

## Basculement du Log Shipping

Ce document indique brièvement les étapes de restauration d'urgence d'un secondaire de log shipping sur Microsoft SQL Server lorsque le principal est défaillant.

Le code SQL de ce document peut être copié depuis un gist github à cette adresse :  
<https://gist.github.com/rudi-bruchez/6db250f4ac9dadbad1b13ed3dcb151b>  
(<https://goo.gl/uyVJCU>).

### *Les principes*

Le log shipping est une solution de reprise sur incident basée sur un ou plusieurs serveurs de standby qui récupère de manière planifiée des sauvegardes de journal de transactions planifiés sur une base vivante, nommée la base principale.

Il s'agit d'une solution assez simple et basique, qui n'offre pas le même niveau de sophistication que des solutions de haute disponibilité modernes, mais qui permet de disposer à peu de frais d'un serveur de secours en cas de défaillance du serveur principal.

En cas de défaillance du serveur principal, vous devez agir pour rendre le serveur secondaire disponible. Le log shipping n'offre aucune fonction de basculement automatique. Vous devez également rediriger manuellement les connexions clientes.

### *Le basculement manuel*

Si votre base principale est indisponible, vous devez agir manuellement. La base secondaire est soit en état de récupération, soit en état de veille. Vous devez agir sur le secondaire, de la façon suivante :

1. vous assurer que toutes les sauvegardes de journal de transaction sont restaurées sur le secondaire ;
2. rendre le secondaire disponible ;
3. rediriger les connexions clientes.

Ces étapes sont détaillées ci-dessous.

### *Vérifier l'état de la restauration sur le serveur secondaire*

**Sur le serveur SQL secondaire :**

Obtenez d'abord les métadonnées du log shipping sur votre serveur SQL secondaire, à l'aide de la requête suivante :

```
SELECT sd.secondary_database,
```

```

s.primary_server,
s.primary_database,
s.backup_source_directory,
s.backup_destination_directory,
sd.last_restored_file,
sd.last_restored_date,
s.last_copied_file,
s.last_copied_date
FROM msdb.dbo.log_shipping_secondary s
JOIN msdb.dbo.log_shipping_secondary_databases sd
ON s.secondary_id = sd.secondary_id;

```

Cette requête retourne les métadonnées de toutes les bases de données secondaires inscrites sur le serveur. Voici un exemple de résultat :

secondary_database	primary_server	primary_database	backup_source_directory	backup_destination_directory
Secondaire	SQL2012	Principal	\\SQL2012\logshipping	C:\sqldata\MSSQL11.MSSQLSERVER\MSSQL\DATA\second...
last_copied_file	last_copied_date	last_restored_file	last_restored_date	
C:\sqldata\MSSQL11.MSSQLSERVER\MSSQL\DATA\second...	2016-04-02 23:27:24.863	C:\sqldata\MSSQL11.MSSQLSERVER\MSSQL\DATA\second...	2016-04-02 23:27:50.283	

Les colonnes retournées sont les suivantes :

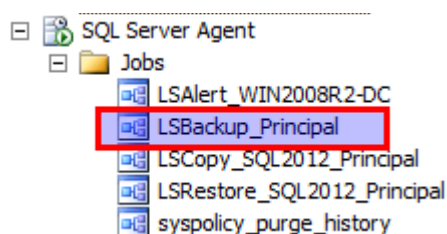
Colonne	Signification
secondary_database	Le nom de la base de données secondaire sur le serveur SQL secondaire. C'est cette base de données que vous voulez activer.
primary_server	Le nom du serveur SQL principal, c'est ce serveur qui est défaillant, si vous voulez activer le secondaire.
primary_database	Le nom de la base de données principale sur le serveur SQL principal. C'est cette base de données qui est défaillante, si vous voulez activer le secondaire.
backup_source_directory	Le partage réseau sur lequel le secondaire va chercher les sauvegardes de journal de transactions effectuées par le serveur SQL principal. Cet emplacement contient peut-être des sauvegardes qui n'ont pas encore été copiées sur votre serveur SQL secondaire. Il faut donc les récupérer. Par contre, si ce partage se situe physiquement sur le serveur principal, le partage n'est peut-être plus disponible.
backup_destination_directory	Le répertoire local au serveur SQL secondaire sur lequel les sauvegardes de journal de transactions sont copiées depuis le partage (backup_source_directory) avant d'être restaurées sur la base de données secondaire (secondary_database).
last_copied_file	Le nom et le chemin du dernier fichier copié dans backup_destination_directory.
last_copied_date	La date de copie du dernier fichier copié dans backup_destination_directory.
last_restored_file	Le nom et le chemin du dernier fichier restauré dans la base de données secondaire.
last_restored_date	La date de la dernière restauration dans la base de données

secondaire.

Le nom des fichiers de sauvegarde de transactions est basé sur l'exemple suivant :

**Principal\_201604022115**10.trn, où **Principal** est le nom de la base de données principale, **20160402** est la date de la sauvegarde au format ISO 8601 sans tirets, c'est-à-dire AAAAMMJJ et **2115** est l'heure GMT (UTC) en HHMM. Notez qu'il s'agit bien de l'heure GMT, il vous faut donc faire la conversion avec l'heure locale pour en déduire l'heure de la sauvegarde. Plus simplement, vous pouvez vous baser sur l'heure indiquée dans `last_copied_date` ou regarder l'heure du fichier dans le gestionnaire de fichiers Windows.

