

How to Restore a SQL Server Database into Azure SQL DB Managed Instances

 brentozar.com/archive/2018/03/how-to-restore-a-sql-server-database-into-azure-sql-db-managed-instances

Brent Ozar

March 16, 2018

You've built your first Managed Instance, and now you wanna play around with real live data. You're going to need a backup in the cloud to do it – SSMS doesn't have a drag-and-drop option.

Big picture, here's what we're going to do:

1. Set up a container in the cloud to hold our backups
2. In our regular SQL Server, set up security so we can back up to that cloud container, and then back up a database to it
3. In our Managed Instance (MI), set up security to access the cloud container, and then restore the backup from it

For steps 1 & 2, follow Steve Thompson's excellent checklist, [Backup SQL Server to an Azure Storage Account](#). I've tried several other checklists, but as of 2018, this was the only one that worked for me out of the box. (He's [@Steve_TSQL](#) and a nice guy.)

The restore syntax in Managed Instances is a little different.

To create the security credential, the syntax is a little different:

Transact-SQL

```
1 CREATE CREDENTIAL
2 [https://MYSTORAGEACCOUNTNAME.blob.core.windows.net/MYCONTAINERNAME]
3 WITH IDENTITY = 'SHARED ACCESS SIGNATURE',
4 SECRET = 'MYBIGLONGKEYSTRINGFROMTHEAZUREPORTAL';
GO
```

The portions you have to change in there are:

- MYSTORAGEACCOUNTNAME
- MYCONTAINERNAME
- MYBIGLONGKEYSTRINGFROMTHEAZUREPORTAL

Then, to restore the backup, the syntax is a little different too:

Transact-SQL

```
1 RESTORE DATABASE MYDATABASENAME
2 FROM URL =
  'https://MYSTORAGEACCOUNTNAME.blob.core.windows.net/MYCONTAINERNAME/MYBACKUPNAME.bak'
```

The portions you have to change:

- MYDATABASENAME – the target, new database you're creating
- MYSTORAGEACCOUNTNAME
- MYBACKUPNAME
- MYCONTAINERNAME

You can also [restore databases asynchronously](#) – meaning, fire off the restore so that it runs even when you're not connected.

Want to check progress? This is about to get technical.

If you don't care about checking progress, you can bail here. Happy testing.

You can't use WITH parameters on the restore, which means you don't get status messages. To check restore progress, use `sp_WhoIsActive`, which works beautifully:

session_id	sql_text	login_name	wait_info
1	SELECT top_data FROM sys.hi_log_read_event_stream @source, @sourceid ...	NT AUTHORITY\SYSTEM	(633ms)P
2	SELECT top_data FROM sys.hi_log_read_event_stream @source, @sourceid ...	DE3-WF-4B3Gm5WkEg3P	(128ms)P
3	RESTORE DATABASE StackOverflow2010 FROM URL = https://www.stackoverflow.com/backup/StackOverflow2010bak ...	Doctor	(271ms)S
4	RESTORE DATABASE StackOverflow2010 FROM URL = https://www.stackoverflow.com/backup/StackOverflow2010bak ...	DE3-WF-4B3Gm5WkEg3P	(40172ms)

Restore in progress, part 1

In that screenshot, session_id 143 is me (Doctor) running the restore command, restoring StackOverflow from backup. If I scroll across to the right of `sp_WhoIsActive`'s output, I can normally check restore progress, but it's null:

status	open_tran_count	percent_complete	host_name	database_name	program_name	start_time	login_time	request_id	collection_time
suspended	0	NULL	DE3	master	TdService	2018-03-15 22:44:58.800	2018-03-15 22:44:58.803	0	2018-03-16 00:15:19.417
suspended	0	NULL	DE3	master	TdService	2018-03-15 22:44:59.813	2018-03-15 22:44:59.820	0	2018-03-16 00:15:19.417
suspended	0	NULL	GatewayDrug	master	Microsoft SQL Server Management Studio - Query	2018-03-16 00:08:25.143	2018-03-15 23:49:53.140	0	2018-03-16 00:15:19.417
suspended	0	81.0911	DE3	master	RestoreService	2018-03-16 00:08:37.313	2018-03-16 00:08:36.690	0	2018-03-16 00:15:19.417

Restore progress, part 2

But it's null.

