Mise en place d'un Haute Disponibilité sur Serveur Broker avec jonction d'une base SQL

Prérequis

- Serveur SQL installé (exemple : SRV-SQL à 192.168.1.109)
- Groupe utilisateurs "Grp_RDS_Brk" créé dans Active Directory avec le serveur Broker en membre
- Serveur Broker RDSHA (exemple : 192.168.1.105), hôte disponible dans DNS
- Ports TCP ouverts (ex : 1433 pour SQL Server)
- Pare-feu configuré pour autoriser les flux nécessaires sur SRV-SQL et SRV-BROKER

Étapes principales

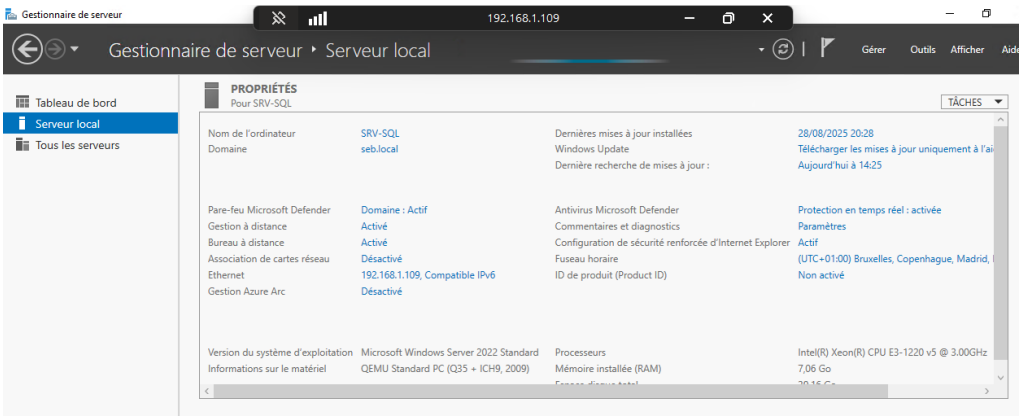
1. **Création et configuration hôte dans DNS**
    - Créer l'hôte A pour le serveur Broker (par exemple 192.168.1.105)
    - Créer un nom DNS cluster comme SRV-BROKER.seb.local utilisé pour les connexions clients RDS
2. **Création du groupe utilisateurs dans Active Directory**
    - Créer un groupe utilisateur (exemple : Grp_RDS_Brk)
    - Ajouter le serveur Broker comme membre de ce groupe
3. **Déploiement base SQL RDSHA**
    - Installer la base de données RDSHA sur le serveur SRV-SQL (192.168.1.109)
    - Configurer la base RDSHA avec les droits nécessaires pour les connexions Broker par le groupe d'utilisateurs
    - Exemple de chaîne de connexion ODBC à utiliser dans la configuration du Broker :

text

DRIVER=ODBC Driver 18 for SQL Server;SERVER=192.168.1.109,1433;DATABASE=RDSHA;Trusted_Connection=Yes;TrustServerCertificate=yes;

4. **Configuration pare-feu**
    - Vérifier les règles pare-feu sur SRV-SQL pour autoriser le port TCP 1433
    - Vérifier les règles sur SRV-BROKER pour autoriser la communication vers la base SQL
5. **Configuration du service Broker en haute disponibilité**
    - Utiliser le nom DNS cluster SRV-BROKER.seb.local dans la configuration des services RDS
    - S'assurer que le Broker utilise la chaîne de connexion ODBC à la base RDSHA
    - Configuration de la redondance et basculement selon la solution mise en place (exemple : failover clustering, load balancing)

Mise en place de la base SQL sur le serveur SRV-SQL en 192.168.1.109

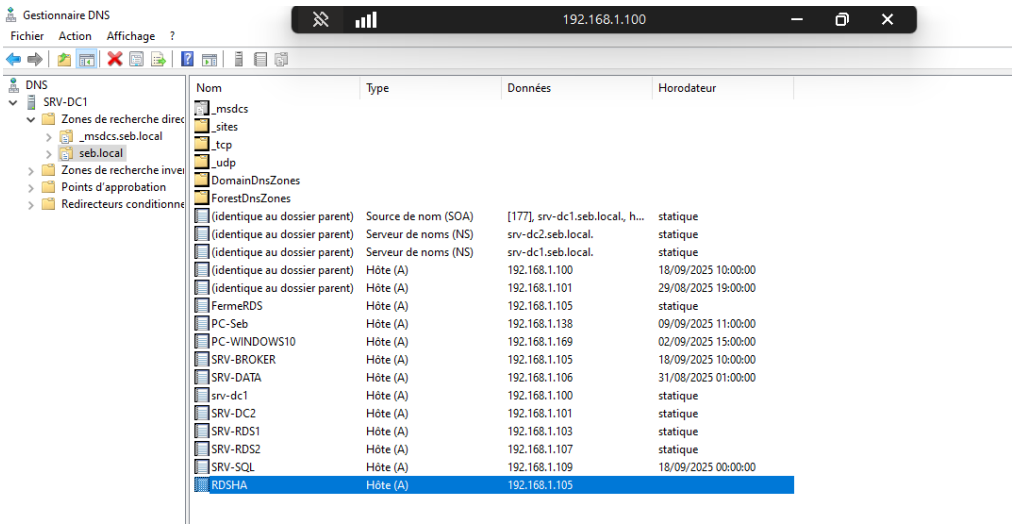


Prérequis : création hôte (A)

RDSHA en 192.168.1.105 pareil au serveur Broker 192.168.1.105

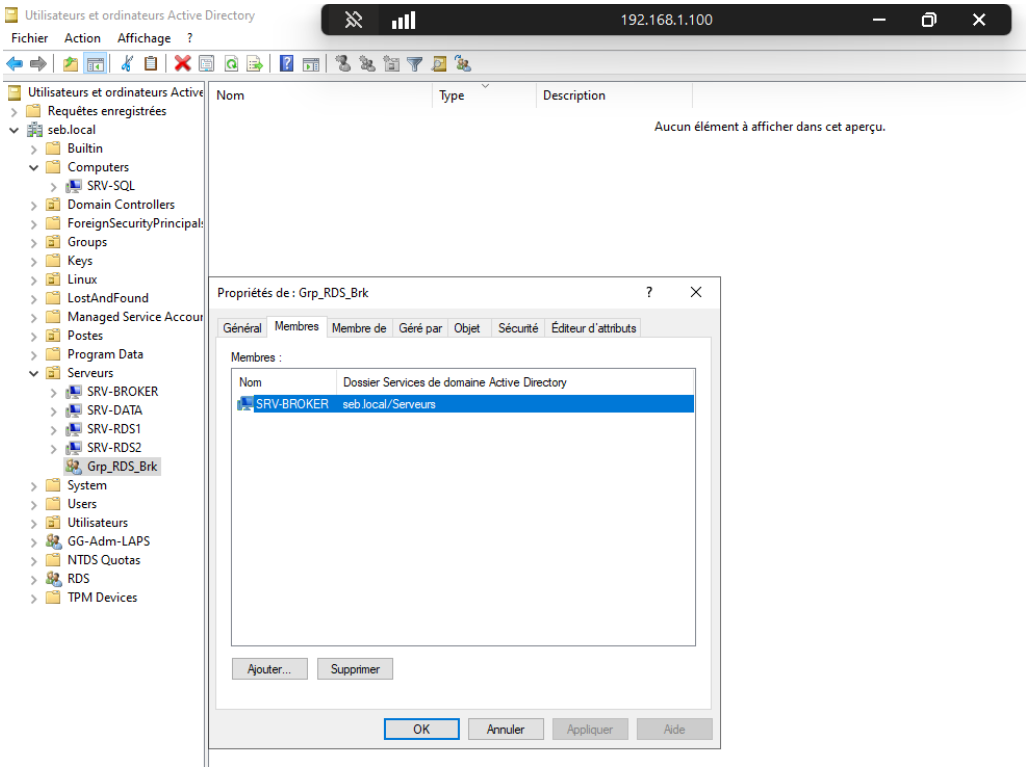
Sur SEB.LOCAL

DNS :

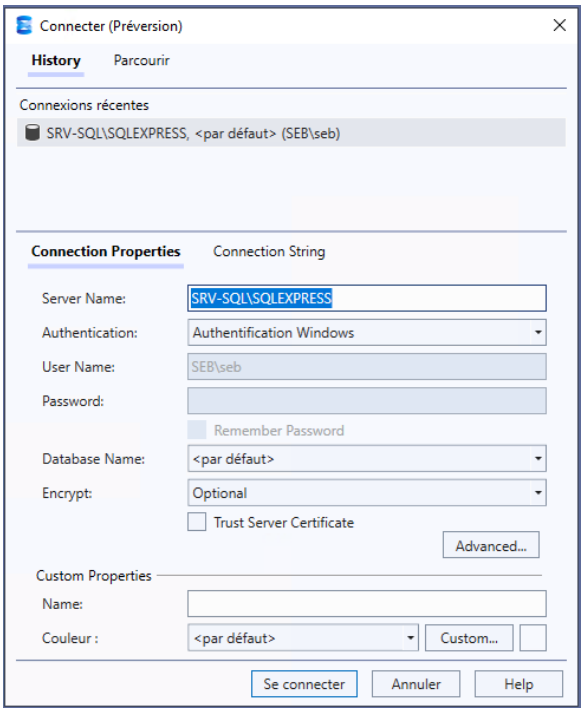


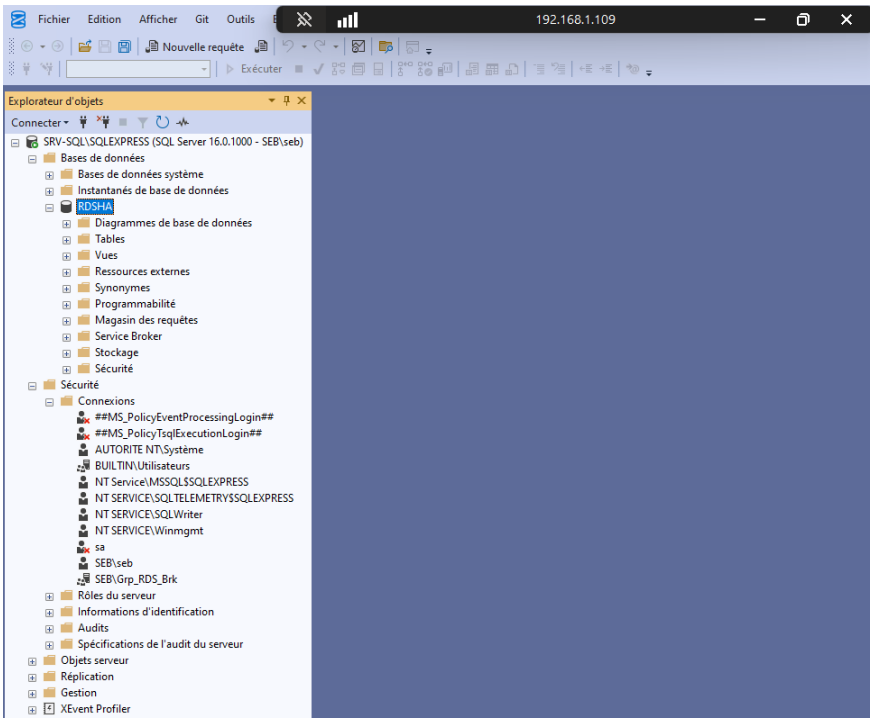
Prérequis : Création groupe utilisateurs avec en membre le serveur Broker

AD :

