité (SOA) et le serveur de noms (NS). Il serait pertinent de créer notre enregistrement intranet pour **gestionfraisintranet.gsb**. Pour ce faire, cliquez avec le bouton droit dans la fenêtre de droite (ou sur le nom de la zone) et choisissez « Nouvel hôte A ou AAAA ». Les enregistrements A sont utilisés pour les adresses IPv4 et les AAAA pour les adresses IPv6 :



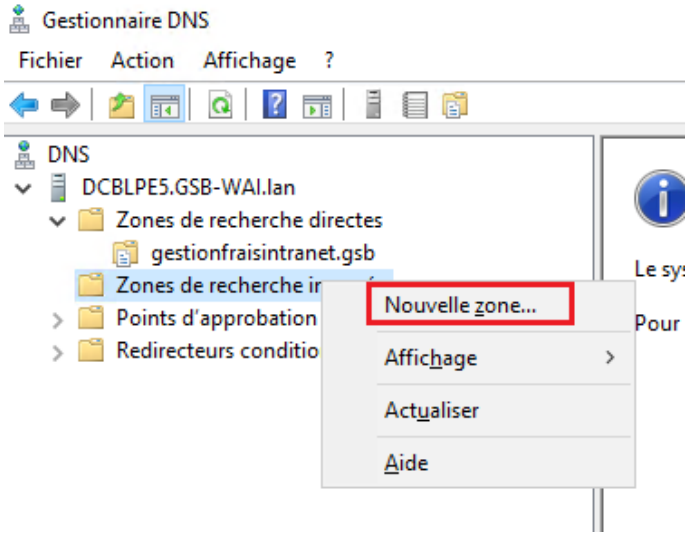
On peut voir ce nouvel enregistrement maintenant :



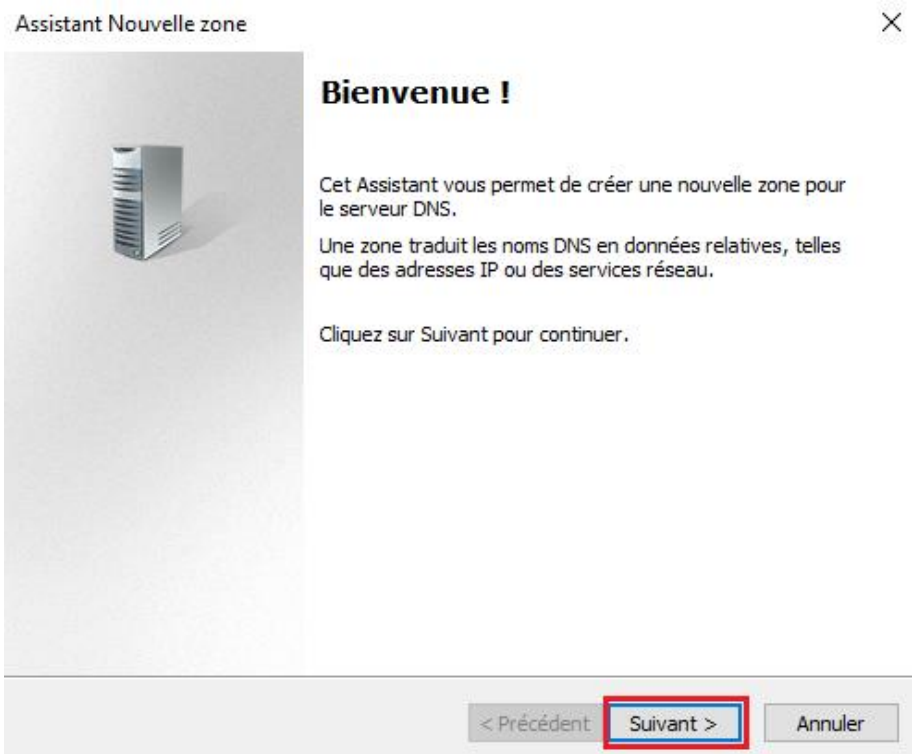
Création d’une zone inversée

Maintenant, nous allons établir une zone inversée. Contrairement à la zone directe, cette dernière associe une adresse IP à un nom, offrant ainsi une approche inverse. Elle vérifie que le nom sélectionné dans une zone directe est bien lié à l'adresse IP. Ainsi, en cas de modification de l'adresse du serveur DNS configuré sur votre serveur DNS, il est possible d'interroger un DNS sur une adresse IP.

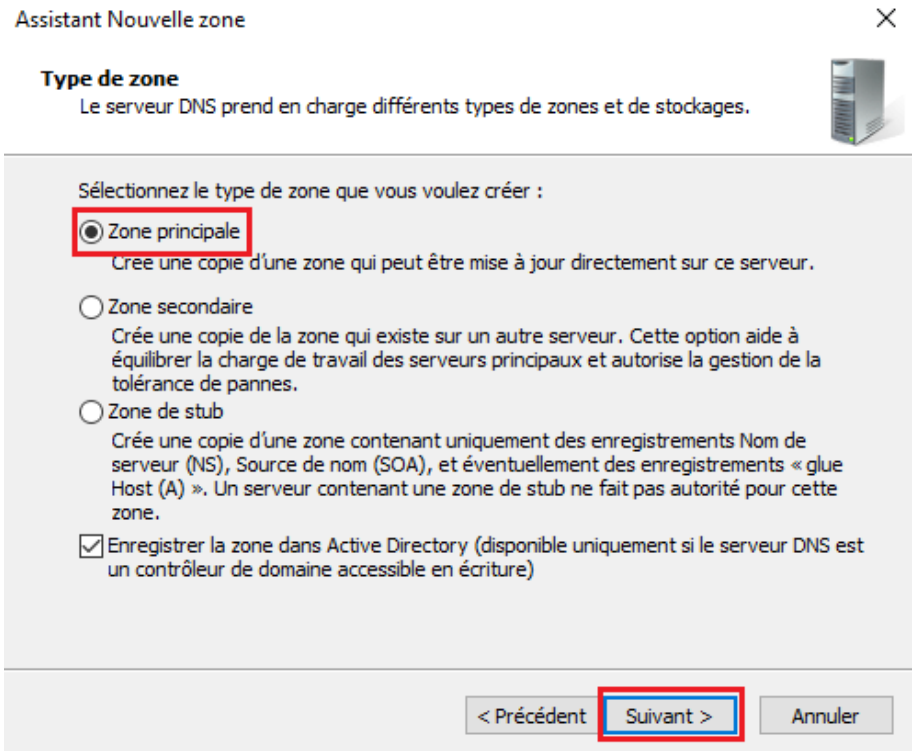
Pour cela, on fait un clic droit sur Zones de recherches inversées et on clique sur « Nouvelle zone » :



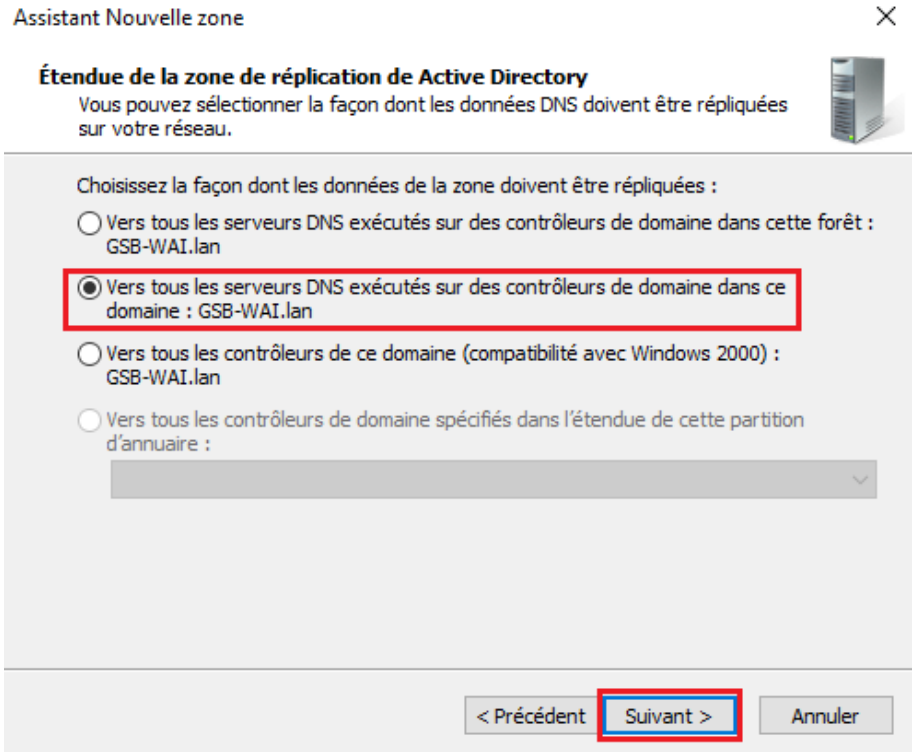
Encore une fois, l'assistant de création de zone s'exécute :



