IDUG

IDUG
2022 NA **Db2** Tech Conference

Demystifying Recovery - Top Questions From Customers Answered

Michael Roecken, IBM

Session Code: C11

Db2 (LUW)

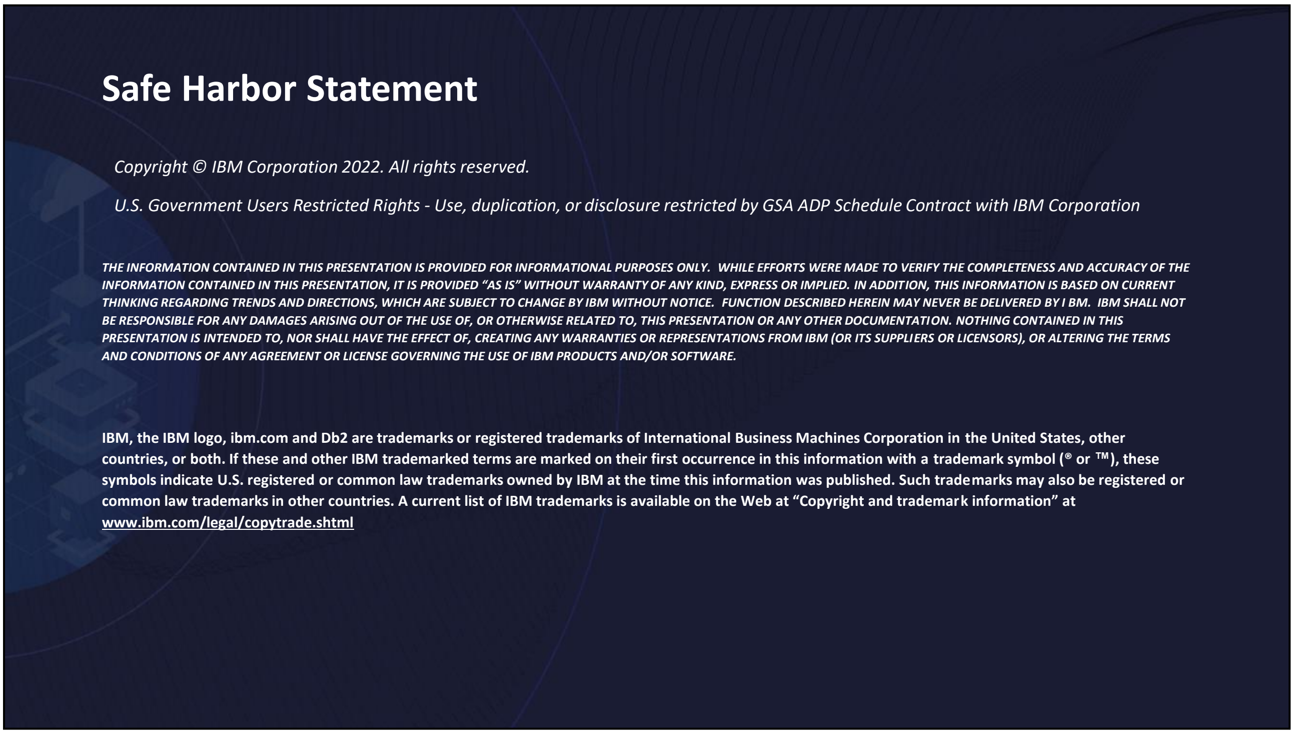
Boston, MA

Thinking about your availability and recovery strategy shouldn't give you sleepless nights. When issues do come up, you want to make sure you have the tools needed to solve problems and ensure you get the most out of your investment in Db2. This presentation will give you some context and insight into the top questions we see from our customers the most in the areas of backup, restore, crash recovery, locking, logging, transaction management and high availability and disaster recovery. The questions and answers discussed here could save you when an unexpected outage arises and/or better prepare you to avoid an outage in the future.

This presentation is valid as of 11.5.7.0.

Michael Roecken is a senior software developer with Db2 for Linux, UNIX, and Windows platforms at the IBM Toronto Lab. Michael has worked since 2000 designing, implementing and supporting various features and capabilities in the areas of: backup/restore, crash/rollforward recovery, high availability/disaster recovery, and logging/transaction management.

Please connect with Michael on Twitter at @roecken and LinkedIn.



