PortailSIO

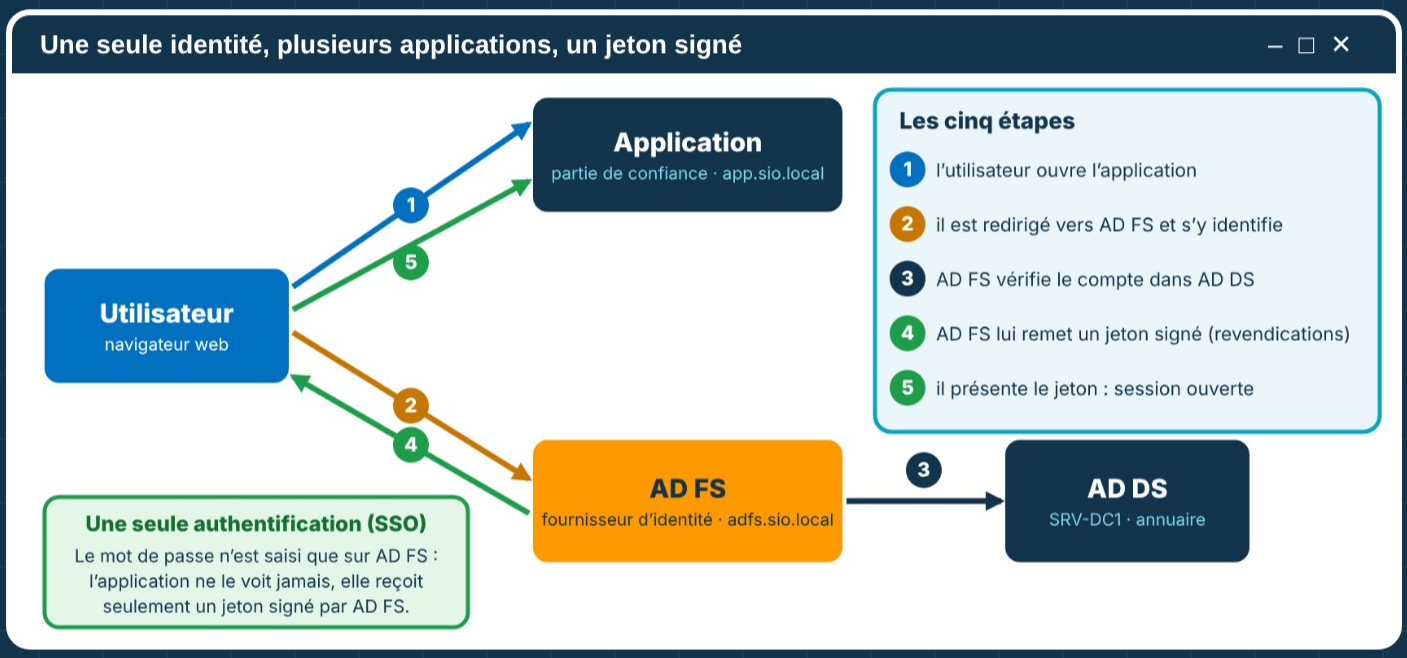
NOUVELLE FORMULE

TUTORIEL

WINDOWS SERVER 2025 · AD FS

Installer et fédérer AD FS

NIVEAU
Expert



1 Comprendre la fédération d'identité

2 Préparer le domaine : DNS et clé KDS

3 Délivrer le certificat SSL du service

4 Installer le rôle et créer la ferme AD FS

5 Tester la connexion et la découverte

6 Fédérer une première application (OIDC)

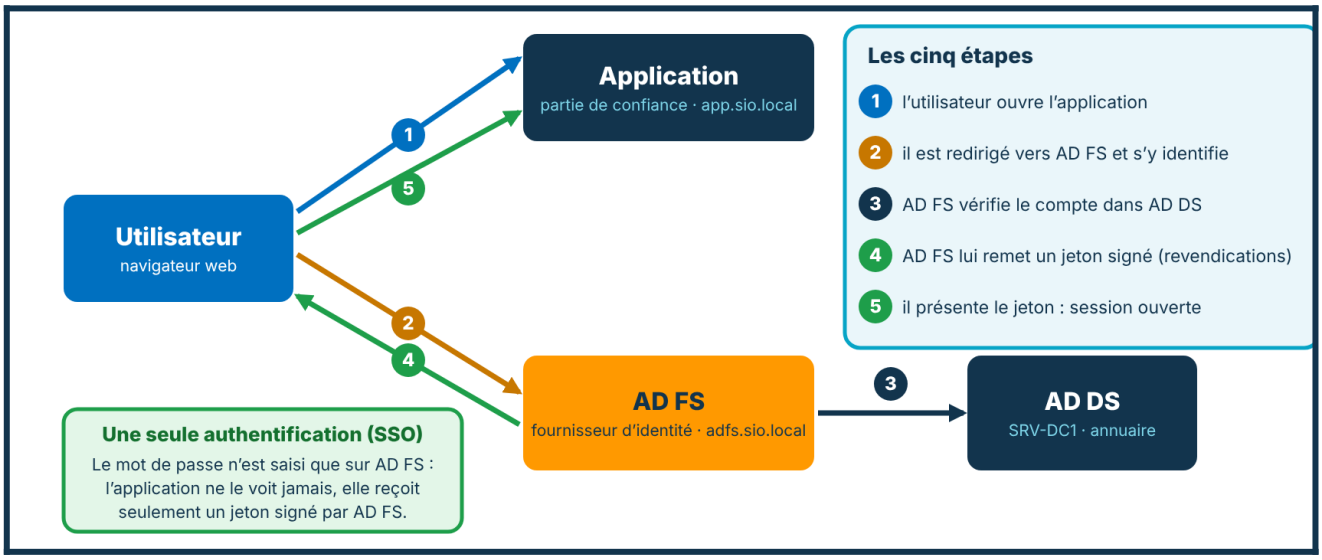
CC BY-NC
Utilisation commerciale interdite
© PortailSIO – 11/2026

Ce tutoriel en ligne
Scannez le QR code. Tous les tutoriels :
portail.sio-ndlp.fr/tutoriels.php

Ce tutoriel a été réalisé avec l'hyperviseur © Proxmox VE. Il peut être réalisé dans un environnement de virtualisation personnel à l'aide des logiciels © VirtualBox ou © VMware Player. Il **suppose que vous disposez du domaine sio.local** (contrôleur **SRV-DC1**, Windows Server 2025) et de l'**autorité de certification SIO-CA** installée dans le tutoriel « Activer LDAPS sur AD DS ».

AD FS (Active Directory Federation Services) est le service de Windows Server qui permet l'**authentification unique** (SSO) : l'utilisateur s'identifie **une seule fois**, avec son compte Active Directory, auprès d'AD FS, et les applications (une application web interne, un outil de gestion, GLPI, Nextcloud…) reçoivent un **jeton signé** qui prouve son identité, sans jamais voir son mot de passe. AD FS comprend les protocoles **SAML 2.0**, **OpenID Connect / OAuth 2.0** et **WS-Federation**.

Dans ce tutoriel, nous préparons le domaine (DNS et clé KDS), nous délivrons à AD FS un **certificat SSL** depuis notre autorité de certification, nous installons la **ferme AD FS**, nous testons la connexion, puis nous **fédérons une première application** en OpenID Connect et nous récupérons un jeton à la main pour comprendre ce qui circule.



