

Distributed AG DR Failover & Drill Runbook

Template: Permanent DR Failover, Drill, and Failback Checklist (On-Prem to Azure over ExpressRoute)

This is a sanitized template for the permanent-DR pattern described in the companion article *“Distributed AG On-Prem to Azure over ExpressRoute”* on powershelldb.de/blog. All server names, AG names, IP addresses, circuit names, and account names have been replaced with generic placeholders from a sample environment. Copy this document, fill in your own environment, and walk through it phase by phase during a scheduled drill or a real DR event.

This is the **failover/drill/failback** counterpart to the *CutOver Runbook Template* published for one-time migrations: use that one if you are decommissioning the on-prem cluster, use this one if the on-prem cluster stays production and Azure is standing DR.

Date: _____ Drill window: _____ – _____

Type of exercise: ☐ Scheduled DR Drill ☐ Real Failover (disaster)

Responsible DBA: _____ Network/ExpressRoute Team: _____ App Team: _____

System Overview (Sample Environment)

Role	Server / Object Name	Remark
On-prem AG primary	ONPREM01	Site A
On-prem AG secondary	ONPREM02	Site B, synchronous
On-prem AG name	AG_OnPrem	Stays production after DR
Azure AG primary	AZVM01	Azure Central US
Azure AG secondary	AZVM02	Azure Central US, synchronous
Azure AG name	AG_Azure	Standby DR leg
Distributed AG	DAG_DR	Async by design; connects both clusters
ExpressRoute circuit	ER-Circuit-01	Private peering, on-prem ↔ Azure vNet
On-prem listener	LST_OnPrem	Moves with the on-prem primary
Azure listener	LST_Azure	Moves with the Azure primary
Service account	CONTOSO\SqlServiceAccount	

Role	Server / Object Name	Remark
Target RPO	_____ minutes	Agreed with the business, not with the circuit
Target RTO	_____ minutes	Decision-to-app-validated, not just the failover command

Prerequisites (verify before every drill or real failover)

- ☐ ExpressRoute circuit ER-Circuit-01: Provider status = Provisioned, both BGP sessions Up
- ☐ AG_Azure: all databases show SYNCHRONIZED, or log_send_queue_size within the agreed RPO budget
- ☐ ONPREM01 confirmed as the current on-prem primary and DAG forwarder
- ☐ Network/ExpressRoute team notified and on standby
- ☐ App team notified and on standby
- ☐ Failback plan (this document, Phase 6) reviewed by whoever is executing
- ☐ For a drill: change approval obtained. For a real event: incident ticket opened

Phase 1: Pre-Failover Health Check (DBA, on ONPREM01)

#	Executor	Action / T-SQL	OK
1	DBA	ONPREM01: SELECT AGs.name, DB_NAME(database_id) AS DBName, synchronization_state_desc, log_send_queue_size, last_hardened_lsn FROM sys.dm_hadr_database_replica_states replicaStates JOIN sys.availability_groups AGs ON AGs.group_id = replicaStates.group_id WHERE (is_distributed = 1) OR (is_primary_replica = 1) <i>-- ALL databases: SYNCHRONIZED, log_send_queue_size</i> <i>at or below RPO budget</i>	☐
2	DBA / Network	Check ExpressRoute circuit health in the Azure Portal (circuit + peering status), or run Get-sqmDistributedAgHealth (sqmSQLTool) for a combined queue + circuit view	☐
3	DBA	Record log_send_queue_size now: this is your realistic RPO estimate if this becomes a real, unplanned failover	☐

Phase 2: Confirm the On-Prem Forwarder (DBA, on ONPREM01/02)

The Distributed AG always fails over from whichever replica is currently the on-prem primary. If ONPREM02 is primary instead of ONPREM01, fail over internally first; do not skip straight to Phase 3 from the wrong node.