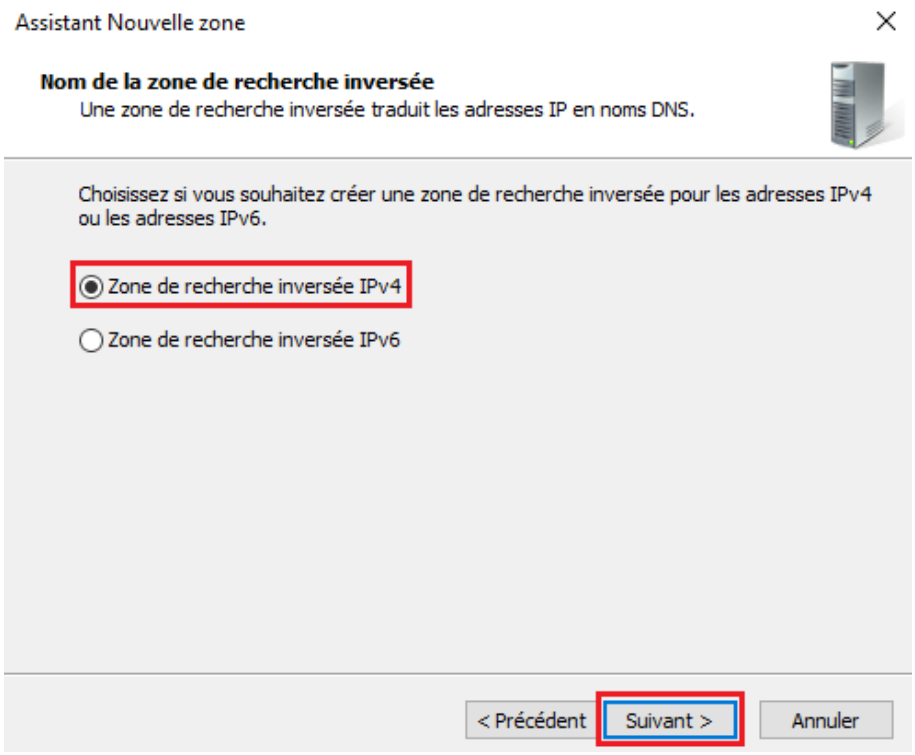
On choisit le type de zone :



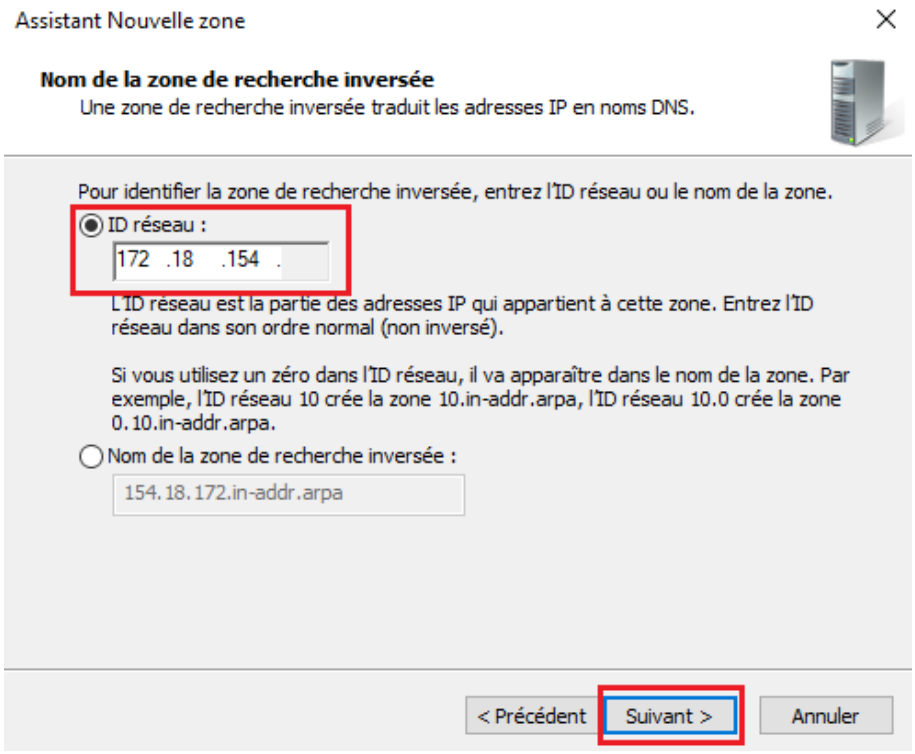
On définit l'étendue de la zone de réplication d'Active Directory :



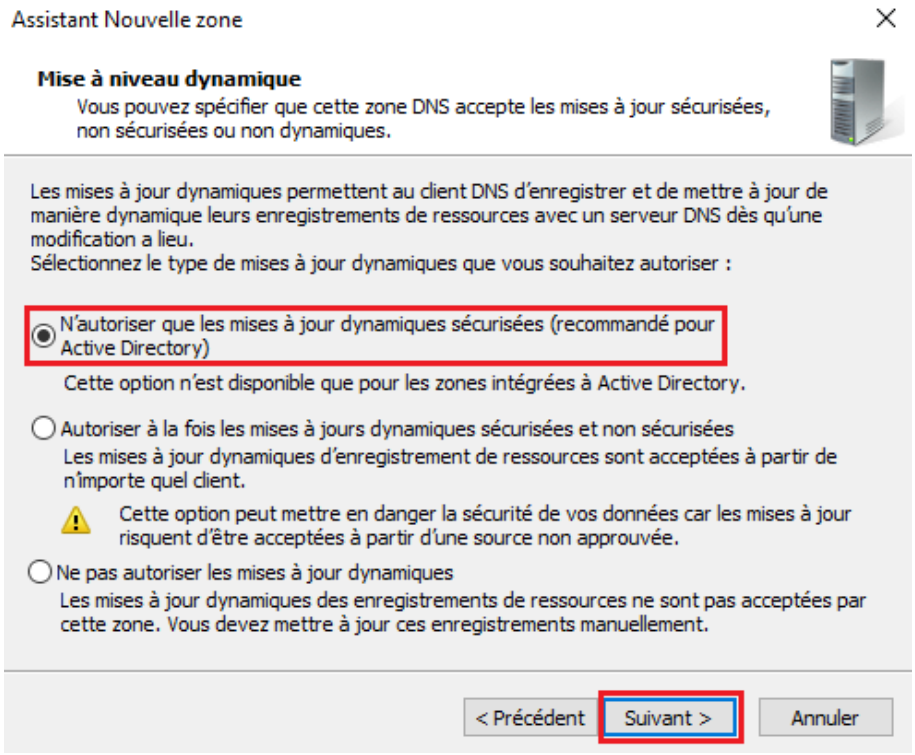
On choisit si on veut créer une zone de recherche inversée pour IPv4 ou IPv6, dans notre cas, ce sera IPv4 :



On définit l'identifiant réseau (@ réseau) qui permettra d'identifier la zone de recherche inversée :