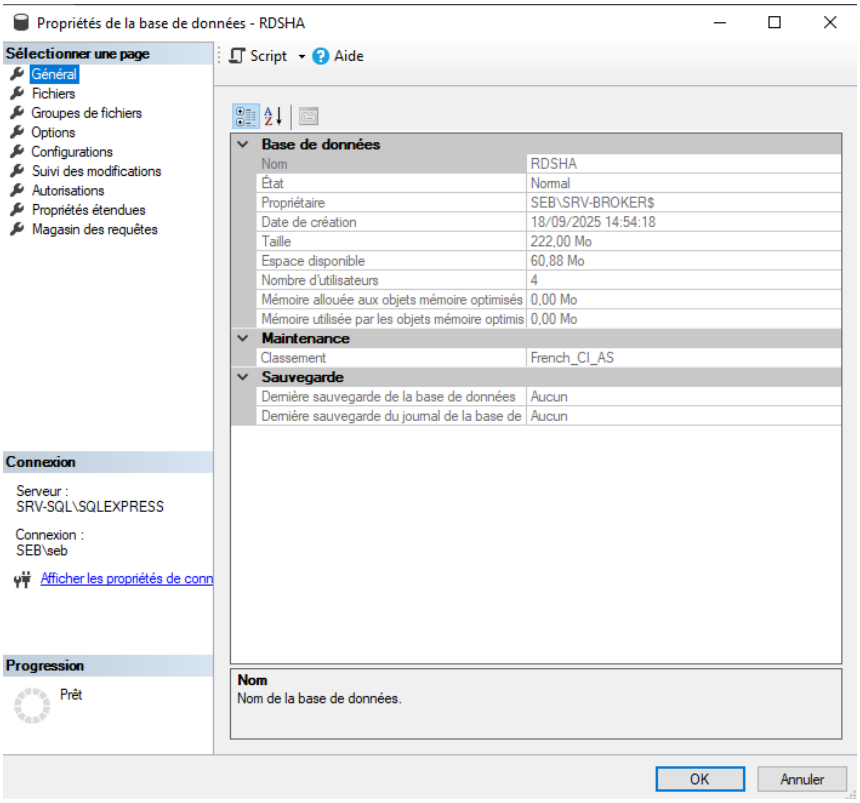


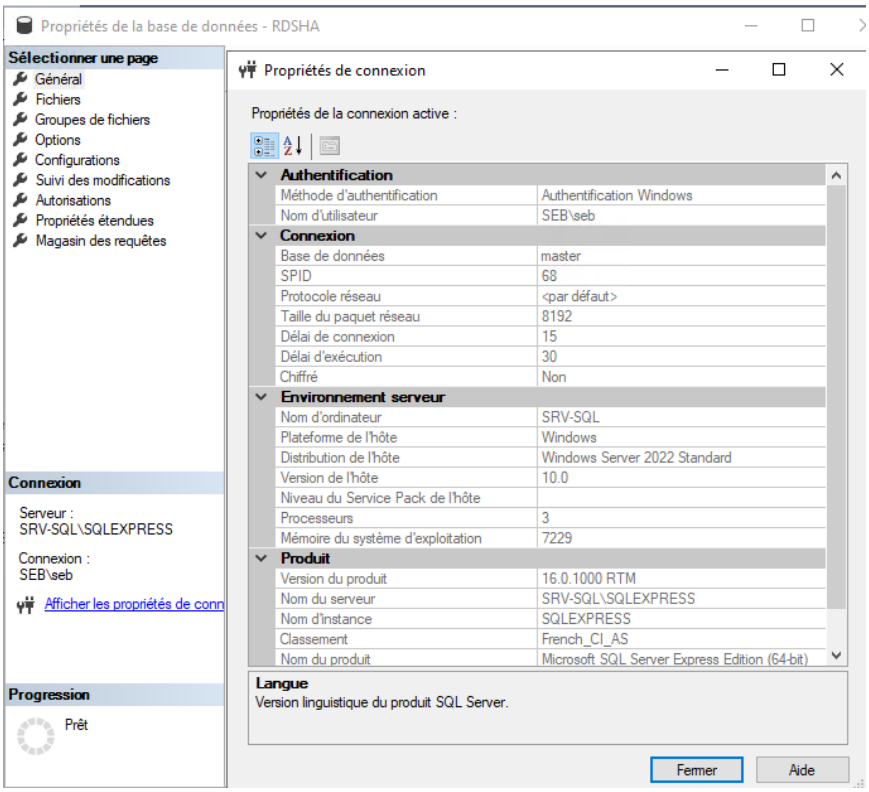
Base SQL :



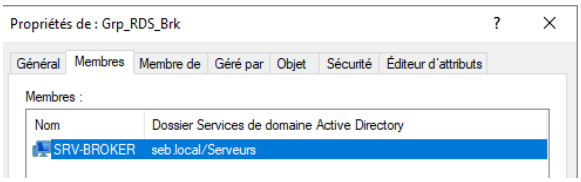


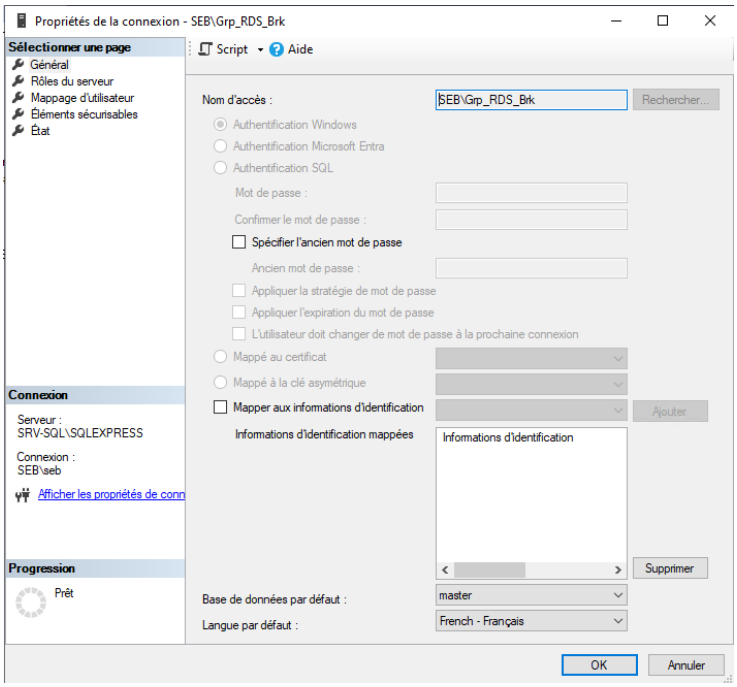
Base de données RDSHA :