Notice how there's another restore running from 127 at the same time, restoring a guid-based database? With the wacko service name, and program_name RestoreService? Azure Managed Instances are the ones really doing the restore, and you can see the percent_complete is populated for their session. They're doing more than just restoring locally, too. In the error log, while a restore runs, you'll also see interesting messages about how Azure is setting up the replication for me to the other nodes:

Date	Source	Message
3/16/2018 12:17:00 AM	spid171a	[Fabric:ReplicaManager: GetServiceDescriptionFromFabric] GetFabricServiceNameHint for fabric service uri: fabric:/Worker/CL/ce39543ce5e5e/SQL LogicalServer/ce...
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: MinLogConsumptionThresholdForThrottlingDueToSeeding is set to 50.000000
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: Registration for config changes of MinLogConsumptionThresholdForThrottlingDueToSeeding succeeded
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: MinDatabaseSizeForThrottlingDueToSeedingInBytes is set to 1000000.000000
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: Registration for config changes of MinDatabaseSizeForThrottlingDueToSeedingInBytes succeeded
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: MinTransferedBytesBeforeThrottling is set to 40000000.000000
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: Registration for config changes of MinTransferedBytesBeforeThrottling succeeded
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: MinSeedingPriorityPerSecond is set to 1000000.000000
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: Registration for config changes of MinSeedingPriorityPerSecond succeeded
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: MinDatabaseSizeForCancelSeedingInBytes is set to 1000000.000000
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: Registration for config changes of MinDatabaseSizeForCancelSeedingInBytes succeeded
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: ResetCancelledSeedingTimeInMsecs is set to 3600000.000000
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: Registration for config changes of ResetCancelledSeedingTimeInMsecs succeeded
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: MaxCancelledSeedingForLogGovernance is set to 5.000000
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: Registration for config changes of MaxCancelledSeedingForLogGovernance succeeded
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: SeedingCompletionThresholdForSkippingCancellation is set to 90.000000
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: Registration for config changes of SeedingCompletionThresholdForSkippingCancellation succeeded
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: LogConsumptionThresholdForCancelSeeding is set to 30.000000
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: Registration for config changes of LogConsumptionThresholdForCancelSeeding succeeded
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: AgeThresholdForKactAbortInMinutes is set to 60.000000
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: Registration for config changes of AgeThresholdForKactAbortInMinutes succeeded
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: LogConsumptionThresholdForKactAbortPct is set to 95.000000
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: Registration for config changes of LogConsumptionThresholdForKactAbortPct succeeded
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: AdditionalLogSizeDueToGeoSecondaryTruncationHoldupPct is set to 5.000000
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: Registration for config changes of AdditionalLogSizeDueToGeoSecondaryTruncationHoldupPct succeeded
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: LogSizeDueToSecondaryTruncationHoldupPct is set to 90.000000
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: Registration for config changes of LogSizeDueToSecondaryTruncationHoldupPct succeeded
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: MaxCatchupQueueSizeThresholdPct is set to 50.000000
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: Registration for config changes of MaxCatchupQueueSizeThresholdPct succeeded
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: LogConsumptionThresholdForThrottlingPct is set to 75.000000
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: Registration for config changes of LogConsumptionThresholdForThrottlingPct succeeded
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: LogManagementPolicyExecutionIntervalInMsecs is set to 10000.000000
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: Registration for config changes of LogManagementPolicyExecutionIntervalInMsecs succeeded
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: ExpectedSendTimeThresholdForGeoReplicatnSecs is set to 60.000000
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: Registration for config changes of ExpectedSendTimeThresholdForGeoReplicatnSecs succeeded
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: ExpectedRedoTimeThresholdInMsecs is set to 10000.000000
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: Registration for config changes of ExpectedRedoTimeThresholdInMsecs succeeded
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: ExpectedSendTimeThresholdInMsecs is set to 10000.000000
3/16/2018 12:16:54 AM	spid172a	Dynamic Configuration: Registration for config changes of ExpectedSendTimeThresholdInMsecs succeeded