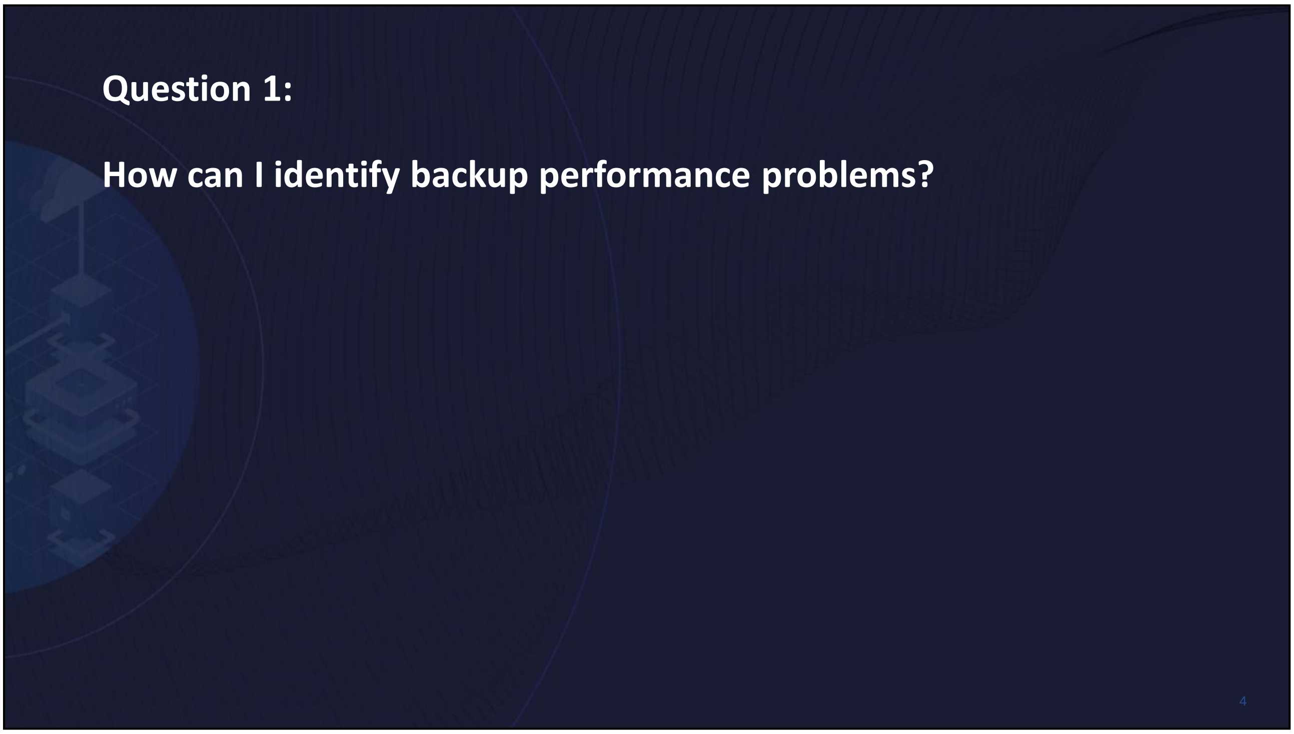
Safe Harbor Statement

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON CURRENT THINKING REGARDING TRENDS AND DIRECTIONS, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. FUNCTION DESCRIBED HEREIN MAY NEVER BE DELIVERED BY I BM. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS AND/OR SOFTWARE.

Agenda

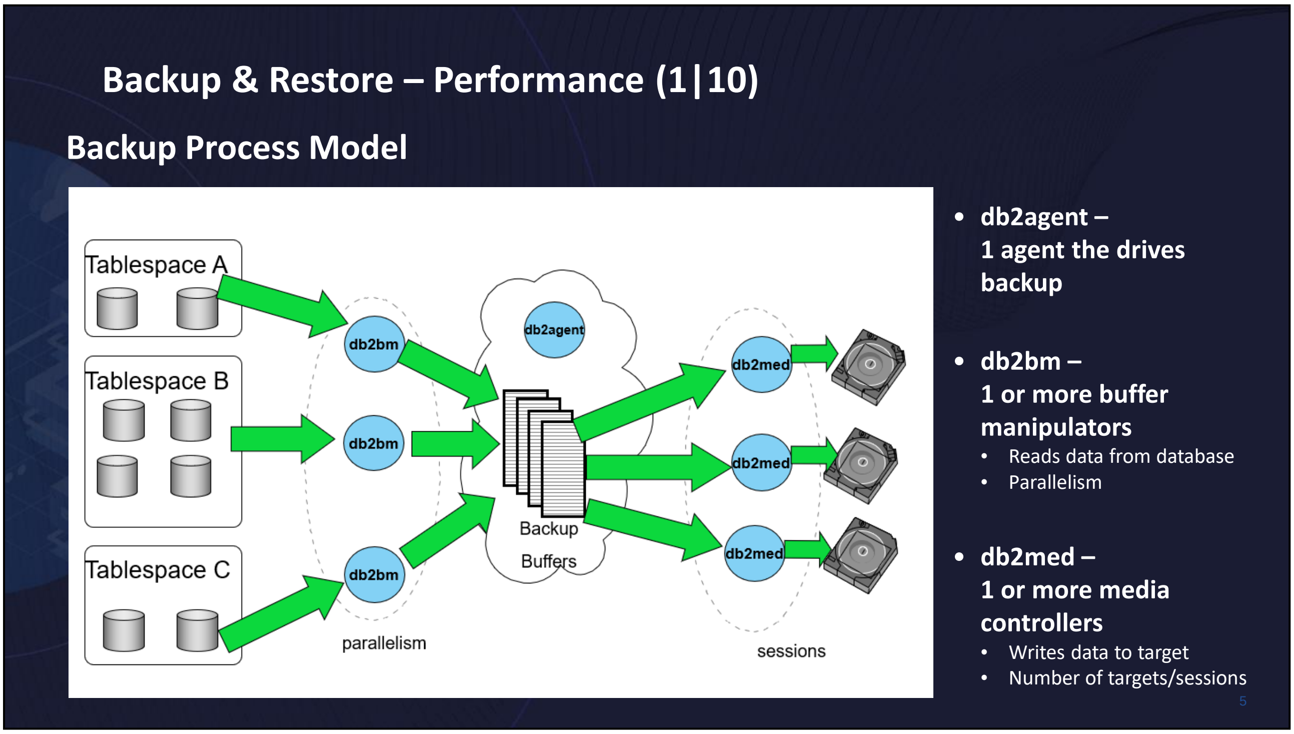
- Backup & Restore
  - Performance
  - Object (Remote) Storage
- Logging
  - Transaction Log Full
  - Disk Full in Archives
- Recovery
  - Point-in-time Recovery
  - Inaccessible Table Spaces
  - Monitor Crash Recovery

