BASIC CONCEPTS & IMPLEMENTATION ON DATABASE MIRRORING

Prepared by: SAIFUL

Prepared for: Knowledge sharing with the DataSoft AML Team.

Date: 10.01.2021

Contents

DATABASE MIRRORING BASIC CONCEPTS

What is Database Mirroring

Database Mirroring is one of the High Availability and Disaster Recovery solution,

which is used to move the database transactions between two copies of a single database that are stored on different server instances of SQL Server Database Engine, while first server instance provides database to the clients and second Server instance act as a standby server that takes over the place of first server instance (either manually or automatically by the help of another third server) in case of any accident occurs.

Database Mirroring Server Roles

Principal Server	**Mirror Server**	**Witness Server (optional)**
The principal server hosts the active copy of the database (referred to as the principal database) and services client requests. The principal server forwards all transactions to the mirror server before it applies them in the principal database.	The mirror server hosts a copy of the principal database (referred to as the mirror database) and applies the transactions forwarded by the principal database to keep the mirror database synchronized with the principal database	The witness server is an optional component of a database mirroring solution. When present, a witness server monitors the principal and mirror servers to ensure continued connectivity and participation in the mirror session (referred to as quorum). If either server loses quorum, the witness server assigns the principal server role, causing automatic failover from the principal server to the mirror server if necessary. A witness server is required for automatic failover; however, one witness server can support several mirror sessions because it is not an intensive job.

Database Mirroring Operating Modes

Database Mirroring has three different Modes

Operating Mode	**Transfer Mechanism**	**Transaction Safety**	**Witness Server**	**Quorum Required**	**Failover Type**	**Operating Mode in SSMS**	**Description**
High Performance	Asynchronous	OFF	N/A	N/A	Forced	High performance (asynchronous)	The same as High-Protection Mode, there is also a need of two servers (principal and mirror server). In this Mode the transaction safety level is set to OFF that results in "Data transfer mechanism between the principal and mirror servers is asynchronous", which means that the principal server does not wait for an acknowledgement from the mirror server that all transaction log records have been recorded on the mirror server and the client application gets confirmation that a transaction is committed as soon as the principal server has written the transaction to the log. If the Principal server in this mode becomes unavailable then we must manually perform the failover since there is no witness server in this mode. Because the transaction safety level is set to OFF, we might lose some transactions in the event of a failover.
High Protection	Synchronous	FULL	N	Y	Manual	High safety without automatic failover (synchronous)	This mode is nearly the same as High-Availability mode, but the difference is that there is a need for two servers only (Principal server and Mirror server). The transaction safety level in this mode is also set to FULL that results in the same High-Availability Mode "Database transfer mechanism between the principal and mirror server is synchronous". Another difference is if the principal server becomes unavailable in this mode then we need to manually perform the failover because there is no witness server in this mode. Because the transaction safety level is set to FULL, we do not lose any committed transactions in the event of a failover.
High Availability	Synchronous	FULL	Y	Y	Automatic /Manual	High safety with automatic failover (synchronous)	In this Mode we need all the three servers, since the transaction safety level is set to FULL that results in the "Database transfer mechanism between the principal and mirror server is synchronous" which means that the principal server waits for an acknowledgement from the mirror server that the transaction log record has been recorded on the mirror server. Then, the client application gets confirmation that the transaction is committed. But if the principal server becomes unavailable then the witness server and the mirror server will form a quorum and perform automatic failover.

Database Mirroring Features

- A **database** is either a principal or **mirrored**, but not both. However, a **SQL Server instance** can be a principal **server** for **one** set **of databases**, and a **mirror** for **another** set.
- It is a mixture of replication and log shipping.
- **Data Transfer:** Individual T-Log records are transferred using TCP endpoints.
- **Server Limitation:**It is one to one. i.e. One principal server to one mirror server.
- **DB Access:**Mirrored DB can only be accessed using snapshot DB.
- **Recovery Model:**Mirroring supports only Full Recovery model.
- **Restoring State:**The restore can be completed using with NORECOVERY.
- **Backup/Restore:**User make backup & Restore manually.
- **Monitor/Distributor/ Witness:**Principal server can't act as both principal server and witness server.
- **Types Of Servers:**All servers should be SQL Server.
- **SQL Server Agent Dependency/Jobs:**:Independent on SQL Server agent.
- **Using With Other Features Or Components:**Database mirroring can be used with Log shipping, Database snapshots , Replication.
- **DDL Operations:**DDL changes are applied automatically.
- **Database Limit:**generally good to have 10 DB's for one server.
- **Latency:**There will not be data transfer latency.
- **Committed /Uncommitted Transactions:**Only committed transactions are transferred to the mirror database.
- **Primary key:**Not required.
- **Individual Articles:**No. Whole database must be selected.
- **FILESTREAM:**Mirroring does not support FILESTREAM.
- **DB Name:**It must be the same name.
- **DB Availability:** In Recovery state, no user can make any operation. You can take snapshot.
- **Warm/ Hot Standby Solution** :When a database mirroring session is synchronized, database mirroring provides a hot standby server that supports rapid failover without a loss of data from committed transactions. When the session is not synchronized, the mirror server is typically available as a warm standby server (with possible data loss).
- **System Data Transferred:** Yes.
- **System Databases:** cannot mirror the Master, msdb, tempdb, or model databases.

DATABASE MIRRORING IMPLEMENTATION

Two types of transport security exist for database mirroring:

- Windows Authentication
- Certificate-based Authentication

There is a limitation with windows account as it is required same user account with same password for all the server included in mirroring, so there is a dependency on windows account. If one of windows account/ password is changed then mirroring will stop working until the user of all other server is not synchronized with same user account and password.

The cleaner solution is certificate based authentication which is independent to windows account and where is not any domain account.

So, we will discuss on mirroring with certificate based authentication here.

Configuring Mirror with certificate-based authentication

Summary to configure mirroring with certificate-based authentication

Step 1. Checking prerequisites

Step 2. Backup and Restore database.

a) Backup database with backup type full as .bak.

b) Backup database with backup type Transaction Log as .trn.

c) restore .bak file database with Recovery State option RESTORE WITH NORECOVERY.

d) restore .trn file database with Recovery State option RESTORE WITH NORECOVERY

Step 3. Configuring Outbound Connections.

Follow these steps on each partner and witness (if includes) server instance :

a) In the master database, **create** a database **master key**.

b) In the master database, **create** an encrypted **certificate** on the server instance.

c) **Create** an **endpoint** for the server instance using its certificate.

d) **Backup** the **certificate** to a file and securely copy it to the other system or systems.

Step 4. Configuring Inbound Connections.

Follow these steps for each partner server and witness (if includes) instance:

In the master database:

a) **Create** a **login** for the other system.

b) **Create** a **user** for that login.

c) **Obtain** the **certificate** for the mirroring endpoint of the other server instance.

d) **Associate** the **certificate** with the user created in step 2.

e) Grant CONNECT permission on the login for that mirroring endpoint.