Dans le cadre de ce tutoriel, nous utilisons les machines suivantes (VLAN SERVEURS, réseau 192.168.20.0 /24) :

Machine	Rôle	Système	Adresse IP
SRV-DC1	Contrôleur de domaine, DNS, **autorité de certification SIO-CA**	Windows Server 2025	192.168.20.250
SRV-ADFS	Serveur de fédération **AD FS** (membre du domaine)	Windows Server 2025	192.168.20.65
SRV-LINUX	Client de test (`curl`), a déjà importé la CA SIO-CA	Debian 13	192.168.20.64

Les adresses, les noms et les mots de passe d'exemple sont à adapter à votre environnement. Les commandes Windows sont exécutées dans **PowerShell** ouvert **en tant qu'administrateur** ; les commandes Debian, avec un compte autorisé à utiliser `sudo`. Pour les tests de connexion, il vous faut aussi un **navigateur** sur un poste du domaine (ou sur SRV-ADFS) et l'utilisateur **amartin** créé dans le tutoriel LDAPS.

01

ÉTAPE 1 SUR 6

Comprendre la fédération d'identité

A – Les acteurs et le vocabulaire

Chaque application qui délègue son authentification à AD FS suit le même scénario (schéma ci-dessus). Quelques termes reviennent tout au long du tutoriel :

Terme	Signification	Dans notre laboratoire
Fournisseur d'identité **(IdP)**	Le service qui authentifie l'utilisateur et délivre le jeton	**AD FS** (SRV-ADFS, `adfs.sio.local`)
Partie de confiance ***(relying party)***	L'application qui fait confiance à l'IdP	Notre application d'essai `app.sio.local`
Jeton ***(token)***	Document **signé** par AD FS qui atteste l'identité de l'utilisateur	Jeton OpenID Connect au format JWT
Revendication ***(claim)***	Une information sur l'utilisateur placée dans le jeton	`upn`, adresse e-mail, nom affiché
Groupe d'applications	Dans AD FS, regroupe le client OpenID Connect et son API web	`app-test`

B – Quel protocole choisir ?

Protocole	Usage typique	Données de configuration à fournir à l'application
SAML 2.0	Applications d'entreprise et SaaS historiques	Fichier de métadonnées `FederationMetadata.xml` d'AD FS
OpenID Connect **(OAuth 2.0)**	Applications web modernes, API, mobiles	Document de découverte, identifiant client, secret, URI de redirection

Nous traitons **OpenID Connect** dans ce tutoriel : c'est le protocole le plus simple à tester en ligne de commande, et celui que proposent la plupart des applications récentes. Le fichier de métadonnées SAML est lui aussi publié par AD FS (étape 5).

C – Les règles à connaître avant de commencer

1. Le **nom du service de fédération** (`adfs.sio.local`) doit être **différent du nom du serveur** (`SRV-ADFS`) : AD FS enregistre un nom de service (SPN) sur son compte de service, qui doit être unique
2. AD FS ne fonctionne qu'en **HTTPS** : le certificat doit contenir le nom du service, être **de confiance** pour les clients et **ne pas utiliser de clé CNG** (étape 3)
3. Pour la connexion intégrée Windows, le nom du service doit pointer vers le serveur par un enregistrement **A** (pas un CNAME)
4. Une ferme AD FS avec la **base interne Windows** (WID) suffit pour un laboratoire ; Microsoft indique qu'au-delà de **100 approbations**, une base **SQL Server** est nécessaire

REMARQUE IMPORTANTE

Ce tutoriel installe un **seul serveur AD FS** pour le laboratoire. La haute disponibilité (plusieurs nœuds), la publication sur Internet (proxy d'application web) et l'authentification multifacteur ne sont pas traitées.

✔ **POINT DE CONTRÔLE**

- Vous savez dire ce que sont un **fournisseur d'identité**, une **partie de confiance**, un **jeton** et une **revendication**
- Vous avez noté la différence entre le **nom du service** (`adfs.sio.local`) et le **nom du serveur** (`SRV-ADFS`)

02 ÉTAPE 2 SUR 6 Préparer le domaine : DNS et clé KDS

A – Créer les enregistrements DNS et la clé racine KDS

Sur **SRV-DC1**, créez d'abord les deux noms du service : `adfs.sio.local` (le service) et `certauth.adfs.sio.local` (utilisé par AD FS pour l'authentification par certificat client). Ils pointent tous deux vers **SRV-ADFS**, par un enregistrement **A** :

POWERSHELL (SRV-DC1)

```
Add-DnsServerResourceRecordA -ZoneName sio.local -Name adfs `
    -IPv4Address 192.168.20.65
Add-DnsServerResourceRecordA -ZoneName sio.local -Name certauth.adfs `
    -IPv4Address 192.168.20.65
```

Ensuite, créez la **clé racine KDS** (*Key Distribution Services*), dont dépend le **compte de service géré de groupe** (gMSA) sous lequel AD FS va s'exécuter : Windows y génère le mot de passe du compte et le renouvelle tout seul. Cette clé est **unique pour la forêt** : la première commande vérifie si elle existe déjà (aucune sortie = pas de clé), la seconde la crée :

POWERSHELL (SRV-DC1)

```
Get-KdsRootKey
Add-KdsRootKey -EffectiveTime (Get-Date).AddHours(-10)
```

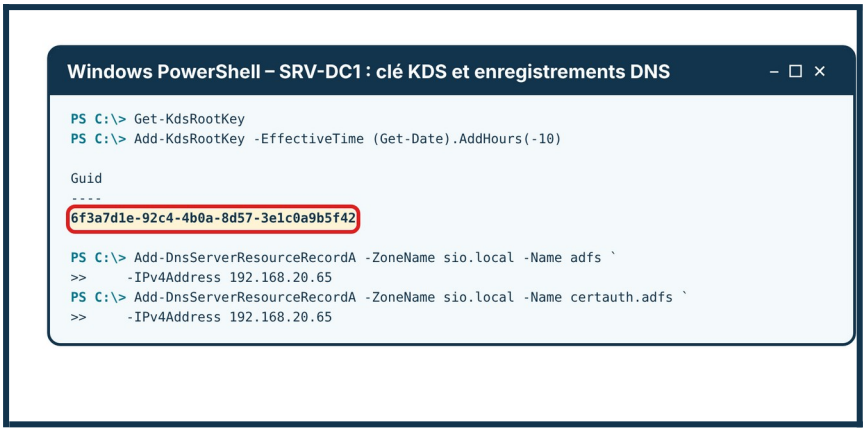


Figure 1 – Enregistrements DNS et clé racine KDS

À SAVOIR

Le GUID affiché confirme la création de la clé. Le paramètre `-EffectiveTime (Get-Date).AddHours(-10)` **antidate** la clé de dix heures, ce qui la rend utilisable tout de suite : c'est la commande que Microsoft indique pour ce cas, adaptée à un laboratoire. En **production**, on crée la clé sans antidater et on attend environ dix heures, le temps que les contrôleurs de domaine se la répliquent.