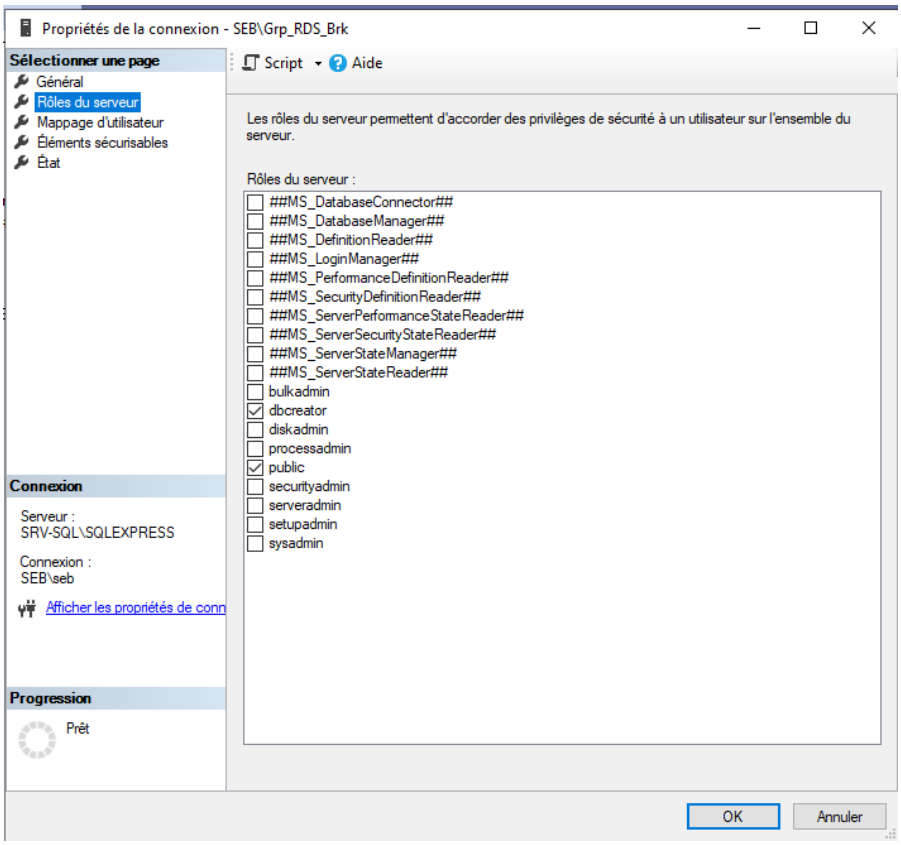


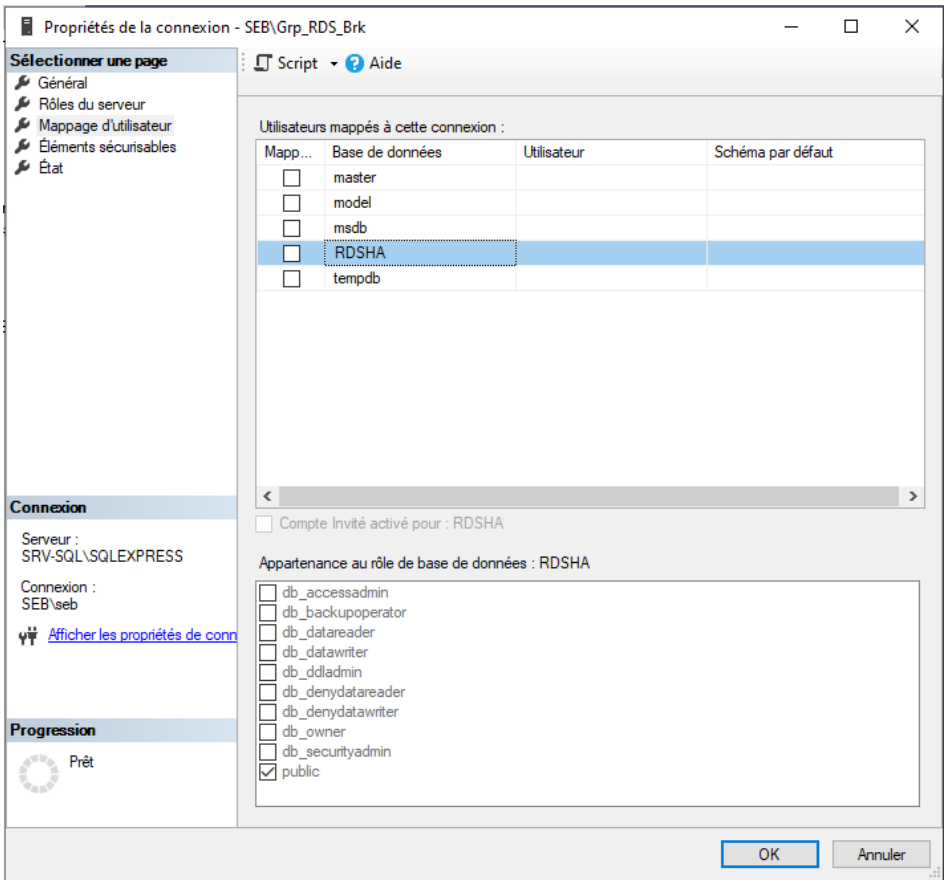


Connexion à la base RDSHA avec Grp_RDS_Brk

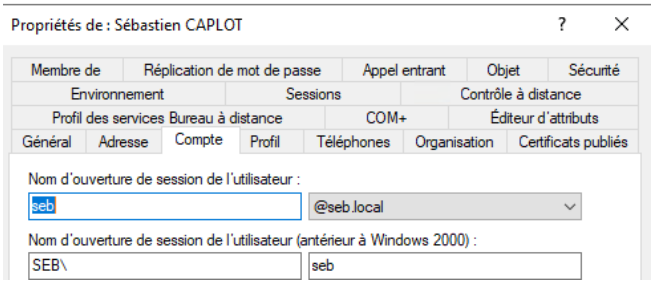


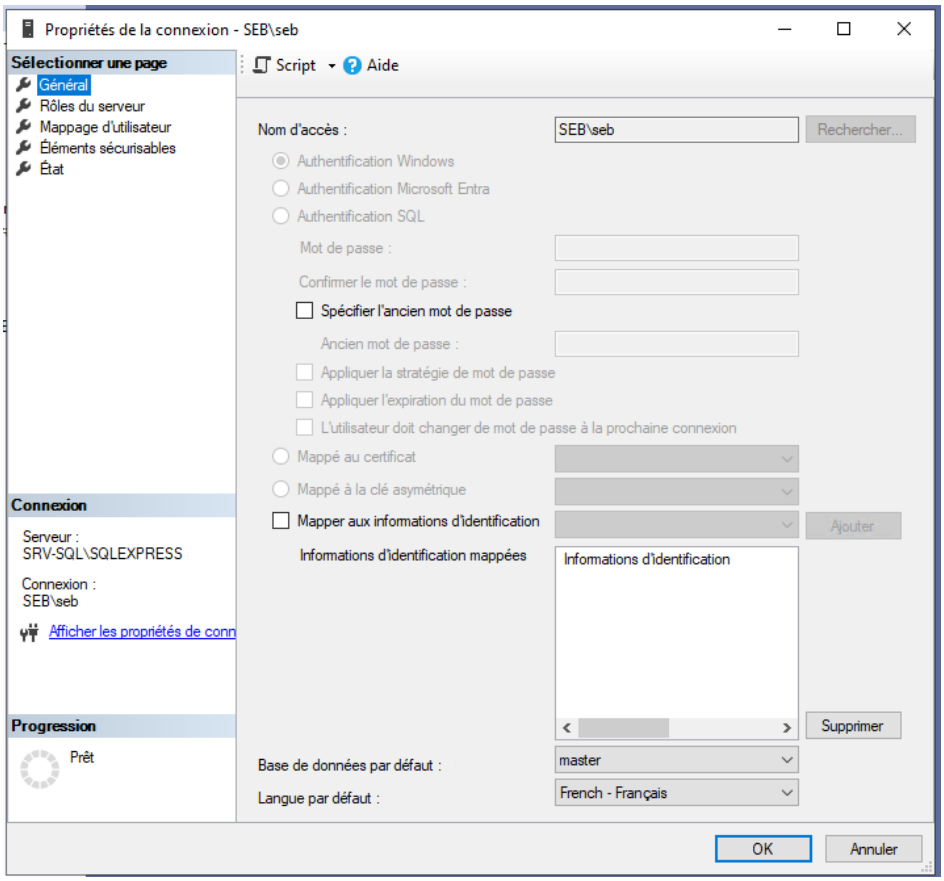


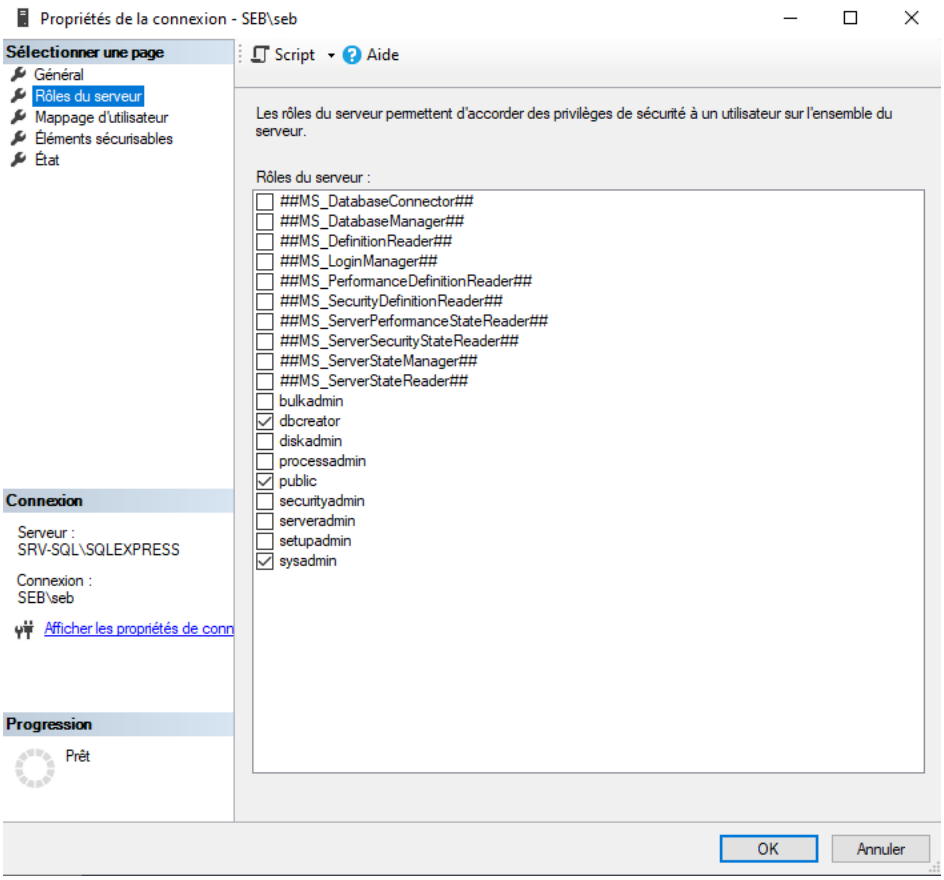

Propriétés de la connexion - SEB\Grp_RDS_Brk
Sélectionner une page
Général
Rôles du serveur
Mappage d'utilisateur
Éléments sécurisables
État
Script
Aide
Les rôles du serveur permettent d'accorder des privilèges de sécurité à un utilisateur sur l'ensemble du serveur.
Rôles du serveur :
##MS_DatabaseConnector##
##MS_DatabaseManager##
##MS_DefinitionReader##
##MS_LoginManager##
##MS_PerformanceDefinitionReader##
##MS_SecurityDefinitionReader##
##MS_ServerPerformanceStateReader##
##MS_ServerSecurityStateReader##
##MS_ServerStateManager##
##MS_ServerStateReader##
bulkadmin
dbcreator
diskadmin
processadmin
public
securityadmin
serveradmin
setupadmin
sysadmin
Connexion
Serveur :
SRV-SQL\SQLEXPRESS
Connexion :
SEB\seb
Afficher les propriétés de conn
Progression
Prêt
OK
Annuler

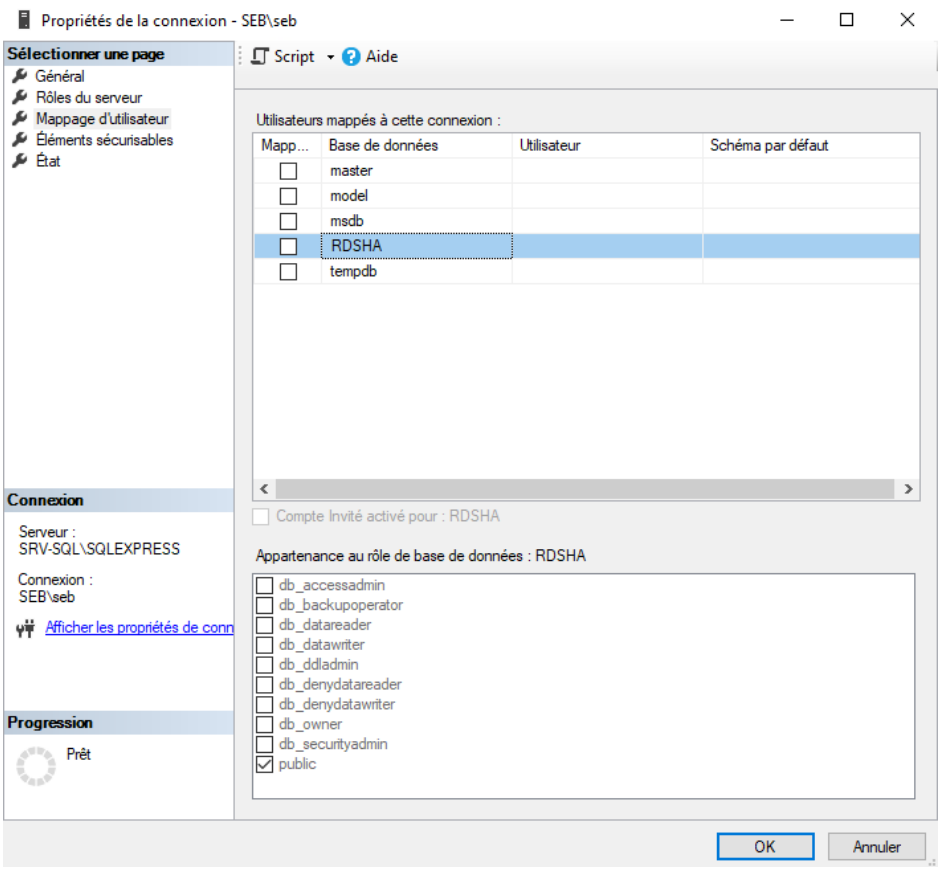

Propriétés de la connexion - SEB\Grp_RDS_Brk
Sélectionner une page
Général
Rôles du serveur
Mappage d'utilisateur
Éléments sécurisables
État
Script
Aide
Utilisateurs mappés à cette connexion :
Mapp...
Base de données
Utilisateur
Schéma par défaut
master
model
msdb
RDSHA
tempdb
Compte Invité activé pour : RDSHA
Appartenance au rôle de base de données : RDSHA
db_accessadmin
db_backupoperator
db_datareader
db_datawriter
db_ddladmin
db_denydatareader
db_denydatawriter
db_owner
db_securityadmin
public
Connexion
Serveur :
SRV-SQL\SQLEXPRESS
Connexion :
SEB\seb
Afficher les propriétés de conn
Progression
Prêt
OK
Annuler