Step 6. Configuring Mirror on Database

In the master database of Mirror Server Instance:

a) Set Principal Partner.

In the master database of Principal Server Instance:

b) Set Mirror Partner.

c) Set Witness.

Step7. Testing with failover.

Details step by step actions to configure mirroring with certificate-based authentication

Following server, port and database have been used for this tutorial:

Principal Server: 192.168.2.237

Mirror Server: 192.168.200.31

Witness Server: 192.168.11.87

Database: VelocityAMLWorking_2021

Port : 5022 [any port can be used if it is not using by any other service]

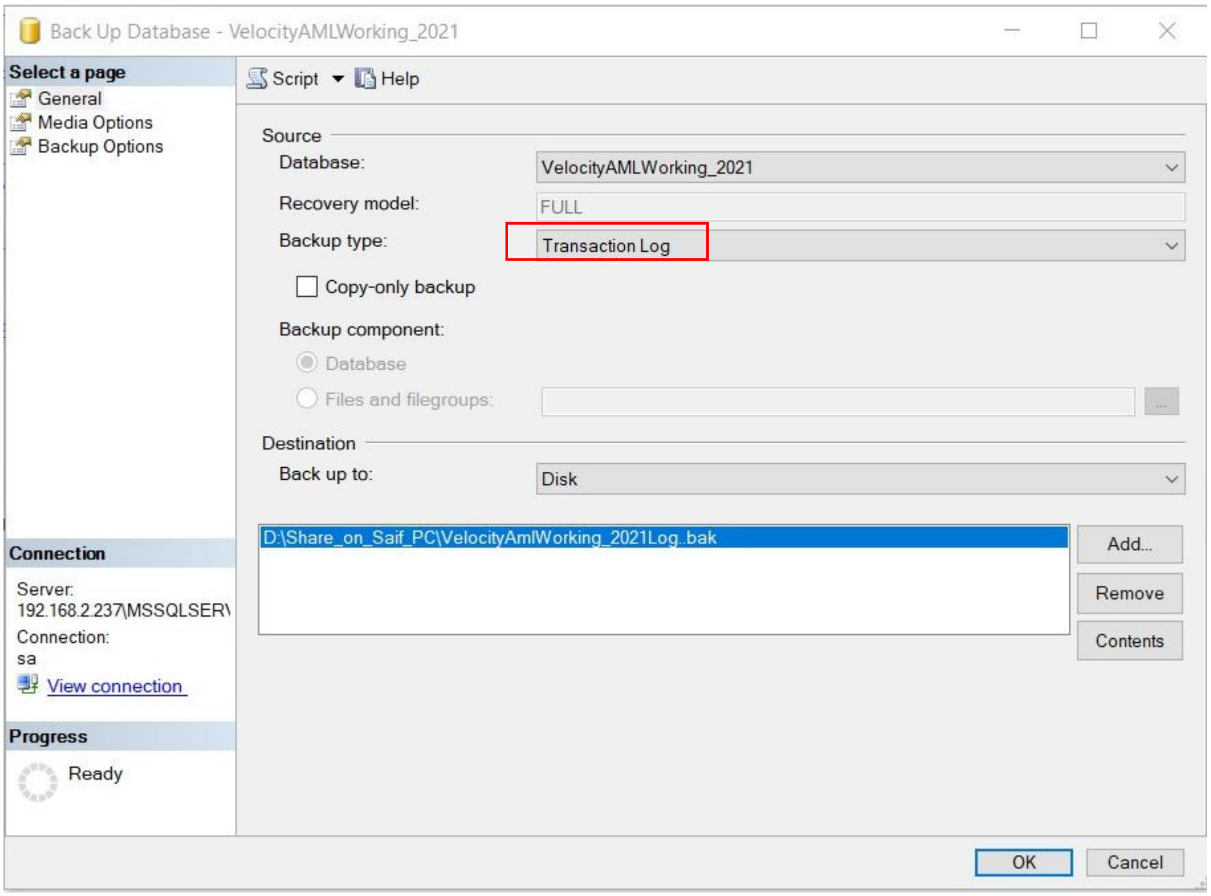
Checking Prerequisites:

1. Ping all server each other included in mirroring. All server each other should response.
2. The port which is used in the mirroring should not be used by any other service and should be opened.

Backup and Restore:

1. By selecting principal database VelocityAMLWorking_2021 --> task --> backup, take full database backup (as VelocityAMLWorking_2021.bak) from principal server for which mirror will be created.

2. As well as, by selecting principal database VelocityAMLWorking_2021 --> --> task --> backup, take transaction log backup (as VelocityAMLWorking_2021TLog.trn) from principal server for which mirror will be created.



Back Up Database - VelocityAMLWorking_2021

Select a page
- General
- Media Options
- Backup Options

Script ▾ | Help

Source
- Database: VelocityAMLWorking_2021
- Recovery model: FULL
- Backup type: Transaction Log
- [ ] Copy-only backup
- Backup component:
  - Database
  - Files and filegroups:

Destination
- Back up to: Disk
- D:\Share_on_Saif_PC\VelocityAmlWorking_2021Log.bak