#	Executor	Action / T-SQL	OK
4	DBA	<i>Only if ONPREM02 is currently primary, run on</i> <i>ONPREM02:</i> ALTER AVAILABILITY GROUP [AG_OnPrem] FAILOVER;	☐

#	Executor	Action / T-SQL	OK
5	DBA	<pre>ONPREM01: SELECT ars.role_desc FROM sys.dm_hadr_availability_replica_states ars JOIN sys.availability_replicas ar ON ar.replica_id = ars.replica_id WHERE ars.is_local = 1 -- Expected: PRIMARY</pre>	☐

Phase 3: Switch to Synchronous & Fail Over the DAG to Azure (DBA)

CRITICAL: After this step, AZVM01 is the new primary. The on-prem side becomes the DR leg. Confirm you are running the right sub-step for a drill (on-prem reachable) versus a real disaster (on-prem unreachable) before you execute.

#	Executor	Action / T-SQL	OK
6	DBA	<pre>ONPREM01 (forwarder), planned drill only, on-prem reachable: ALTER AVAILABILITY GROUP [DAG_DR] MODIFY AVAILABILITY GROUP ON 'AG_OnPrem' WITH (AVAILABILITY_MODE = SYNCHRONOUS_COMMIT), 'AG_Azure' WITH (AVAILABILITY_MODE = SYNCHRONOUS_COMMIT); -- Switch to SYNC before a planned failover, exactly like a local AG</pre>	☐
7	DBA	<pre>Re-run the Phase 1 sync-state query. Wait until synchronization_state_desc = SYNCHRONIZED and log_send_queue_size = 0 before continuing.</pre>	☐
8	DBA	<pre>ONPREM01, planned drill, after confirmed SYNCHRONIZED: ALTER AVAILABILITY GROUP [DAG_DR] SET (ROLE = SECONDARY);</pre>	☐
9	DBA	<pre>AZVM01, planned drill: ALTER AVAILABILITY GROUP [DAG_DR] FAILOVER; AZVM01, REAL DISASTER, on-prem unreachable ONLY: ALTER AVAILABILITY GROUP [DAG_DR] FORCE_FAILOVER_ALLOW_DATA_LOSS; -- Possible data loss up to the last recorded log_send_queue_size (Phase 1, step 3)</pre>	☐
10	DBA	<pre>AZVM01: SELECT ars.role_desc FROM sys.dm_hadr_availability_replica_states ars JOIN sys.availability_replicas ar ON ar.replica_id = ars.replica_id WHERE ars.is_local = 1 -- Expected: PRIMARY</pre>	☐

Phase 4: Application Cutover to Azure (App Team + Network Team)

#	Executor	Action / T-SQL	OK
11	Network	Repoint DNS / connection string from LST_OnPrem to LST_Azure (or update the CNAME if apps connect through a single logical name)	☐
12	App Team	ipconfig /flushdns on all relevant servers and clients	☐
13	App Team	Start applications, perform functional test on all critical workflows	☐
14	DBA	Record actual failover time and any observed data loss in the DR Drill Log below: this is your real RTO/RPO, not the target	☐

DONE: Azure is now primary. Note the time: _____

Phase 5: Confirm DR Posture

- ☐ If this was a DRILL: proceed to Phase 6 (failback) in the same maintenance window, or schedule it explicitly; do not leave Azure as primary indefinitely by accident.
- ☐ If this was a REAL event: Azure is now production. Decide when on-prem becomes available again whether to rebuild AG_OnPrem as the new DAG secondary, or to keep Azure permanent and stand up a new DR leg elsewhere.

Phase 6: Failback to On-Prem (once ONPREM01/02 are available and caught up)

#	Executor	Action / T-SQL	OK
15	DBA	Confirm ONPREM01/02 are online, AG_OnPrem is healthy, and rejoin it to DAG_DR as the secondary if it was ever removed	☐
16	DBA	AZVM01 (current forwarder): ALTER AVAILABILITY GROUP [DAG_DR] MODIFY AVAILABILITY GROUP ON 'AG_OnPrem' WITH (AVAILABILITY_MODE = SYNCHRONOUS_COMMIT), 'AG_Azure' WITH (AVAILABILITY_MODE = SYNCHRONOUS_COMMIT);	☐
17	DBA	Wait for AG_OnPrem to reach SYNCHRONIZED with log_send_queue_size = 0 (Phase 1 query, run from AZVM01's perspective)	☐
18	DBA	AZVM01: ALTER AVAILABILITY GROUP [DAG_DR] SET (ROLE = SECONDARY); ONPREM01: ALTER AVAILABILITY GROUP [DAG_DR] FAILOVER;	☐
19	Network	Repoint DNS / connection string back to LST_OnPrem, flush DNS on clients	☐
20	App Team	Validate applications against on-prem primary	☐
21	DBA	ONPREM01 (forwarder), return to steady-state DR posture: ALTER AVAILABILITY GROUP [DAG_DR] MODIFY AVAILABILITY GROUP ON 'AG_OnPrem' WITH (AVAILABILITY_MODE = ASYNCHRONOUS_COMMIT), 'AG_Azure' WITH (AVAILABILITY_MODE = ASYNCHRONOUS_COMMIT); <i>-- The DAG Link is asynchronous by product design; this just restores the steady state</i>	☐