Connexions à la base RDSHA avec SEB\seb

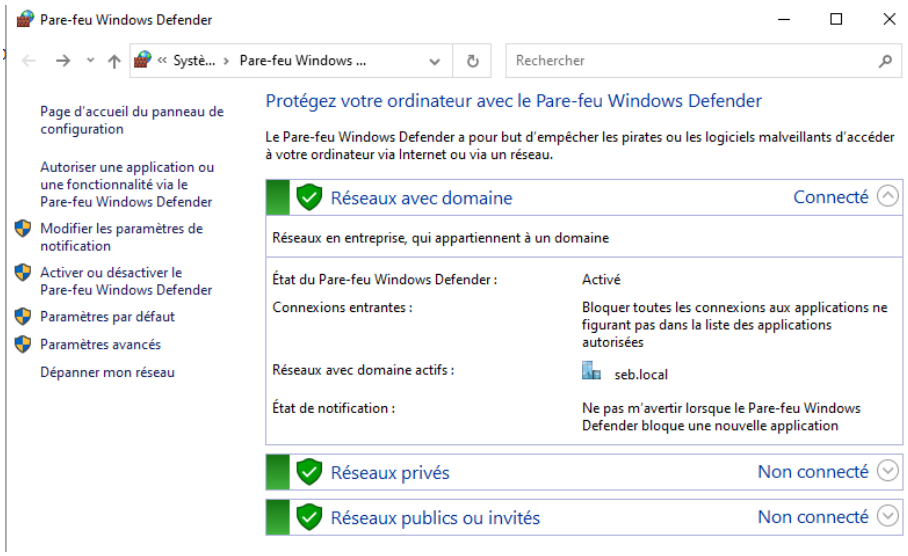




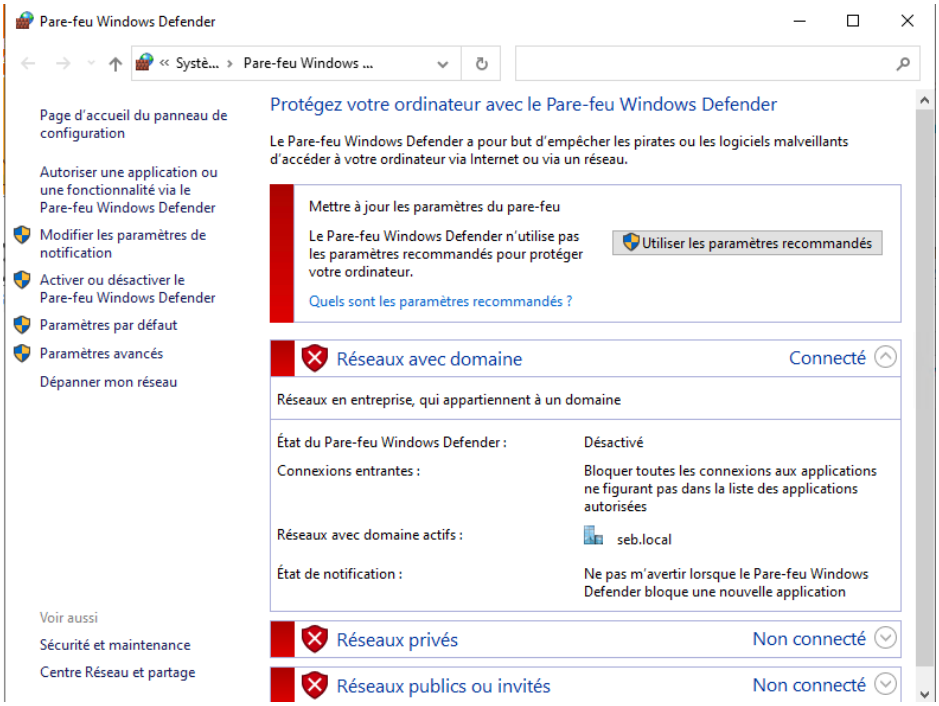

Propriétés de la connexion - SEB\seb
Sélectionner une page
Général
Rôles du serveur
Mappage d'utilisateur
Éléments sécurisables
État
Script
Aide
Les rôles du serveur permettent d'accorder des privilèges de sécurité à un utilisateur sur l'ensemble du serveur.
Rôles du serveur :
##MS_DatabaseConnector##
##MS_DatabaseManager##
##MS_DefinitionReader##
##MS_LoginManager##
##MS_PerformanceDefinitionReader##
##MS_SecurityDefinitionReader##
##MS_ServerPerformanceStateReader##
##MS_ServerSecurityStateReader##
##MS_ServerStateManager##
##MS_ServerStateReader##
bulkadmin
dbcreator
diskadmin
processadmin
public
securityadmin
serveradmin
setupadmin
sysadmin
Connexion
Serveur :
SRV-SQL\SQLEXPRESS
Connexion :
SEB\seb
Afficher les propriétés de conn
Progression
Prêt
OK
Annuler


Propriétés de la connexion - SEB\seb
Sélectionner une page
Général
Rôles du serveur
Mappage d'utilisateur
Éléments sécurisables
État
Script
Aide
Utilisateurs mappés à cette connexion :
Mapp...
Base de données
Utilisateur
Schéma par défaut
master
model
msdb
RDSHA
tempdb
Compte Invité activé pour : RDSHA
Appartenance au rôle de base de données : RDSHA
db_accessadmin
db_backupoperator
db_datareader
db_datawriter
db_ddladmin
db_denydatareader
db_denydatawriter
db_owner
db_securityadmin
public
Connexion
Serveur :
SRV-SQL\SQLEXPRESS
Connexion :
SEB\seb
Afficher les propriétés de conn
Progression
Prêt
OK
Annuler

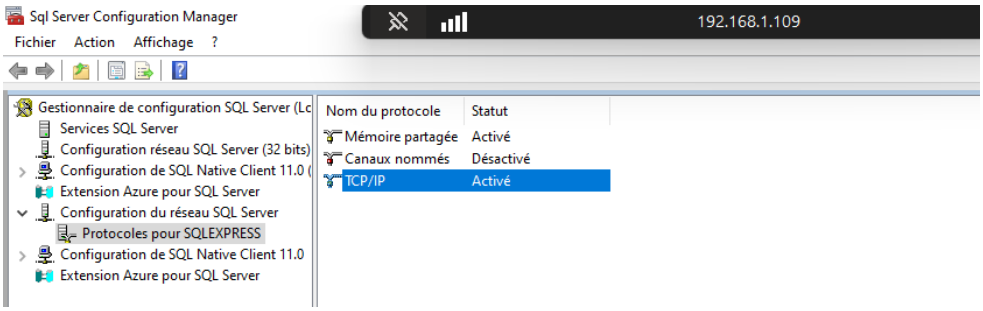
Pare feu Windows sur SRV-SQL

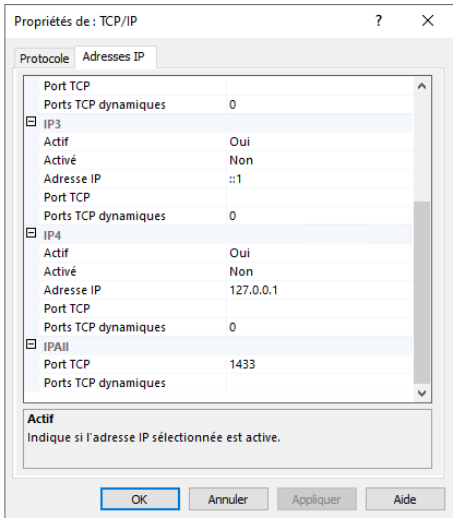


Pare feu sur SRV-BROKER



Port TCP sur SRV-SQL :





IPAII : Port TCP en 1433 et Port TCP Dynamiques : ne rien mettre

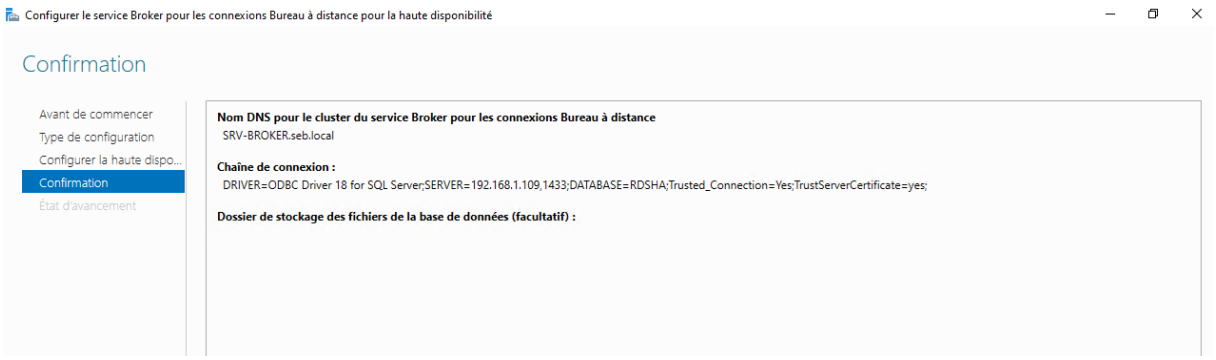
Paramétrage de la mise en Haute Disponibilité du Broker

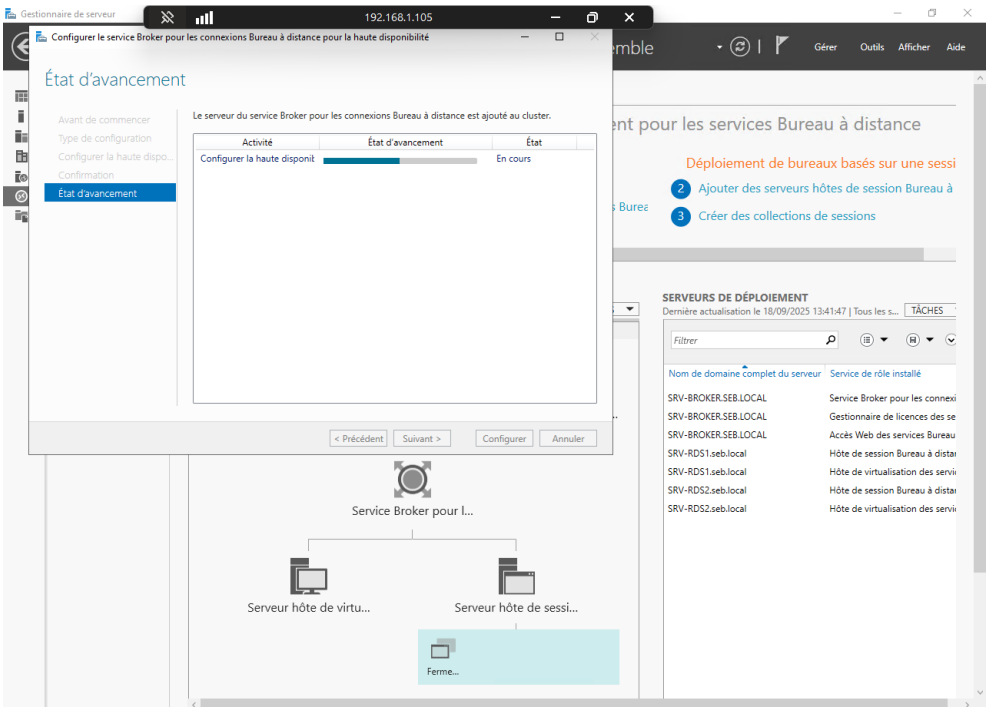
Nom DNS cluster du service broker pour les connexions Bureau à Distance :