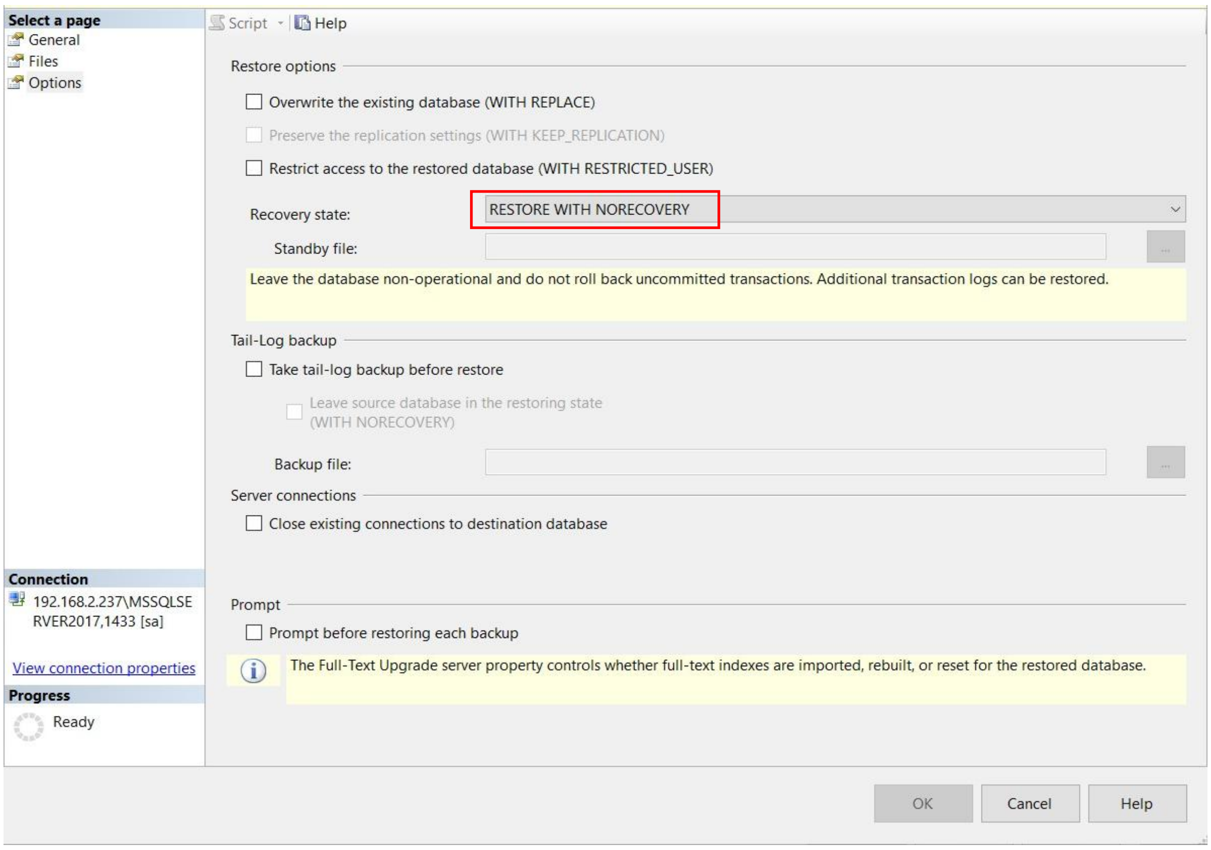
Add... | Remove | Contents

Connection
Server:
192.168.2.237\MSSQLSER\
Connection:
sa
View connection

Progress
Ready

OK | Cancel

3. By selecting mirror instance's Databases-->Restore Database, restore backup (which is taken by step 2) to Mirror Server with RESTOREWITHNORECOVERY option.



Select a page
- General
- Files
- Options

Script ▾ | Help

Restore options
- [ ] Overwrite the existing database (WITH REPLACE)
- [ ] Preserve the replication settings (WITH KEEP_REPLICATION)
- [ ] Restrict access to the restored database (WITH RESTRICTED_USER)

Recovery state: RESTORE WITH NORECOVERY

Standby file:

Leave the database non-operational and do not roll back uncommitted transactions. Additional transaction logs can be restored.

Tail-Log backup
- [ ] Take tail-log backup before restore
  - [ ] Leave source database in the restoring state (WITH NORECOVERY)

Backup file:

Server connections
- [ ] Close existing connections to destination database

Prompt
- [ ] Prompt before restoring each backup

The Full-Text Upgrade server property controls whether full-text indexes are imported, rebuilt, or reset for the restored database.

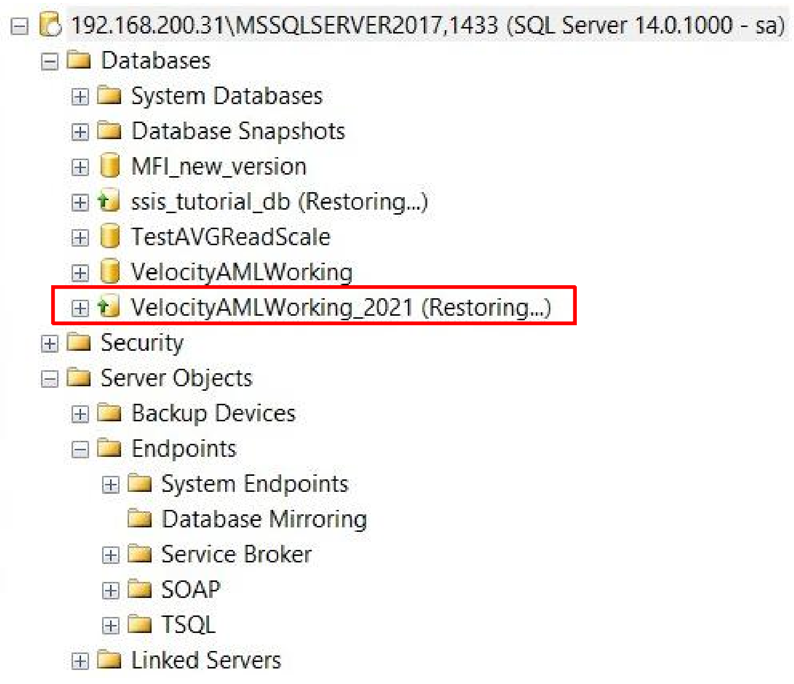
Connection
192.168.2.237\MSSQLSERVER2017,1433 [sa]

View connection properties

Progress
Ready

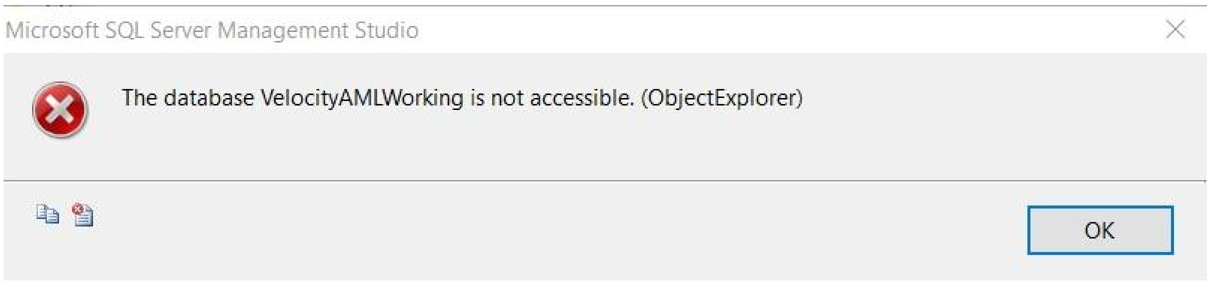
OK | Cancel | Help

VelocityAMLWorking_2021 database will be restored in mirror server as restoring mode as following screen.



```
192.168.200.31\MSSQLSERVER2017,1433 (SQL Server 14.0.1000 - sa)
  Databases
    System Databases
    Database Snapshots
    MFI_new_version
    ssis_tutorial_db (Restoring...)
    TestAVGReadScale
    VelocityAMLWorking
    VelocityAMLWorking_2021 (Restoring...)
  Security
  Server Objects
    Backup Devices
    Endpoints
      System Endpoints
      Database Mirroring
      Service Broker
      SOAP
      TSQL
    Linked Servers
```

