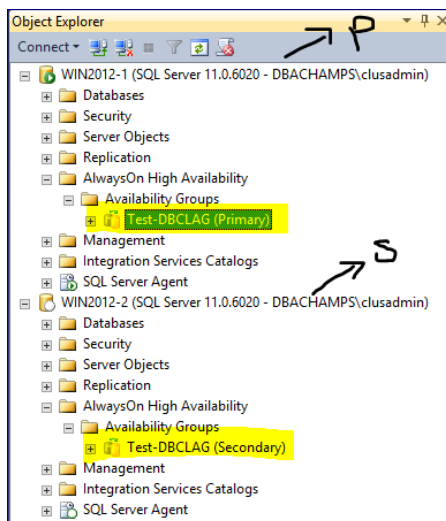


Steps to Enable TDE (Transparent Data Encryption) on Availability Group – SQL Server

- Transparent Data Encryption (TDE) encrypts SQL Database files by encrypting data at rest.
- In a situation where our physical media such as data, log and backup files get stolen, the malicious person can restore/attach the database and retrieve data.
- TDE protects this by not letting the database restored/attached without the associated certificate and master key.
- **Note: When enabling TDE, we need to make sure to backup the certificate and the key associated with the certificate.**
- **Without this certificate, we will never be able to restore/attach the database to a different server.**
- **The certificate should be available even if the TDE is disabled, part of the transaction log may remain still protected and the certificate may be required until a full database backup is taken.**
- Enabling TDE isn't as straightforward as it is for a database outside of an availability group.
- Databases that are in the availability group requires certain considerations and precautions to enable TDE which is explained step by step as follows:

1. Here we have a 2-node cluster and can see only 1 test database (DB Name: Test) on the PRIMARY REPLICA which is not part of the availability group yet. When enabling TDE on Always-On, we need to make sure that our database has been removed from the AG or else the database on the secondary nodes will change to SUSPECT MODE.



ON THE PRIMARY NODE – Verify that the primary node has a Database Master Key (DMK) in the master database.

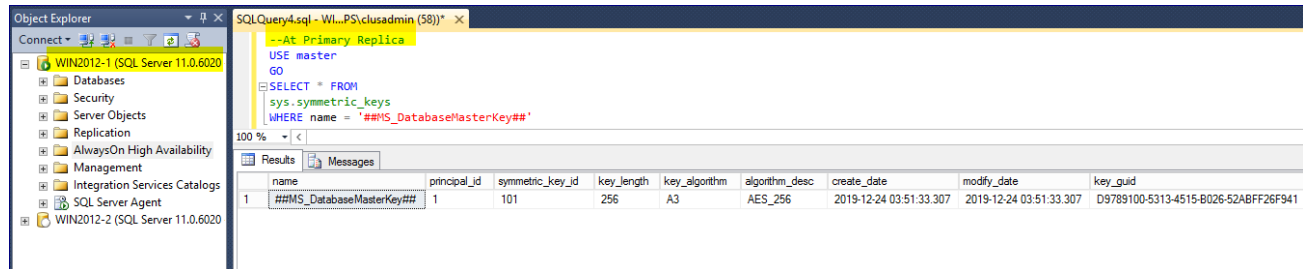
USE master

GO

SELECT * FROM

sys.symmetric_keys

WHERE name = '##MS_DatabaseMasterKey##'



2. Use Create Master Key if it doesn't exist (or) use Alter Master Key if it already exists and change the password if it is not known.

Note: The Service Master Key is the root of SQL Server's Encryption Hierarchy. There can only be 1 service master key per SQL Server instance. The service master key is used to protect (encrypt) other keys, mainly the database master keys. It cannot be used directly to encrypt data. And we can't create one our self.

USE master;

GO

CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'P@\$word1234';

GO

(Or) – Below is used to Alter the existing master key and provide the new password.

use master

go

alter master key add encryption by password = 'P@\$word1234'