B – Préparer SRV-ADFS

Sur le serveur **SRV-ADFS** (Windows Server 2025 fraîchement installé), donnez-lui son adresse fixe, son nom et rattachez-le au domaine. Les deux contrôleurs sont ses serveurs DNS. Adaptez l'interface (`Ethernet`) et la passerelle à votre réseau :

POWERSHELL (SRV-ADFS)

```
New-NetIPAddress -InterfaceAlias "Ethernet" -IPAddress 192.168.20.65 `
    -PrefixLength 24 -DefaultGateway 192.168.20.254
Set-DnsClientServerAddress -InterfaceAlias "Ethernet" `
    -ServerAddresses 192.168.20.250,192.168.20.251
Rename-Computer -NewName SRV-ADFS -Restart
```

1. Une fois le serveur redémarré, joignez-le au domaine avec le compte **Administrateur** du domaine, puis redémarrez de nouveau :

POWERSHELL (SRV-ADFS)

```
Add-Computer -DomainName sio.local -Credential SIO\Administrateur -Restart
```

2. Ouvrez une session sur **SRV-ADFS** avec **SIO\Administrateur**, puis contrôlez que le serveur est dans le domaine et qu'il résout bien le nom du service :

POWERSHELL (SRV-ADFS)

```
Get-CimInstance Win32_ComputerSystem | Select-Object Name, Domain, PartOfDomain
Resolve-DnsName adfs.sio.local -Type A
```

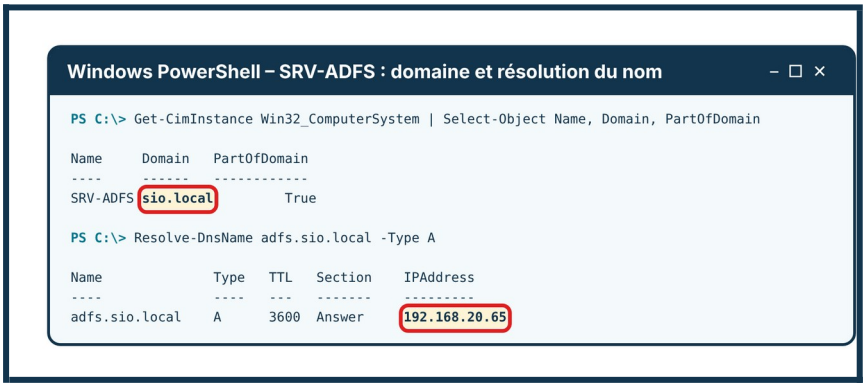


Figure 2 – SRV-ADFS dans le domaine, nom du service résolu

À SAVOIR

Si `Resolve-DnsName` ne renvoie rien, vérifiez que les deux contrôleurs sont bien les serveurs DNS de SRV-ADFS et que l'enregistrement `adfs` existe dans la zone `sio.local` (console DNS).

✔ POINT DE CONTRÔLE

- Les enregistrements **adfs** et **certauth.adfs** (type A) pointent vers **192.168.20.65**
- `Get-KdsRootKey` renvoie une clé
- **SRV-ADFS** est membre de **sio.local** et résout `adfs.sio.local`

03

ÉTAPE 3 SUR 6

Délivrer le certificat SSL du service

Le certificat SSL d'AD FS est celui que présentent le service et la page de connexion. Les exigences sont précises, et l'une d'elles pose souvent problème :

Exigence	Pourquoi
Le nom du service (`adfs.sio.local`) **dans le CN ou dans le nom alternatif (SAN)**	Le client compare ce nom à celui qu'il a contacté
Délivré par une autorité de confiance **pour tous les clients**	Notre CA d'entreprise convient : les postes du domaine lui font déjà confiance (Debian l'a importée dans le tutoriel LDAPS)
Une clé privée **dans le magasin « Ordinateur local »**	AD FS doit pouvoir s'en servir
Pas de clé CNG, mais un fournisseur de chiffrement hérité **(CSP)**	Sinon la configuration d'AD FS échoue : « The certificates with the CNG private key are not supported »
Optionnel : `certauth.adfs.sio.local` **dans le SAN**	Pour l'authentification par certificat client ; sans effet pour le reste du laboratoire

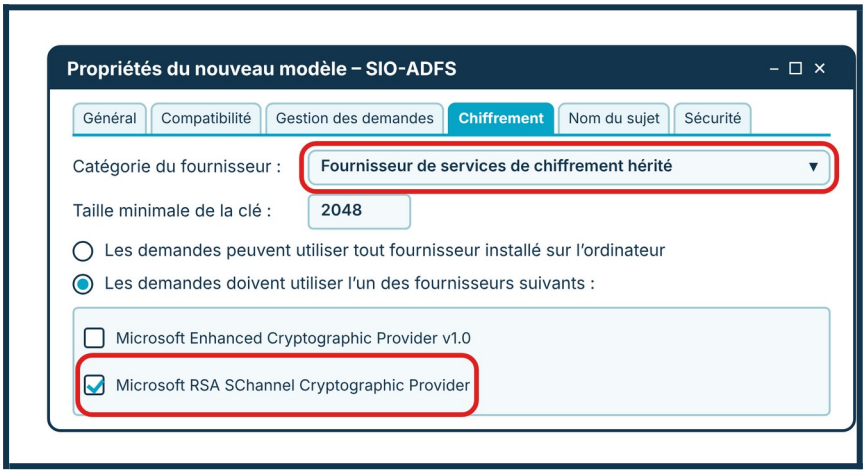
Le modèle « Serveur Web » de la CA utilise déjà un fournisseur de chiffrement hérité, mais ses droits d'inscription ne sont pas ouverts aux serveurs. Nous en créons donc une **copie**, **SIO-ADFS**, dont nous contrôlons le fournisseur de chiffrement et les droits.

A – Créer le modèle de certificat SIO-ADFS

Sur **SRV-DC1**, ouvrez la console des modèles de certificats (`certtmpl.msc`), faites un clic droit sur « **Serveur Web** » puis « **Dupliquer le modèle** ». Dans les propriétés du nouveau modèle :

1. Onglet « **Général** » : nom complet du modèle **SIO-ADFS** (le nom du modèle est alors `SIO-ADFS`, sans espace)
2. Onglet « **Compatibilité** » : laissez les valeurs proposées, ou choisissez « Windows Server 2016 » pour l'autorité et le destinataire

3 Onglet « **Chiffrement** » : catégorie « **Fournisseur de services de chiffrement hérité** », taille minimale **2048**, option « Les demandes doivent utiliser l'un des fournisseurs suivants », puis cochez uniquement **Microsoft RSA SChannel Cryptographic Provider** :



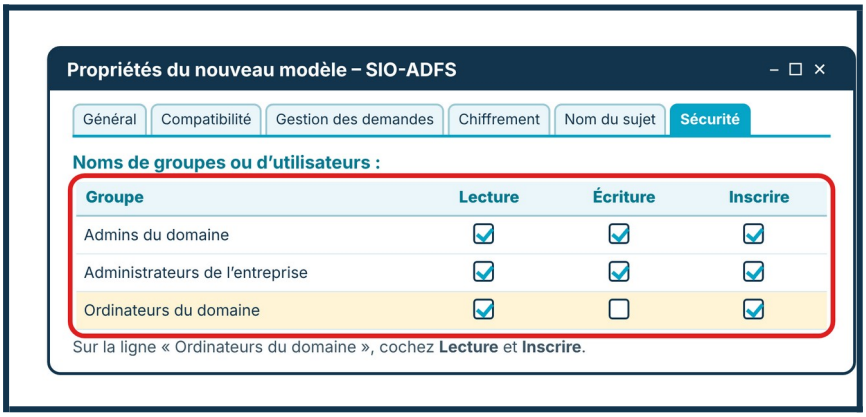
À SAVOIR

C'est le réglage décisif : une clé créée par le fournisseur « hérité » est acceptée par AD FS, une clé du fournisseur de stockage de clés (KSP, CNG) est refusée.

Figure 3 – Modèle SIO-ADFS : fournisseur de chiffrement hérité

4 Onglet « **Nom du sujet** » : laissez « **Fournir dans la demande** » (valeur du modèle d'origine) : c'est le serveur qui indique son nom

5 Onglet « **Sécurité** » : cliquez « **Ajouter** », saisissez **Ordinateurs du domaine**, puis cochez **Lecture** et **Inscrire** pour cette ligne, et validez avec « **OK** » :



À SAVOIR

En production, remplacez « Ordinateurs du domaine » par un groupe qui ne contient que le ou les serveurs AD FS. Si vous prévoyez de copier le certificat sur un second serveur AD FS, cochez aussi « Autoriser l'exportation de la clé privée » dans l'onglet « Gestion des demandes ».

