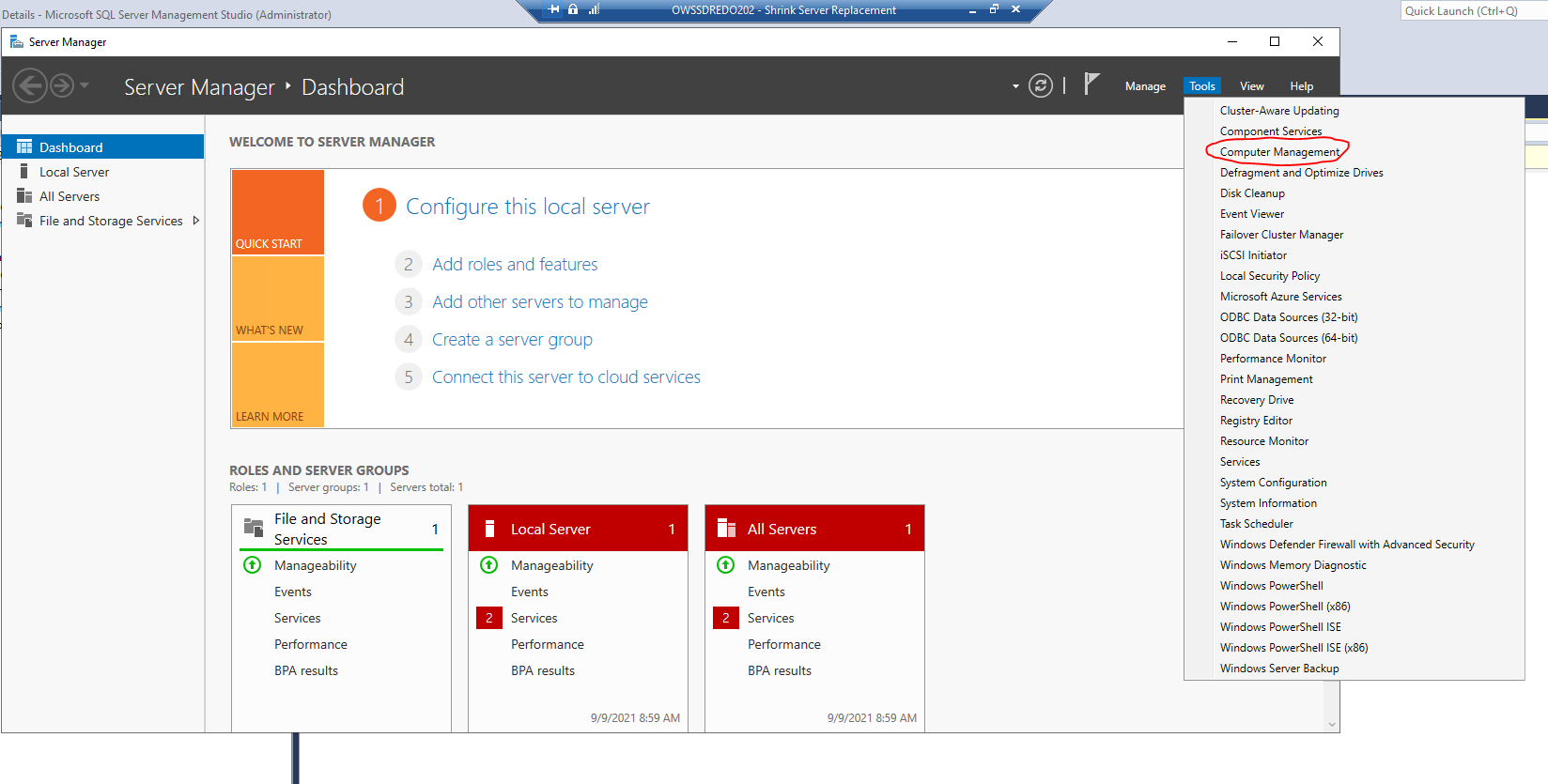
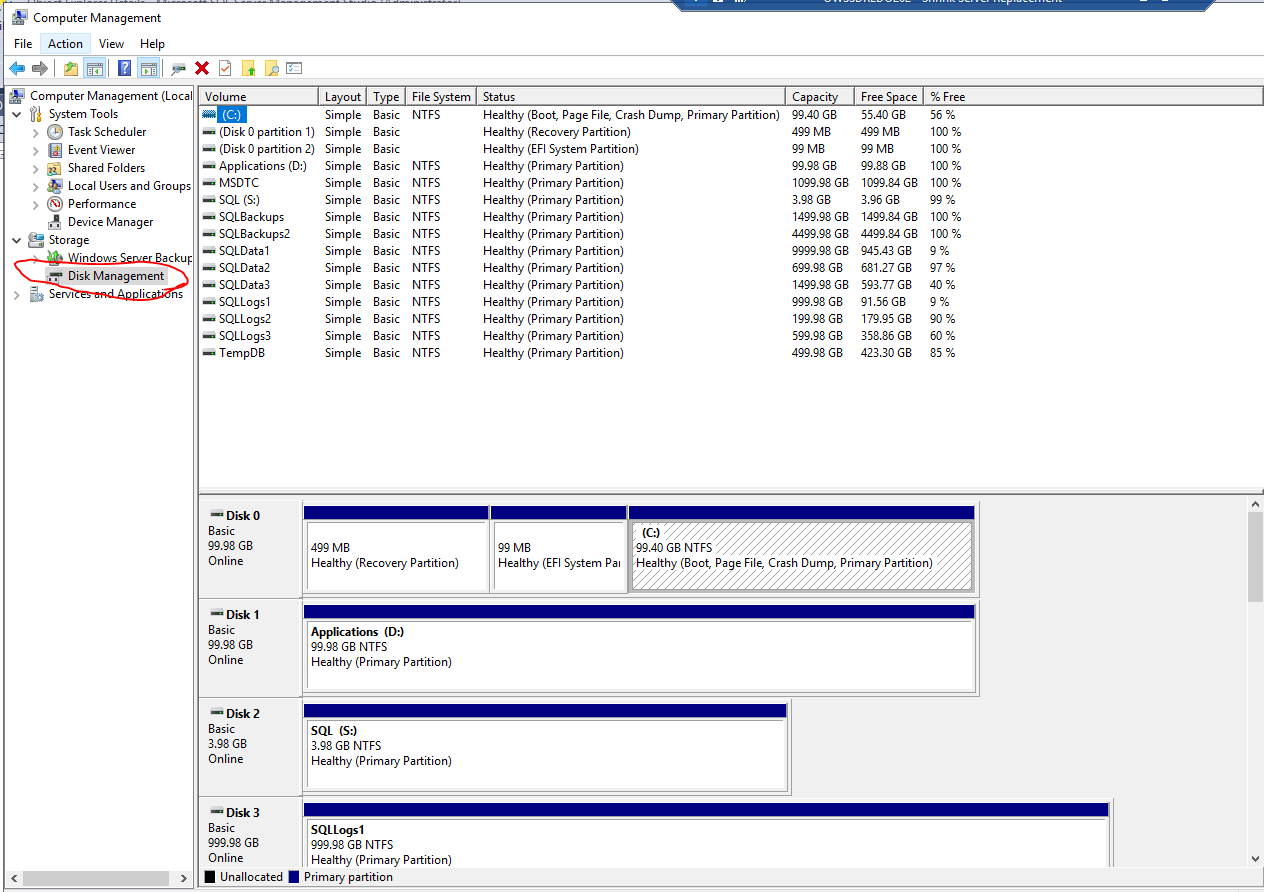
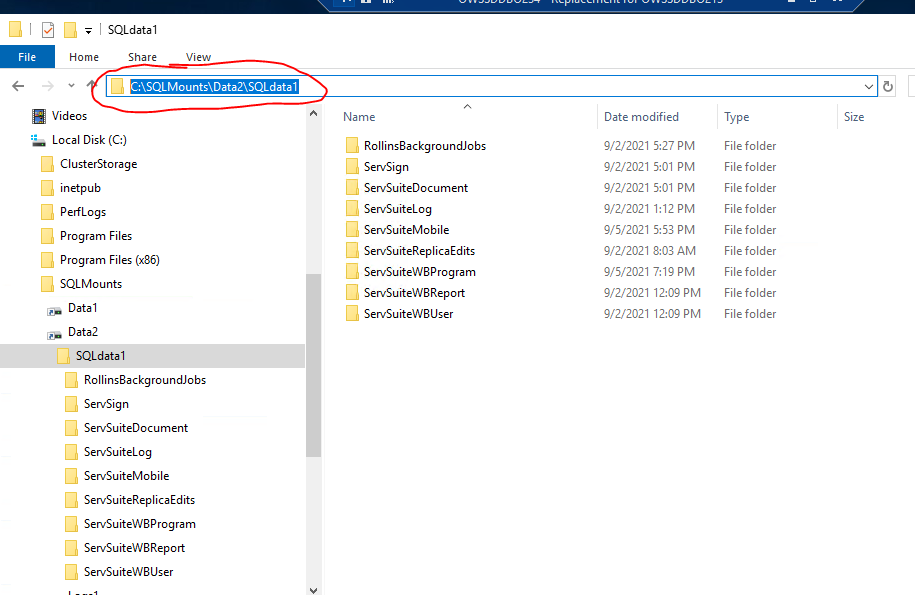
**MSSQL Server Create Databases On Replica Server Using BACKUP & RESTORE Method**