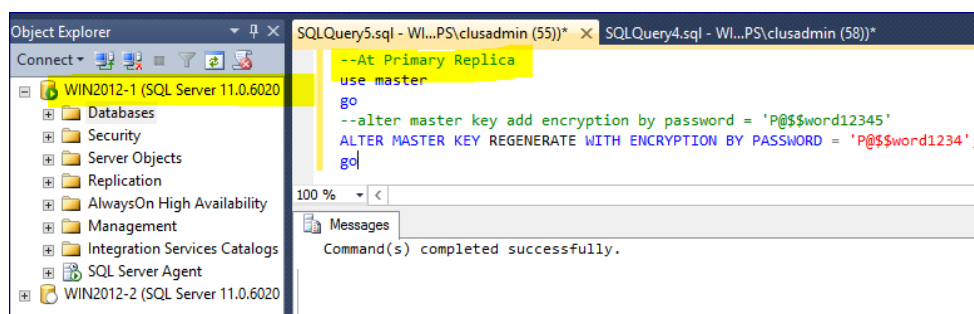
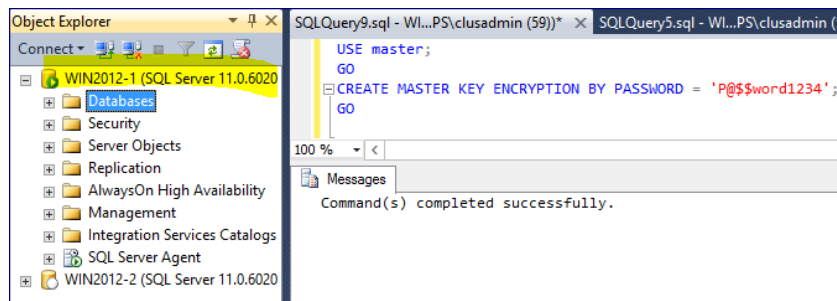
go

(Or) – Below is used to Overwrite (Re-Generate) the existing master key and provide the new password.

Use master

go

ALTER MASTER KEY REGENERATE WITH ENCRYPTION BY PASSWORD = 'P@\$word1234';



Note: Make sure to use a complex password and store it in a password vault to avoid any risk of compromise.

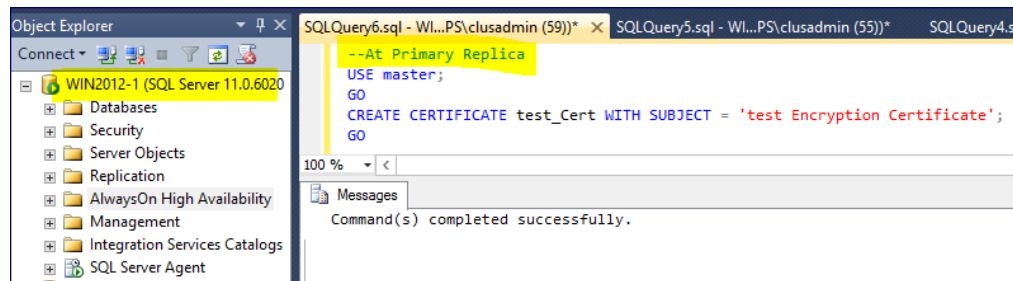
3. Create the Certificate for the test database

USE master;

GO

CREATE CERTIFICATE test_Cert WITH SUBJECT = 'test Encryption Certificate';

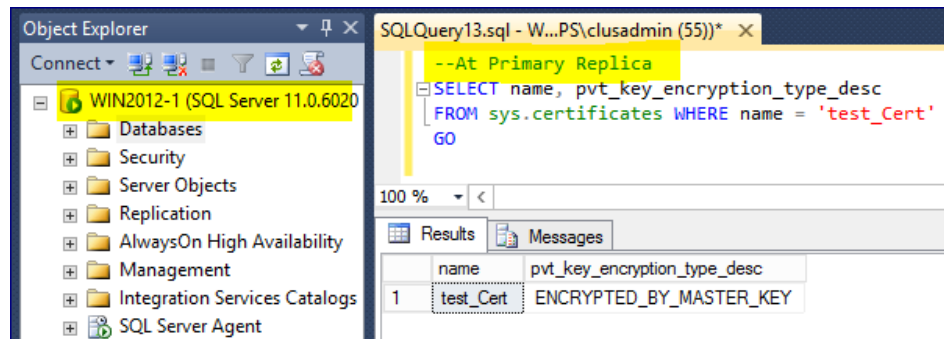
GO



4. Run the following script to check if the certificate was created

SELECT name, pvt_key_encryption_type_desc FROM sys.certificates WHERE name = 'test_Cert'