Figure 4 – Modèle SIO-ADFS : droit d'inscription des ordinateurs

B – Publier le modèle sur l'autorité

Un modèle n'est utilisable que s'il est **publié** sur la CA. Ouvrez `certsrv.msc`, faites un clic droit sur « **Modèles de certificats** » de **SIO-CA** > « **Nouveau** » > « **Modèle de certificat à délivrer** », sélectionnez **SIO-ADFS** puis « **OK** » :

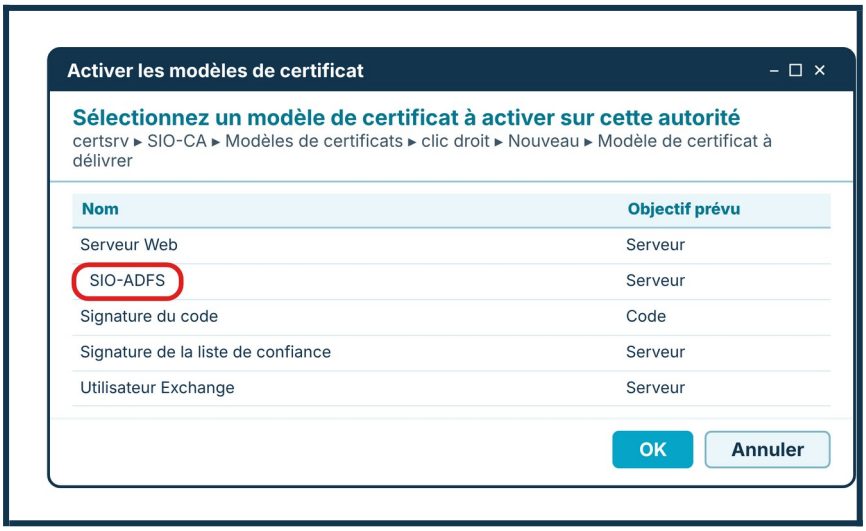


Figure 5 – Publication du modèle SIO-ADFS sur SIO-CA

À SAVOIR

Le modèle apparaît ensuite dans la liste des modèles de certificats de l'autorité. La réplication de l'annuaire peut demander une à deux minutes avant que les serveurs le voient.

C – Demander le certificat depuis SRV-ADFS

Sur **SRV-ADFS**, demandez le certificat avec le modèle **SIO-ADFS**. Le nom du service est dans le sujet et les deux noms DNS dans le SAN ; la commande range le certificat dans le magasin de l'ordinateur. La dernière ligne contrôle le fournisseur de la clé privée : elle doit afficher **Microsoft RSA SChannel Cryptographic Provider** :

POWERSHELL (SRV-ADFS)

```
$cert = (Get-Certificate -Template SIO-ADFS -SubjectName "CN=adfs.sio.local" `
    -DnsName adfs.sio.local, certauth.adfs.sio.local `
    -CertStoreLocation Cert:\LocalMachine\My).Certificate
$cert | Format-List Subject, DnsNameList, Issuer, NotAfter, Thumbprint
$cert.PrivateKey.CspKeyContainerInfo.ProviderName
```



Figure 6 – Le certificat SSL du service est délivré

À SAVOIR

Gardez cette fenêtre PowerShell ouverte : la variable `$cert` est réutilisée à l'étape suivante. Si vous l'avez fermée, retrouvez le certificat avec `$cert = Get-ChildItem Cert:\LocalMachine\My | Where-Object { $_.Subject -eq "CN=adfs.sio.local" }`.

À SAVOIR

Si `Get-Certificate` échoue, faites la demande à la main : `certlm.msc` > Personnel > clic droit > Toutes les tâches > Demander un nouveau certificat > stratégie « Active Directory » > modèle **SIO-ADFS** > lien « Des informations supplémentaires sont requises », puis renseignez le nom commun `adfs.sio.local` et les noms DNS.

✔ POINT DE CONTRÔLE

- Le modèle **SIO-ADFS** est publié sur **SIO-CA**, avec les droits Lecture et Inscrire pour les ordinateurs
- Le certificat de **adfs.sio.local** est dans « Ordinateur local > Personnel » de **SRV-ADFS**, délivré par **SIO-CA**
- Le fournisseur de la clé privée est **Microsoft RSA SChannel Cryptographic Provider** (pas de clé CNG)

04 ÉTAPE 4 SUR 6 Installer le rôle et créer la ferme AD FS

A – Installer le rôle

Sur **SRV-ADFS**, ouvrez PowerShell **avec le compte SIO\Administrateur** : la création de la première ferme exige des droits d'administrateur de domaine, car AD FS crée dans l'annuaire un conteneur pour ses clés et son compte de service. Installez le rôle **Services de fédération Active Directory** :

POWERSHELL (SRV-ADFS)

```
Install-WindowsFeature ADFS-Federation -IncludeManagementTools
```

B – Créer la première ferme

La commande `Install-AdfsFarm` configure le service de fédération. Sans option particulière, elle utilise la **base interne Windows** (WID). Le compte de service **gmsa-adfs** est un gMSA : le `$` final de son nom est **obligatoire**, et AD FS le **crée** s'il n'existe pas encore :

POWERSHELL (SRV-ADFS)

```
Import-Module ADFS
Install-AdfsFarm -CertificateThumbprint $cert.Thumbprint `
    -FederationServiceName adfs.sio.local `
    -FederationServiceDisplayName "SIO – Connexion" `
    -GroupServiceAccountIdentifier 'SIO\gmsa-adfs$'
```

Paramètre	Valeur	Rôle
`CertificateThumbprint`	`$cert.Thumbprint`	Empreinte du certificat SSL de l'étape 3 (liste : `dir Cert:\LocalMachine\My`)
`FederationServiceName`	`adfs.sio.local`	Nom du service : doit figurer dans le certificat
`FederationServiceDisplayName`	`SIO – Connexion`	Nom affiché en haut de la page de connexion
`GroupServiceAccountIdentifier`	`SIO\gmsa-adfs$`	gMSA sous lequel s'exécute le service (le `$` final est obligatoire)



Figure 7 – Installation du rôle et création de la ferme

À SAVOIR

L'état « Success » signifie que la ferme est configurée. Redémarrez ensuite **SRV-ADFS** par précaution (`Restart-Computer`). En cas d'erreur, relancez la commande avec `-Verbose` et lisez le message : les causes les plus fréquentes (certificat CNG, clé KDS, nom du service absent du certificat) figurent dans le tableau de dépannage.