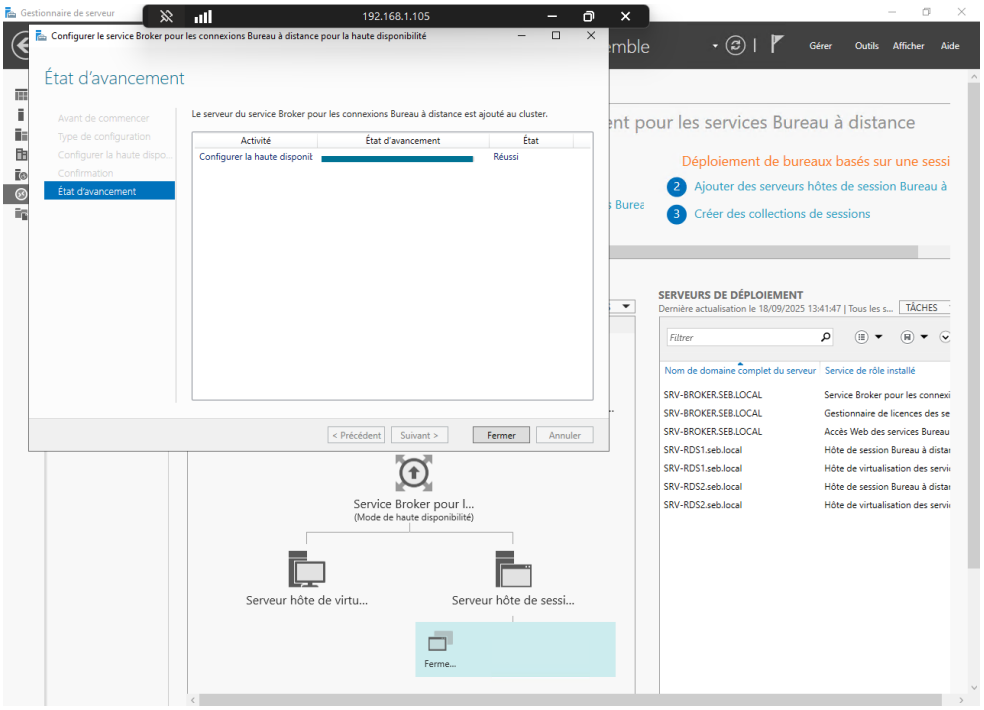
SRV-BROKER.seb.local

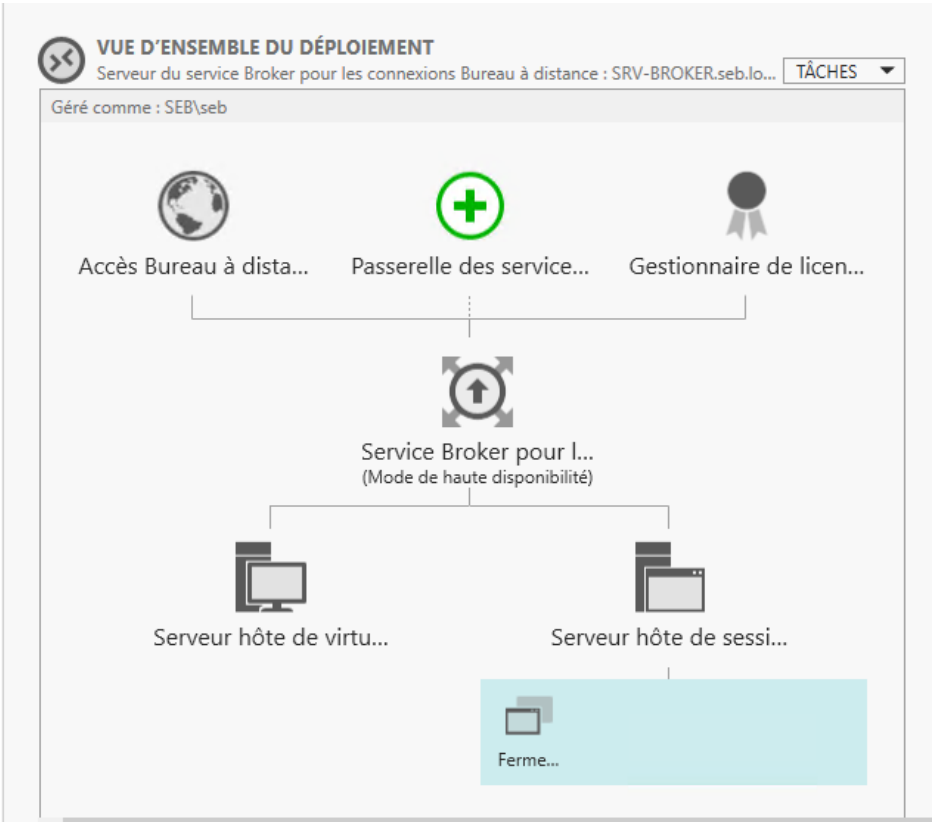
Chaîne de connexion :

DRIVER=ODBC Driver 18 for SQL Server;SERVER=192.168.1.109,1433;DATABASE=RDSHA;Trusted_Connection=Yes;TrustServerCertificate=yes;




Gestionnaire de serveur
192.168.1.105
Configurer le service Broker pour les connexions Bureau à distance pour la haute disponibilité
État d'avancement
Avant de commencer
Type de configuration
Configurer la haute dispo...
Confirmation
État d'avancement
Le serveur du service Broker pour les connexions Bureau à distance est ajouté au cluster.
Activité
État d'avancement
État
Configurer la haute disponit
En cours
< Précédent
Suivant >
Configurer
Annuler
Service Broker pour l...
Serveur hôte de virtu...
Serveur hôte de sessi...
Ferme...
Gérer
Outils
Afficher
Aide
nt pour les services Bureau à distance
Déploiement de bureaux basés sur une sessi
2 Ajouter des serveurs hôtes de session Bureau à
3 Créer des collections de sessions
SERVEURS DE DÉPLOIEMENT
Dernière actualisation le 18/09/2025 13:41:47 | Tous les s...
TÂCHES
Filtrer
Nom de domaine complet du serveur
Service de rôle installé
SRV-BROKER.SEB.LOCAL
Service Broker pour les connexi
SRV-BROKER.SEB.LOCAL
Gestionnaire de licences des se
SRV-BROKER.SEB.LOCAL
Accès Web des services Bureau
SRV-RDS1.seb.local
Hôte de session Bureau à distar
SRV-RDS1.seb.local
Hôte de virtualisation des servi
SRV-RDS2.seb.local
Hôte de session Bureau à distar
SRV-RDS2.seb.local
Hôte de virtualisation des servi


Gestionnaire de serveur
192.168.1.105
Configurer le service Broker pour les connexions Bureau à distance pour la haute disponibilité
État d'avancement
Avant de commencer
Type de configuration
Configurer la haute dispo...
Confirmation
État d'avancement
Le serveur du service Broker pour les connexions Bureau à distance est ajouté au cluster.
Activité
État d'avancement
État
Configurer la haute disponit
Réussi
< Précédent
Suivant >
Fermer
Annuler
Service Broker pour l...
(Mode de haute disponibilité)
Serveur hôte de virtu...
Serveur hôte de sessi...
Ferme...
Gérer
Outils
Afficher
Aide
nt pour les services Bureau à distance
Déploiement de bureaux basés sur une sessi
2 Ajouter des serveurs hôtes de session Bureau à
3 Créer des collections de sessions
SERVEURS DE DÉPLOIEMENT
Dernière actualisation le 18/09/2025 13:41:47 | Tous les s...
TÂCHES
Filtrer
Nom de domaine complet du serveur
Service de rôle installé
SRV-BROKER.SEB.LOCAL
Service Broker pour les connexi
SRV-BROKER.SEB.LOCAL
Gestionnaire de licences des se
SRV-BROKER.SEB.LOCAL
Accès Web des services Bureau
SRV-RDS1.seb.local
Hôte de session Bureau à distar
SRV-RDS1.seb.local
Hôte de virtualisation des servi
SRV-RDS2.seb.local
Hôte de session Bureau à distar
SRV-RDS2.seb.local
Hôte de virtualisation des servi


VUE D'ENSEMBLE DU DÉPLOIEMENT
Serveur du service Broker pour les connexions Bureau à distance : SRV-BROKER.seb.lo...
TÂCHES
Géré comme : SEB\seb
Accès Bureau à dista...
Passerelle des service...
Gestionnaire de licen...
Service Broker pour l...
(Mode de haute disponibilité)
Serveur hôte de virtu...
Serveur hôte de sessi...
Ferme...

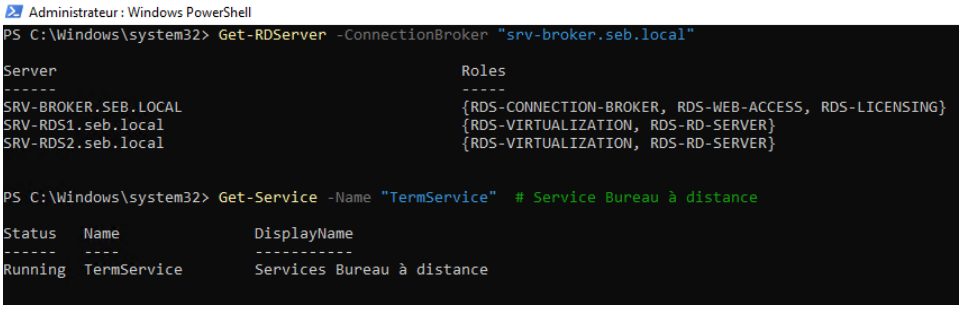
Commandes et contrôles pour tester la haute disponibilité

Contrôle du service Broker RDS

- Vérifier que le service Broker RDS est en état "Running":

powershell

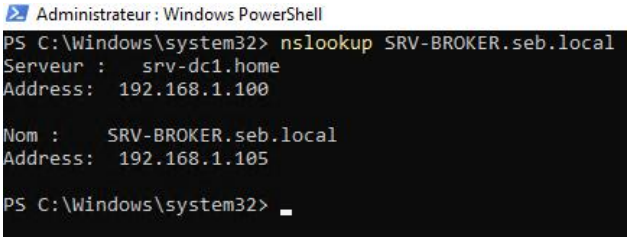
Get-Service -Name "TermService" # Service Bureau à distance



- Test de résolution DNS du cluster Broker :

text

nslookup SRV-BROKER.seb.local



- Ping sur le nom DNS cluster pour vérifier la connectivité réseau :

text

ping SRV-BROKER.seb.local

Vérification des connexions à la base SQL RDSHA

Vérification DNS et réseau

- Vérifier la résolution du nom DNS cluster du broker :

powershell

Resolve-DnsName SRV-BROKER.seb.local

- Tester la connectivité réseau vers le broker :

powershell

Test-Connection SRV-BROKER.seb.local

Contrôle des connexions TCP sur le port SQL

- **Lister les connexions TCP vers le port 1433 (SQL Server) sur la machine locale :**

powershell

Get-NetTCPConnection -RemotePort 1433

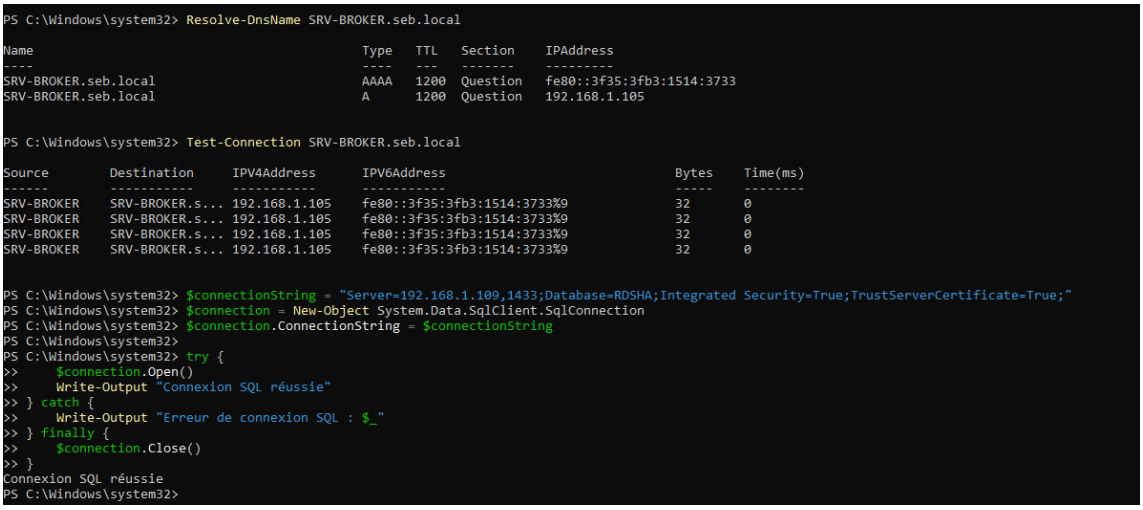
Tester la connexion à la base SQL avec PowerShell (en utilisant .NET)

Exemple de script simple qui teste la connexion à SQL Server via la chaîne de connexion utilisée par le broker RDS :

powershell

```
$connectionString = "Server=192.168.1.109,1433;Database=RDSHA;Integrated Security=True;TrustServerCertificate=True;"
$connection = New-Object System.Data.SqlClient.SqlConnection
$connection.ConnectionString = $connectionString

try {
    $connection.Open()
    Write-Output "Connexion SQL réussie"
} catch {
    Write-Output "Erreur de connexion SQL : $_"
} finally {
    $connection.Close()
}
```



Vérifier les sessions SQL actives liées au Broker :

Sur SRV-SQL

SELECT *session_id, login_name,* ***status FROM*** *sys.dm_exec_sessions* ***WHERE*** *login_name LIKE '%Broker%'*

Tests fonctionnels haute disponibilité

- Arrêter le service Broker sur un nœud et vérifier la bascule vers un autre nœud si configuré en cluster :

powershell

Stop-Service -Name BrokerServiceName # Adapter selon le nom exact du service broker

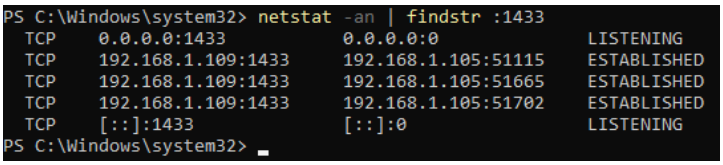
- Surveiller à la fois le fonctionnement des connexions RDS via le cluster DNS et la base SQL en bascule.
-