- Upgrade



Question 1:

How can I identify backup performance problems?



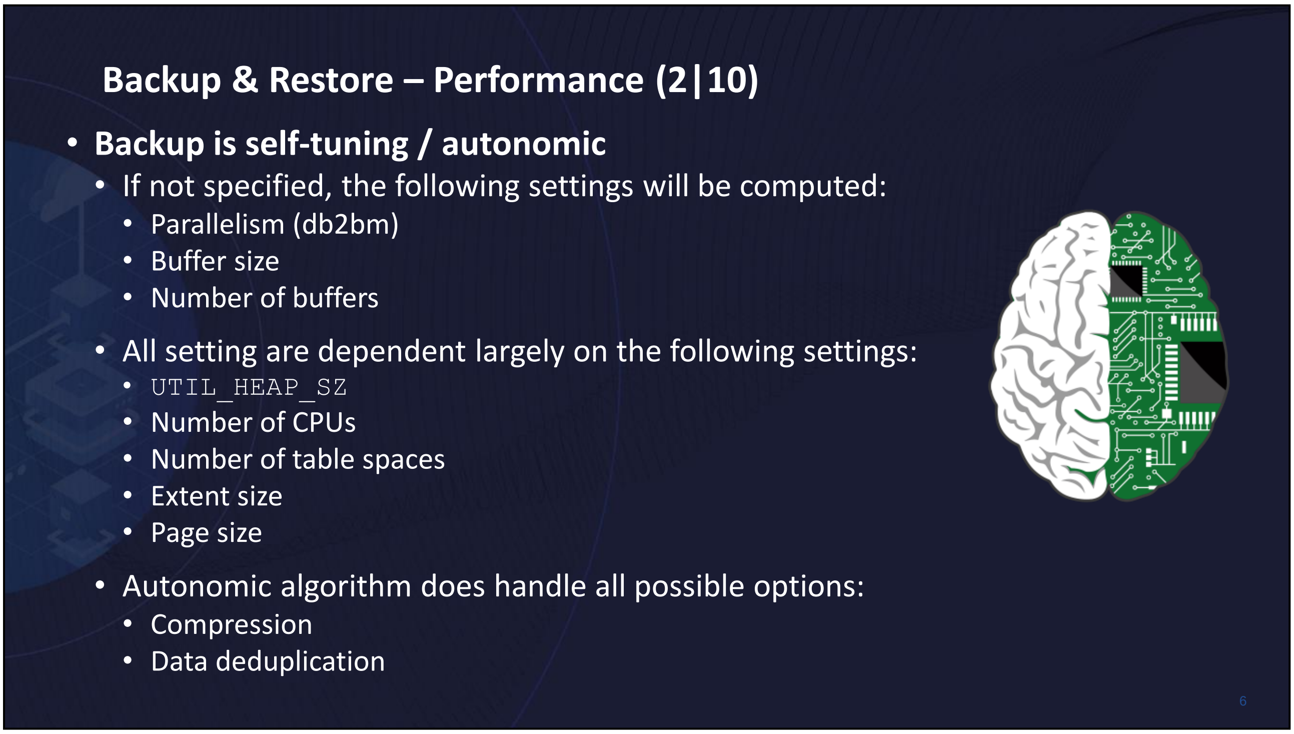
Backup & Restore – Performance (1|10)