GO



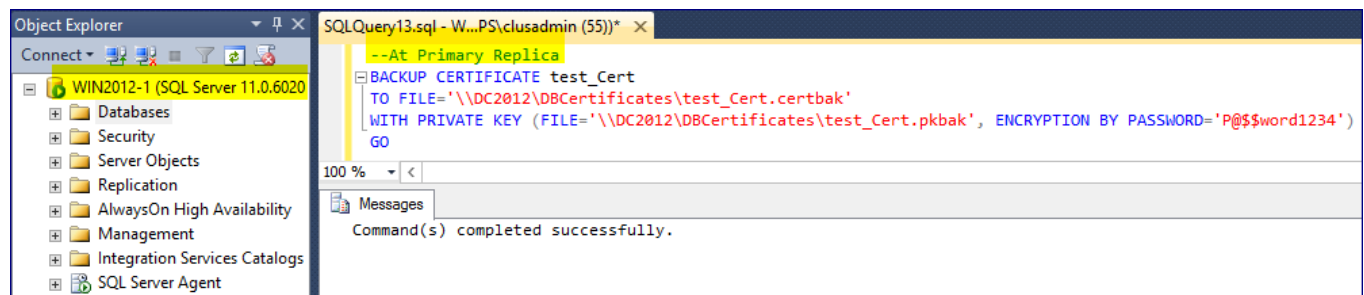
5. Backup the certificate on a shared location where all other nodes has access and keep it in a secure place

BACKUP CERTIFICATE test_Cert

TO FILE='\\DC2012\DBCertificates\test_Cert.cerbak'

WITH PRIVATE KEY (FILE='\\DC2012\DBCertificates\test_Cert.pkbak', ENCRYPTION BY PASSWORD='P@\$word1234')

GO



6. Create AES_256 encryption using the certificate on the required database to be added in AG.

USE test;

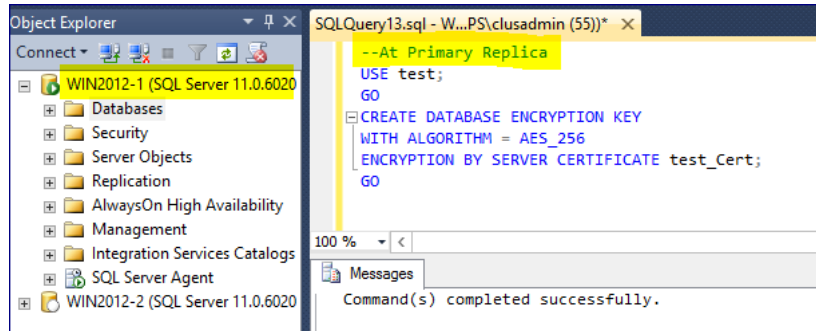
GO

CREATE DATABASE ENCRYPTION KEY

WITH ALGORITHM = AES_256

ENCRYPTION BY SERVER CERTIFICATE test_Cert;

GO



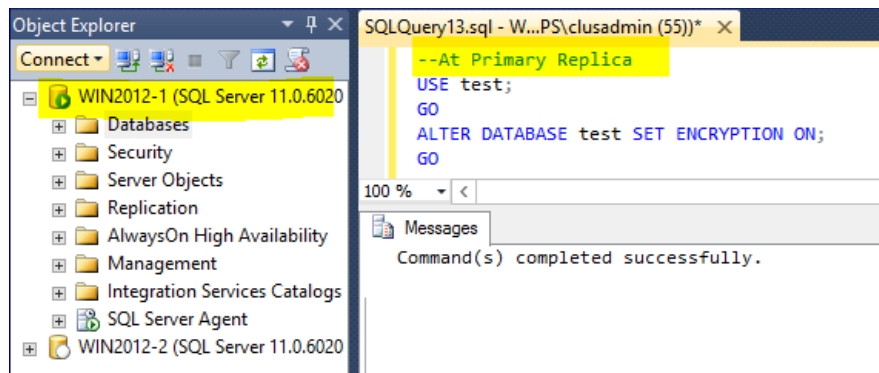
7. Enable the encryption on the required database to be added in AG.