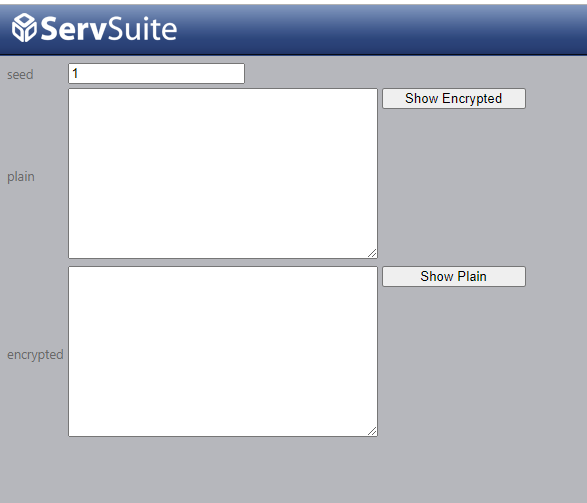
This document contains the step-by-step task required to create a replica of a Microsoft SQL Server databases from the Primary node to the a Replica node.

1. **Check the disk space allocation on the source server.**
   1. Log into the source server and navigate to Computer Management. 
   2. Navigate to Disk Management and take a screenshot of the Volume list. This will be used to ensure the target server has sufficient space to hold the databases.
2. **Create the target directories for the database files.**
   1. The two text files containing the commands to create the necessary directiories are located here: [\\RWSQLMGMTH202\c$\PowerSh\Rollins\DBA\_Admin\database transfer\Transfer DBs to New Server](file:///\\RWSQLMGMTH202\c$\PowerSh\Rollins\DBA_Admin\database%20transfer\Transfer%20DBs%20to%20New%20Server)
      1. Create\_System\_Db\_Directories.txt
      2. Create\_User\_Db\_Directories.txt
3. **Backup the target databases on the Primary node.**
   1. Open your desired tool for creating database backups.
   2. Choose a location for the backup file that will be accessible by the target server.
      1. When using the MSSQL native backup method, choose a shared network directory that can be reached by the other server(s).
         1. Make sure the directory is large enough to hold the file. If it isn’t, contact the appropriate department to request more space.
      2. Third-party tools, such as EMC’s DDBoost, use a shared backup server, so space and accessibility shouldn’t be an issue.
   3. Give the file a descriptive name that is easily identifiable.
   4. Choose the FULL & COPY-ONLY options.
   5. Set the retention period to an acceptable, realistic value; you don’t need to keep this backup after the next regularly scheduled full backup.
   6. Once the FULL backup is complete:
      1. Open a new query window connected to the database.
      2. Execute CHECKPOINT.
      3. Perform a Transaction Log backup. This will be applied to the Replica node during the database restore steps.
4. **Restore the database(s) onto the Replica node.**
   1. Once the backup has completed, log into the target Replica server and **REPEAT STEP 1, A & B FROM ABOVE.**
   2. Compare the Volume sizes of the source and target.
      1. Ideally, these values will be exactly the same, including the names. If not, you can use the MOVE option in the database restore to place the files in their proper location on the target server.
   3. Open the backup utility and navigate to the RESTORE portion.
      1. Locate the backups previously created in **Step 2**.
      2. Follow the necessary steps for restoring the full & log backup of your database(s)
         1. Set the proper locations for each file using the standard path & naming convention.
            1. For example, new DEV boxes will store data files in C:\SQLMounts\Data2\SQLdata1\<\_\_Database Name\_\_> 
      3. Select the FULL Backup created in the previous steps.
      4. Choose the “**WITH NORECOVERY**” option for the restore. This allows additional logs to be applied before adding the database to an AG Group.
   4. Once the restore has completed, perform a TRANSACTION Log Restore of your database with the “**WITH NORECOVERY**”. This is necessary for adding the database to an AG Group.
5. **Copy the logins, passwords, and database permissions from the Primary node.**