C – Contrôler la ferme

Après le redémarrage, ouvrez PowerShell sur **SRV-ADFS** : le service **adfssrv** doit tourner, la ferme porter le bon nom, et AD FS disposer de ses trois certificats : celui de **communication** (le certificat SSL) et deux certificats **générés tout seuls** pour la **signature** et le **déchiffrement** des jetons :

POWERSHELL (SRV-ADFS)

```
Get-Service adfssrv | Select-Object Name, Status, StartType
Get-AdfsProperties | Select-Object HostName, HttpsPort, EnableIdPInitiatedSignonPage
Get-AdfsCertificate | Select-Object CertificateType, Thumbprint
```

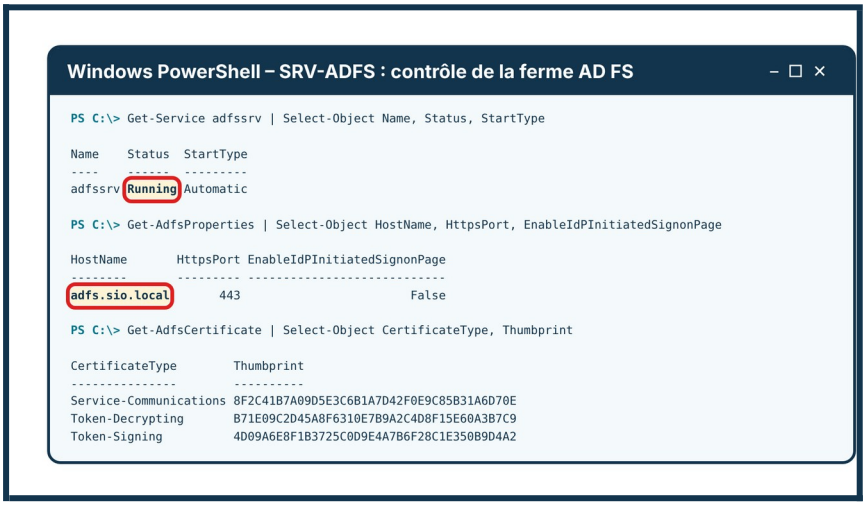


Figure 8 – La ferme AD FS est en service

À SAVOIR

Le compte **gmsa-adfs** est visible dans `dsa.msc`, dossier « Managed Service Accounts ». La console de gestion d'AD FS s'ouvre avec `C:\Windows\ADFS\Microsoft.IdentityServer.msc` (elle est aussi dans les outils d'administration du Gestionnaire de serveur).

✔ POINT DE CONTRÔLE

- `Install-AdfsFarm` se termine avec l'état **Success**
- Le service **adfssrv** est **Running**
- `Get-AdfsCertificate` liste les certificats **Service-Communications**, **Token-Decrypting** et **Token-Signing**

05 ÉTAPE 5 SUR 6 — Tester la connexion et la découverte

A – Activer la page de test et repérer les points de terminaison

AD FS propose une page de connexion de test, **désactivée par défaut**. Activez-la, puis listez les adresses (les *endpoints*) que les applications utiliseront :

POWERSHELL (SRV-ADFS)

```
Set-AdfsProperties -EnableIdPInitiatedSignonPage $true
Get-AdfsEndpoint | Select-Object FullUrl |
    Select-String "openid-configuration|FederationMetadata"
```

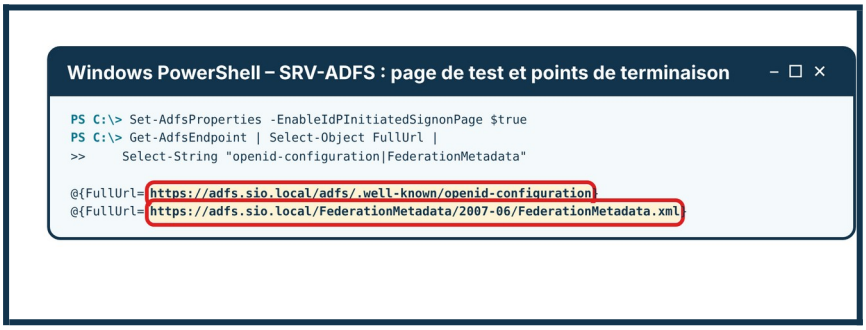


Figure 9 – Page de test activée et points de terminaison

À SAVOIR

Deux adresses sont à retenir : le **document de découverte OpenID Connect** (`/adfs/.well-known/openid-configuration`) et le **fichier de métadonnées SAML** (`/FederationMetadata/2007-06/FederationMetadata.xml`).

B – Se connecter avec un compte de l'annuaire

Depuis un **navigateur** d'un poste du domaine (ou de SRV-ADFS), ouvrez `https://adfs.sio.local/adfs/ls/idpinitiatedsignon.aspx`. La page de connexion s'affiche avec le nom « SIO – Connexion » : saisissez le compte **amartin@sio.local** et son mot de passe :

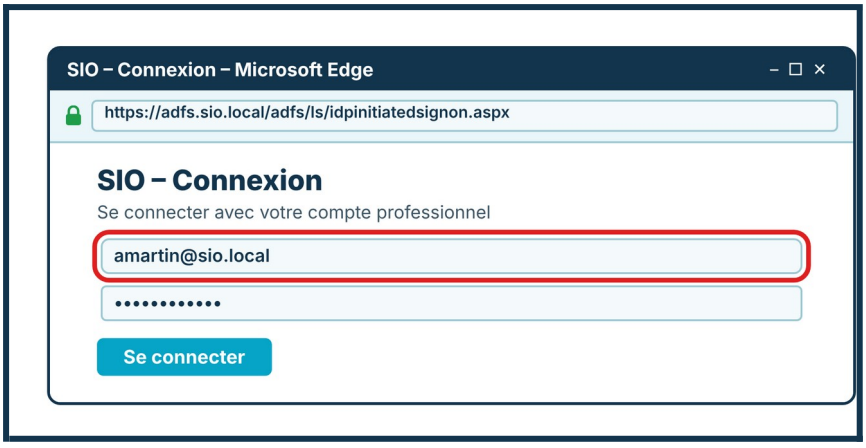


Figure 10 – Page de connexion d'AD FS

À SAVOIR

Le cadenas du navigateur doit être **fermé** : il prouve que le certificat est de confiance. Une alerte de certificat signale que le poste ne fait pas confiance à SIO-CA (poste hors domaine : importez le certificat de la CA).

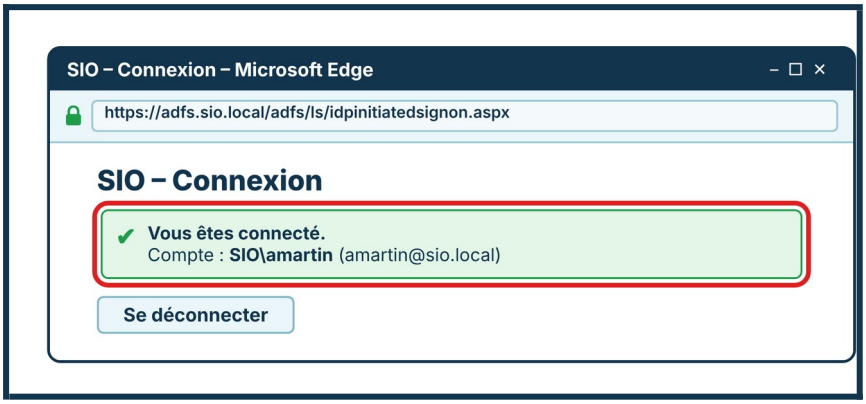


Figure 11 – Connexion réussie

À SAVOIR

Un compte de l'annuaire a été authentifié par AD FS. Sur un poste du domaine, selon le navigateur et sa zone « Intranet local », la page peut s'ouvrir **sans demander de mot de passe** : AD FS réutilise la session Windows (authentification intégrée), ce qui est précisément le principe du SSO.

C – Lire le document de découverte depuis Debian

Un client OpenID Connect commence par lire ce document, qui décrit le service (adresses, algorithmes, clés de signature). Sur **SRV-LINUX**, qui connaît déjà la CA SIO-CA et utilise les contrôleurs comme DNS (tutoriel LDAPS), récupérez-le :

TERMINAL

```
curl -s https://adfs.sio.local/adfs/.well-known/openid-configuration \
    | python3 -m json.tool \
    | grep -E '"(issuer|authorization_endpoint|token_endpoint|jwks_uri)"'
```



Figure 12 – Debian lit le document de découverte

À SAVOIR

La commande réussit sans `-k` : preuve que le certificat est reconnu par Debian. Si `curl` répond « SSL certificate problem: unable to get local issuer certificate », reprenez l'étape 5 du tutoriel LDAPS (import de la CA).