USE test;

GO

ALTER DATABASE test SET ENCRYPTION ON;

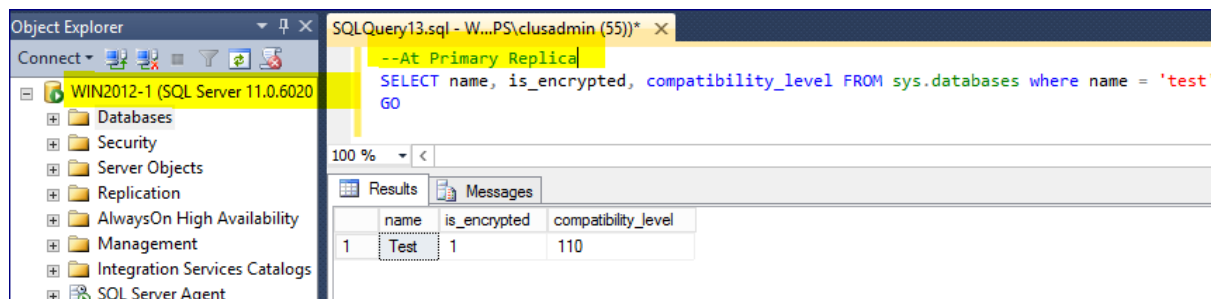
GO



8. Check to verify the database is encrypted ?

SELECT name, is_encrypted, compatibility_level FROM sys.databases where name = 'test'

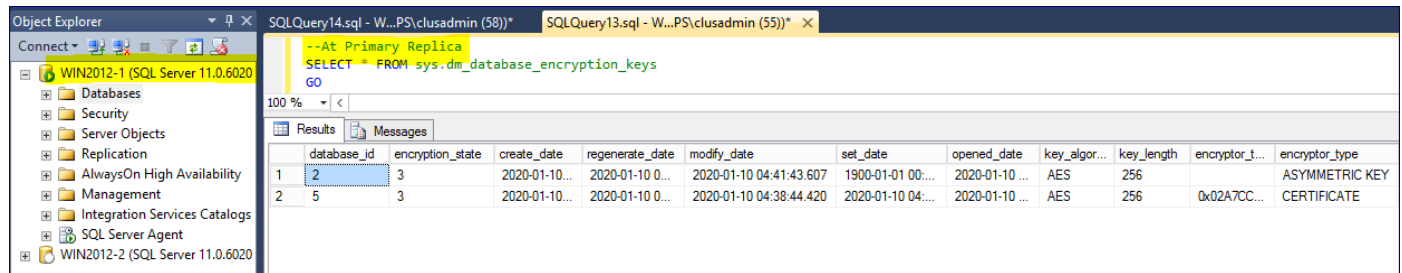
GO



How to find details about each database that is encrypted ?

```
SELECT * FROM sys.dm_database_encryption_keys
```

GO



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Object Explorer' pane displays the server hierarchy for 'WIN2012-1 (SQL Server 11.0.6020)'. The 'Databases' folder is expanded. The main pane shows the results of the query 'SELECT * FROM sys.dm_database_encryption_keys'. The results are displayed in a table with the following columns: database_id, encryption_state, create_date, regenerate_date, modify_date, set_date, opened_date, key_algorithm, key_length, encryptor_type, and encryptor_type. The table contains two rows of data.

database_id	encryption_state	create_date	regenerate_date	modify_date	set_date	opened_date	key_algorithm	key_length	encryptor_type	encryptor_type
1	2	2020-01-10...	2020-01-10 0...	2020-01-10 04:41:43.607	1900-01-01 00:...	2020-01-10 ...	AES	256	ASYMMETRIC KEY	ASYMMETRIC KEY
2	5	2020-01-10...	2020-01-10 0...	2020-01-10 04:38:44.420	2020-01-10 04:...	2020-01-10 ...	AES	256	0x02A7CC...	CERTIFICATE

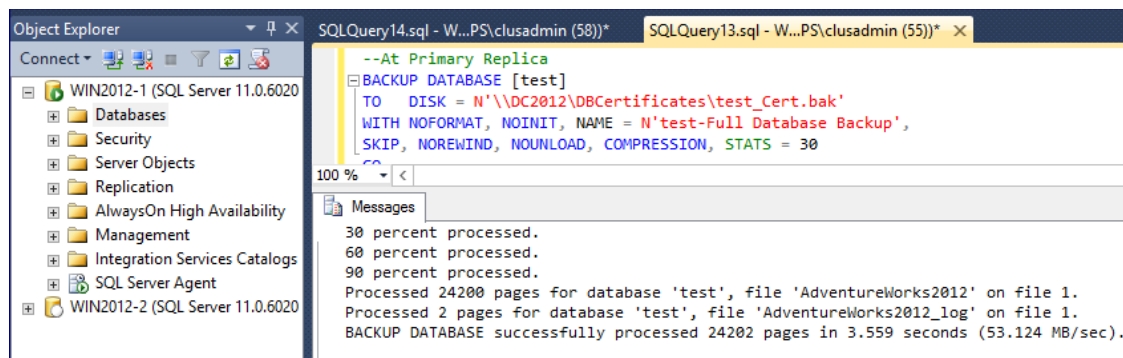
9. Take a full backup of the database. Note: Perform copy-only backup if it's part of a backup maintenance plan so that we don't break the existing backup chain if you need to revert at any time incase.

```
BACKUP DATABASE [test]
```

```
TO DISK = N'\\DC2012\DBCertificates\test_Cert.bak' WITH NOFORMAT, NOINIT, NAME = N'test-Full Database Backup',
```

```
SKIP, NOREWIND, NOUNLOAD, COMPRESSION, STATS = 30
```

