## Upgrade SQL Server with Minimum disruption and Risk (In-Place upgrade from SQL Server 2017 to SQL Server 2019)

# Document Summary: This document is about [Upgrading our SQL Server 2017 to SQL Server 2019 with AlwaysOn Availability Groups](https://sqlundercover.com/2017/06/30/upgrading-our-test-lab-from-sql-2016-to-2017-ctp-with-always-on-availability-groups/) (Automatically using Power Shell Programming).

NOTE: Below Pre-Requisites is Mandatory for the Data Recovery and Smooth In-Place upgrade of SQL Server 2017 to SQL Server 2019

1. Verify that backups and Log backups exist for all databases (user and system). Verify that these backups can be restored.
2. Make sure you have Admin Rights over the Instances for proper Installation (To avoid Access Denied issues)
3. Make sure you script out all important SQL Server Jobs, SQL Server Logins
4. Make sure you script out all Transactional Replication Jobs, Publishers, Subscribers 5. Script out any and all necessary system objects.
5. Script out any and all necessary SSIS packages (either from MSDB or as flat files).

NOTE: USE ONLY the upper version ConfigurationFile.ini in case of "/ACTION = UPGRADE"

[**Upgrading our SQL Server 2017 to SQL Server 2019 with Always On Availability Groups**](https://sqlundercover.com/2017/06/30/upgrading-our-test-lab-from-sql-2016-to-2017-ctp-with-always-on-availability-groups/)

Carrying out an in place upgrade with minimal downtime, below are the series of steps followed:

Here is the very basic setup for this scenario; we have Availability Group **AOAG1** with 3 Replicas.

We have 3 SQL replicas ‘**SQL01**’, ‘**SQL02**’, ‘**SQL03**’ and we will be starting this scenario with SQL01 acting as the primary replica, so let’s get started. **Now would be a good time to take any Full backups that may be missing or inaccessible**

Before you start be sure to check you have sufficient log drive space for potential log file growth in case the SQL Server installations take longer than expected.

1. Let’s get the secondary replica (**SQL03**) switched from Synchronous mode to Asynchronous mode, we need to ensure that this is set so that transactions on the primary can be committed without waiting for confirmation from the secondary replica, later the SQL service on the secondary will be restarted as part of the upgrade and the server itself will be rebooted.

USE [master]

GO

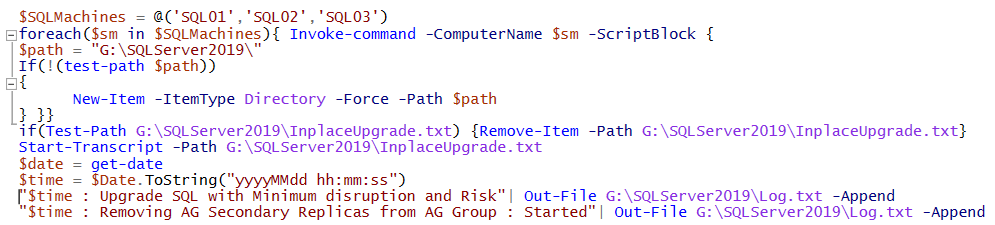
 ALTER AVAILABILITY GROUP [AOAG1]

MODIFY REPLICA ON N'SQL03'

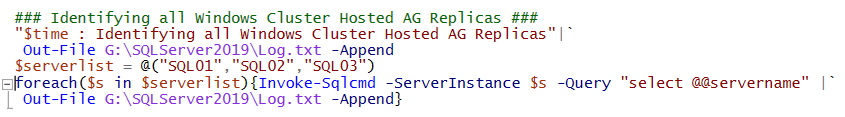
WITH ( AVAILABILITY\_MODE = ASYNCHRONOUS\_COMMIT );

**2.** Connect to the Secondary server using which ever method you use to access the machine , in our case we are connecting to **SQL03** (you may have more than one so this will be the first secondary that you are planning to upgrade). Let’s do it automatically by using the Windows Power shell Programming:

1. Creating timestamp and log file and Media path in all AG Replicas



1. Identifying the SQL Server AOAG Replicas



1. Copying SQL Server 2019 Installation media and Starting the In Place Upgrade from SQL Server 2017 --> SQL Server 2019

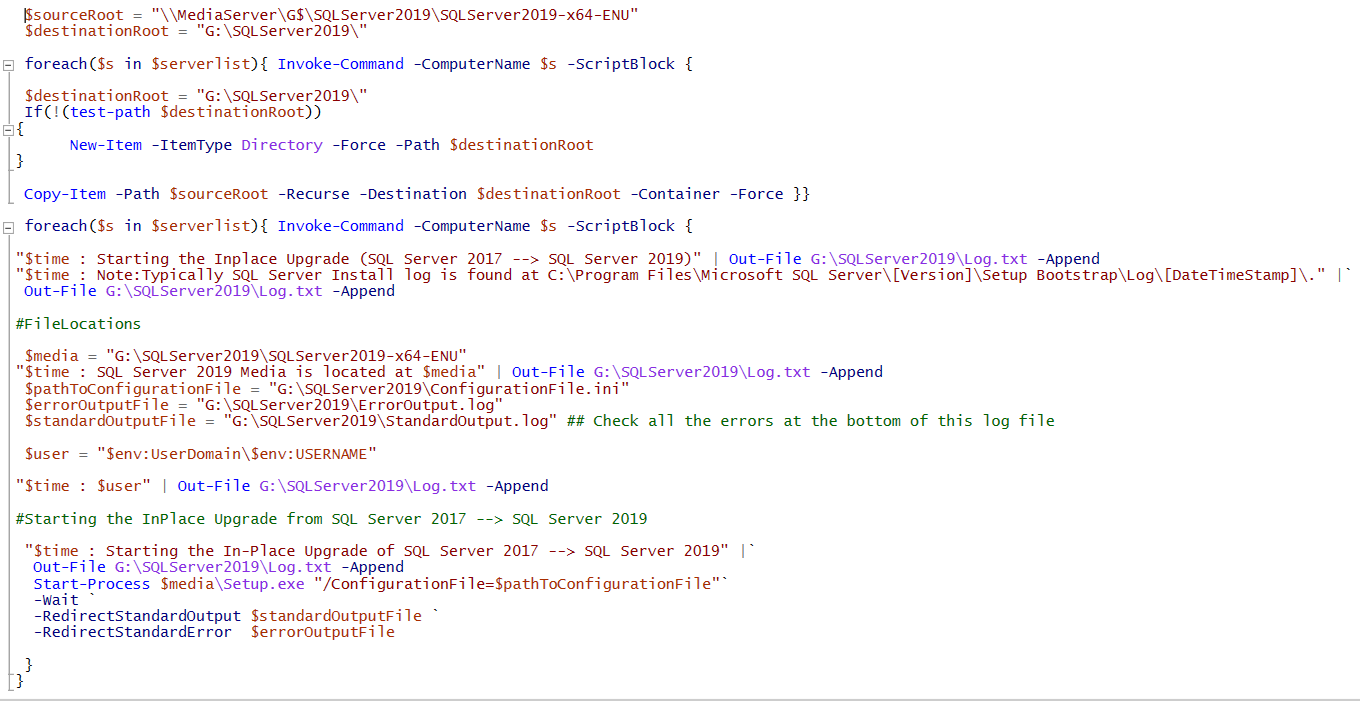
NOTE: During In Place Upgrade below ConfigurationFile.ini Parameters are MANDATORY.

1. The **/IAcceptSQLServerLicenseTerms** command line parameter is missing or has not been set to true. It is a required parameter for the setup action you are running.

NOTE: During In Place Upgrade below ConfigurationFile.ini Parameters are **NOT ACCEPTED** when the value of setting 'ACTION' is 'Upgrade'.

The setting FEATURES, INSTALLSHAREDDIR, INSTALLSHAREDWOWDIR, INSTANCEDIR, AGTSVCACCOUNT,AGTSVCSTARTUPTYPE,COMMFABRICPORT,COMMFABRICNETWORKLEVEL,COMMFABRICENCRYPTION,MATRIXCMBRICKCOMMPORT,SQLSVCSTARTUPTYPE,FILESTREAMLEVEL,SQLMAXDOP,SQLMAXMEMORY,SQLMINMEMORY,ENABLERANU,SQLCOLLATION,SAPWDSQLSVCACCOUNT,SQLSVCINSTANTFILEINIT,SQLSYSADMINACCOUNTS,SQLTEMPDBFILECOUNT,SQLTEMPDBFILESIZE,SQLTEMPDBLOGFILEGROWTH,ADDCURRENTUSERASSQLADMIN,TCPENABLED,TCPENABLED,NPENABLED,BROWSERSVCSTARTUPTYPE is not allowed when the value of setting 'ACTION' is 'Upgrade'.