Backup Process Model

- **db2agent –**
  1 agent the drives backup
- **db2bm –**
  1 or more buffer manipulators
  - Reads data from database
  - Parallelism
- **db2med –**
  1 or more media controllers
  - Writes data to target
  - Number of targets/sessions

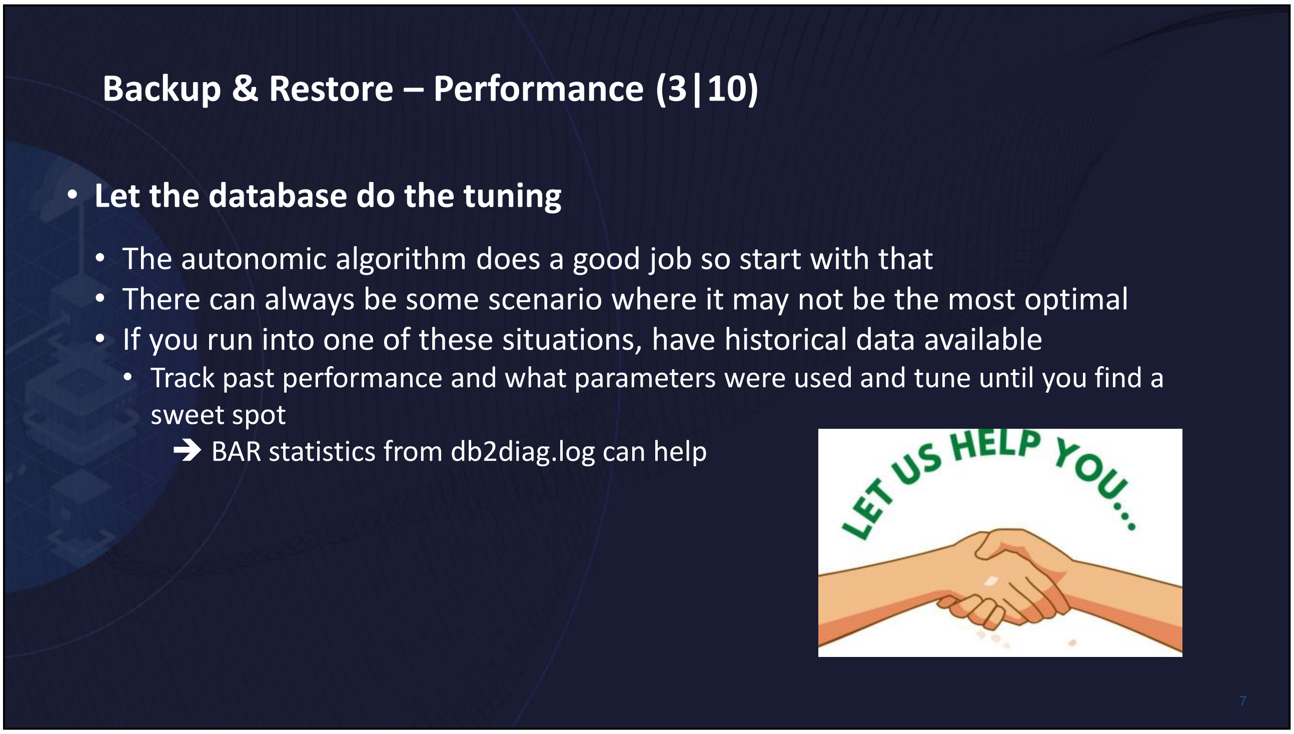
BACKUP DATABASE comma

- https://www.ibm.com/docs/en/db2/11.5?topic=commands-backup-database



Backup & Restore – Performance (2|10)

- **Backup is self-tuning / autonomic**
  - If not specified, the following settings will be computed:
    - Parallelism (db2bm)
    - Buffer size
    - Number of buffers
  - All setting are dependent largely on the following settings:
    - `UTIL_HEAP_SZ`
    - Number of CPUs
    - Number of table spaces
    - Extent size
    - Page size
  - Autonomic algorithm does handle all possible options:
    - Compression
    - Data deduplication



Backup & Restore – Performance (3|10)