GO



The screenshot shows the SQL Server Enterprise Manager interface. The 'Object Explorer' pane is on the left. The main pane shows the execution of the 'BACKUP DATABASE [test]' command. The 'Messages' pane displays the progress of the backup operation.

```
--At Primary Replica
BACKUP DATABASE [test]
TO DISK = N'\\DC2012\DBCertificates\test_Cert.bak'
WITH NOFORMAT, NOINIT, NAME = N'test-Full Database Backup',
SKIP, NOREWIND, NOUNLOAD, COMPRESSION, STATS = 30
```

30 percent processed.
60 percent processed.
90 percent processed.
Processed 24200 pages for database 'test', file 'AdventureWorks2012' on file 1.
Processed 2 pages for database 'test', file 'AdventureWorks2012_log' on file 1.
BACKUP DATABASE successfully processed 24202 pages in 3.559 seconds (53.124 MB/sec).

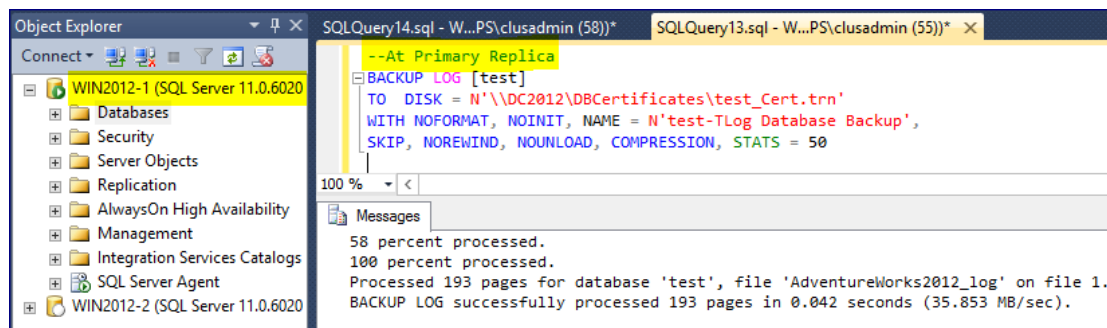
10. Take a log backup (To make it a part of an AG with TDE enabled a log backup is required)

```
--At Primary Replica
```

```
BACKUP LOG [test]
```

```
TO DISK = N'\\DC2012\DBCertificates\test_Cert.trn'
```

```
WITH NOFORMAT, NOINIT, NAME = N'test-TLog Database Backup', SKIP, NOREWIND, NOUNLOAD, COMPRESSION, STATS = 50
```



The screenshot shows the SQL Server Enterprise Manager interface. The 'Object Explorer' pane is on the left. The main pane shows the execution of the 'BACKUP LOG [test]' command. The 'Messages' pane displays the progress of the log backup operation.

```
--At Primary Replica
BACKUP LOG [test]
TO DISK = N'\\DC2012\DBCertificates\test_Cert.trn'
WITH NOFORMAT, NOINIT, NAME = N'test-TLog Database Backup',
SKIP, NOREWIND, NOUNLOAD, COMPRESSION, STATS = 50
```

58 percent processed.
100 percent processed.
Processed 193 pages for database 'test', file 'AdventureWorks2012_log' on file 1.
BACKUP LOG successfully processed 193 pages in 0.042 seconds (35.853 MB/sec).

11. ON THE SECONDARY NODES – Create the same Database Master Key (DMK) in the master database that was created on Node 1 for Node 2.