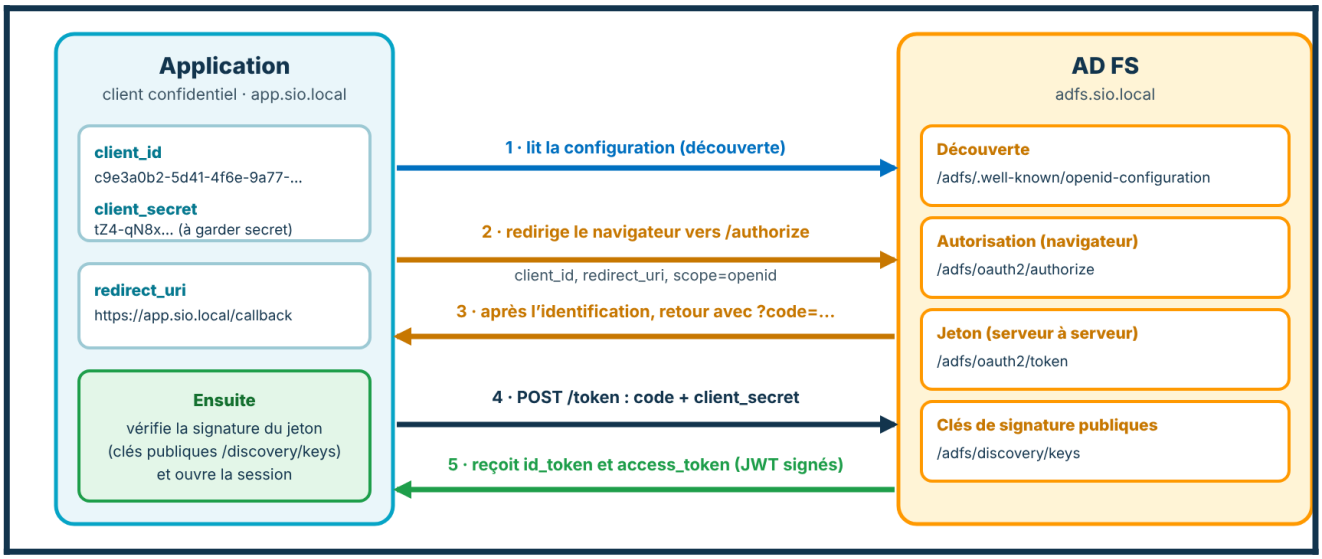
✔ POINT DE CONTRÔLE

- `https://adfs.sio.local/adfs/ls/idpinitiatedsignon.aspx` s'ouvre **sans alerte de certificat**
- La connexion avec **amartin@sio.local** réussit
- `curl` depuis Debian renvoie l'**issuer** et les adresses `authorize`, `token` et `jwks_uri`

06 ÉTAPE 6 SUR 6
Fédérer une première application (OIDC)

A – Le flux OpenID Connect

Une application **confidentielle** (qui peut garder un secret côté serveur) utilise le **flux « code d'autorisation »** : elle n'obtient jamais le mot de passe de l'utilisateur, seulement un **code** à usage unique qu'elle échange, avec son secret, contre des jetons signés :



Valeur	Exemple	D'où elle vient
`client_id`	`c9e3a0b2-5d41-4f6e-…`	Identificateur du client, généré par AD FS à l'étape B
`client_secret`	`tZ4-qN8x…`	Secret partagé généré à l'étape B : **à copier tout de suite**
`redirect_uri`	`https://app.sio.local/callback`	Adresse de retour de l'application, **exactement** comme déclarée
`scope`	`openid`	Ce que l'application demande (`openid` = identité de l'utilisateur)

B – Créer le groupe d'applications

Un **groupe d'applications** regroupe le client OpenID Connect (l'application) et l'**API web** (la ressource pour laquelle AD FS délivre les jetons). Sur **SRV-ADFS**, ouvrez la console AD FS (`C:\Windows\ADFS\Microsoft.IdentityServer.msc`) :

1 Dans l'arborescence, faites un clic droit sur « **Groupes d'applications** » puis « **Ajouter un groupe d'applications…** » :

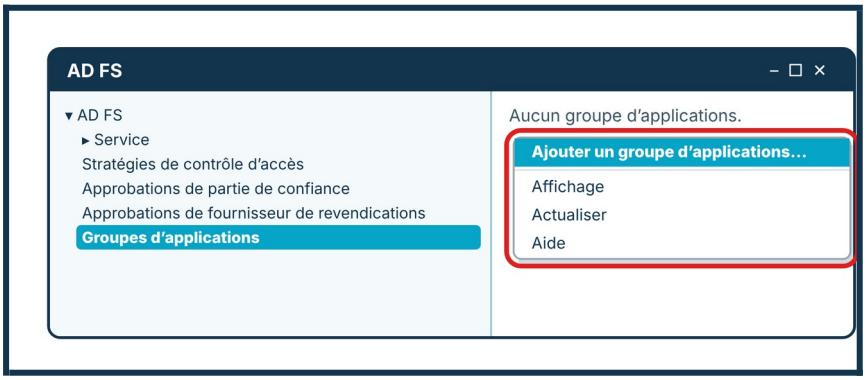


Figure 13 – Console AD FS : Ajouter un groupe d'applications

À SAVOIR

L'assistant se déroule en huit pages : Bienvenue, Application serveur, Informations d'identification, API web, Stratégie de contrôle d'accès, Autorisation d'application, Résumé, Terminer.

2 Page « **Bienvenue** » : nom **app-test**, modèle « **Application serveur accédant à une API web** » (dans la rubrique « Applications client-serveur »), puis « **Suivant** » :

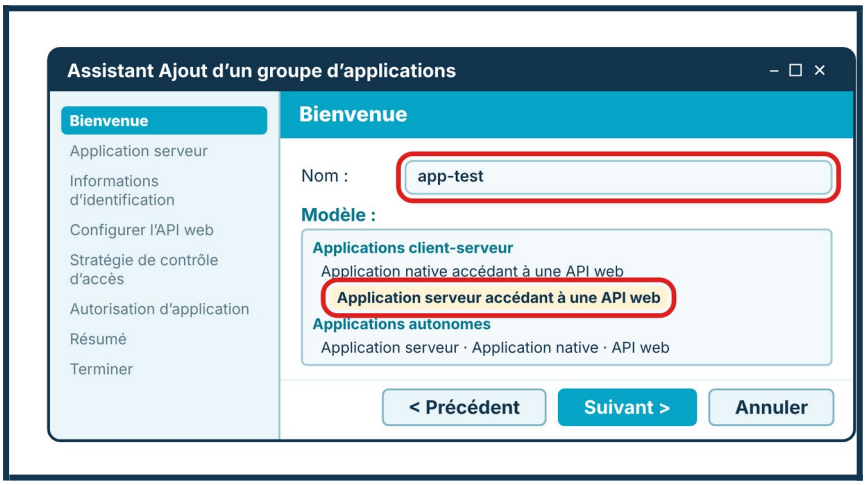


Figure 14 – Assistant : nom et modèle

À SAVOIR

Ce modèle correspond à une application web qui tourne sur un serveur et sait garder un secret : c'est le cas de la plupart des applications auto-hébergées.