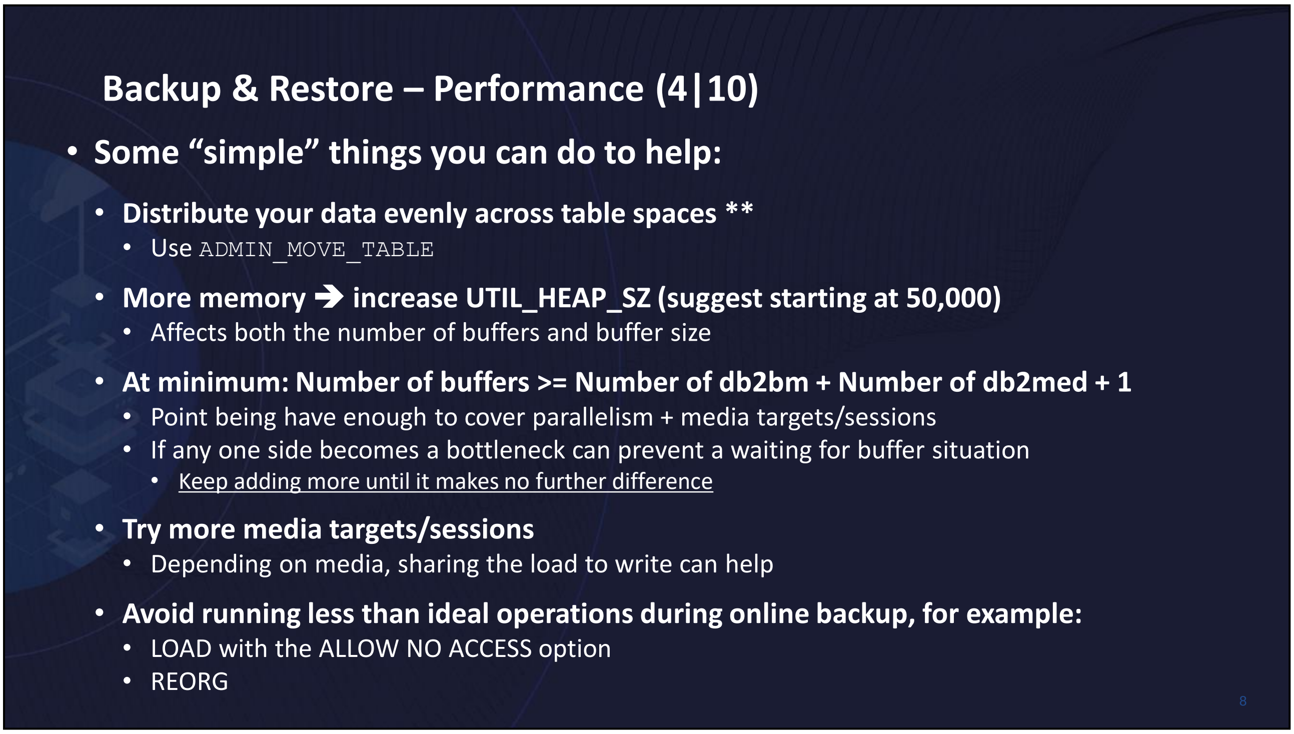
- **Let the database do the tuning**
  - The autonomic algorithm does a good job so start with that
  - There can always be some scenario where it may not be the most optimal
  - If you run into one of these situations, have historical data available
    - Track past performance and what parameters were used and tune until you find a sweet spot

      ➔ BAR statistics from db2diag.log can help

** In the future this may be less of a need due to Intra Tablespace Parallelism feature

Optimizing backup performance

- https://www.ibm.com/docs/en/db2/11.5?topic=backup-optimizing-performance



Backup & Restore – Performance (4|10)

- Some "simple" things you can do to help:
  - Distribute your data evenly across table spaces **
    - Use `ADMIN_MOVE_TABLE`
  - More memory ➔ increase UTIL_HEAP_SZ (suggest starting at 50,000)
    - Affects both the number of buffers and buffer size
  - At minimum: Number of buffers >= Number of db2bm + Number of db2med + 1
    - Point being have enough to cover parallelism + media targets/sessions
    - If any one side becomes a bottleneck can prevent a waiting for buffer situation
      - Keep adding more until it makes no further difference
  - Try more media targets/sessions
    - Depending on media, sharing the load to write can help
  - Avoid running less than ideal operations during online backup, for example:
    - LOAD with the ALLOW NO ACCESS option
    - REORG