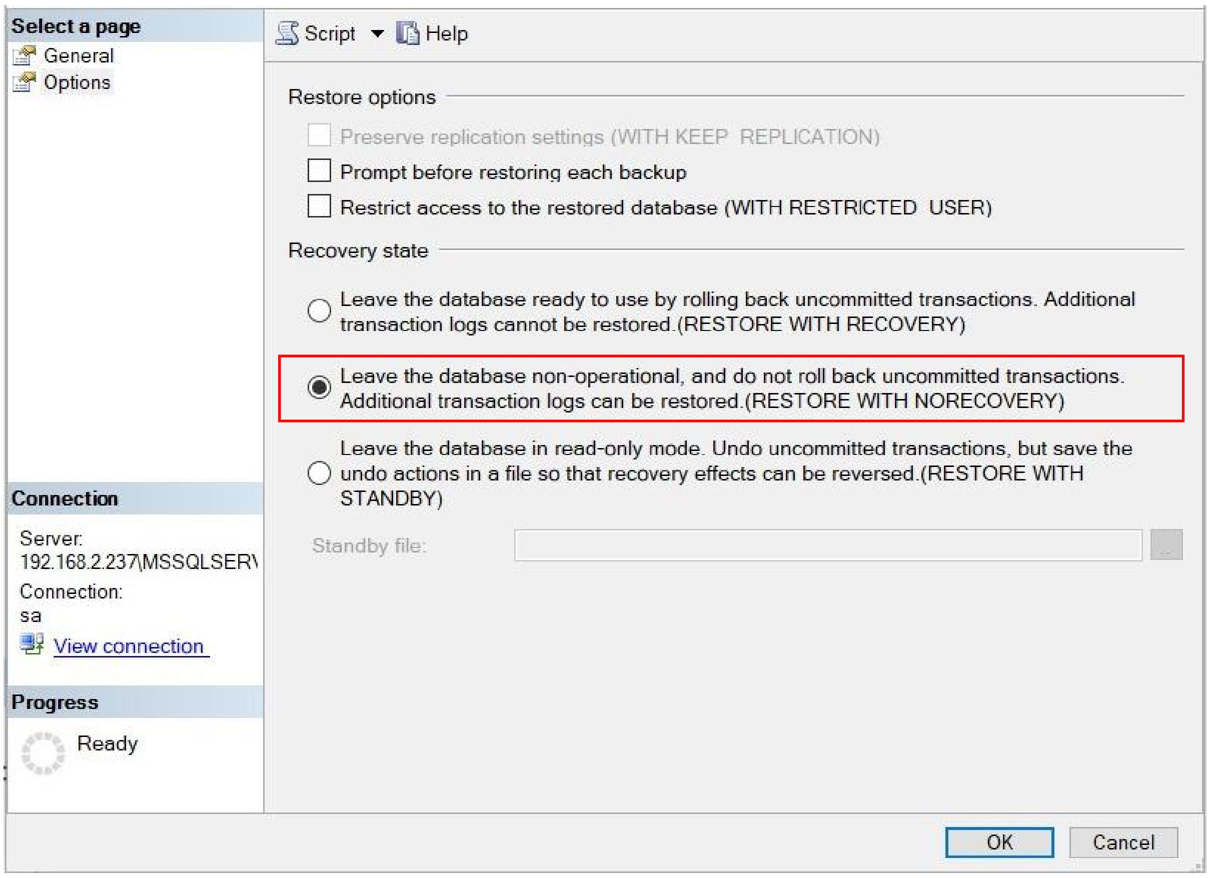
Which is not accessible to anyone.



Microsoft SQL Server Management Studio

The database VelocityAMLWorking is not accessible. (ObjectExplorer)

4. Restore transaction log backup with RESTORE WITH NORECOVERY option, by selecting VelocityAMLWorking_2021(Restoring…)-->Task-->Restore-->Transaction Log…



Restore options

[ ] Preserve replication settings (WITH KEEP REPLICATION)

[ ] Prompt before restoring each backup

[ ] Restrict access to the restored database (WITH RESTRICTED USER)

Recovery state

○ Leave the database ready to use by rolling back uncommitted transactions. Additional transaction logs cannot be restored.(RESTORE WITH RECOVERY)

● Leave the database non-operational, and do not roll back uncommitted transactions. Additional transaction logs can be restored.(RESTORE WITH NORECOVERY)

○ Leave the database in read-only mode. Undo uncommitted transactions, but save the undo actions in a file so that recovery effects can be reversed.(RESTORE WITH STANDBY)

Standby file:

Start Mirroring:

Execute following command on Principal, Mirror, Witness (if includes) server as written and as serial wise.

```
--------------------------------------------------------------------------------------------------------
/* Principal Server Instance */
--------------------------------------------------------------------------------------------------------
----Step 1: (a) Create Master Key----
USE master
GO
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'aml@123';
GO

----Step 1: (b) Create Certificate----
CREATE CERTIFICATE PrincipalServerCertificate WITH SUBJECT = 'Principal Server Certificate for Mirror',
START_DATE = '20210101', EXPIRY_DATE = '20211231'
GO

----Step 1: (c) Create Endpoint----
CREATE ENDPOINT MirroringEndpoint
STATE = STARTED AS TCP(LISTENER_PORT = 5022, LISTENER_IP = ALL)
FOR DATABASE_MIRRORING (AUTHENTICATION = CERTIFICATE PrincipalServerCertificate,
ENCRYPTION = REQUIRED ALGORITHM RC4, ROLE = ALL);
GO

----Step 1: (d) Backup Certificate----
BACKUP CERTIFICATE PrincipalServerCertificate TO FILE = 'D:\PrincipalServerCertificate.cer'
--------------------------------------------------------------------------------------------------------
```