Repeat these steps after all databases have been restored on each server.

The scripts 01\_Create\_procs\_to\_transfer\_logins\_to\_new\_server.sql and 02\_Generate SQL\_For\_Logins.sql are used to generate the scripts that will be used to recreate the users on the target MSSQL Server instance.

* 1. Copy the files to a location accessible by the source/original server.
     1. On the source/original server, execute the script 01\_Create\_procs\_to\_transfer\_logins\_to\_new\_server.sql
     2. Copy the contents of the “**Messages**” tab to a **new query window on the new target server** and execute.
     3. On the source/original server, execute the script 02\_Generate SQL\_For\_Logins.sql
     4. Copy the contents of the “**Results**” tab to a **new query window on the new target server** and execute.
     5. All logins from the source server have been created on the new server with their existing hashed passwords.
  2. Verify that all restored databases are accessible and all users have been recreated.

1. **You may now proceed with the steps necessary for Adding A Database To An Always-On Group After Database Has Been Restored.**
2. **If the BOSS database ServSuiteWBUser has been restored to a new server, the encrypted values in the table ServSuiteWBUser.dbo.tdatabase must be changed to reflect the new server and AG Listener names.**
   1. Navigate to the page for [DEV BOSS Encryption Tool](http://rwaseah21.rollins.local:84/encryption.aspx). 
   2. Create encrypted values for the AG Listener and Replica server and place them in the appropriate sections of the following query and execute on new Primary:

-------------------------------------------------------------------------------

-- <\_\_New AG Listener Name\_\_> = Encrypted value for new AG Listener

-- <\_\_New Replica Server Name\_\_> = Encrypted value for new Replica Server Name

-------------------------------------------------------------------------------

UPDATE [ServSuiteWBUser].[dbo].[tdatabase]

SET programserver = '<\_\_New AG Listener Name\_\_>',

logserver = '<\_\_New AG Listener Name\_\_>',

helpserver = '<\_\_New AG Listener Name\_\_>',

replicateserver = '<\_\_New Replica Server Name\_\_>'

GO

1. Verify the new values have been Populated.
2. Verify existence of all necessary logins, database mappings, and database ownership.
3. Verify Secondary replica nodes are set to **Readable secondary Yes**. 