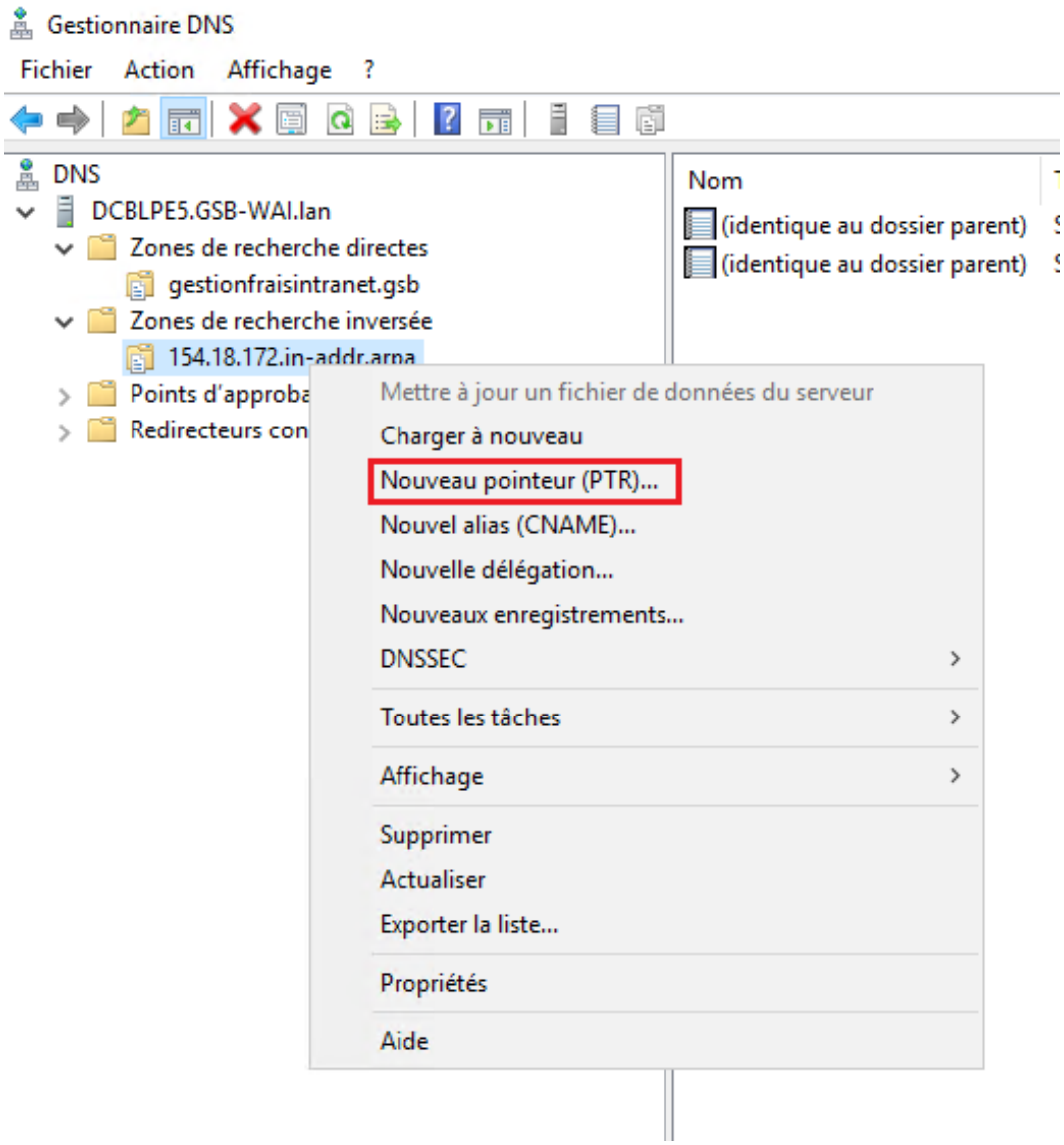
On définit la politique de mise à niveau (dynamique) :



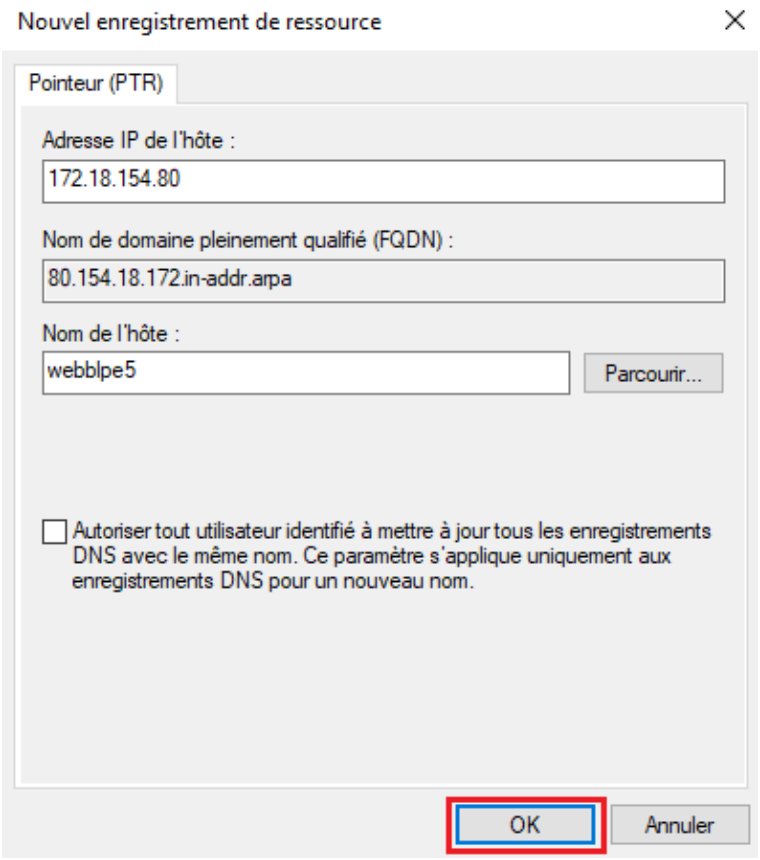
L'assistant indique de la zone inversée à correctement était configuré, on clique sur terminer :

Création d’un nouvel enregistrement PTR

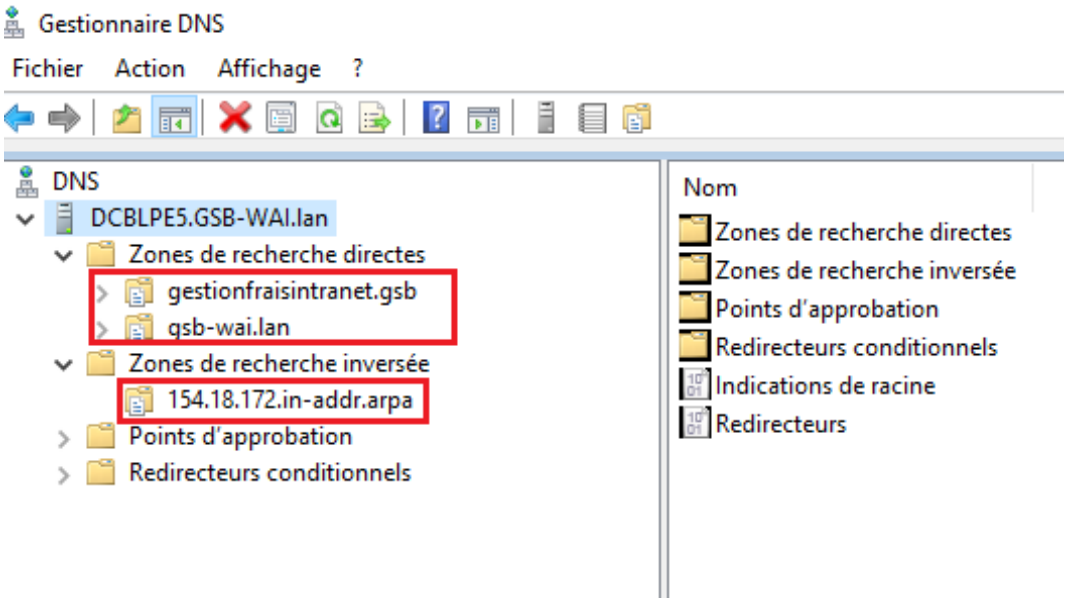
De manière similaire à la configuration d'une zone directe, notre configuration par défaut ne comporte que deux enregistrements. Ainsi, nous allons créer un enregistrement PTR pour notre serveur Web. Pour ce faire, il suffit de faire un clic droit sur la nouvelle zone de recherche inversée, puis de sélectionner « Nouveau pointeur » :



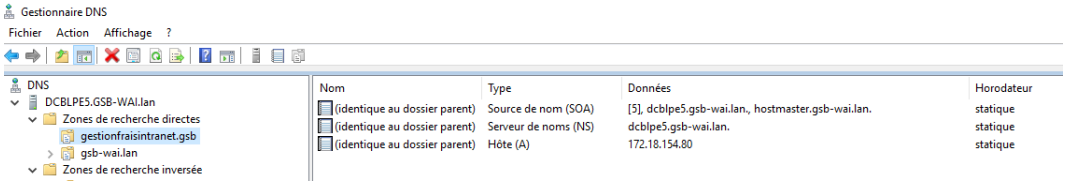
On rentre l'adresse IP de l'hôte et son nom :



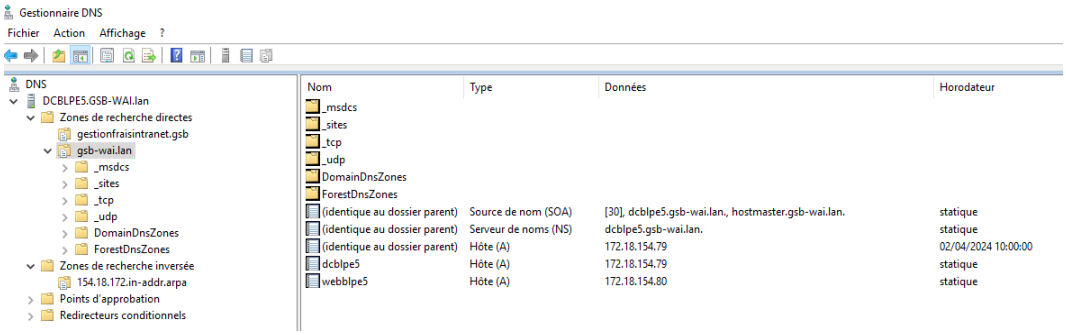
Pour notre situation, nous avons 2 zones de recherche directes et 1 zone de recherche inversée :