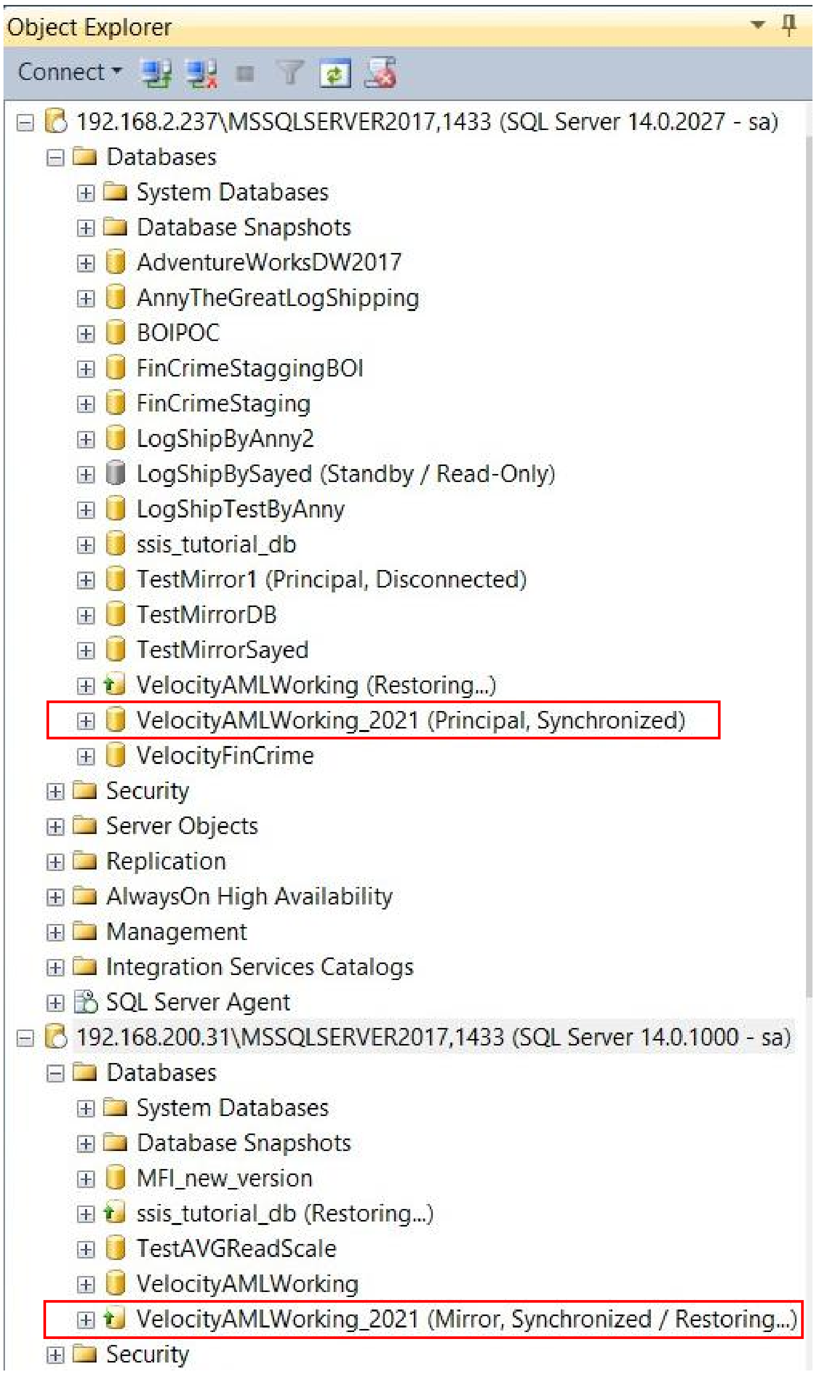
```
--------------------------------------------------------------------------------------------------------
/* Mirror Server Instance */
--------------------------------------------------------------------------------------------------------
----Step 2: (a) Create Master Key----
USE master
GO
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'aml@123';
GO
----Step 2: (b) Create Certificate----
CREATE CERTIFICATE MirrorServerCertificate WITH SUBJECT = 'Mirror Server Certificate for Mirror',
START_DATE = '20210101', EXPIRY_DATE = '20211231'
GO
----Step 2: (c) Create Endpoint----
CREATE ENDPOINT MirroringEndpoint
STATE = STARTED AS TCP(LISTENER_PORT = 5022, LISTENER_IP = ALL)
FOR DATABASE_MIRRORING (AUTHENTICATION = CERTIFICATE MirrorServerCertificate,
ENCRYPTION = REQUIRED ALGORITHM RC4, ROLE = ALL);
GO
----Step 2: (d) Backup Certificate----
BACKUP CERTIFICATE MirrorServerCertificate TO FILE = 'D:\MirrorServerCertificate.cer'
--------------------------------------------------------------------------------------------------------
```

```sql
----[If Witness Server Includes]

-----------------------------------------------------------------------------------------------------------
/* Witness Server Instance */
-----------------------------------------------------------------------------------------------------------
----Step 3: (a) Create Master Key----
USE master
GO
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'aml@123'
GO
----Step 3: (b) Create Certificate----
CREATE CERTIFICATE WitnessServerCertificate WITH SUBJECT = 'Witness Server Certificate for Mirror',
START_DATE = '20210101', EXPIRY_DATE = '20211231'
GO
----Step 3: (c) Create Endpoint----
CREATE ENDPOINT MirroringEndpoint
STATE = STARTED AS TCP(LISTENER_PORT = 5022, LISTENER_IP = ALL)
FOR DATABASE_MIRRORING (AUTHENTICATION = CERTIFICATE WitnessServerCertificate,
ENCRYPTION = REQUIRED ALGORITHM RC4, ROLE = ALL);
GO
----Step 3: (d) Backup Certificate----
BACKUP CERTIFICATE WitnessServerCertificate TO FILE = 'D:\WitnessServerCertificate.cer'
-----------------------------------------------------------------------------------------------------------


-----------------------------------------------------------------------------------------------------------
/* Principal Server Instance */
-----------------------------------------------------------------------------------------------------------
----Step 4: (a) Create Login----
USE master
GO
CREATE LOGIN MirrorLogin WITH PASSWORD = 'aml@123'
GO
----Step 4: (b) Create User----
CREATE USER MirrorUser FOR LOGIN MirrorLogin
GO
----Step 4: (c) Obtain Certificate----
----Copy the MirrorServerCertificate.cer certificate file (step 2(d) ) from Mirror Server to Principal server
----[If Witness Server Includes]
----Copy the WitnessServerCertificate.cer certificate file (step 3(d) ) from Witness Server to Principal server

----Step 4: (d) Assign Certificate to User ----
CREATE CERTIFICATE MirrorServerCertificate
AUTHORIZATION MirrorUser FROM FILE = 'D:\MirrorServerCertificate.cer'
GO
----[If Witness Server Includes]
CREATE CERTIFICATE WitnessServerCertificate
AUTHORIZATION MirrorUser FROM FILE = 'D:\WitnessServerCertificate.cer'
GO
----Step 4: (e) Grant Connect Permission----
GRANT CONNECT ON ENDPOINT::MirroringEndpoint TO [MirrorLogin]
```

```
---------------------------------------------------------------------------------------------------------
/* Mirror Server Instance */
---------------------------------------------------------------------------------------------------------
----Step 5: (a) Create Login----
USE master
GO
CREATE LOGIN MirrorLogin WITH PASSWORD = 'aml@123';
GO
----Step 4: (b) Create User----
CREATE USER MirrorUser FOR LOGIN MirrorLogin
GO
----Step 5: (c) Obtain Certificate----
----Copy the PrincipalServerCertificate.cer certificate file (step 1(d) ) from Principal Server to Mirror server
----[If Witness Server Includes]
----Copy the WitnessServerCertificate.cer certificate file (step 3(d) ) from Witness Server to Mirror server
----Step 5: (d) Assign Certificate to User ----
CREATE CERTIFICATE PrincipalServerCertificate
AUTHORIZATION MirrorUser FROM FILE = 'D:\PrincipalServerCertificate.cer'
GO
----[If Witness Server Includes]
CREATE CERTIFICATE WitnessServerCertificate
AUTHORIZATION MirrorUser FROM FILE = 'D:\WitnessServerCertificate.cer'
GO
----Step 5: (e) Grant Connect Permission----
GRANT CONNECT ON ENDPOINT::MirroringEndpoint TO [MirrorLogin]
---------------------------------------------------------------------------------------------------------

----[If Witness Server Includes]
---------------------------------------------------------------------------------------------------------
/* Witness Server Instance */
---------------------------------------------------------------------------------------------------------
----Step 6: (a) Create Login----
USE master
GO
CREATE LOGIN MirrorLogin WITH PASSWORD = 'aml@123';
GO
----Step 6: (b) Create User----
CREATE USER MirrorUser FOR LOGIN MirrorLogin
GO
----Step 6: (c) Obtain Certificate----
----Copy the PrincipalServerCertificate.cer certificate file (step 1(d) ) from Principal Server to Witness server
----Copy the MirrorServerCertificate.cer certificate file (step 3(d) ) from Mirror Server to Witness server

----Step 6: (d) Assign Certificate to User ----
CREATE CERTIFICATE PrincipalServerCertificate
AUTHORIZATION MirrorUser FROM FILE = 'D:\PrincipalServerCertificate.cer'
GO
CREATE CERTIFICATE MirrorServerCertificate
AUTHORIZATION MirrorUser FROM FILE = 'D:\MirrorServerCertificate.cer'
GO
----Step 6: (e) Grant Connect Permission----
GRANT CONNECT ON ENDPOINT::MirroringEndpoint TO [MirrorLogin]
---------------------------------------------------------------------------------------------------------
```

```
------------------------------------------------------------------------------------------------------
/* Mirror Server Instance */
------------------------------------------------------------------------------------------------------
----Step 7: (a) To create mirroring set Principal Server as partner to Mirror Server----
USE master
GO
ALTER DATABASE VelocityAMLWorking_2021 SET PARTNER = 'TCP://192.168.2.237:5022'
------------------------------------------------------------------------------------------------------


------------------------------------------------------------------------------------------------------
/* Principal Server Instance */
------------------------------------------------------------------------------------------------------
----Step 7: (b) To create mirroring set Mirror Server as partner to Principal Server----
USE master
GO
ALTER DATABASE VelocityAMLWorking_2021 SET PARTNER = 'TCP://192.168.200.31:5022'
GO
----[If Witness Server Includes]
----Step 7: (c) To create mirroring set Witness Server as Witness to Principal Server----
ALTER DATABASE VelocityAMLWorking_2021 SET WITNESS = 'TCP://192.168.11.87:5022'
------------------------------------------------------------------------------------------------------
```