Contrôle des ports TCP utilisés

- Vérifier le port TCP 1433 en écoute sur le serveur SQL :

text

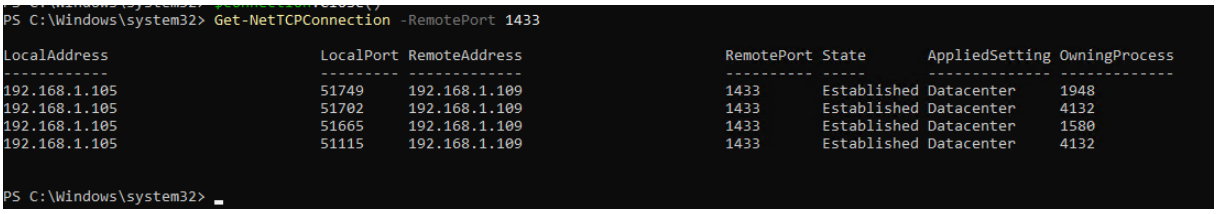
netstat -an | findstr :1433



- Vérifier les connexions ouvertes entre Broker et SQL sur le port 1433 :

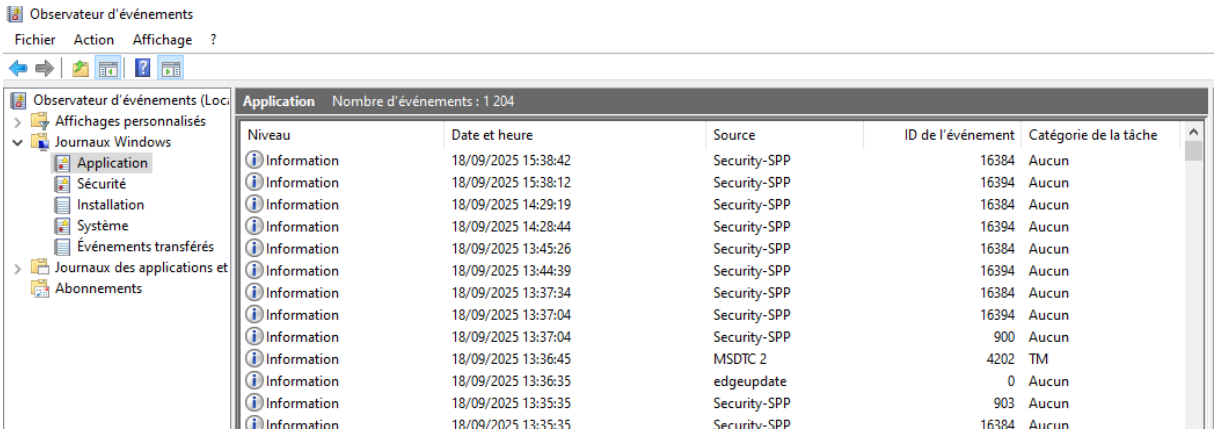
powershell

Get-NetTCPConnection -RemotePort 1433

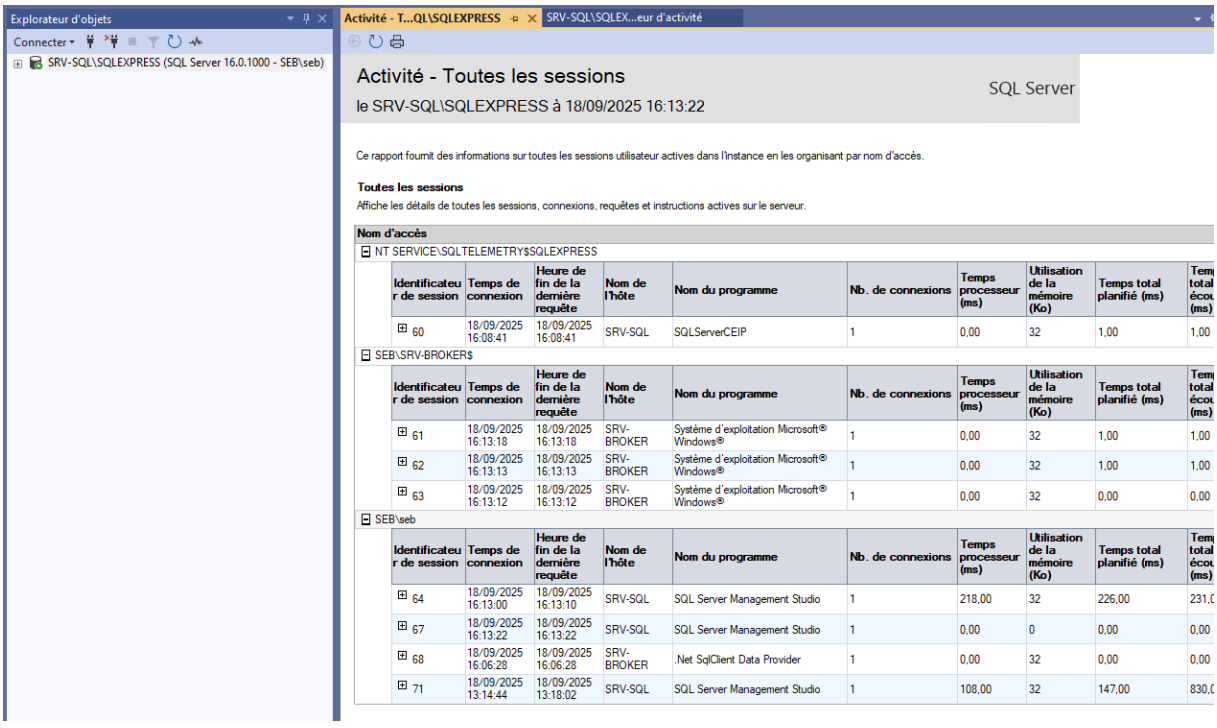


Logs à vérifier

- Logs événements Windows pour Broker RDS :
    - Ouvrir Observateur d'événements → Journaux Windows → Application et Système



- Logs SQL Server pour erreurs et connexions

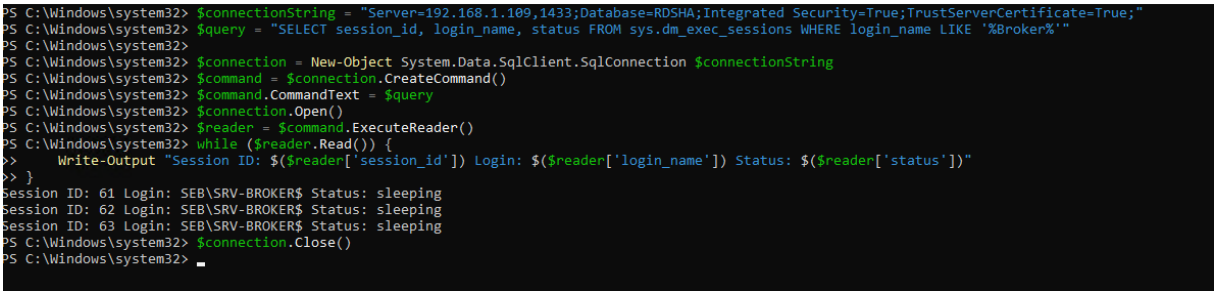


- Logs système et application su SRV-BROKER pour bascule et erreurs

Pour exécuter une requête SQL dans PowerShell, il faut d'abord établir une connexion SQL Server via les objets .NET :

```
$connectionString = "Server=192.168.1.109,1433;Database=RDSHA;Integrated Security=True;TrustServerCertificate=True;"
$query = "SELECT session_id, login_name, status FROM sys.dm_exec_sessions WHERE login_name LIKE '%Broker%'"

$connection = New-Object System.Data.SqlClient.SqlConnection $connectionString
$command = $connection.CreateCommand()
$command.CommandText = $query
$connection.Open()
$reader = $command.ExecuteReader()
while ($reader.Read()) {
    Write-Output "Session ID: $($reader['session_id']) Login: $($reader['login_name']) Status: $($reader['status'])"
}
$connection.Close()
```



La capture montre que la commande PowerShell utilisant .NET pour interroger SQL Server fonctionne parfaitement. Le script établi la connexion à la base de données, exécute la requête SQL sur sys.dm_exec_sessions et retourne les sessions actives du broker avec succès. Les résultats affichent les IDs de session, le login utilisé (SEB\SRV-BROKER$) et l'état de chaque session (sleeping), ce qui confirme que la liaison PowerShell/SQL Server est fonctionnelle et permet de superviser le broker RDS efficacement.

Phase 2 : SRV-BROKER2

Actuellement je ne possède qu'un seul broker donc pas possible de tester le basculement :
Pour tester la haute disponibilité d'un Broker RDS, il est effectivement nécessaire d'avoir au moins deux Brokers configurés en mode haute disponibilité.

Pourquoi un deuxième Broker est nécessaire ?

- La haute disponibilité consiste à assurer la continuité de service même si un serveur Broker tombe en panne.
- Avec un seul Broker, il n'y a pas de redondance, donc pas de bascule possible en cas de problème.
- En ajoutant un deuxième Broker, il est possible de configurer un cluster ou un mécanisme de bascule automatique pour répartir la charge et assurer la continuité.
- Les clients utilisent un nom DNS cluster (ex : SRV-BROKER.seb.local) qui pointe vers ce groupe de Brokers.

Configuration typique pour un deuxième Broker

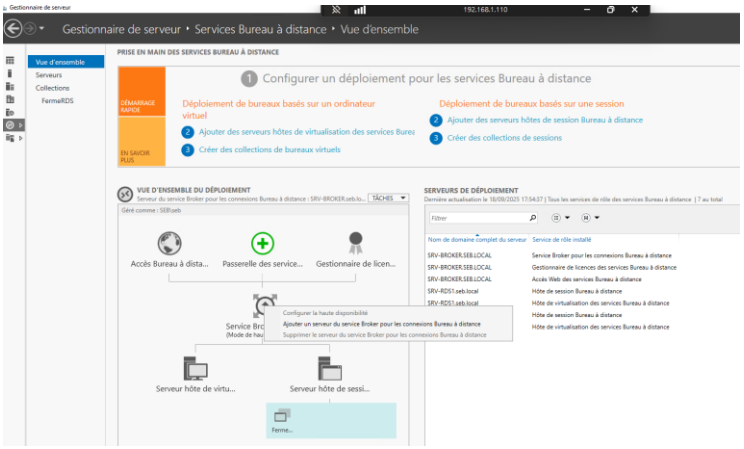
- Installer et configurer le même rôle Broker sur un deuxième serveur.
- Ajouter ce serveur Broker au groupe AD prévu (ex : Grp_RDS_Brk).
- Configurer la base SQL (RDSHA) pour accepter les connexions des deux Brokers.
- Utiliser le nom DNS cluster commun pour la connexion aux brokers.
- Mettre en place le mécanisme de bascule côté infrastructure réseau ou par cluster.

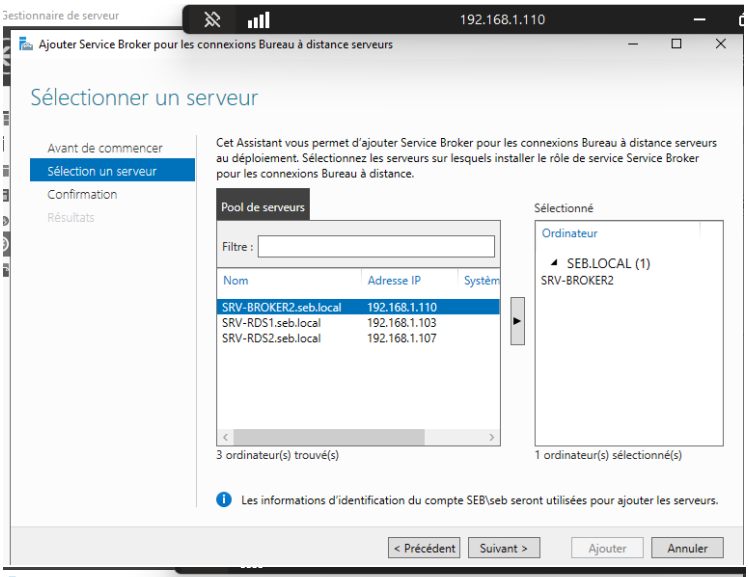
Test de la haute disponibilité

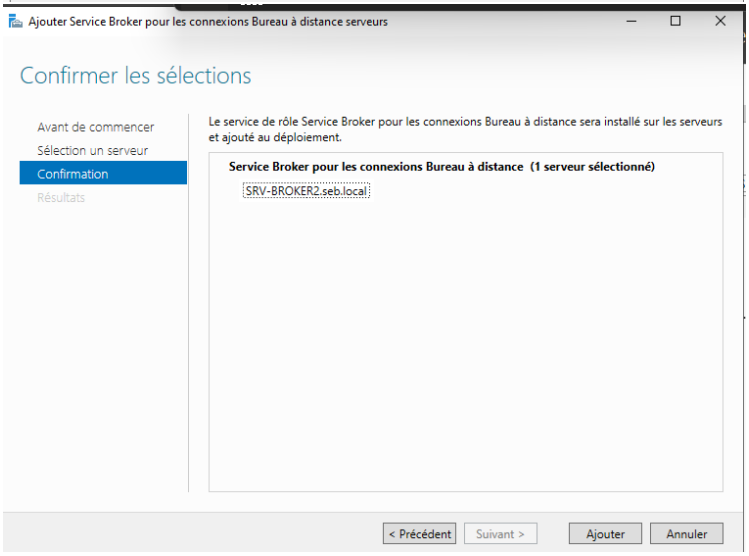
- Avec deux Brokers actifs, tester la connexion via le nom DNS cluster.
- Arrêter l'un des Brokers et vérifier que la connexion et la prise en charge des sessions par l'autre Broker continuent sans interruption.
- Surveiller les journaux d'événements et la base SQL pour vérifier l'activité et la répartition.