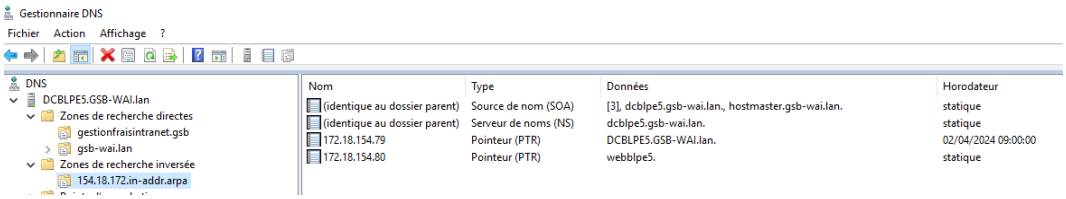
Un enregistrement (supplémentaire) dans la zone directe gestionfraisintranet.gsb pour le site intranet de GSB :



3 enregistrements (supplémentaire) dans la zone directe gsb-wai.lan pour le domaine de GSB :



Et 2 enregistrements (supplémentaire) pour les pointeurs du DC et du Serveur Web dans la zone de recherche inversée 172.18.154.0 :



Conclusion

En résumé, l'installation et la configuration réussies d'un serveur DNS sous Windows Server permet de mettre en place un système de résolution de noms de domaine efficace au sein d'un réseau. Cette configuration essentielle garantit aux utilisateurs un accès facile aux ressources et services disponibles. Grâce à la configuration minutieuse des zones, des enregistrements et des paramètres du serveur DNS, le réseau dispose désormais d'une infrastructure solide et fiable qui favorisera la communication et la connectivité au sein d'un environnement Windows et plus.

Mode Opératoire Serveur Web sous Debian12

Dans ce mode opératoire, nous allons explorer la configuration d'un serveur Web "LAMP" sous Debian 12. Ce serveur sera prêt à héberger divers types de contenus tels que des sites Internet ou des applications.

Le terme "LAMP" fait référence à un ensemble de logiciels essentiels pour créer un serveur Web robuste. Il se compose de quatre composants principaux :

- **L pour Linux** : Cela désigne le système d'exploitation sur lequel le serveur est installé. Dans notre cas, nous utiliserons Debian 11.
- **A pour Apache** : Il s'agit du serveur Web qui gère les requêtes HTTP et sert les fichiers Web aux utilisateurs.
- **M pour MySQL/MariaDB** : Ce composant est un système de gestion de base de données relationnelle. Il stocke et organise les données nécessaires au fonctionnement des sites Web et des applications.
- **P pour PHP** : PHP est un langage de script côté serveur largement utilisé pour générer des contenus dynamiques sur les sites Web. Il est souvent utilisé en combinaison avec des bases de données pour créer des applications Web interactives.

En combinant ces quatre éléments, nous mettrons en place un environnement stable et fonctionnel pour répondre aux besoins d'hébergement du laboratoire.

Installation du serveur Apache

La première étape est la mise à jour des paquets. La commande « **apt update && apt upgrade** » permet de mettre à jour la liste des paquets disponibles et à installer les mises à jour disponibles pour les paquets installés sur un système Debian. Mettre à jour les paquets disponibles est essentiel pour garantir la sécurité, la stabilité et les performances du système. Les mises à jour peuvent inclure des correctifs de sécurité pour protéger le système contre les vulnérabilités connues, des améliorations de fonctionnalités pour optimiser les performances, ainsi que des corrections de bogues pour garantir la stabilité du système. En maintenant les logiciels à jour, on réduit les risques de failles de sécurité et on assure le bon fonctionnement global du système. Pour commencer, on va passer en mode super utilisateur (root) :

```
adm_n.wai-lune@webblpe5:~$ su -
Mot de passe :
root@webblpe5:~#
```

Utiliser l'utilisateur **root** plutôt que « sudo » est préférable dans certains cas car cela donne un accès direct et complet au système sans les restrictions de privilèges imposées par « sudo », mais cela nécessite une prudence accrue pour éviter les actions accidentelles ou malveillantes qui pourraient endommager le système.

Maintenant on peut utiliser la commande évoquer précédemment :

```
root@webblpe5:~# apt update && apt upgrade
```

Ensuite, on installe le paquet « apache2 » avec le paramètre « -y » pour une installation automatique et silencieuse :

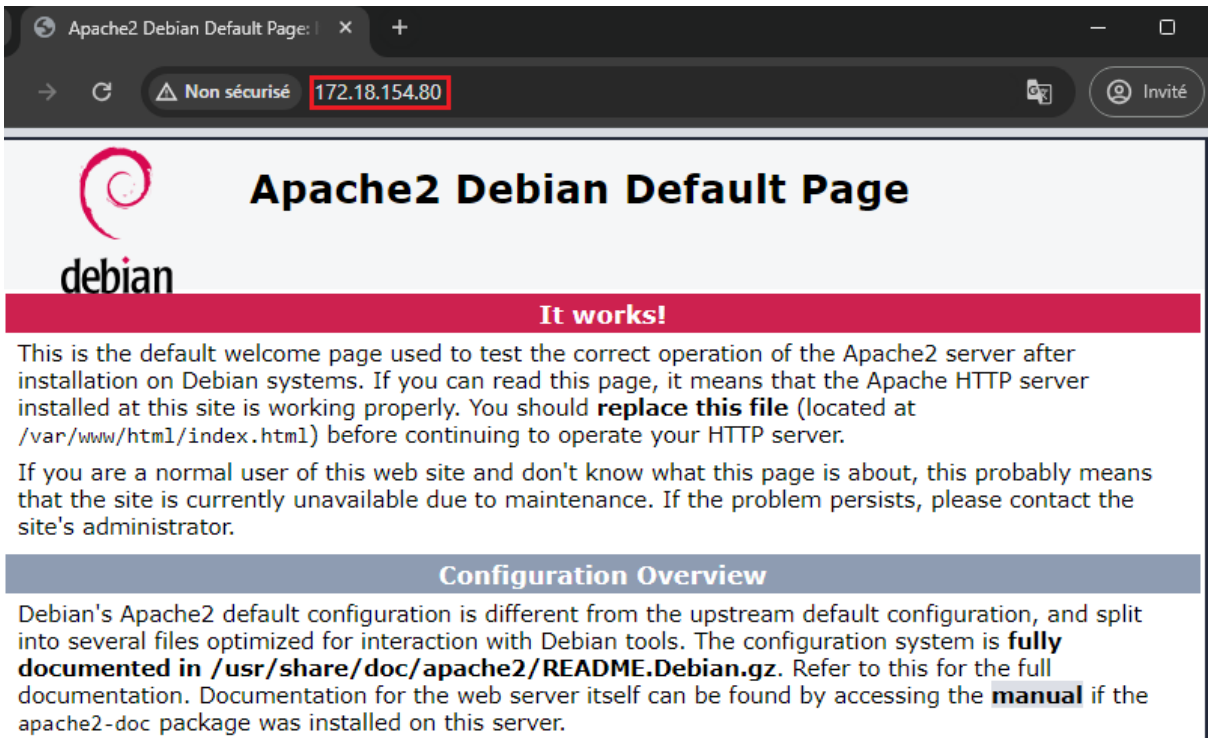
```
root@webblpe5:~# apt install apache2 -y
```

On vient utiliser la commande « systemctl enable apache2 ». Cette dernière configure le système pour démarrer automatiquement le service Apache (apache2) au démarrage du système. Cela garantit que le serveur web sera démarré dès que le système sera lancé, assurant ainsi sa disponibilité plus ou moins continue :

```
root@webblpe5:~# systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
```

On peut d'or et déjà accéder à la page par défaut d'Apache. Pour cela, il suffit de récupérer l'adresse IP du serveur et d'y accéder par un navigateur Web. Par exemple :

```
root@webblpe5:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:50:56:a2:07:25 brd ff:ff:ff:ff:ff:ff
    altname enp11s0
    inet 172.18.154.80/21 brd 172.18.159.255 scope global ens192
       valid_lft forever preferred_lft forever
```



Il faut maintenant activer quelques modules d'Apache indispensables pour faire tourner un site Internet. Ces modules sont les suivants :

```
root@webblpe5:~# a2enmod rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
  systemctl restart apache2
```

```
root@webblpe5:~# a2enmod deflate
Considering dependency filter for deflate:
Module filter already enabled
Module deflate already enabled
```

```
root@webblpe5:~# a2enmod headers
Enabling module headers.
To activate the new configuration, you need to run:
  systemctl restart apache2
```

```
root@webblpe5:~# a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
Module ssl already enabled
```

La commande « **a2enmode** » active les modules Apache « rewrite », « deflate », « headers » et « ssl ». Ces modules ajoutent des fonctionnalités supplémentaires au serveur Apache, telles que :

- **rewrite** : Réécrit les URLs, souvent utilisé pour la gestion des permaliens et les redirections.
- **deflate** : Compresse le contenu pour améliorer les performances en réduisant la taille des fichiers envoyés au client.
- **headers** : Permet la manipulation des en-têtes HTTP, utile pour configurer la sécurité, le cache et les contrôles d'accès.
- **ssl** : Fournit la prise en charge du protocole SSL/TLS pour des connexions sécurisées, permettant la configuration et la gestion des certificats SSL/TLS pour assurer la confidentialité et l'intégrité des données.