After executing all command successfully, refresh both instance, If mirroring successful then you will find VelocityAMLWorking_2021(Principal, Synchronized) in Principal Server (192.168.2.237) and VelocityAMLWorking_2021(Mirror, Synchronized/Restoring…) in Mirror Server (192.168.200.31) as following screen.

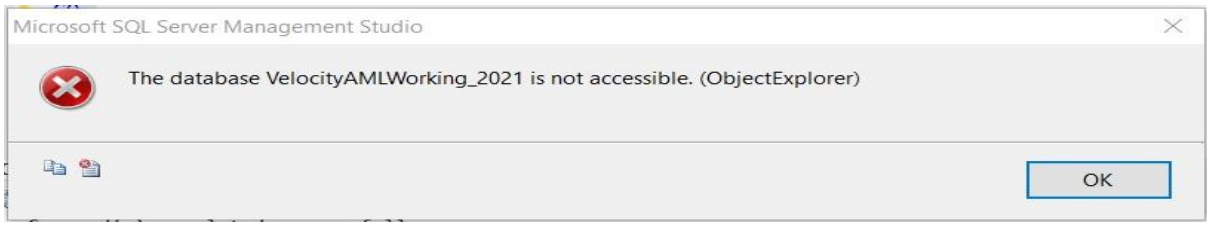


Object Explorer

Connect

```
192.168.2.237\MSSQLSERVER2017,1433 (SQL Server 14.0.2027 - sa)
  Databases
    System Databases
    Database Snapshots
    AdventureWorksDW2017
    AnnyTheGreatLogShipping
    BOIPOC
    FinCrimeStaggingBOI
    FinCrimeStaging
    LogShipByAnny2
    LogShipBySayed (Standby / Read-Only)
    LogShipTestByAnny
    ssis_tutorial_db
    TestMirror1 (Principal, Disconnected)
    TestMirrorDB
    TestMirrorSayed
    VelocityAMLWorking (Restoring...)
    VelocityAMLWorking_2021 (Principal, Synchronized)
    VelocityFinCrime
  Security
  Server Objects
  Replication
  AlwaysOn High Availability
  Management
  Integration Services Catalogs
  SQL Server Agent
192.168.200.31\MSSQLSERVER2017,1433 (SQL Server 14.0.1000 - sa)
  Databases
    System Databases
    Database Snapshots
    MFI_new_version
    ssis_tutorial_db (Restoring...)
    TestAVGReadScale
    VelocityAMLWorking
    VelocityAMLWorking_2021 (Mirror, Synchronized / Restoring...)
  Security
```

When you click on Mirror database then you will following screen will be displayed as mirror db is not accessible and you cannot do anything with mirror db.



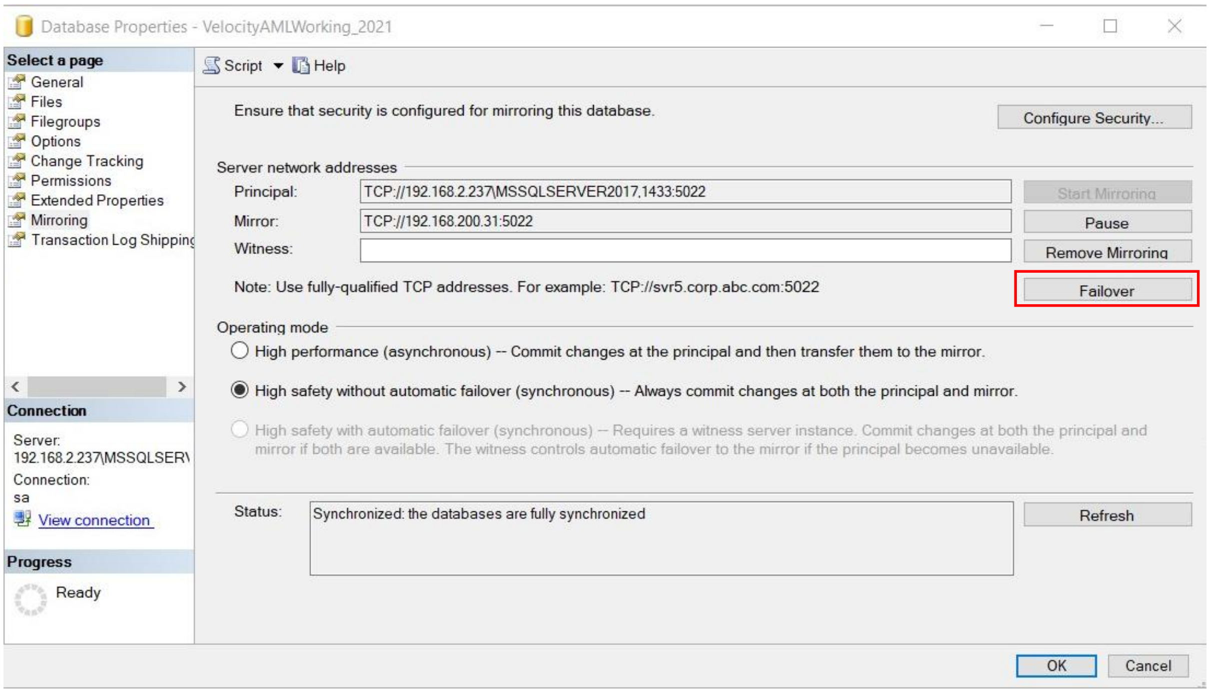
Microsoft SQL Server Management Studio

The database VelocityAMLWorking_2021 is not accessible. (ObjectExplorer)

OK

Test Mirror with Failover

Right click on principal database and click mirroring and then click Failover Button. It will make mirror database to principal.