Seeding the secondaries

And even system databases get new names – master is replicatedmaster, for example:

12:39 PM	Backup	Log was backed up. Database: replicatedmaster, creation date(time): 2018/03/15(17:1
12:39 PM	spid142	Backup(replicatedmaster): Scheduling incremental checkpoint as a background job
12:39 PM	spid142	[INFO] [CKPT] HkHostDbCbt::RegisterHkTruncationLsnChange(): Database ID: [32763
12:39 PM	spid142	Backup(replicatedmaster): Writing the end of backup set
12:39 PM	spid142	Backup(replicatedmaster): Writing the trailing metadata
12:39 PM	spid142	Backup(replicatedmaster): Copying transaction log is complete
12:39 PM	spid142	Backup(replicatedmaster): 100 percent (57344/57344 bytes) processed
12:39 PM	spid142	Backup(replicatedmaster): Copying transaction log
12:39 PM	spid142	Backup(replicatedmaster): Writing the leading metadata
12:39 PM	spid142	Backup(replicatedmaster): Last LSN: 219:2080:1
12:39 PM	spid142	Backup(replicatedmaster): Work estimation is complete
12:39 PM	spid142	Backup(replicatedmaster): Estimated total size = 57344 bytes (data size = 0 bytes, log si
12:39 PM	spid142	Backup(replicatedmaster): First LSN: 219:1968:1
12:39 PM	spid142	Backup(replicatedmaster): Start LSN: 219:1912:3, SERepl LSN: 0:0:0
12:39 PM	spid142	Backup(replicatedmaster): Effective options: Checksum=1, Compression=1, Encryption=
12:39 PM	spid142	Backup(replicatedmaster): The media set is ready for backup
12:39 PM	spid142	Backup(replicatedmaster): Preparing the media set for writing
12:39 PM	spid142	Backup(replicatedmaster): The backup media set is open
12:39 PM	spid142	Backup(replicatedmaster): Opening the backup media set
12:39 PM	spid142	Backup(replicatedmaster): Skipped SafeCheckpoint
12:39 PM	spid142	Backup(replicatedmaster): Synchronizing with other operations on the database is comp
12:39 PM	spid142	Backup(replicatedmaster): Acquiring bulk-op lock on the database
12:39 PM	spid142	Backup(replicatedmaster): Acquiring S lock on the database
12:39 PM	spid142	Backup(replicatedmaster): BACKUP LOG started

Replicatedmaster

And they really do mean replicated – as of this writing, you can create tables in master, load them full of data, and get your instance into trouble on allocated disk space. (You get 32GB by default, and master is included in that number.)

Want to go from cloud to on-premises?

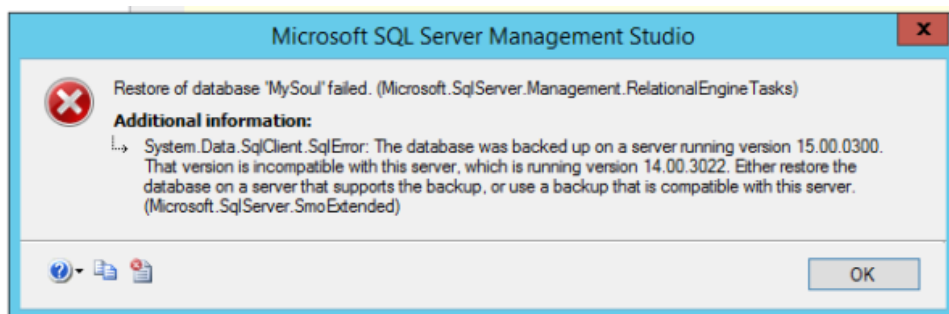
You can go in the other direction, too: since both your on-premises SQL Server and Azure SQL Managed Instances can access the same Azure Blob Storage, you can take a copy-only backup in Azure SQL Managed Instance:

Transact-SQL

- 1 BACKUP DATABASE MySoul
- 2 TO URL = 'https://mountchimborazo.blob.core.windows.net/sqlbackup/MySoul.bak'
- 3 WITH COPY_ONLY;

(This syntax requires me to have already created the credential as shown at the start of this post, a SHARED ACCESS SIGNATURE credential pointing to this URL.)