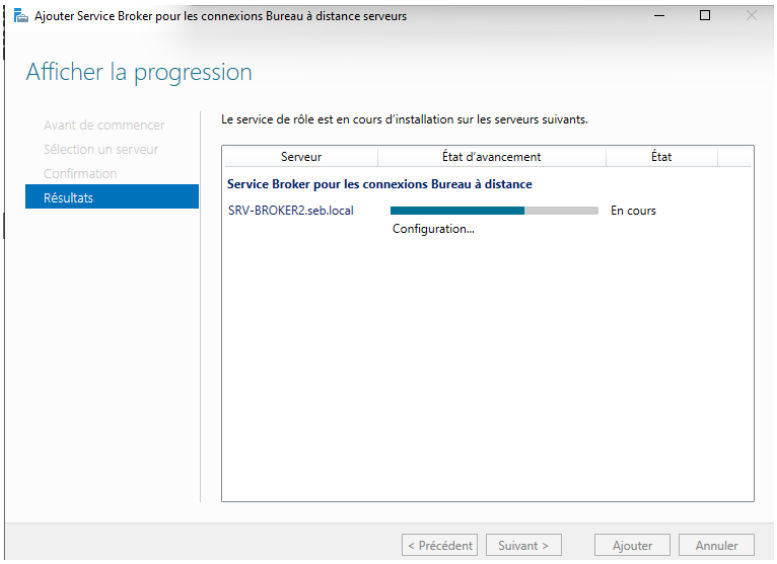
Création de la VM SRV-BROKER2 en 192.168.1.110
Commissionner le SRV-BROKER2 dans le domaine seb.local
Ajout du broker2 dans le groupe utilisateur de l'AD
Ajout de l'hôte dans le DNS en RDSHA 192.168.1.110
Installation du rôle Service Bureau à distance puis du service de rôle Broker
Ajout des Serveurs à gérer : RDS1, RDS2, Broker1 et 2
Ajout du client SQL : msodbcsql
Ajout du Broker 2 dans le service broker pour les connexions bureau à distance

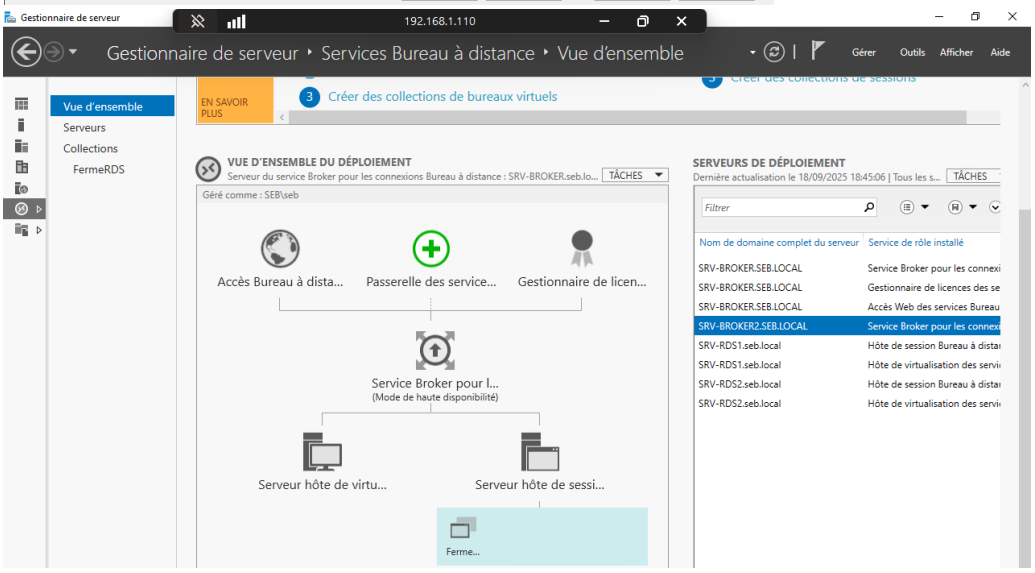
- Utilise le Gestionnaire de serveur **> Services Bureau à distance > Vue d'ensemble**, fais un clic droit sur "Service Broker pour les connexions Bureau à distance", puis choisis "**Ajouter le serveur du service Broker pour les connexions Bureau à distance**" et sélectionne SRV-BROKER2.seb.local dans l'assistant.
- L'assistant va intégrer SRV-BROKER2 dans le pool HA existant (même base SQL, même adresse DNS cluster)


Gestionnaire de serveur • Services Bureau à distance • Vue d'ensemble
Vue d'ensemble
Serveurs
Collections
FermeRDS
PRISE EN MAIN DES SERVICES BUREAU À DISTANCE
DÉMARRAGE RAPIDE
EN SAVOIR PLUS
1 Configurer un déploiement pour les services Bureau à distance
Déploiement de bureaux basés sur un ordinateur virtuel
2 Ajouter des serveurs hôtes de virtualisation des services Bureau
3 Créer des collections de bureaux virtuels
Déploiement de bureaux basés sur une session
2 Ajouter des serveurs hôtes de session Bureau à distance
3 Créer des collections de sessions
VUE D'ENSEMBLE DU DÉPLOIEMENT
SERVEURS DE DÉPLOIEMENT
Accès Bureau à dista...
Passerelle des service...
Gestionnaire de licen...
Configurer la haute disponibilité
Ajouter un serveur du service Broker pour les connexions Bureau à distance
Supprimer le serveur du service Broker pour les connexions Bureau à distance
Serveur hôte de virtu...
Serveur hôte de sessi...
Ferme...

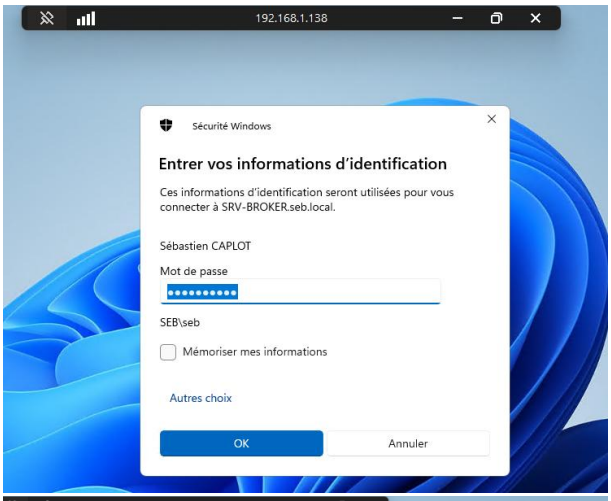

192.168.1.110
Ajouter Service Broker pour les connexions Bureau à distance serveurs
Sélectionner un serveur
Avant de commencer
Sélection un serveur
Confirmation
Résultats
Cet Assistant vous permet d'ajouter Service Broker pour les connexions Bureau à distance serveurs au déploiement. Sélectionnez les serveurs sur lesquels installer le rôle de service Service Broker pour les connexions Bureau à distance.
Pool de serveurs
Filtre :
Nom
Adresse IP
Système d'exploitation
SRV-BROKER2.seb.local
192.168.1.110
SRV-RDS1.seb.local
192.168.1.103
SRV-RDS2.seb.local
192.168.1.107
3 ordinateur(s) trouvé(s)
Sélectionné
Ordinateur
SEB.LOCAL (1)
SRV-BROKER2
1 ordinateur(s) sélectionné(s)
Les informations d'identification du compte SEB\seb seront utilisées pour ajouter les serveurs.
< Précédent
Suivant >
Ajouter
Annuler

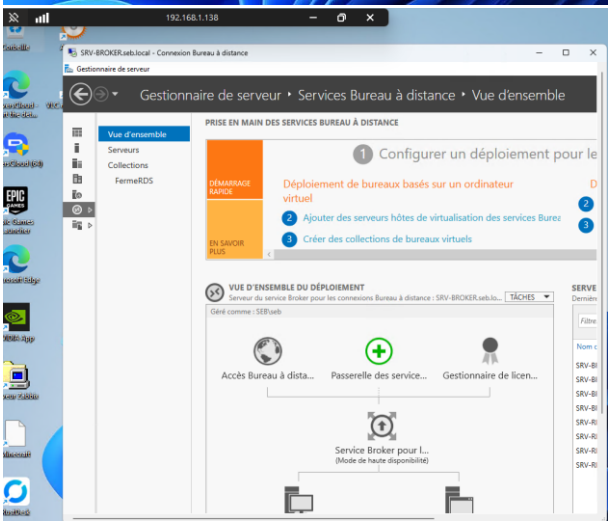

Ajouter Service Broker pour les connexions Bureau à distance serveurs
Confirmer les sélections
Avant de commencer
Sélection un serveur
Confirmation
Résultats
Le service de rôle Service Broker pour les connexions Bureau à distance sera installé sur les serveurs et ajouté au déploiement.
Service Broker pour les connexions Bureau à distance (1 serveur sélectionné)
SRV-BROKER2.seb.local
< Précédent
Suivant >
Ajouter
Annuler


Ajouter Service Broker pour les connexions Bureau à distance serveurs
Afficher la progression
Avant de commencer
Sélection un serveur
Confirmation
Résultats
Le service de rôle est en cours d'installation sur les serveurs suivants.
Serveur
État d'avancement
État
Service Broker pour les connexions Bureau à distance
SRV-BROKER2.seb.local
En cours
Configuration...
< Précédent
Suivant >
Ajouter
Annuler


Gestionnaire de serveur
192.168.1.110
Gestionnaire de serveur ‣ Services Bureau à distance ‣ Vue d'ensemble
Gérer
Outils
Afficher
Aide
Vue d'ensemble
Serveurs
Collections
FermeRDS
EN SAVOIR PLUS
3 Créer des collections de bureaux virtuels
VUE D'ENSEMBLE DU DÉPLOIEMENT
TÂCHES
Géré comme : SEB\seb
Accès Bureau à dista...
Passerelle des service...
Gestionnaire de licen...
Service Broker pour l...
(Mode de haute disponibilité)
Serveur hôte de virtu...
Serveur hôte de sessi...
Ferme...
SERVEURS DE DÉPLOIEMENT
Filtrer
Nom de domaine complet du serveur
Service de rôle installé
SRV-BROKER.SEB.LOCAL
SRV-BROKER.SEB.LOCAL
SRV-BROKER.SEB.LOCAL
SRV-BROKER2.SEB.LOCAL
SRV-RDS1.seb.local
SRV-RDS1.seb.local
SRV-RDS2.seb.local
SRV-RDS2.seb.local

Test de connexion RDP sur : SRV-BROKER.seb.local à partir d'un pc relié au domaine seb.local