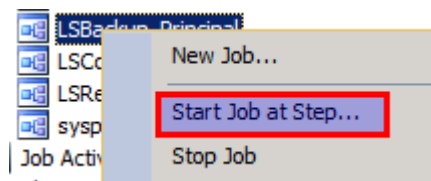
### *Si vous avez encore accès au serveur SQL principal*



Si vous avez encore accès au serveur SQL principal vous pouvez lancer le travail de sauvegarde pour récupérer les dernières transactions de votre journal. Le travail dans l'agent SQL sur serveur SQL principal est surligné dans la copie d'écran suivante :

Le nommage est simple : **LSBackup\_Principal** : **LS** pour Log Shipping, **Backup** pour l'action du travail et **Principal** est le nom de la base de données principale.

Pour lancer le travail, faites un clic droit sur le travail et choisissez « démarrez la travail à l'étape... » ou « start job at step... », comme surligné sur la copie d'écran suivante.



Vous pouvez vérifier les informations du log shipping sur le serveur principal à l'aide de la requête suivante :

```
SELECT
    primary_database,
    backup_directory,
    backup_share,
    last_backup_file,
    last_backup_date
FROM msdb.dbo.log_shipping_primary_databases;
```

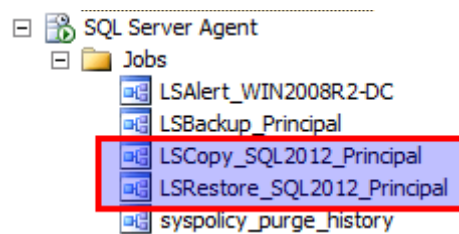
Les colonnes retournées sont les suivantes :

Colonne	Signification
primary_database	Le nom de la base de données principale sur le serveur SQL principal.
backup_directory	Le répertoire local au serveur SQL principal sur lequel les sauvegardes de journal de transactions sont déposées, si le partage

	du log shipping est situé sur le serveur principal.
backup_share	Le partage réseau sur lequel le secondaire vient chercher les sauvegardes de journal de transactions.
last_backup_file	Le nom et le chemin du dernier fichier de sauvegarde de journal de transactions pour le log shipping.
last_backup_date	La date de la dernière sauvegarde de journal de transactions pour le log shipping.

## Récupérer les dernières sauvegardes de journal de transactions

Vous savez maintenant quel est le dernier fichier de sauvegarde de journal de transactions restauré sur votre base secondaire. Vous devez savoir si c'est bien la dernière sauvegarde disponible. Si votre serveur principal est inaccessible, regardez simplement dans le répertoire backup\_destination\_directory si vous avez des sauvegardes plus récentes. Plus simplement ; vous pouvez lancer manuellement les travaux de l'agent SQL sur le serveur SQL secondaire qui copie et restaurent les sauvegardes.



La copie d'écran ci-dessus montre les deux travaux de l'agent. Le nommage est simple : **LSCopy\_SQL2012\_Principal** : **LS** pour Log Shipping, **Copy** ou **Restore** pour l'action du travail, **SQL2012** correspond au serveur SQL principal et **Principal** est la base de données principale qui est copiée sur le serveur secondaire.

Lancez d'abord le travail de copie. Lorsqu'il est terminé, même s'il y a une erreur parce qu'il ne parvient pas à atteindre le partage sur le principal, lancez le travail de restauration.

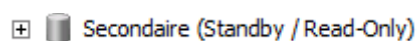
Pour lancer les travaux, faites un clic droit sur chaque travail et choisissez « démarrez la travail à l'étape... » ou « start job at step... »

Vérifiez ensuite à l'aide de la requête de métadonnées indiqué plus haut que vous avez bien restauré la dernière sauvegarde.

Vous pouvez désactiver les travaux de copie et de restauration, ils tomberont en erreur de toute manière lorsque vous effectuerez l'étape suivante pour rendre la base secondaire disponible.

## Rendre la base de données secondaire disponible

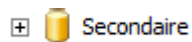
Lorsque vous avez la certitude que toutes les sauvegardes de journal sont restaurées sur le serveur SQL secondaire, vous devez rendre la base de données secondaire disponible. Celle-ci est toujours en mode de récupération ou en mode veille, comme dans la copie d'écran suivante :



Ouvrez une fenêtre de requête et lancez la requête suivante. **Attention, remplacez bien le nom « Secondaire » par le nom de votre base de données secondaire.**

```
USE master;  
RESTORE DATABASE Secondaire WITH RECOVERY;
```

Rafraîchissez ensuite l'affichage de l'explorateur d'objets de SQL Server Management Studio. Vous devez maintenant voir la base de données disponible, comme ceci :



### ***Redirigez le connexions clientes***

Vous devez maintenant vous assurez que les clients accèdent au nouveau serveur. Pour ce faire, vous pouvez modifier la chaîne de connexion sur votre serveur applicatif ou serveur web si vous avez une application web ou middleware. Vous pouvez aussi modifier l'adresse DNS de l'ancien serveur pour qu'elle pointe sur le nouveau serveur.

Attention à une chose, si votre base de données comporte des utilisateurs liés à des connexions SQL (login SQL), vous devez avoir préalablement créé les connexions sur le serveur secondaire. Vous devez ensuite vous assurer que le lien est établi à l'aide du SID entre le login et l'utilisateur. Vous pouvez le faire avec un code comme ceci :

```
USE MaBase ;  
ALTER USER MonUtilisateur WITH login = MonLogin;
```