3 Page « **Application serveur** » : nom **app-test-serveur**. **Copiez l'identificateur du client** généré (un GUID) dans un fichier. Saisissez l'**URI de redirection** `https://app.sio.local/callback` et cliquez « **Ajouter** » :

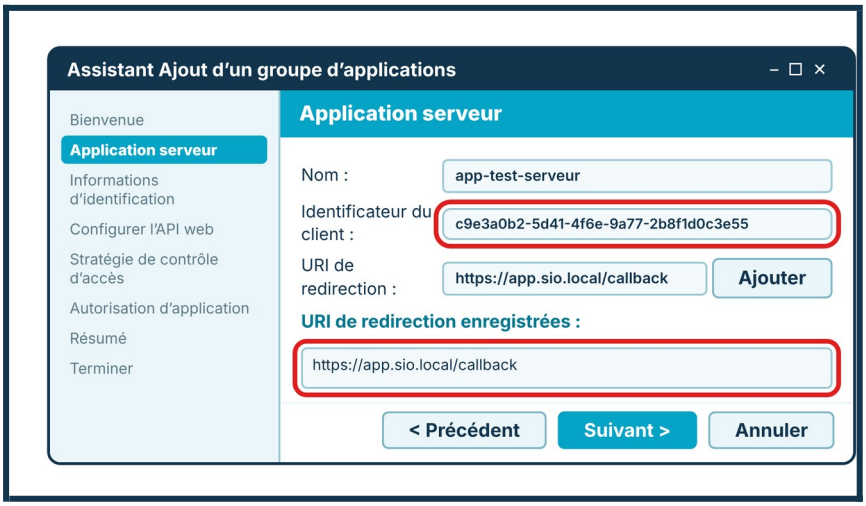


Figure 15 – Assistant : identificateur du client et URI de redirection

À SAVOIR

L'URI de redirection doit correspondre **exactement** à celle que l'application enverra (schéma, nom, chemin, barre finale éventuelle). Pour une vraie application, reprenez l'adresse indiquée dans sa documentation (la plupart l'affichent dans leur page de configuration OpenID Connect).

4 Page « **Informations d'identification** » : choisissez « **Générer une clé secrète partagée** » et **copiez le secret immédiatement** dans votre fichier : AD FS ne le réaffichera plus :



Figure 16 – Assistant : secret partagé de l'application

À SAVOIR

Traitez ce secret comme un mot de passe : ne le publiez pas dans un dépôt ou un document partagé. Si vous le perdez, il faudra en générer un nouveau.

5 Page « **Configurer l'API web** » : nom **app-test-api**, et comme **identificateur** collez le **même GUID que l'identificateur du client**, puis « **Ajouter** » :

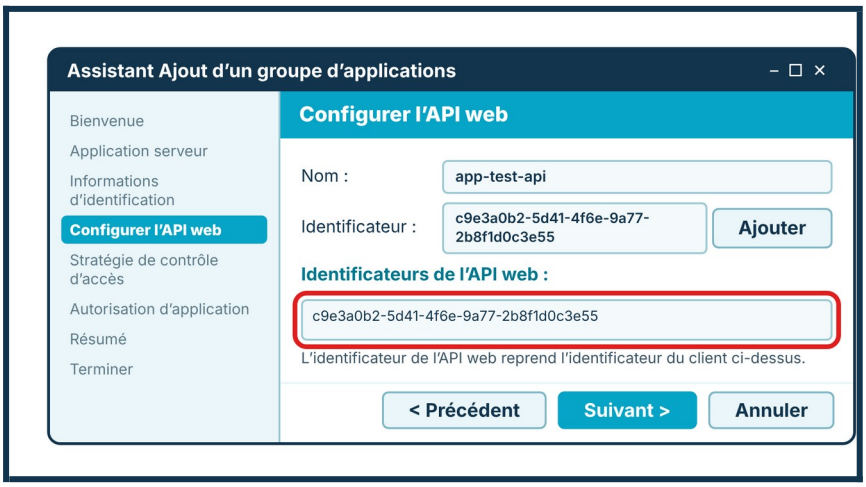


Figure 17 – Assistant : identificateur de l'API web

À SAVOIR

Reprendre l'identificateur du client comme identificateur de l'API web est la pratique habituelle des applications qui ne demandent que l'identité de l'utilisateur (jeton `id_token`).

6 Page « **Stratégie de contrôle d'accès** » : laissez « **Autoriser tout le monde** » pour le laboratoire. En production, choisissez une stratégie qui limite l'accès à un groupe de l'annuaire

7 Page « **Autorisation d'application** » : cochez les étendues **openid** et **allatclaims**, puis « **Suivant** », vérifiez le résumé et cliquez « **Fermer** » à la fin :

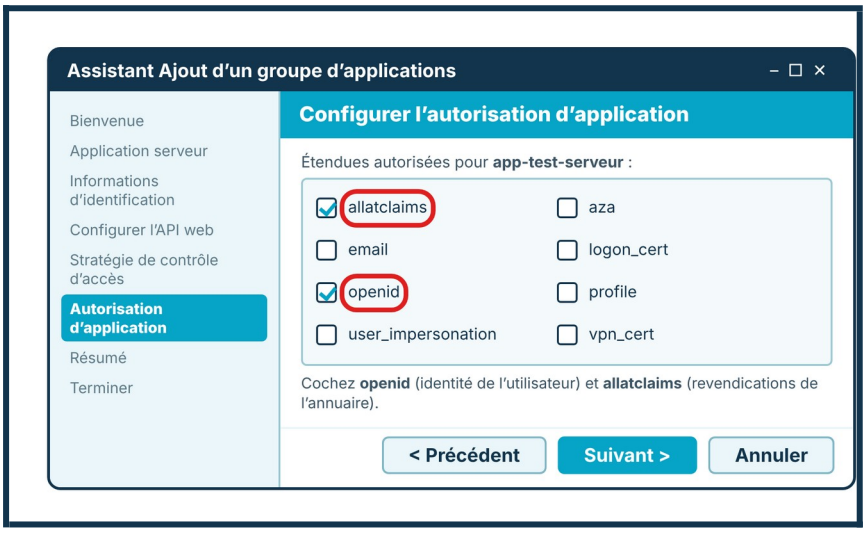


Figure 18 – Assistant : étendues autorisées

À SAVOIR

openid autorise la demande d'identité OpenID Connect ; **allatclaims** permet d'inclure dans le jeton les revendications définies à l'étape suivante.

C – Choisir les revendications placées dans le jeton

Un jeton ne contient que les revendications que l'API web est **configurée** pour émettre. Nous demandons à AD FS de lire dans l'annuaire l'identifiant de connexion (`userPrincipalName`), l'adresse e-mail et le nom affiché de l'utilisateur. La règle est écrite dans le langage de revendications d'AD FS, puis appliquée à l'API web ; la dernière commande vérifie que le groupe d'applications existe :

POWERSHELL (SRV-ADFS)

```
$rules = @'
@RuleTemplate = "LdapClaims"
@RuleName = "Attributs AD"
c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname",
   Issuer == "AD AUTHORITY"]
 => issue(store = "Active Directory",
    types = ("http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn",
             "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress",
             "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name"),
    query = ";userPrincipalName,mail,displayName;{0}", param = c.Value);
'@
Set-AdfsWebApiApplication -TargetName "app-test-api" -IssuanceTransformRules $rules
Get-AdfsApplicationGroup -Name app-test | Select-Object Name, Description
```

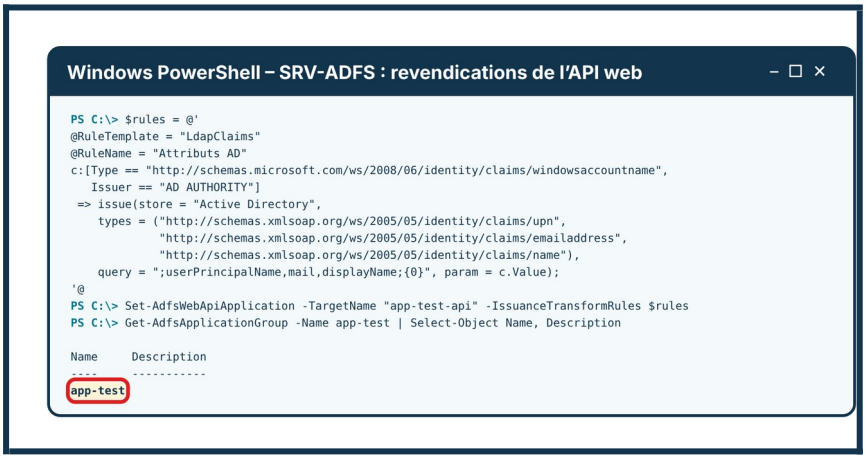


Figure 19 – Règle de revendications et contrôle du groupe d'applications

À SAVOIR

Le marqueur `'@` qui ferme le bloc doit être **seul au début d'une ligne**. Cette règle remplace les règles existantes de l'API web ; vous pouvez aussi la saisir dans la console (propriétés de l'API web > « Règles de transformation d'émission » > « Envoyer les attributs LDAP en tant que revendications »).