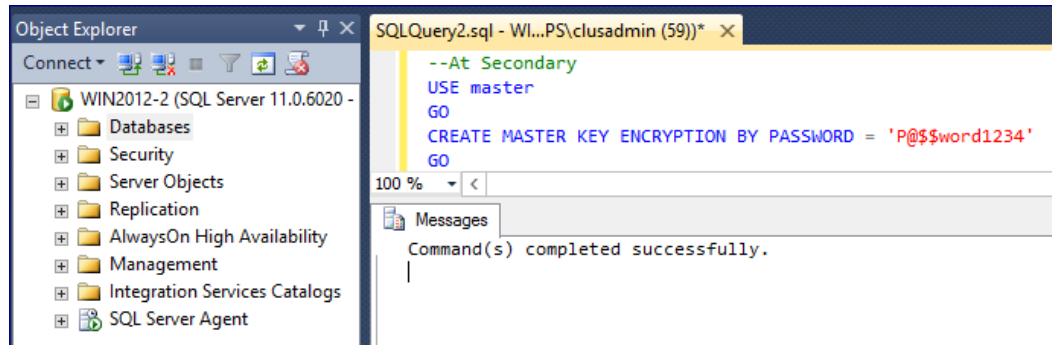
Run this script on all secondary nodes (at Node 2)

USE master

GO

CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'P@\$sword1234'

GO



12. Transfer the certificate from the certificate backup on secondary node: Node 2

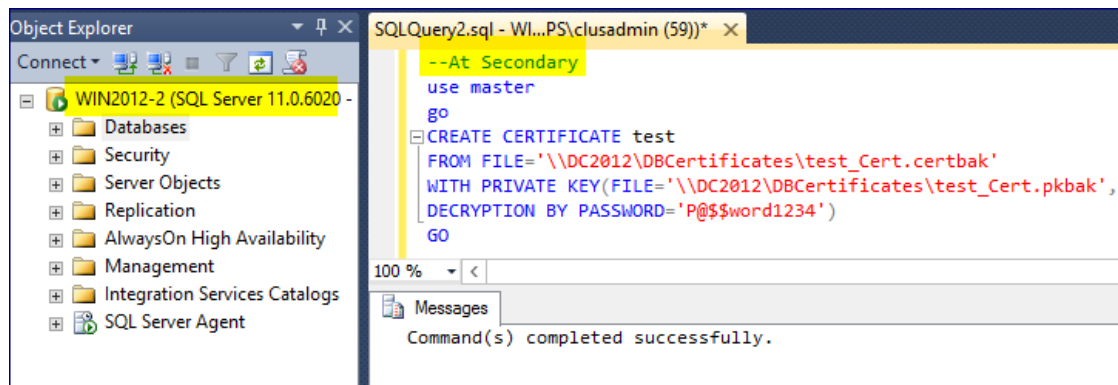
CREATE CERTIFICATE test

FROM FILE='\\DC2012\DBCertificates\test_Cert.certbak'

WITH PRIVATE KEY(FILE='\\DC2012\DBCertificates\test_Cert.pkbak',

DECRYPTION BY PASSWORD='P@\$sword1234')

GO



13. Restore the full backup followed by the log backup of the encrypted test database with No-Recovery mode. We want the database to be in restoring mode so we can join it later to the Availability Group via script.

Note: On secondary node, we are using the backups that we took earlier on the primary node.

--At Secondary

USE [master]

GO

RESTORE DATABASE [test]

FROM DISK = '\\DC2012\DBCertificates\test_Cert.bak' WITH FILE = 1, NORECOVERY, NOUNLOAD, STATS = 1