Après avoir activé ou désactivé un module, ou modifié la configuration d'Apache, il faut toujours redémarrer/rechargé le service apache2 :

```
root@webblpe5:~# systemctl restart apache2.service
```

On va également venir installer le paquet « apache2-utils » car ce dernier offre divers outils utiles pour la gestion, le dépannage et l'évaluation des performances du serveur Apache et notamment la gestion des fichiers de mots de passe utilisés pour l'authentification **HTTP** de base :

```
root@webblpe5:~# apt install apache2-utils -y
```

Installation du serveur MariaDB

MariaDB est une alternative populaire à MySQL, résultant d'un fork communautaire. L'avantage principal de MariaDB réside dans son statut open source et sa licence GPL, offrant ainsi une transparence et une liberté d'utilisation que MySQL, propriétaire chez Oracle, ne peut garantir malgré sa gratuité. MariaDB bénéficie d'un suivi de développement actif et d'une communauté engagée, ce qui en fait un système robuste et performant. Pour installer MariaDB sur notre serveur, on utilise la commande suivante :

```
root@webblpe5:~# apt install mariadb-server -y
```

Ensuite, une bonne pratique est d'exécuter le script « mariadb-secure-installation » afin de garantir une configuration de base sécurisée de MariaDB dès son installation, réduisant ainsi les risques potentiels liés à des paramètres par défaut moins sécurisés. Le script "mariadb-secure-installation" est un utilitaire fourni avec MariaDB qui permet de sécuriser une installation fraîche de MariaDB en suivant plusieurs étapes. Ces étapes incluent généralement :

1. Définition d'un nouveau mot de passe pour le compte "root" de la base de données.
2. Suppression des comptes d'utilisateurs anonymes.
3. Désactivation de la connexion root à distance pour des raisons de sécurité.
4. Suppression de la base de données de test, qui est généralement utilisée pour les tests de développement.
5. Rechargement des privilèges pour s'assurer que les modifications prennent effet immédiatement.

Pour l'exécuter on utilise la commande suivante :

```
root@webblpe5:~# mariadb-secure-installation
```

Il est ensuite possible de configurer les différents éléments énoncer précédemment. Pour cela, il faut répondre aux questions comme ci-dessous :

```
NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      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] n
```

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

Change the root password? [Y/n] n
```

```
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.

Remove anonymous users? [Y/n] y
```

```
Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] y
```

```
By default, MariaDB comes with a database named 'test' that anyone can
access.  This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] y
```

```
Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
```

```
All done!  If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
```

Pour se connecter à l'instance MariaDB, on utilise la syntaxe « **mariadb -u user -p** ». Le mot de passe de « user » vous sera ensuite demander afin de rentrer dans la console MariaDB, c'est là où vous pourrez exécuter vos requêtes SQL. S'il on est connecter avec le super utilisateur on peut y accéder simplement comme ci-dessous :

```
root@webblpe5:~# mariadb
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 38
Server version: 10.11.6-MariaDB-0+deb12u1 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.
```

Installation de PHP

Maintenant, on va venir installer le langage de programmation PHP sur notre serveur. PHP est un langage de script côté serveur largement utilisé pour développer des applications web dynamiques. Une fois installé, PHP permet au serveur web d'interpréter et d'exécuter des scripts PHP, ce qui permet de créer des sites web interactifs et dynamiques. Ce dernier va venir se greffer sur notre serveur Apache, comme une extension, afin de pouvoir traiter les scripts intégrés aux pages «***.php*** **»**. Pour ce faire, on utilise la commande suivante :

```
root@webblpe5:~# apt install php -y
```

On peut vérifier la version de PHP qui vient d'être installer avec la commande suivante :

```
root@webblpe5:~# php -v
PHP 8.2.18 (cli) (built: Apr 11 2024 22:07:45) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.2.18, Copyright (c) Zend Technologies
    with Zend OPcache v8.2.18, Copyright (c), by Zend Technologies
```

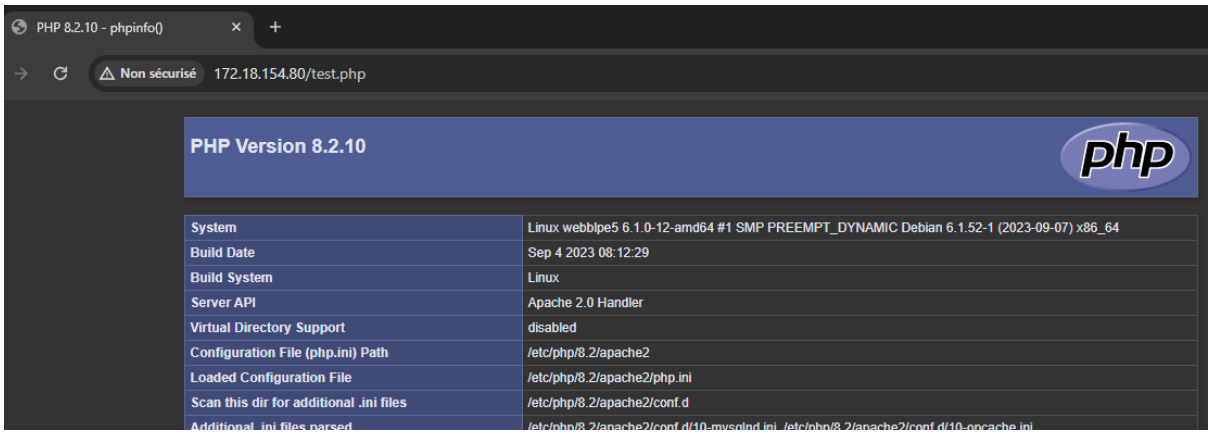
On va venir tester que notre moteur de script PHP est bien opérationnel en créant un fichier « **test.php »** dans le répertoire suivant : (nvim est un éditeur de texte tout comme nano utilisés dans les environnements Unix/Linux)

```
root@webblpe5:~# nvim /var/www/html/test.php
```

On va y définir la fonction « **phpinfo()** » qui génère et affiche des informations détaillées sur la configuration de PHP installée sur un serveur. Lorsque vous appelez cette fonction dans un script PHP et exécutez ce script dans un navigateur, vous obtenez une page HTML détaillant divers aspects de la configuration PHP, tels que les paramètres du serveur, les modules activés, les versions des logiciels, les chemins d'accès, les directives de configuration, etc :