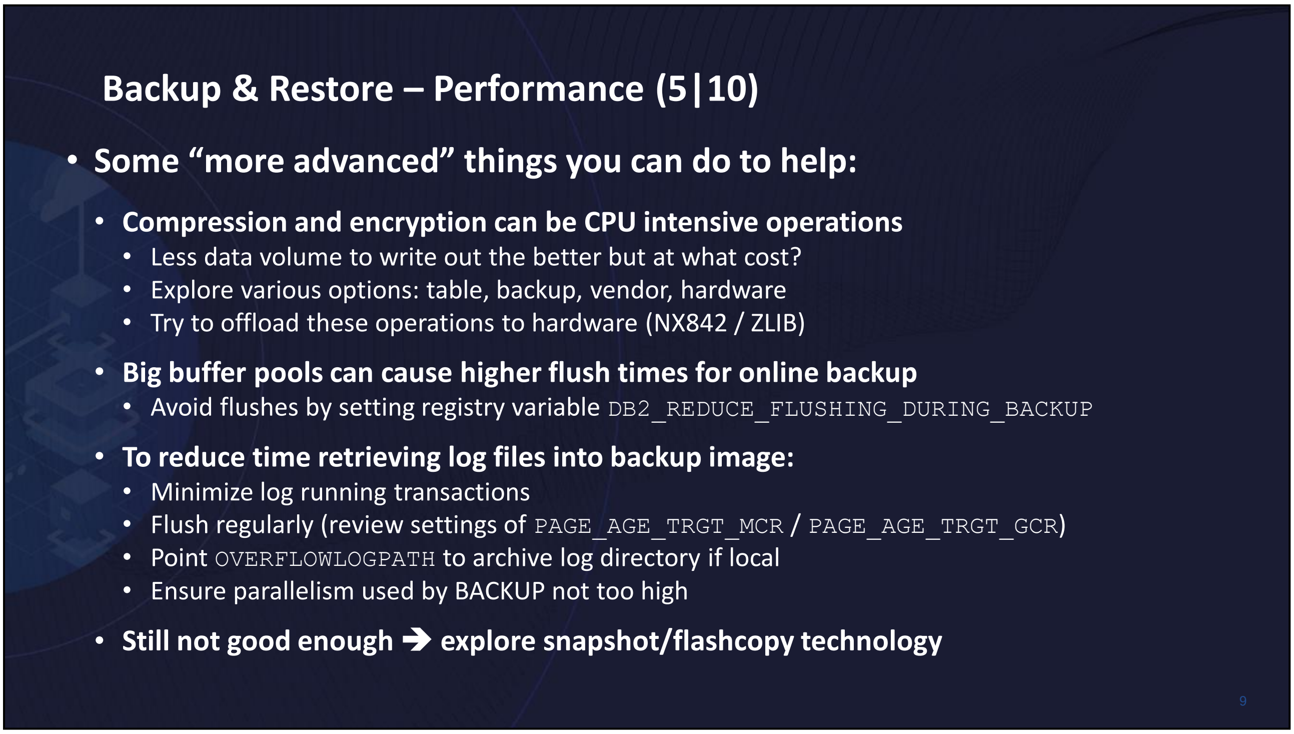
** In the future this may be less of a need due to Intra Tablespace Parallelism feature

Optimizing backup performance

- https://www.ibm.com/docs/en/db2/11.5?topic=backup-optimizing-performance

Compatibility of online backup and other utilities

- https://www.ibm.com/docs/en/db2/11.5?topic=backup-compatibility-online-other-utilities



Backup & Restore – Performance (5|10)