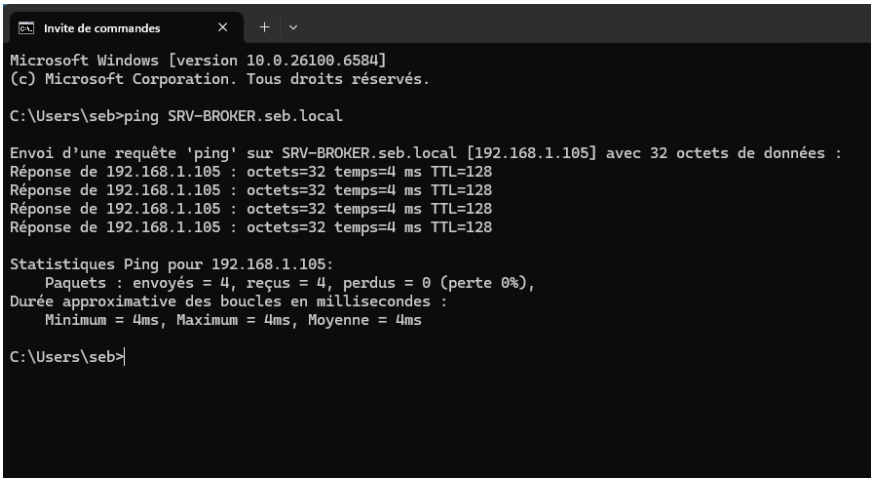
Points à vérifier :

- **Ping et résolution** : Sur le client ou serveur test, exécute :

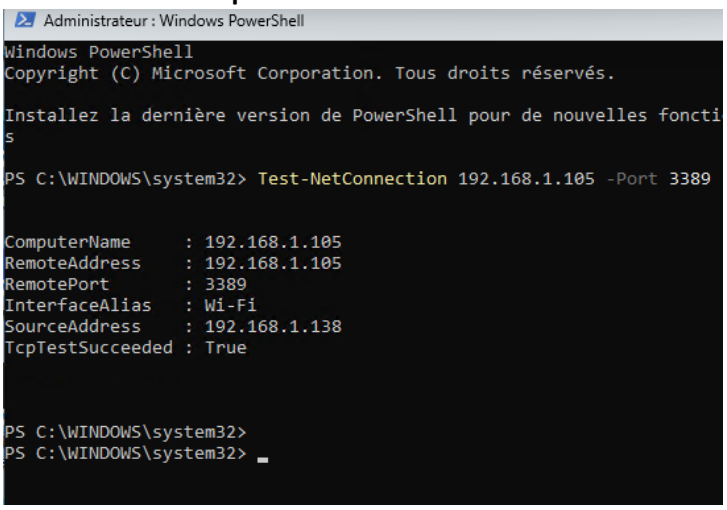
text

ping SRV-BROKER.seb.local

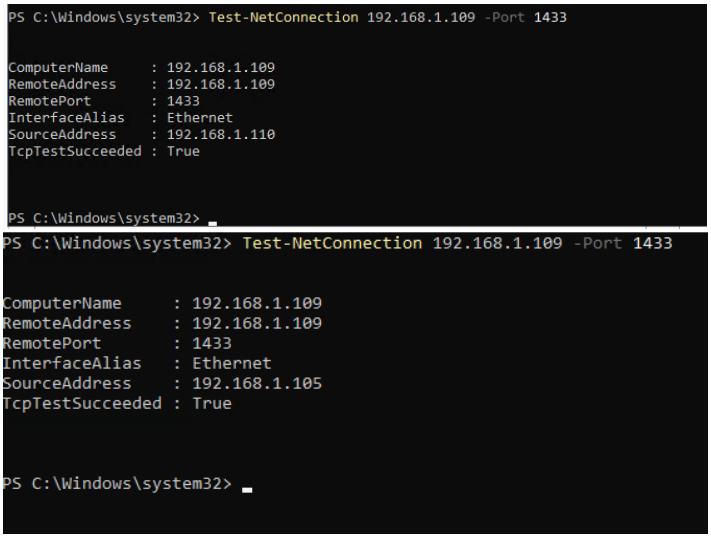
Si la résolution retourne bien 192.168.1.105, le DNS fonctionne.



Test **Ouverture du port RDP sur l'hôte 192.168.1.105** :



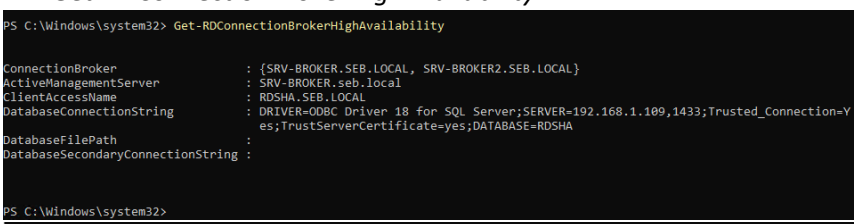
- Tester la connectivité SQL depuis chaque broker vers la base RDSHA :

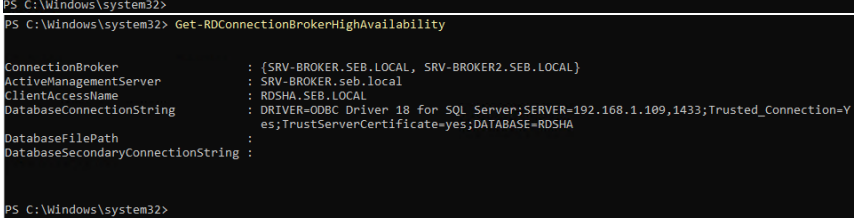


Commandes pour vérifier la synchronisation et la configuration RDS Broker HA

1. Sur un des brokers, lancer la commande PowerShell suivante pour lister l'état des serveurs broker dans la ferme HA :

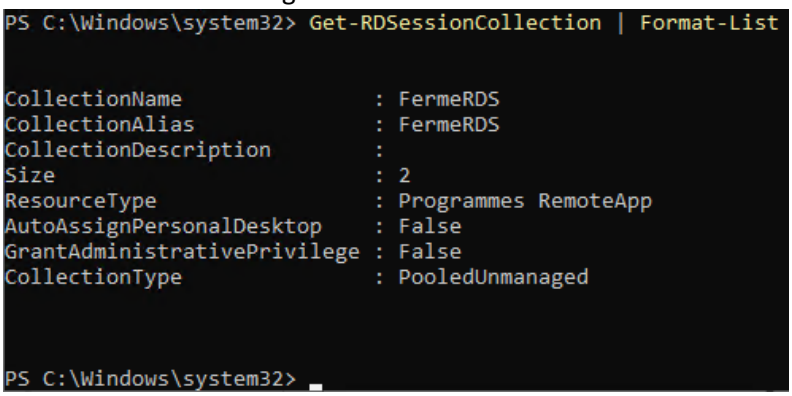
Get-RDConnectionBrokerHighAvailability

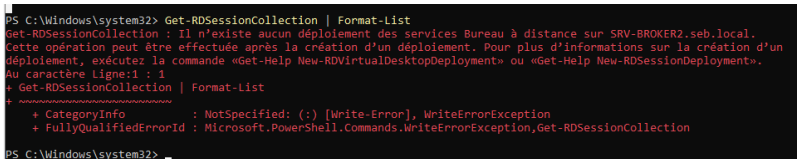




Elle retourne la liste des brokers en HA avec leur statut.

2. Vérifier la configuration des collections RDS et les brokers associés :





C'est normal en environnement RDS haute disponibilité : la commande PowerShell Get-RDSessionCollection ne retourne de résultats que depuis **le broker actif de la ferme** ou via l'adresse DNS du cluster HA, car la configuration, les collections et les déploiements sont pilotés à travers le point d'entrée logique (le cluster Broker), et non via l'accès direct à un broker secondaire.

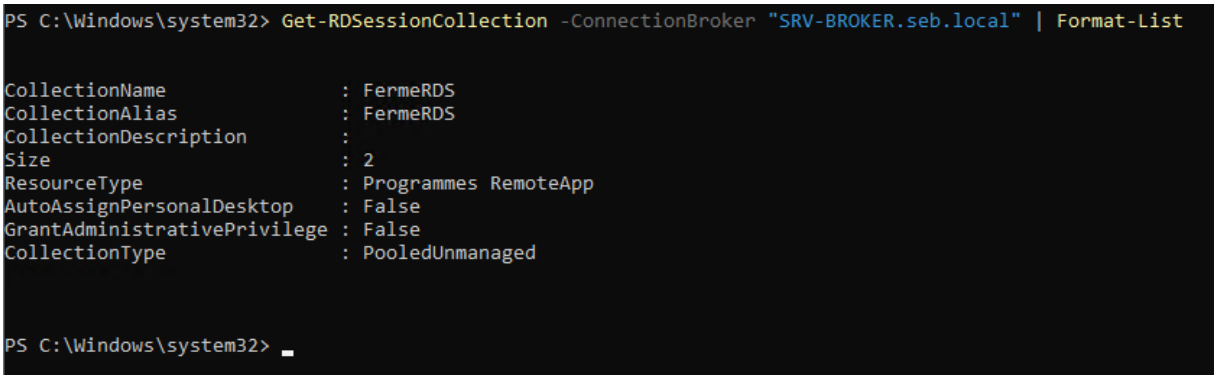
Points à retenir :

- **Administration et consultation des collections** se font via le cluster HA, pas par chaque broker indépendamment.
- L'erreur obtenue ne remet pas en cause la fonctionnalité HA : elle indique simplement qu'il ne faut pas piloter isolément chaque broker pour les opérations de ferme ou de collections.

Pour lister les collections et valider la HA :

- Exécute la commande PowerShell **depuis le broker principal/actif ou via une session PowerShell pointée vers le DNS cluster** :

Get-RDSessionCollection -ConnectionBroker "SRV-BROKER.seb.local" | Format-List



Les captures montrent que tout fonctionne désormais correctement :

- Tu es connecté avec succès via RDP à « SRV-BROKER.seb.local », en utilisant la haute disponibilité du broker, sur l'IP 192.168.1.105.
- Le gestionnaire de serveur RDS indique bien le broker en **mode de haute disponibilité**.
- Le test PowerShell Test-NetConnection 192.168.1.105 -Port 3389 affiche "TcpTestSucceeded : True", donc le port Bureau à distance est ouvert et accessible.

Conclusion

Ce laboratoire a permis de mettre en œuvre un service Broker de connexions Bureau à distance en haute disponibilité au sein d'une infrastructure Active Directory virtuelle, déployée sur un serveur Fujitsu TX1330 M2 avec plusieurs machines virtuelles. Cette expérimentation a confirmé la continuité, la résilience et l'efficacité du déploiement RDS en haute disponibilité.

Le projet a inclus notamment :

- La configuration d'une base SQL centralisée, utilisée par les brokers pour externaliser les informations de session.
- La création d'un cluster DNS autorisant un point d'entrée unique via l'adresse SRV-BROKER.seb.local.
- La préparation et la vérification des droits Active Directory, via la création du groupe « Grp_RDS_Brk » et son assignation à tous les brokers virtualisés.
- L'installation et la configuration des pilotes ODBC nécessaires à la connexion SQL sur chaque broker.
- L'intégration et la synchronisation des brokers secondaires dans la ferme RDS en haute disponibilité, pilotée via le Gestionnaire de Serveur et PowerShell.
- La résolution des problématiques classiques liées à la réplication AD, aux permissions RPC, et à la configuration DNS, assurant un fonctionnement stable et fiable.

La validation a reposé sur des tests exhaustifs de bout en bout, depuis la connectivité réseau entre les brokers et la base SQL jusqu'à l'accessibilité RDP via le DNS cluster. Tous les indicateurs démontrent la robustesse de la solution, offrant un accès transparent et tolérant aux pannes pour les bureaux virtuels publiés.

Ce laboratoire constitue une base solide pour envisager des évolutions vers un équilibrage de charge plus avancé, la gestion centralisée des certificats SSL et l'implémentation d'audits renforcés pour la supervision et la sécurité. Il démontre la faisabilité et la pertinence d'un RDS en haute disponibilité, même dans un environnement virtuel de test.

Cette maîtrise technique dans un contexte laboratoire sur serveur Fujitsu TX1330 M2 prépare sereinement le passage à des environnements d'entreprise plus complexes, avec un cadre de gestion centralisée et un service Bureau à distance hautement disponible et scalable.