```
<?php phpinfo(); ?>
```

Ce qui donne, en rajoutant « **/test.php** » à l'URL de la précédente page de notre navigateur :



La page générer fournit une multitude d'informations détaillées sur la configuration de PHP ainsi que sur le serveur Apache. Cependant, son accès devrait être restreint aux moments où ces données sont nécessaires. En d'autres termes, il est crucial de ne pas laisser cette page accessible à tout le monde, car elle peut contenir des informations sensibles sur votre configuration serveur. Il est donc recommandé de restreindre l'accès à cette page uniquement aux utilisateurs autorisés, afin de prévenir toute exposition non désirée de données sensibles.

Conclusion

En résumé, la mise en place réussie d'un environnement de serveur LAMP (Linux, Apache, MySQL/MariaDB, PHP) sur un système Debian constitue une étape cruciale dans la création d'une plateforme web robuste. Cette configuration permet désormais de fournir des services web, d'héberger des applications et de gérer des bases de données de manière sécurisée et efficace. Avec Linux comme fondation, Apache comme serveur web, MariaDB pour la gestion des données, et PHP pour la dynamique des applications web, ce serveur LAMP est parfaitement équipé pour répondre aux exigences de futures projets web, tout en offrant une flexibilité et une fiabilité plus ou moins optimales.

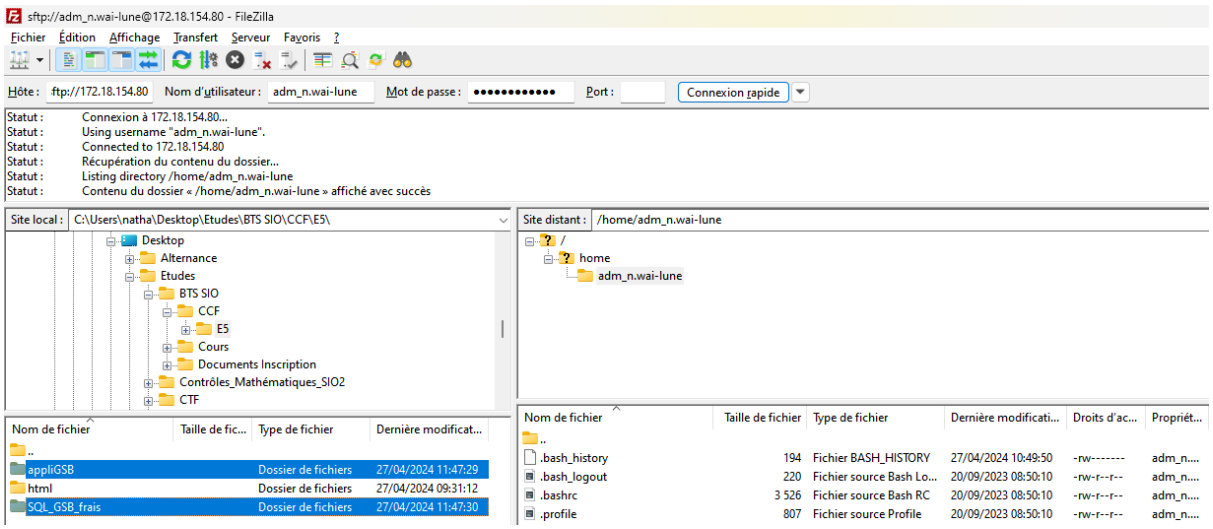
Mode Opératoire Virtual Host

Maintenant que nous avons réussi à mettre en place un environnement de serveur LAMP (Linux, Apache, MySQL/MariaDB, PHP) sur notre système Debian, nous sommes prêts à explorer une fonctionnalité essentielle pour la gestion des sites web : la configuration des Virtual hosts. Mais un Virtual host qu'est-ce que c'est ? Un Virtual host, ou hôte virtuel, est une méthode utilisée par le serveur web Apache pour héberger plusieurs sites web sur une seule machine physique. Chaque Virtual host peut avoir sa propre configuration distincte, permettant ainsi à plusieurs sites web de coexister sur le même serveur tout en étant isolés les uns des autres. Cette isolation permet aux administrateurs système de gérer efficacement plusieurs sites web avec des configurations et des contenus différents sur une seule infrastructure serveur.

Nous configurerons un hôte virtuel la solution Web ***gestionfraisintranet.gsb*** du laboratoire Galaxy Swiss Bourdin.

Il convient de noter qu'Apache est initialement configuré avec un hôte virtuel par défaut, qui pointe vers le répertoire « /var/www/html ». Bien que cela serve de point de départ pratique, cette configuration de base s'avère limitée pour héberger plusieurs sites web.

Dans des circonstances habituelles, la création d'un répertoire dans « /var/www » aurait été nécessaire. Cependant, dans ce contexte, ce répertoire m'a été fourni. Si vous êtes dans une situation similaire où vous avez déjà les dossiers de votre site, vous pouvez les importer en utilisant un client FTP tel que **FileZilla** :



Et les déplacer dans le bon répertoire :

```
root@webblpe5:~# ls -rtl
total 8
drwxr-xr-x 2 adm_n.wai-lune adm_n.wai-lune 4096 27 avril 12:08 SQL_GSB_frais
drwxr-xr-x 5 adm_n.wai-lune adm_n.wai-lune 4096 27 avril 12:08 appliGSB
```

```
root@webblpe5:~# mv /root/appliGSB/ /var/www/
```

Nous allons ultérieurement utiliser les fichiers situés dans le répertoire « **/root/SQL_GSB_frais** ». Ces fichiers sont essentiels au bon fonctionnement de la solution Web.

⚠ Je les ai dans le cadre de la mise en place de ma solution. Si vous suivez ce mode opératoire, à moins d'être dans le même contexte que moi, vous n'en aurez pas besoin ⚠

Pour créer l'hôte virtuel, on se déplace dans le répertoire « **/etc/apache2/sites-available** » comme ci-dessous :

```
root@webblpe5:~# cd /etc/apache2/sites-available/
root@webblpe5:/etc/apache2/sites-available# ls -rtl
total 12
-rw-r--r-- 1 root root 1286 27 avril 09:48 000-default.conf
-rw-r--r-- 1 root root 6195 27 avril 09:48 default-ssl.conf
```

Tous les nouveaux sites doivent être créer dans ce répertoire. Nous établirons un lien symbolique par la suite avec le répertoire « **/etc/apache2/sites-enabled** ». Pour créer un nouveau fichier de configuration on utilise la commande suivante :

```
root@webblpe5:/etc/apache2/sites-available# nvim gestionfraisintranet.gsb.conf
```

Voici un exemple de configuration d'un Virtual host Apache :

```
<VirtualHost *:80>
   ServerName example.com
   DocumentRoot /var/www/example
</VirtualHost>
```

Pour ma part je ne vais pas modifier le fichier que je viens de créer, du moins pour l'instant. La solution que je dois mettre en place doit utiliser le protocole HTTPS car ce dernier offre une sécurité supplémentaire par rapport à HTTP en chiffrant les données lors de leur transfert, ce qui les rend appropriées pour les sites Web nécessitant une protection des données sensibles. Pour cela, je vais créer un répertoire avec des permissions spécifiques et un propriétaire spécifique (vous pouvez évidemment faire de même) :

```
root@webblpe5:~# mkdir -p /etc/apache2/ssl
root@webblpe5:~# chmod 700 /etc/apache2/ssl/
root@webblpe5:~# chown -R root:root /etc/apache2/ssl/
```

Ces commandes sont utilisées pour créer un répertoire sécurisé et garantir que les fichiers de certificat et de clé privée sont accessibles uniquement par l'utilisateur et le groupe appropriés, tout en sécurisant les permissions pour prévenir les accès non autorisés.

Je vais ensuite générer une nouvelle clé privée RSA. En effet, une clé RSA est utilisée dans le processus SSL/TLS pour chiffrer les données, garantir leur intégrité et authentifier les parties impliquées dans la communication sécurisée sur Internet. Dans mon cas, j'utilise la commande ci-dessous pour générer une nouvelle clé privée RSA de 2048 bits et l'écrire dans le fichier spécifié :

```
root@webblpe5:~# openssl genrsa -out /etc/apache2/ssl/server.key 2048
```

Je fais ensuite une demande de signature de certificat (CSR - Certificate Signing Request), nécessaire pour obtenir un certificat SSL/TLS signé par une autorité de certification (CA). CSR est une demande formelle envoyée à une autorité de certification pour obtenir un certificat SSL/TLS. Elle contient des informations sur l'entité demandant le certificat ainsi que la clé publique correspondant à la clé privée qui sera utilisée pour chiffrer les données. Une fois que la CSR est signée par la CA, elle devient un certificat SSL/TLS valide qui peut être utilisé pour sécuriser les communications sur le site Web. Pour cela j'utilise la commande suivante et je complète le questionnaire :

```
root@webblpe5:~# openssl req -new -key /etc/apache2/ssl/server.key -out /etc/apache2/ssl/server.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
```

```
Country Name (2 letter code) [AU]:FR
State or Province Name (full name) [Some-State]:Réunion
Locality Name (eg, city) []:Saint-Denis
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Galaxy Swiss Bourdin
Organizational Unit Name (eg, section) []:GSB
Common Name (e.g. server FQDN or YOUR name) []:GSB
Email Address []:
```

```
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Maintenant, je génère le certificat auto-signé. Générer un certificat SSL auto-signé peut être utile pour des besoins de développement, de test ou dans des environnements isolés où la confiance peut être établie différemment. Cependant, il est important de comprendre que les certificats auto-signés ne fournissent pas le même niveau de confiance et de sécurité que les certificats émis par une autorité de certification publique. Ils ne sont donc pas recommandés pour une utilisation en production sur des sites Web accessibles au public ou traitant des informations sensibles. Ils peuvent être utiles pour des cas d'utilisation spécifiques, mais leur utilisation doit être soigneusement évaluée en fonction des exigences de sécurité et de confiance de votre application ou service.