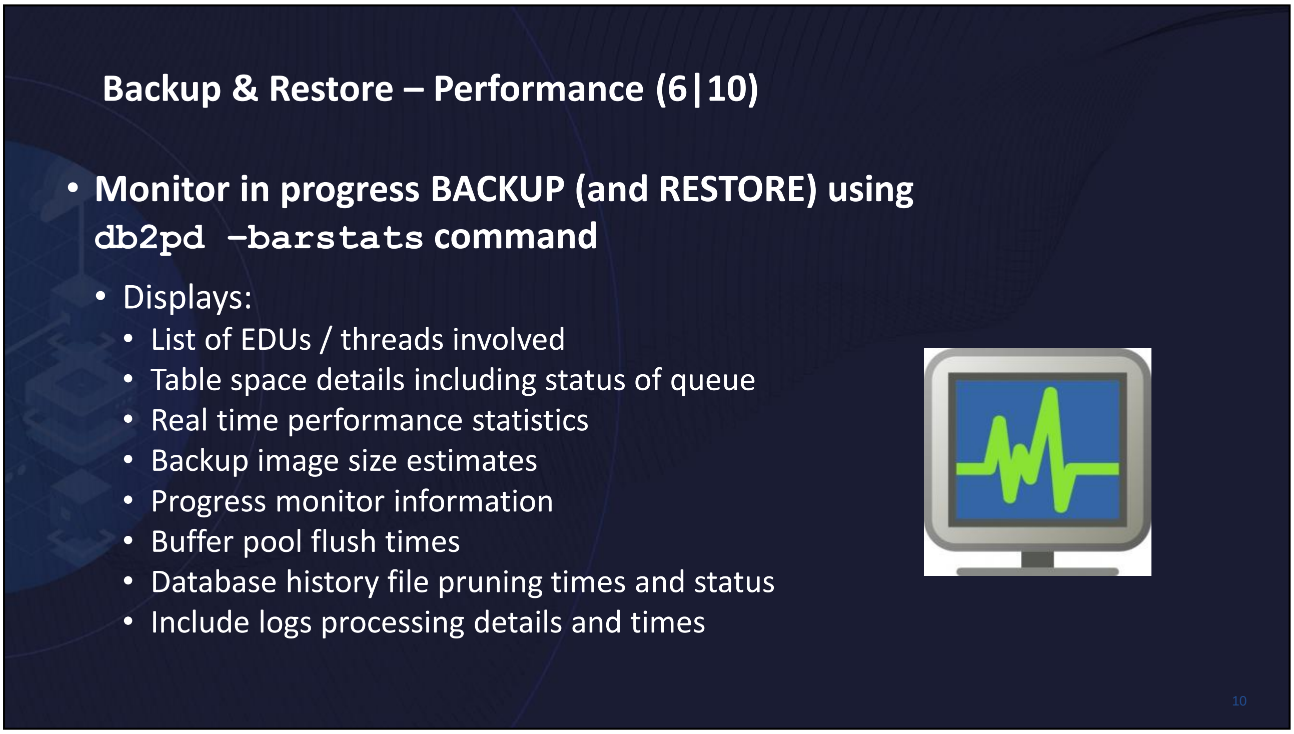
- **Some "more advanced" things you can do to help:**
  - **Compression and encryption can be CPU intensive operations**
    - Less data volume to write out the better but at what cost?
    - Explore various options: table, backup, vendor, hardware
    - Try to offload these operations to hardware (NX842 / ZLIB)
  - **Big buffer pools can cause higher flush times for online backup**
    - Avoid flushes by setting registry variable `DB2_REDUCE_FLUSHING_DURING_BACKUP`
  - **To reduce time retrieving log files into backup image:**
    - Minimize log running transactions
    - Flush regularly (review settings of `PAGE_AGE_TRGT_MCR` / `PAGE_AGE_TRGT_GCR`)
    - Point `OVERFLOWLOGPATH` to archive log directory if local
    - Ensure parallelism used by BACKUP not too high
  - **Still not good enough ➔ explore snapshot/flashcopy technology**

Optimizing backup performance

- https://www.ibm.com/docs/en/db2/11.5?topic=backup-optimizing-performance

Hardware accelerated backup and log file compression

- https://www.ibm.com/docs/en/db2/11.5?topic=considerations-hardware-accelerated-backup-log-file-compression



Backup & Restore – Performance (6|10)

- **Monitor in progress BACKUP (and RESTORE) using `db2pd –barstats` command**
  - Displays:
    - List of EDUs / threads involved
    - Table space details including status of queue
    - Real time performance statistics
    - Backup image size estimates
    - Progress monitor information
    - Buffer pool flush times
    - Database history file pruning times and status
    - Include logs processing details and times

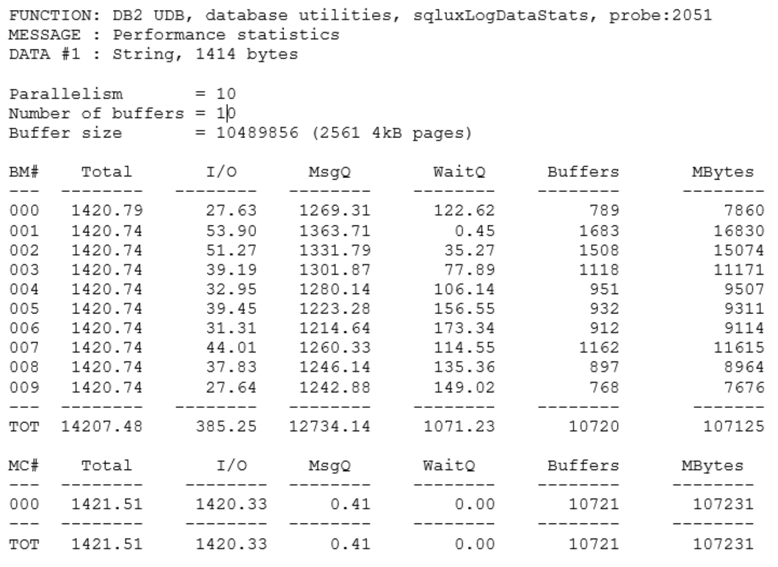
Monitoring backup and restore performance with db2pd -barstats

- https://www.ibm.com/docs/en/db2/11.5?topic=strategy-backup-restore-monitoring-db2pd-barstats
- https://www.ibm.com/docs/en/db2/11.5?topic=barstats-example-output-backup-performance-monitoring#c0070441