GO

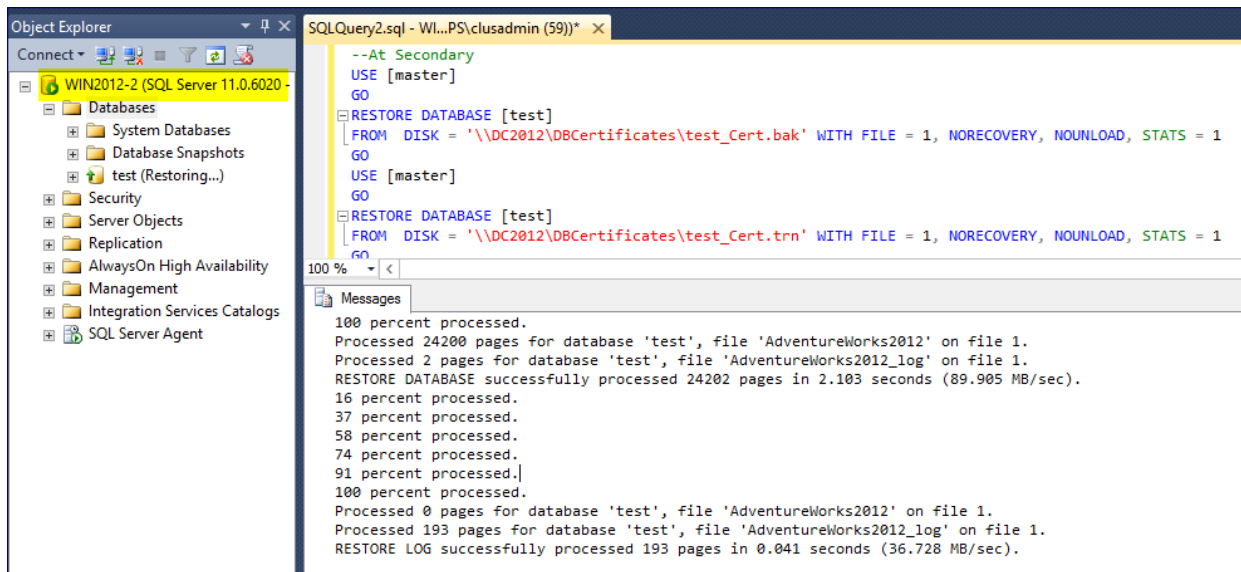
USE [master]

GO

RESTORE DATABASE [test]

FROM DISK = '\\DC2012\DBCertificates\test_Cert.trn' WITH FILE = 1, NORECOVERY, NOUNLOAD, STATS = 1

GO



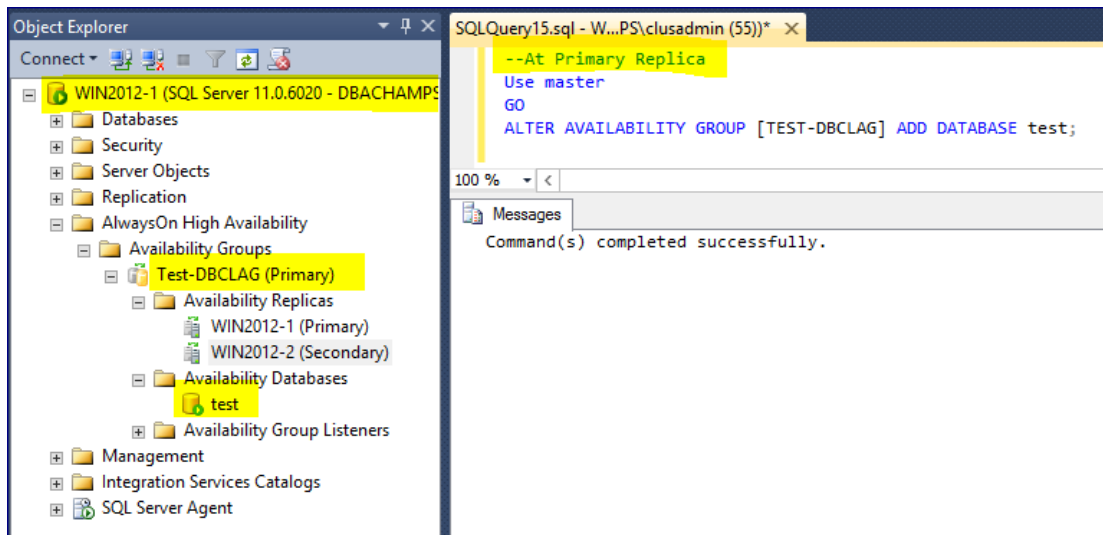
Finally, Go back to the Primary Replica/Node:

14. Go back to the Primary node and add the database to the availability group. Refresh all of the nodes and we will see that the test database has been successfully added to the AG.

Use master

GO

ALTER AVAILABILITY GROUP [TEST-DBCLAG] ADD DATABASE test;



With this step, we have successfully added an TDE Enabled database into Always-on AG Group.

Re-modified by – Praveen Madupu

Sr. SQL Server DBA

Whatsapp: +91-98661-30093

Direct Call: +91-81972-93434