Ensure that security is configured for mirroring this database.

Server network addresses

Principal: TCP://192.168.2.237\MSSQLSERVER2017,1433:5022

Mirror: TCP://192.168.200.31:5022

Witness:

Note: Use fully-qualified TCP addresses. For example: TCP://svr5.corp.abc.com:5022

Operating mode

( ) High performance (asynchronous) -- Commit changes at the principal and then transfer them to the mirror.

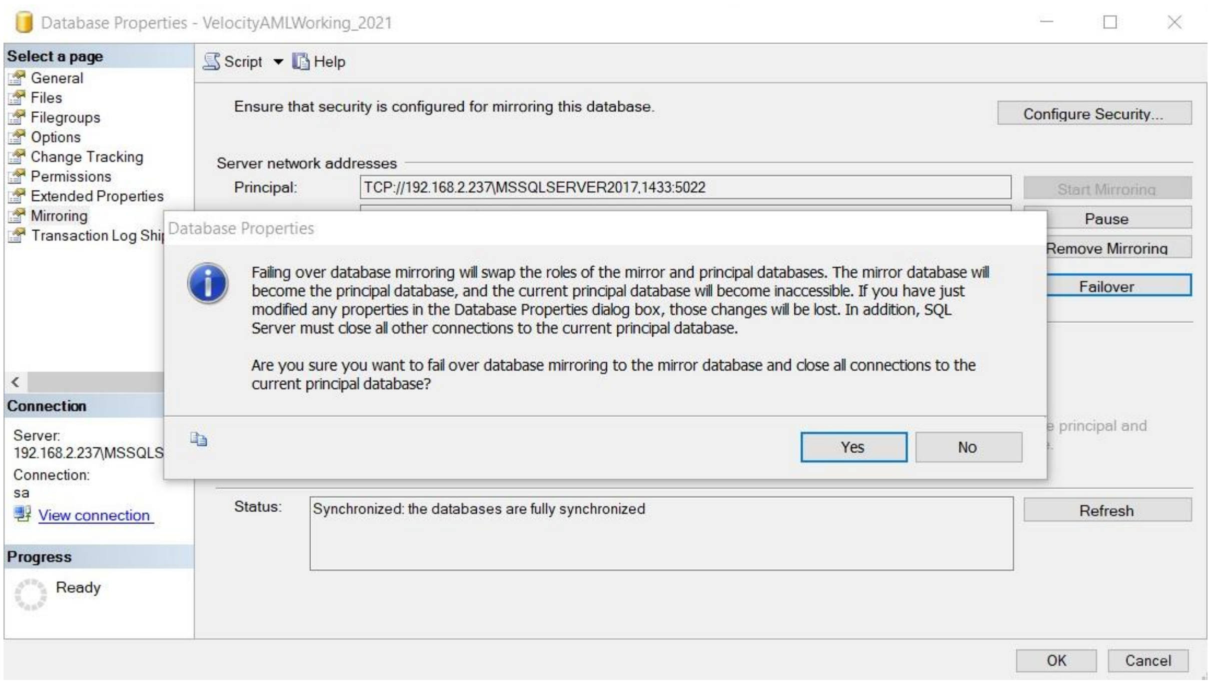
(●) High safety without automatic failover (synchronous) -- Always commit changes at both the principal and mirror.

( ) High safety with automatic failover (synchronous) -- Requires a witness server instance. Commit changes at both the principal and mirror if both are available. The witness controls automatic failover to the mirror if the principal becomes unavailable.