Below is the code for the Copying SQL Server 2019 Installation media and Starting the In Place Upgrade from SQL Server 2017 --> SQL Server 2019.



1. This bit will take a little while for the upgrade to be completed.
2. At this point in the installation the SQL Service will be restarted on the secondary server (SQL02) you are working on, this is where our Asynchronous switch comes into its own.
3. If you happen to be watching the Availability group Dashboard/s then you will notice that the Replica you are working on will be showing as Critical with Warnings – this is fine we can ignore these as they are planned and will not be affecting the Primary node from a service availability point of view, transactions will be happily committing between the Primary and any other Secondaries you may have,  but for now the queue will be building for this server until it comes back online and back into sync.
4. Service has been restarted as part of the installation and now the installation has completed, if you are still watching the Availability group dashboard you will notice that things should have re-synced back up again with some lovely looking green ticks (these won’t stay for long unfortunately) , it’s now time to Restart that secondary – Lets restart the secondary replica box…
5. When SQL02 comes back up you will notice that the dashboard will be showing those green ticks again as things are back in Sync, a quick connection to SQL02 from management studio will reveal that the version is now changed from 14.0.1000.169 🡪 15.0.2000.5.
6. **Final Step:** Time to set the Synchronization mode back to Synchronous Commit on the Secondary in preparation for Failing the Availability group over to the Secondary Server so that we can upgrade the primary with the least impact on the front end users.

* **If you have more than one Secondary Replica then you should repeat all of the previous steps and only proceed past this step once everything is in sync and just the Primary upgrade remains.**

1. Give the Availability group a quick once over to check that it’s completely ‘Synchronized’ and not showing as ‘not synchronizing’.
2. You should now consider taking backups of ALL databases including system databases, the method you use is up to you just be sure that if you are taking Differential or Log backups that the Full is available and that the log chain is intact.
3. Don’t forget to backup, this is our only lifeline if you need to go back to 2017, once the database/s are upgraded to 2019 all subsequent backups can only be restored using SQL Server 2019.
4. Once we feel that we have contingency in place in the form of backups then you should feel confident in proceeding with the failover checking.
5. Be sure to check you Log drive space as there is a chance Logs could grow if the installation hangs or takes a long time if Logs are reaching their space allocation on busy databases.
6. A quick sanity check here to be absolutely certain that there will be no data loss…

In Recovery!?

1. Don’t panic, this is still going as planned as this is totally expected and below is the reason why: We are now in a situation where we have the current Primary Replica server (**SQL02**) running SQL Server 2019 but one or possibly more than one for you running still on SQL Server 2017 version, it’s not possible for the 2017 server to synchronize as its databases have not been upgraded yet, they will therefore be stuck in recovery but we are about to fix that very soon as the whole upgrade completes.
2. It is recommended to run sp\_FailedLogins and [Sp\_FailedJobs](https://sqlundercover.com/2017/06/16/undercover-toolbox-sp_failedjobs-the-quick-way-to-check-for-failed-agent-jobs/) periodically on our new primary (SQL02) whilst the upgrade of SQL03, SQL01 is in progress and also afterwards too just to be sure that we are on top of our game.
3. Once the upgrade is complete successfully on SQL03, SQL01 and the box restarted you should see the dashboard showing a warning triangle rather than the Red X, but still not in Sync yet.
4. If we expand the Availability Databases tree we have the answer – our Data movement was Paused so until we resume the data movement the Synchronization state will remain as ‘Not Synchronizing’, Let’s get this moving…
5. A Quick check of the SQL Server Error log on the Old Primary – now secondary (SQL01) shows that the Databases has been successfully upgraded , Nice .. Now we are back in business!
6. Open the Availability group dashboard so that you can witness green and healthy dashboard….
7. Lets puts things back to the way they were – we set the synchronization mode back to Synchronous Commit on SQL01, SQL02 and Asynchronous commit mode to SQL03.

We are fully upgraded SQL Server 2017 to SQL Server 2019 with minimal impact to the end users