**DD Boost Backup and Restore PowerShell Scripts**

**Backup-MT\_DDBoost\_DB**

Synopsis:

Function to take backup of any database on any server which has ddboost installed on it.

Parameters:

*-Servername(mandatory):* Hostname\InstanceName of Backup server where you want to take backup

*-database:* Database name to perform backup. If no value provided all databases on the specified instance will be taken backup.

*Syntax:*

***Backup-MT\_DDBoost\_DB -servername BOWSQLTEST2019A\BOMSSTEST2019A -database Hamster1***

*Description:*

Upon being passed single server and single database name command will perform backup of database by collecting information from ddboost info table of servername and saves information in log file on C:\Temp\Backup-MT\_DDBoost\_YYYYMMDD\_HHMMSS.log. Server Name provided should be a full path i.e., hostname\instancename. If No Name for database is provided by default all databases will be backed up.

Parameters:

**Servername**

Hostname\Instancename - ServerName for backup

**Database**

Database name for backup. If value is NULL all databases in instance are backed up.

**Restore-MT\_DDBoost\_DB**

*Synopsis*

Function to restore a backup of database from one server to another server using DDBoost.

*Syntax*

***Restore-MT\_DDBoost\_DB -backupserver BOWSQLTEST2019A\BOMSSTEST2019A -backupDB Hamster1 -RestoreServer BOWSQLTEST2019B\BOMSSTEST2019B -RestoreDB Hamster1\_Test1 -timestamp "12/08/2021 10:00:00 AM"***

*Description*

Upon passing backupserver and backupDB, Script collects data about data domain server from backupserver.

Prepares Dynamic CMD Command for restoration of Restore DB on RestoreServer. TimeStamp parameter makes it possible to restore database to particular

time frame given log backups are present on server. File Mapping is automatically done if restore database exists on Restoreserver. Detailed log is stored in location C:\Temp\Restore-MT\_DDBOOST\_YYYYMMDD\_HHMMSS.log.

*Parameters:*

BackupServer(Mandatory)

Hostname\Instancename - to collect information about ddboost data domain where source database is present for restoration

backupDB(Mandatory)

Single value of database name in string for backup file retrieval from DDBoost.

RestoreServer(Mandatory)

Hostname\Instancename - Destination server name where database should be restored.

RestoreDB(Mandatory)

Database name in string for restore. Can be same or different from source database name.

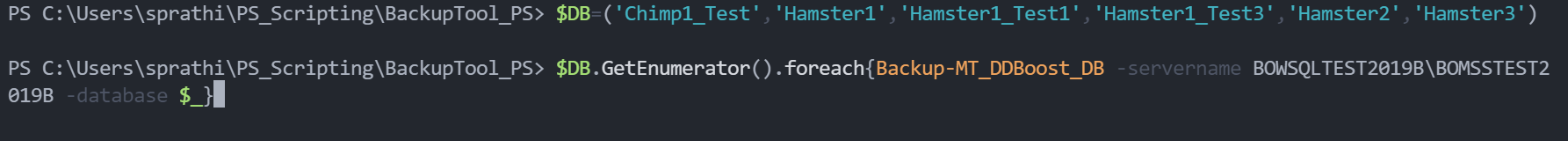
timestamp

Timestamp in format "MM/DD/YYYY HH:MM:SS AM\PM" (quotes mandatory). If log backups are present for given timestamp. Command automatically picks all backups until that timestamp for restore.

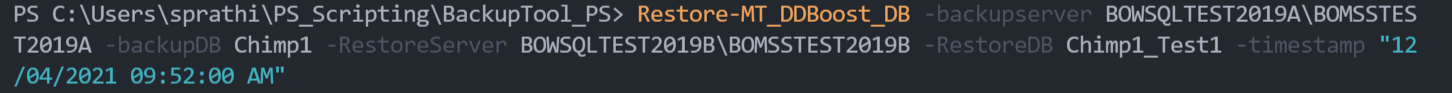
If no backups are present for given timestamp. Script throws an error.

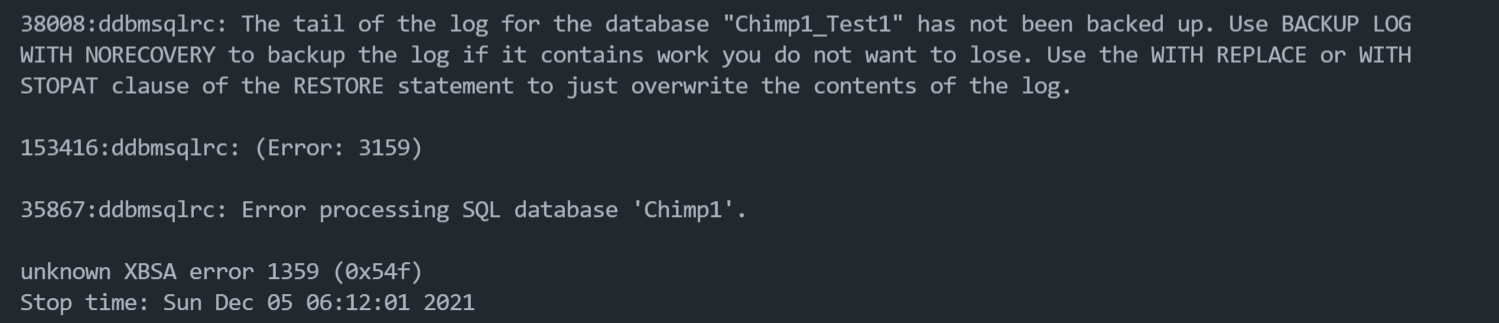
Working:

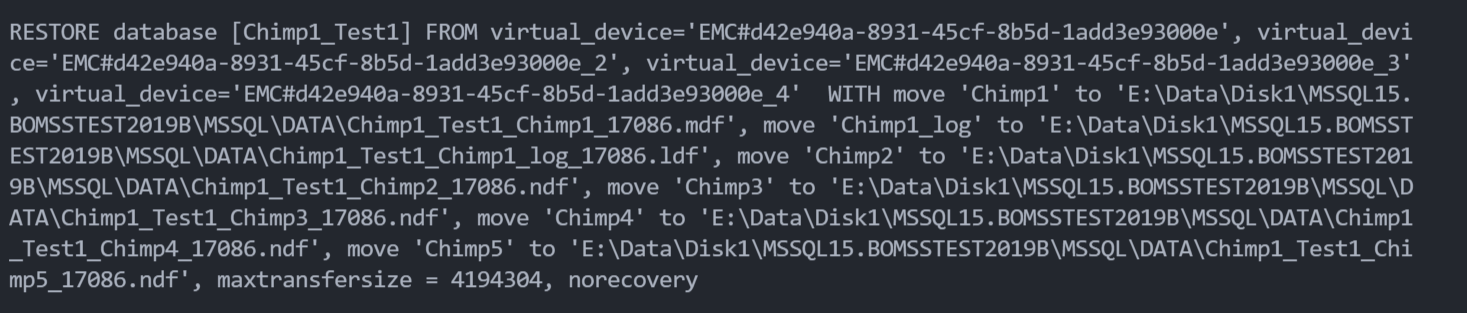
* Make sure you DDboost is installed on both backupserver and restore server.
* Copy Backup-MT\_DDBoost\_DB.ps1,Restore-MT\_DDBoost.ps1 and Write-log.ps1 to same folder. Otherwise script throws an error.
* Open Powershell and navigate to above folder location where you copied all three files. Run all three files so that they are loaded into powershell.
* Once they are loaded use above syntax to backup and restore any database which has ddboost installed on them.
* Open log file for each script and for more details. (C:\Temp)
* If you want to run these scripts for more multiple databases(empty value into database parameter will backup all databases. If you want to take backup of certain set of databases) use below syntax. Put all database names that belong to same server in single array variable. For example $DB.



* If you want to take backup of destination database before restore. Run Backup-MT\_DDBoost\_DB for source server and source db before you run restore command.
* Backup only performs full back and restore only performs with recovery. Incase of T-log backups and No Recovery restores requirement use Backup-MT\_SQLDB and Restore-MT\_SQLDB scripts.
* If script fails because it is unable to take tail-log backup of destination database. Either drop database and restore it or copy T-SQL from log or from PS screen and edit it to restore with replace.







*RESTORE database [Chimp1\_Test1] FROM virtual\_device='EMC#d42e940a-8931-45cf-8b5d-1add3e93000e', virtual\_device='EMC#d42e940a-8931-45cf-8b5d-1add3e93000e\_2', virtual\_device='EMC#d42e940a-8931-45cf-8b5d-1add3e93000e\_3', virtual\_device='EMC#d42e940a-8931-45cf-8b5d-1add3e93000e\_4' WITH*

*move 'Chimp1' to 'E:\Data\Disk1\MSSQL15.BOMSSTEST2019B\MSSQL\DATA\Chimp1\_Test1\_Chimp1\_17086.mdf',*

*move 'Chimp1\_log' to 'E:\Data\Disk1\MSSQL15.BOMSSTEST2019B\MSSQL\DATA\Chimp1\_Test1\_Chimp1\_log\_17086.ldf',*

*move 'Chimp2' to 'E:\Data\Disk1\MSSQL15.BOMSSTEST2019B\MSSQL\DATA\Chimp1\_Test1\_Chimp2\_17086.ndf',*

*move 'Chimp3' to 'E:\Data\Disk1\MSSQL15.BOMSSTEST2019B\MSSQL\DATA\Chimp1\_Test1\_Chimp3\_17086.ndf',*

*move 'Chimp4' to 'E:\Data\Disk1\MSSQL15.BOMSSTEST2019B\MSSQL\DATA\Chimp1\_Test1\_Chimp4\_17086.ndf',*

*move 'Chimp5' to 'E:\Data\Disk1\MSSQL15.BOMSSTEST2019B\MSSQL\DATA\Chimp1\_Test1\_Chimp5\_17086.ndf',*

*maxtransfersize = 4194304, norecovery, REPLACE*