D – Tester le flux complet à la main

Nous jouons le rôle de l'application. Sur **SRV-LINUX**, préparons les valeurs (remplacez le GUID par **votre** identificateur du client), puis affichons l'adresse d'autorisation :

TERMINAL

```
BASE=https://adfs.sio.local/adfs/oauth2/authorize
ID=c9e3a0b2-5d41-4f6e-9a77-2b8f1d0c3e55
CB=https://app.sio.local/callback
echo "$BASE?response_type=code&client_id=$ID&redirect_uri=$CB&scope=openid"
```

1. Copiez l'adresse affichée dans le **navigateur** et ouvrez-la : AD FS demande l'identification (ou réutilise la session), puis **redirige** vers `https://app.sio.local/callback?code=…`
2. La page d'arrivée ne s'affichera pas (notre application n'existe pas) : c'est **normal**. **Copiez la valeur du paramètre** `code` dans la barre d'adresse : ce code est **à usage unique** et de courte durée

3 Sur Debian, échangez ce code, avec le secret, contre les jetons (faites-le sans tarder) :

TERMINAL

```
SECRET='<le secret copié à l'étape B>'
CODE='<la valeur du paramètre code>'
curl -s -X POST https://adfs.sio.local/adfs/oauth2/token \
    -d grant_type=authorization_code -d client_id="$ID" \
    -d client_secret="$SECRET" -d redirect_uri="$CB" \
    -d code="$CODE" | python3 -m json.tool | cut -c1-72
```



À SAVOIR

Le jeton `id_token` est la preuve d'identité destinée à l'application. C'est un **JWT** : trois parties séparées par des points. La deuxième contient les revendications ; pour la lire, copiez-la dans `echo "<partie 2>" | tr '_-' '/+' | base64 -d`.

Figure 20 – Debian échange le code contre des jetons

E – Brancher une vraie application

Pour une application réelle, on ne refait pas ce test à la main : on lui donne les valeurs de ce tutoriel, dans son écran de configuration « OpenID Connect » (la plupart des applications récentes savent le faire) :

Champ de l'application	Valeur à saisir
Adresse du fournisseur / *discovery URL*	`https://adfs.sio.local/adfs/.well-known/openid-configuration`
Identifiant client (*client ID*)	L'identificateur du client copié à l'étape B
Secret client (*client secret*)	Le secret partagé copié à l'étape B
Étendues (*scopes*)	`openid` (et `email` ou `profile` si l'application les demande et que vous les avez autorisées)
URI de redirection	Celle de l'application, **déjà déclarée** dans l'assistant (étape B)

L'application doit aussi **faire confiance à SIO-CA** (import du certificat racine, comme pour LDAPS), sinon elle refusera de contacter `adfs.sio.local`.

✔ POINT DE CONTRÔLE

- Le groupe d'applications **app-test** existe, avec le client **app-test-serveur** et l'API web **app-test-api**
- Vous avez noté l'**identificateur du client** et le **secret** dans un endroit sûr
- Le flux à la main renvoie un **id_token** (le code d'autorisation a bien été échangé)

Notre domaine **sio.local** dispose maintenant d'un véritable **fournisseur d'identité** : une ferme AD FS, protégée par un certificat de notre autorité, et un modèle de groupe d'applications reproductible pour chaque nouvelle application. Il ne reste qu'à saisir ces valeurs dans les applications du laboratoire.

Dépannage : les symptômes les plus fréquents

Symptôme	Cause probable	Solution
`Install-AdfsFarm` : « **certificates with the CNG private key are not supported** »	Le certificat utilise une clé CNG (fournisseur KSP)	Refaire le certificat avec le modèle **SIO-ADFS** (fournisseur hérité, étape 3) et vérifier `ProviderName`
`Install-AdfsFarm` : **erreur sur le compte gMSA**	Clé racine KDS absente ou pas encore valable, ou compte non joignable	`Get-KdsRootKey` sur un contrôleur ; recréer la clé antidatée (étape 2) ; vérifier le domaine et le DNS
Le nom du service n'est pas accepté	`adfs.sio.local` absent du CN et du SAN du certificat	Refaire la demande avec `-SubjectName` et `-DnsName` corrects
`Get-Certificate` : **modèle introuvable ou accès refusé**	Modèle non publié, droit **Inscrire** manquant, réplication en cours	Contrôler la publication (3-B) et les droits (3-A) ; attendre la réplication ; `gpupdate /force`
Navigateur : alerte de certificat sur `adfs.sio.local`	Le poste ne fait pas confiance à SIO-CA	Poste du domaine : `gpupdate /force` ; hors domaine : importer le certificat racine
`idpinitiatedsignon.aspx` : **page introuvable**	La page de test est désactivée	`Set-AdfsProperties -EnableIdPInitiatedSignonPage $true` (étape 5-A)
`curl` : « **unable to get local issuer certificate** »	Debian ne connaît pas SIO-CA	Importer la CA (étape 5 du tutoriel LDAPS)
Erreur après l'identification (client ou URI de redirection refusés)	La `redirect_uri` envoyée diffère de celle de l'assistant	Comparer caractère par caractère (casse, barre finale) et corriger dans le groupe d'applications
Échange du code refusé (`invalid_grant`**…)**	Code déjà utilisé, expiré, ou secret erroné	Refaire l'autorisation pour obtenir un nouveau code, puis relancer `curl` aussitôt
Le jeton ne contient ni e-mail ni nom	Règle de revendications absente, ou étendue **allatclaims** non cochée	Refaire l'étape 6-C et vérifier les étendues (étape 6-B)

Pense-bête : où regarder

Où / commande	Utilité
`Microsoft.IdentityServer.msc` **(dans** `C:\Windows\ADFS`**)**	Console de gestion d'AD FS : groupes d'applications, approbations, stratégies
`Get-AdfsProperties`	Nom du service, ports et options de la ferme
`Get-AdfsEndpoint`	Liste des points de terminaison (découverte, métadonnées, jetons)
`Get-AdfsCertificate`	Certificats de communication, de signature et de déchiffrement
`Get-AdfsApplicationGroup`, `Get-AdfsWebApiApplication`	Contrôle des groupes d'applications et de leurs API web
`Get-KdsRootKey`	Vérification de la clé racine nécessaire au gMSA
Journal « Observateur d'événements > AD FS > Admin »	Erreurs détaillées du service de fédération
`https://adfs.sio.local/adfs/.well-known/openid-configuration`	Document de découverte OpenID Connect
`https://adfs.sio.local/FederationMetadata/2007-06/FederationMetadata.xml`	Métadonnées SAML à fournir aux applications SAML