Abort a Drill Before Cutover

If you abort before step 8 (before setting ONPREM01 to SECONDARY): no action needed. Switch the availability mode back to ASYNCHRONOUS_COMMIT (same syntax as Phase 6, step 21) and stand down.

If you abort after step 8 but before step 9 completes on Azure: ONPREM01 is SECONDARY with no primary yet. Re-run ALTER AVAILABILITY GROUP [DAG_DR] FAILOVER; on ONPREM01 to reclaim the primary role, then switch back to async commit.

DR Drill / Failover Log

Date	Type	RTO Achieved	RPO Achieved (data loss)	Issues	Signed off by
	☐ Drill ☐ Real				
	☐ Drill ☐ Real				
	☐ Drill ☐ Real				
	☐ Drill ☐ Real				

Recommended cadence: quarterly drill at minimum; more often if ExpressRoute capacity, workload, or the RPO/RTO target has changed since the last one.

Common Issues & Troubleshooting

ExpressRoute circuit degraded or down during the drill

Cause: provider-side maintenance, BGP flap, or a fiber cut. Solution: check circuit + peering status in the Azure Portal before blaming SQL Server; if a VPN failover path is configured for the circuit, confirm it engaged. Do not force a failover on a stale, unreachable link unless the RTO SLA requires it; forcing does not fix the network.

log_send_queue_size never reaches zero before the drill window

Cause: sustained write load exceeds what the ExpressRoute circuit can drain in real time. Solution: re-baseline the RPO target around the observed steady-state queue rather than assuming zero, or size up the circuit; do not force a failover through a large queue just to keep the schedule.

Failover succeeds but applications cannot connect to Azure

Cause: SPNs for LST_Azure were never registered, or are duplicated on the old computer account. Solution: run Get-sqmSpnReport (sqmSQLTool) against the Azure listener and instances, apply the generated setspn commands, then have clients re-authenticate.

Distributed AG not visible on Azure after a real, forced failover

Cause: the on-prem side was unreachable when FORCE_FAILOVER_ALLOW_DATA_LOSS ran, so it never received the role change. Solution: this is expected. Once on-prem is back, Phase 6 rejoins it as a fresh secondary; do not attempt to reconcile the two sides manually.

TIP (sqmSQLTool): Test-sqmDistributedAgReadiness checks sync state and queue size before you touch Phase 3. Get-sqmDistributedAgHealth gives an ongoing combined view of the DAG queue and (where exposed) the ExpressRoute circuit, worth scheduling as a recurring job outside of drill windows. Invoke-sqmDistributedFailover runs the Phase 3 sequence for both drills and real events instead of typing the ALTER statements under pressure. Get-sqmSpnReport covers Phase 4, step 11's Kerberos risk ahead of time.

Appendix: RPO/RTO Worksheet

Use this after each drill or real event to track whether your actual numbers are converging on the agreed targets, or drifting away from them as workload grows.

Date	log_send_queue_size at failover (KB)	Estimated data loss	Decision-to-validated time (RTO)	Notes

How This Differs from the Migration CutOver Runbook

The companion *CutOver Runbook Template* (same downloads folder) walks a one-time migration: seed, catch up, fail over once, delete the old AG and the Distributed AG, decommission the old cluster. This document assumes the opposite: the on-prem cluster stays production indefinitely, Azure is standing DR, and the Distributed AG is never torn down. It fails over, fails back, and returns to steady-state async on a recurring basis. Use the migration runbook if you're retiring on-prem; use this one if you're not.