Pour générer le certificat, j'utilise la commande suivante :

```
root@webblpe5:~# openssl x509 -req -days 365 -in /etc/apache2/ssl/server.csr -signkey /etc/apache2/ssl/server.key -out /etc/apache2/ssl/server.crt
Certificate request self-signature ok
subject=C = FR, ST = R\C3\83\C2\A9union, L = Saint-Denis, O = Galaxy Swiss Bourdin, OU = GSB, CN = GSB
```

Je configure mes Virtuals Hosts de la manière suivante :

```
<VirtualHost *:80>
    ServerName gestionfraisintranet.gsb
    Redirect permanent / https://gestionfraisintranet.gsb
</VirtualHost>

<VirtualHost *:443>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    ServerName gestionfraisintranet.gsb
    DocumentRoot /var/www/appliGSB

    <Directory /var/www/appliGSB>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
        DirectoryIndex cAccueil.php
    </Directory>

    SSLEngine on
    SSLCertificateFile /etc/apache2/ssl/server.crt
    SSLCertificateKeyFile /etc/apache2/ssl/server.key

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

```
<VirtualHost 172.18.154.71:80>
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/appliGSB

        <Directory /var/www/appliGSB>
                Options Indexes FollowSymLinks
                AllowOverride All
                Require all granted
                DirectoryIndex cAccueil.php
        </Directory>

        RewriteEngine On
        RewriteCond %{HTTPS} off
        RewriteRule ^ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301]

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>

<VirtualHost 172.18.154.71:443>
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/appliGSB

        <Directory /var/www/appliGSB>
                Options Indexes FollowSymLinks
                AllowOverride All
                Require all granted
                DirectoryIndex cAccueil.php
        </Directory>

        SSLEngine on
        SSLCertificateFile /etc/apache2/ssl/server.crt
        SSLCertificateKeyFile /etc/apache2/ssl/server.key

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>
```

Le premier Virtual Host admet configuration Apache qui définit un hôte virtuel pour le trafic HTTP (port 80) du domaine **gestionfraisintranet.gsb**. Voici ce que chaque ligne de cette configuration fait :

- **<VirtualHost *:80>** : Déclare un hôte virtuel pour le trafic HTTP sur toutes les interfaces (*) et le port 80.
- **ServerName gestionfraisintranet.gsb** : Spécifie le nom du serveur pour cet hôte virtuel, dans ce cas, gestionfraisintranet.gsb.
- **Redirect permanent / https://gestionfraisintranet.gsb** : Cette directive indique à Apache d'effectuer une redirection permanente (statut 301) pour tout le trafic arrivant sur cet hôte virtuel. La redirection envoie le navigateur du client vers l'URL https://gestionfraisintranet.gsb, ce qui signifie que tout le trafic HTTP est automatiquement redirigé vers HTTPS pour une connexion sécurisée.

Le second admet une configuration Apache qui définit un hôte virtuel pour le trafic HTTPS (port 443) du domaine **gestionfraisintranet.gsb**. Voici une explication de chaque élément de cette configuration :

- **<VirtualHost *:443>** : Déclare un hôte virtuel pour le trafic HTTPS sur toutes les interfaces (`*`) et le port 443.
- **ServerName gestionfraisintranet.gsb** : Spécifie le nom du serveur pour cet hôte virtuel, dans ce cas, gestionfraisintranet.gsb.
- **ServerAdmin webmaster@localhost** : Définit l'adresse e-mail de l'administrateur du serveur.
- **DocumentRoot /var/www/appliGSB** : Définit le répertoire racine du site Web, c'est-à-dire l'emplacement où les fichiers du site sont stockés.
- **<Directory /var/www/appliGSB>** : Configure les options spécifiques au répertoire, telles que les permissions et les autorisations d'accès. Dans ce cas, cela permet l'accès au répertoire « /var/www/appliGSB » et spécifie « cAccueil.php » comme fichier index par défaut.
- **SSLEngine on** : Active le moteur SSL pour ce virtualhost, indiquant qu'il doit gérer le trafic HTTPS.
- **SSLCertificateFile /etc/apache2/ssl/server.crt** : Spécifie le chemin du fichier de certificat SSL utilisé pour ce virtualhost.
- **SSLCertificateKeyFile /etc/apache2/ssl/server.key** : Spécifie le chemin du fichier de clé privée correspondant au certificat SSL.
- **ErrorLog ${APACHE_LOG_DIR}/error.log** : Spécifie le fichier de journal des erreurs pour cet hôte virtuel.
- **CustomLog ${APACHE_LOG_DIR}/access.log combined** : Spécifie le fichier de journal des accès pour cet hôte virtuel.

Le troisième admet une configuration Apache qui définit un hôte virtuel pour le trafic HTTP (port 80) provenant de l'adresse IP **172.18.154.71** (hébergeur). Voici une explication de chaque élément de cette configuration :

- **<VirtualHost 172.18.154.71:80>** : Déclare un hôte virtuel pour le trafic HTTP sur l'adresse IP 172.18.154.71 et le port 80.
- **ServerAdmin webmaster@localhost** : Définit l'adresse e-mail de l'administrateur du serveur.
- **DocumentRoot /var/www/appliGSB** : Définit le répertoire racine du site Web, c'est-à-dire l'emplacement où les fichiers du site sont stockés.
- **<Directory /var/www/appliGSB>** : Configure les options spécifiques au répertoire, telles que les permissions et les autorisations d'accès. Dans ce cas, cela permet l'accès au répertoire « /var/www/appliGSB » et spécifie « cAccueil.php » comme fichier index par défaut.
- **RewriteEngine On** : Active le module de réécriture d'URL pour ce virtualhost.
- **RewriteCond %{HTTPS} off** : Définit une condition pour la règle de réécriture suivante : si la connexion n'est pas sécurisée (HTTPS est désactivé).
- **RewriteRule ^ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301]** : Redirige toutes les requêtes HTTP vers HTTPS en utilisant une redirection permanente (statut 301). Cela garantit que tout le trafic HTTP est automatiquement redirigé vers HTTPS pour une connexion sécurisée.
- **ErrorLog ${APACHE_LOG_DIR}/error.log** : Spécifie le fichier de journal des erreurs pour cet hôte virtuel.
- **CustomLog ${APACHE_LOG_DIR}/access.log combined** : Spécifie le fichier de journal des accès pour cet hôte virtuel.

Le quatrième admet une configuration Apache qui définit un hôte virtuel pour le trafic HTTPS (port 443) provenant de l'adresse IP **172.18.154.71**. Voici une explication de chaque élément de cette configuration :

- **<VirtualHost 172.18.154.71:443>** : Déclare un hôte virtuel pour le trafic HTTPS sur l'adresse IP 172.18.154.71 et le port 443.
- **ServerAdmin webmaster@localhost** : Définit l'adresse e-mail de l'administrateur du serveur.
- **DocumentRoot /var/www/appliGSB** : Définit le répertoire racine du site Web, c'est-à-dire l'emplacement où les fichiers du site sont stockés.
- **<Directory /var/www/appliGSB>** : Configure les options spécifiques au répertoire, telles que les permissions et les autorisations d'accès. Dans ce cas, cela permet l'accès au répertoire « /var/www/appliGSB » et spécifie « cAccueil.php » comme fichier index par défaut.
- **SSLEngine on** : Active le moteur SSL pour ce virtualhost, indiquant qu'il doit gérer le trafic HTTPS.
- **SSLCertificateFile /etc/apache2/ssl/server.crt** : Spécifie le chemin du fichier de certificat SSL utilisé pour ce virtualhost.
- **SSLCertificateKeyFile /etc/apache2/ssl/server.key** : Spécifie le chemin du fichier de clé privée correspondant au certificat SSL.

- **ErrorLog ${APACHE_LOG_DIR}/error.log** : Spécifie le fichier de journal des erreurs pour cet hôte virtuel.
- **CustomLog ${APACHE_LOG_DIR}/access.log combined** : Spécifie le fichier de journal des accès pour cet hôte virtuel.

Vous pouvez ensuite enregistrer votre fichier de configuration et utiliser la commande suivante pour simplifier le processus d'activation des sites web sur Apache en créant les liens symboliques appropriés et en permettant à Apache de charger les fichiers de configuration des sites web lors de son démarrage :

```
root@webblpe5:/etc/apache2/sites-available# ls -rtl
total 16
-rw-r--r-- 1 root root 1286 27 avril 09:48 000-default.conf
-rw-r--r-- 1 root root 6195 27 avril 09:48 default-ssl.conf
-rw-r--r-- 1 root root 1837 27 avril 12:58 gestionfraisintranet.gsb.conf
```