Then back on-premises, just do a restore pointing to that file and...awww....



SadTrombone.com

That's fair – MIs are a newer version of SQL Server than what we have on-premises. I wouldn't be surprised if we got access to a FeatureSwitch to enable exporting data easier at some point in the future, though – Lord knows there's enough backup feature switches in sys.dm_hadr_fabric_config_parameters:

```
SELECT * FROM sys.dm_hadr_fabric_config_parameters
WHERE section_name LIKE '%backup%'
OR parameter_name LIKE '%backup%';
```

100 %

Results Messages

	section_name	parameter_name	value	is_enc
183	FeatureSwitches	DiffBackupWatchdog	on	0
184	FeatureSwitches	DifferentTroubleshootingEventsForBackupRestore	on	0
185	FeatureSwitches	DisableBackupCleanup	off	0
186	FeatureSwitches	DisableBackupsWhenLogNearFull	off	0
187	FeatureSwitches	DoLogBackupIfEmptyAndTruncationHoldupIsLogBackup	off	0
188	FeatureSwitches	DoNotUpdateLastLogBackupAttemptedTimeWhenBac...	on	0
189	FeatureSwitches	DoubleCheckLatestLogBackupTime	on	0
190	FeatureSwitches	DropDatabaseBeforeRestoreFullBackupForCMS	off	0
191	FeatureSwitches	EnableBackupSizeInRecoveryPointMetaData	on	0
192	FeatureSwitches	EnableBackupThroughPhysicalMasterForCloudLifter	on	0
193	FeatureSwitches	EnableBackupWatchdogUtilCacheCredentialProvider	off	0
194	FeatureSwitches	EnableCleanupSeparateBackupMetadataContainerAnd...	off	0
195	FeatureSwitches	EnableDynamicBackupStripeCount	off	0
196	FeatureSwitches	EnableLTRStampBackupBeforeCopy	on	0
197	FeatureSwitches	EnableLTRStartCopyFromNewBackup	off	0
198	FeatureSwitches	EnableLTRUpdateBackupMetadataIfMissing	on	0
199	FeatureSwitches	EnableMarkFirstBackupTime	on	0
200	FeatureSwitches	EnableParallelFullBackups	off	0
201	FeatureSwitches	EnablePerSLOBackupStripes	on	0
202	FeatureSwitches	EnablePerSLODiffBackupInterval	on	0
203	FeatureSwitches	EnablePhysicalSeparateBackupMetadataContainerAnd...	off	0
204	FeatureSwitches	EnableSeparateBackupMetadataContainerAndBackup...	off	0
205	FeatureSwitches	EnableSeparateBackupWatchdogMetadataContainerA...	off	0
206	FeatureSwitches	ExternalBackupRestoreGeoDrEngine	on	0
207	FeatureSwitches	ExternalBackupRestoreV2Enabled	on	0
208	FeatureSwitches	FetchBackupMetadataUpfront	off	0
209	FeatureSwitches	FixCopyLogBackupToPartialSet	off	0
210	FeatureSwitches	ForceFullBackupForBrokenLogChain	on	0
211	FeatureSwitches	ForceFullBackupWhenNotExists	on	0
212	FeatureSwitches	FullBackupRetryBackoff	on	0
213	FeatureSwitches	FullBackupStaggerUsingLogicalDatabaseId	off	0
214	FeatureSwitches	GeoRestoreBackupSearchInSecondaryFirst	on	0
215	FeatureSwitches	GeoRestoreBackupSearchShouldNotLookInPrimary	off	0

DBCC WITHOUTATRAACE

But that's a story for another post, and [GroupBy](#) is about to start.

Speaking of feature switches, [what trace flags are in use on Managed Instances?](#)