Status: Synchronized: the databases are fully synchronized

Server: 192.168.2.237\MSSQLSER\

Connection: sa



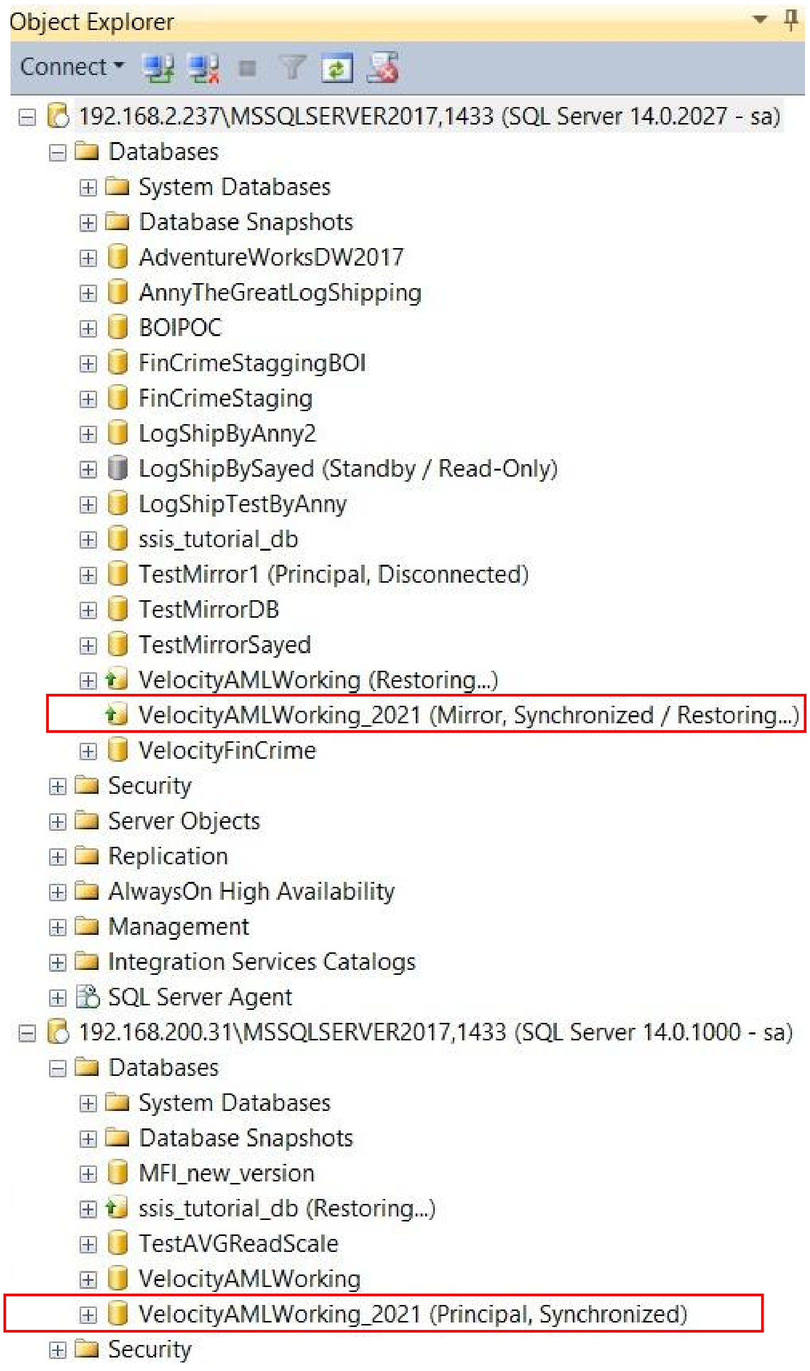
Database Properties

Failing over database mirroring will swap the roles of the mirror and principal databases. The mirror database will become the principal database, and the current principal database will become inaccessible. If you have just modified any properties in the Database Properties dialog box, those changes will be lost. In addition, SQL Server must close all other connections to the current principal database.

Are you sure you want to fail over database mirroring to the mirror database and close all connections to the current principal database?

Click on yes button.

Refresh both instance you will find VelocityAMLWorking_2021(Principal, Synchronized) in Mirror Server (192.168.200.31) and VelocityAMLWorking_2021(Mirror, Synchronized/Rsestoring…) in Principal Server (192.168.2.237) as following screen.

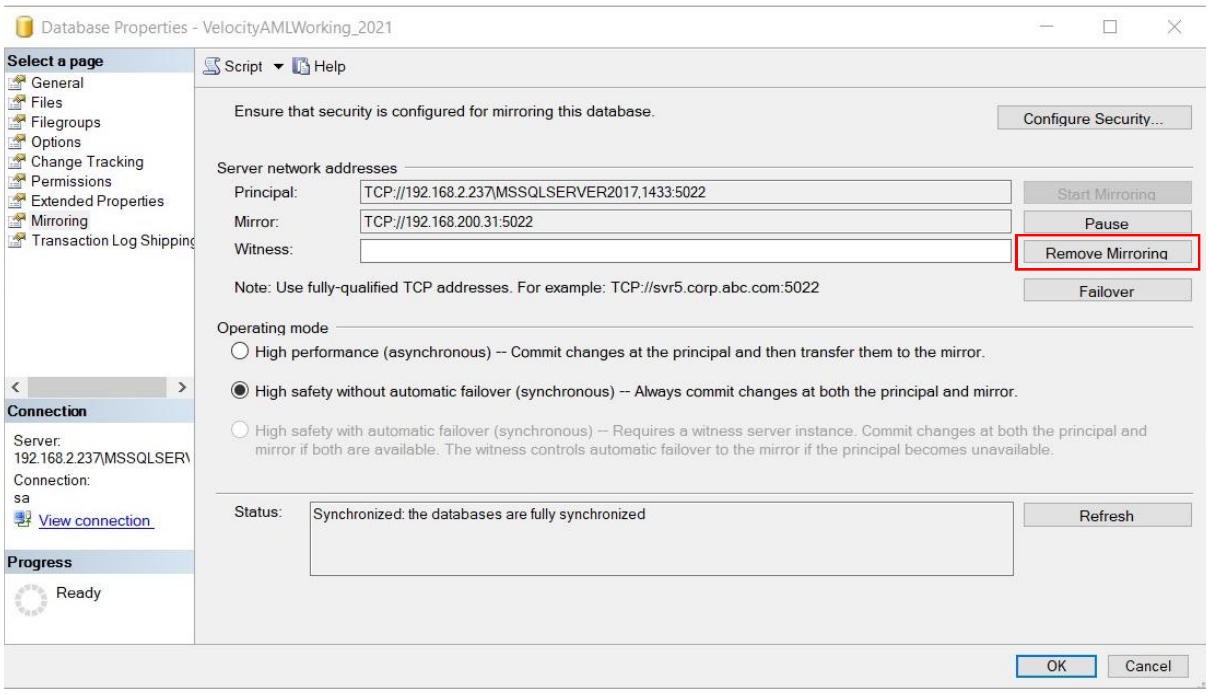


```
192.168.2.237\MSSQLSERVER2017,1433 (SQL Server 14.0.2027 - sa)
  Databases
    System Databases
    Database Snapshots
    AdventureWorksDW2017
    AnnyTheGreatLogShipping
    BOIPOC
    FinCrimeStaggingBOI
    FinCrimeStaging
    LogShipByAnny2
    LogShipBySayed (Standby / Read-Only)
    LogShipTestByAnny
    ssis_tutorial_db
    TestMirror1 (Principal, Disconnected)
    TestMirrorDB
    TestMirrorSayed
    VelocityAMLWorking (Restoring...)
    VelocityAMLWorking_2021 (Mirror, Synchronized / Restoring...)
    VelocityFinCrime
  Security
  Server Objects
  Replication
  AlwaysOn High Availability
  Management
  Integration Services Catalogs
  SQL Server Agent
192.168.200.31\MSSQLSERVER2017,1433 (SQL Server 14.0.1000 - sa)
  Databases
    System Databases
    Database Snapshots
    MFI_new_version
    ssis_tutorial_db (Restoring...)
    TestAVGReadScale
    VelocityAMLWorking
    VelocityAMLWorking_2021 (Principal, Synchronized)
  Security
```

To Failback do the same as above for mirror database and refresh both the instance.

Clean up Everything Created

Right click on principal database and click mirroring and then click Remove Mirroring Button



Ensure that security is configured for mirroring this database.

Server network addresses

Principal: TCP://192.168.2.237\MSSQLSERVER2017,1433:5022

Mirror: TCP://192.168.200.31:5022

Witness:

Note: Use fully-qualified TCP addresses. For example: TCP://svr5.corp.abc.com:5022

Operating mode

○ High performance (asynchronous) -- Commit changes at the principal and then transfer them to the mirror.

● High safety without automatic failover (synchronous) -- Always commit changes at both the principal and mirror.

○ High safety with automatic failover (synchronous) -- Requires a witness server instance. Commit changes at both the principal and mirror if both are available. The witness controls automatic failover to the mirror if the principal becomes unavailable.

Status: Synchronized: the databases are fully synchronized

Server:
192.168.2.237\MSSQLSER\

Connection:
sa

Then,
To clean up only mirroring run following command in each server
DROP ENDPOINT MirrorEndpoint

To clean up everything run following command in each server

```
DROP ENDPOINT MirrorEndpoint
DROP CERTIFICATE PrincipalServerCertificate
DROP CERTIFICATE MirrorServerCertificate
DROP CERTIFICATE WitnessServerCertificate
DROP MASTER KEY
DROP LOGIN MirrorLogin
DROP USER MirrorUser
```