```
root@webblpe5:/etc/apache2/sites-available# a2ensite gestionfraisintranet.gsb.conf
Enabling site gestionfraisintranet.gsb.
To activate the new configuration, you need to run:
  systemctl reload apache2
```

Assurez-vous de recharger le service Apache2. Maintenant, vous pouvez voire votre fichier dans le répertoire « /etc/apache2/sites-enabled » :

```
root@webblpe5:~# cd /etc/apache2/sites-enabled/
root@webblpe5:/etc/apache2/sites-enabled# ls -rtl
total 0
lrwxrwxrwx 1 root root 35  2 oct.   2023 000-default.conf -> ../sites-available/000-default.conf
lrwxrwxrwx 1 root root 35  2 oct.   2023 default-ssl.conf -> ../sites-available/default-ssl.conf
lrwxrwxrwx 1 root root 48 27 avril 13:20 gestionfraisintranet.gsb.conf -> ../sites-available/gestionfraisintranet.gsb.conf
```

Importation fichier SQL

Je vais maintenant importer les fichiers présents dans le répertoire « **/root/SQL_GSB_frais** » sur mon serveur MariaDB. L'importation de fichiers SQL est un outil essentiel pour gérer efficacement les données dans une base de données, que ce soit pour la migration, la sauvegarde, la synchronisation ou le déploiement de votre application. Pour cela, j'utilise la commande suivante :

```
root@webblpe5:~# mysql -u votre_utilisateur -p votre_base_de_donnees < /chemin/vers/votre/fichier.sql
```

Il faudra évidemment remplacez « votre_utilisateur » par votre nom d'utilisateur MySQL, « votre_base_de_donnees » par le nom de votre base de données et « /chemin/vers/votre/fichier.sql » par le chemin du fichier SQL que vous souhaitez importer. Dans mon cas, j'ai utilisé les commandes suivantes :

```
root@webblpe5:~# ls -rtl
total 4
drwxr-xr-x 2 adm_n.wai-lune adm_n.wai-lune 4096 27 avril 12:08 SQL_GSB_frais
```

```
root@webblpe5:~# cd SQL_GSB_frais/
root@webblpe5:~/SQL_GSB_frais# ls -rtl
total 8
-rw-r--r-- 1 adm_n.wai-lune adm_n.wai-lune 3340 27 avril 12:08 gsb_frais_structure.sql
-rw-r--r-- 1 adm_n.wai-lune adm_n.wai-lune 3745 27 avril 12:08 gsb_frais_insert_tables_statiques.sql
```

```
root@webblpe5:~/SQL_GSB_frais# mysql -u root < /root/SQL_GSB_frais/gsb_frais_structure.sql
```

```
root@webblpe5:~/SQL_GSB_frais# mysql -u root < /root/SQL_GSB_frais/gsb_frais_insert_tables_statiques.sql
```

Je peux vérifier si l'importation c'est bien effectuer via les commandes suivantes :

```
root@webblpe5:~# mariadb
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 41
Server version: 10.11.6-MariaDB-0+deb12u1 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)]> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| gsb_frais          |
| information_schema |
| mysql              |
| ocs                |
| performance_schema |
| sys                |
+--------------------+
6 rows in set (0,001 sec)
```

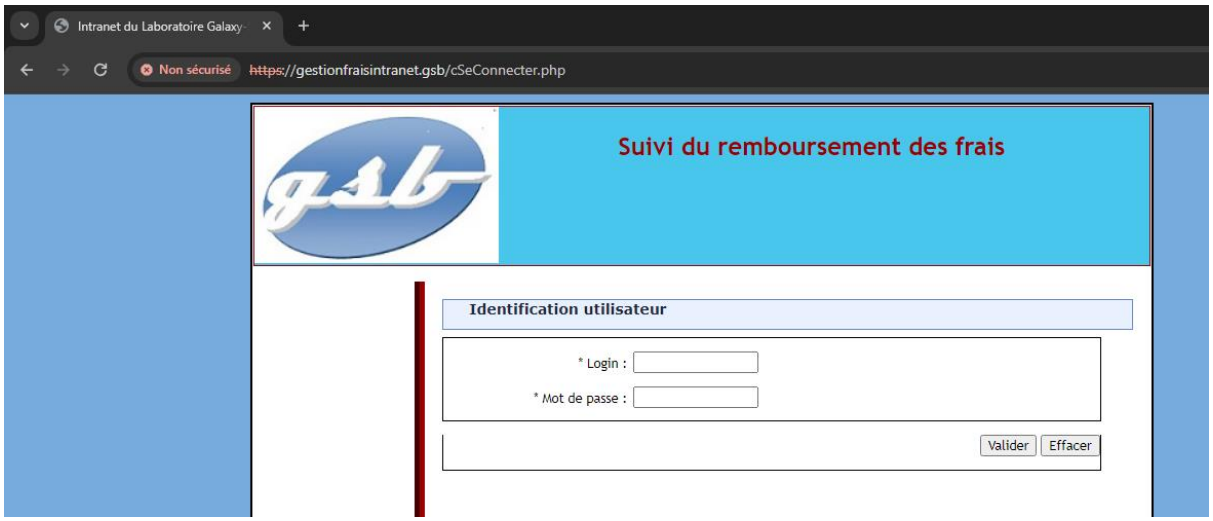
```
MariaDB [(none)]> USE gsb_frais;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [gsb_frais]> SHOW TABLES;
+------------------------+
| Tables_in_gsb_frais    |
+------------------------+
| Etat                   |
| FicheFrais             |
| FraisForfait           |
| LigneFraisForfait      |
| LigneFraisHorsForfait  |
| Visiteur               |
+------------------------+
6 rows in set (0,001 sec)
```

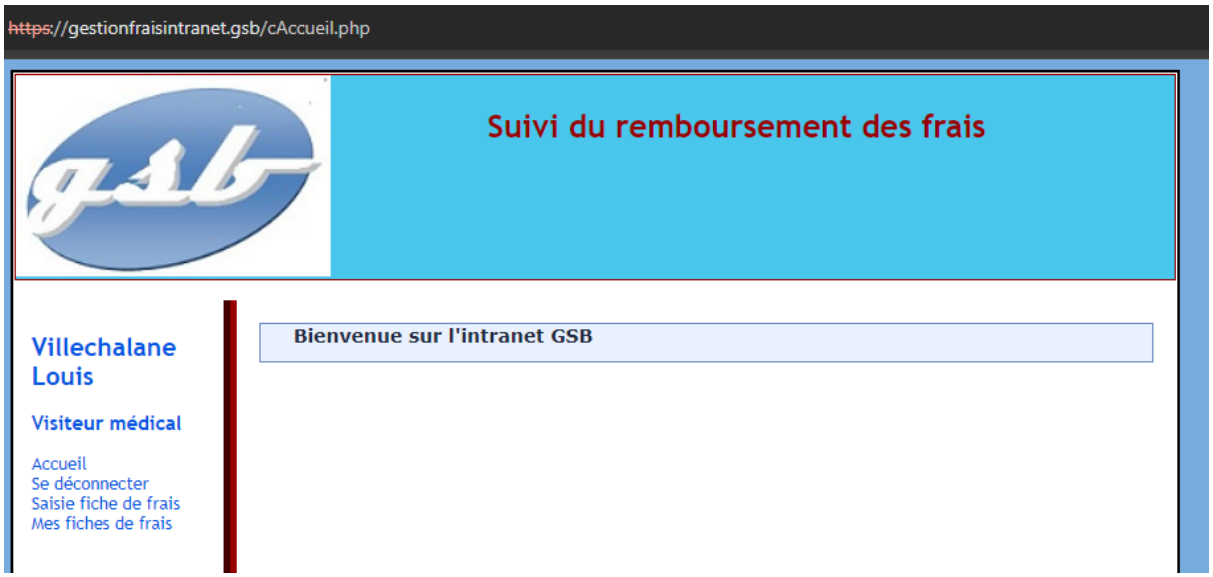
```
MariaDB [gsb_frais]> SELECT login, mdp from Visiteur;
+--------------+-------+
| login        | mdp   |
+--------------+-------+
| lvillachane  | jux7g |
```

Avant de tester l'accès à la solution Web :

```
root@webblpe5:~# systemctl restart mariadb.service
root@webblpe5:~# systemctl restart apache2.service
```

Ce qui donne :





Conclusion

En conclusion, la mise en place réussie d'un virtual host dans Apache constitue une étape cruciale pour rendre une solution web accessible de manière fiable et sécurisée. Grâce à cette configuration, le serveur est en mesure de diriger le trafic vers le bon répertoire de contenu, offrant ainsi aux utilisateurs une expérience en ligne fluide et optimale.