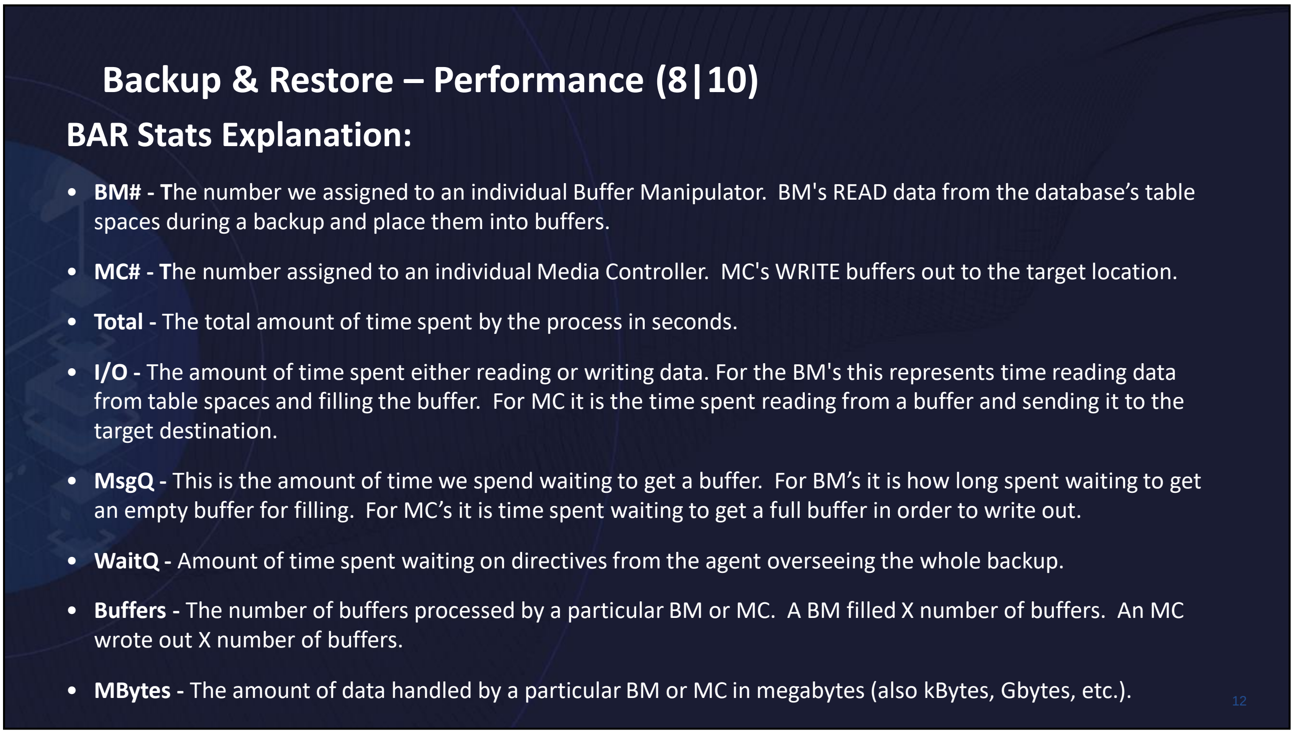
Backup & Restore – Performance (7|10)

BAR Stats Example:

- Backup to vendor
- Parallelism of 10 (BMs)
- 1 session (MC)
  - Media target
- util_heap_sz 30000



```
FUNCTION: DB2 UDB, database utilities, sqluxLogDataStats, probe:2051
MESSAGE : Performance statistics
DATA #1 : String, 1414 bytes

Parallelism       = 10
Number of buffers = 10
Buffer size       = 10489856 (2561 4kB pages)

BM#    Total      I/O      MsgQ      WaitQ      Buffers      MBytes
---  --------  --------  --------  --------   --------    --------
000   1420.79     27.63   1269.31    122.62        789        7860
001   1420.74     53.90   1363.71      0.45       1683       16830
002   1420.74     51.27   1331.79     35.27       1508       15074
003   1420.74     39.19   1301.87     77.89       1118       11171
004   1420.74     32.95   1280.14    106.14        951        9507
005   1420.74     39.45   1223.28    156.55        932        9311
006   1420.74     31.31   1214.64    173.34        912        9114
007   1420.74     44.01   1260.33    114.55       1162       11615
008   1420.74     37.83   1246.14    135.36        897        8964
009   1420.74     27.64   1242.88    149.02        768        7676
---  --------  --------  --------  --------   --------     -------
TOT  14207.48    385.25  12734.14   1071.23      10720      107125

MC#    Total       I/O     MsgQ      WaitQ      Buffers     MBytes
---  --------   --------  --------  --------   --------   --------
000   1421.51    1420.33      0.41      0.00      10721     107231
---  --------   --------  --------  --------   --------   --------
TOT   1421.51    1420.33      0.41      0.00      10721     107231
```

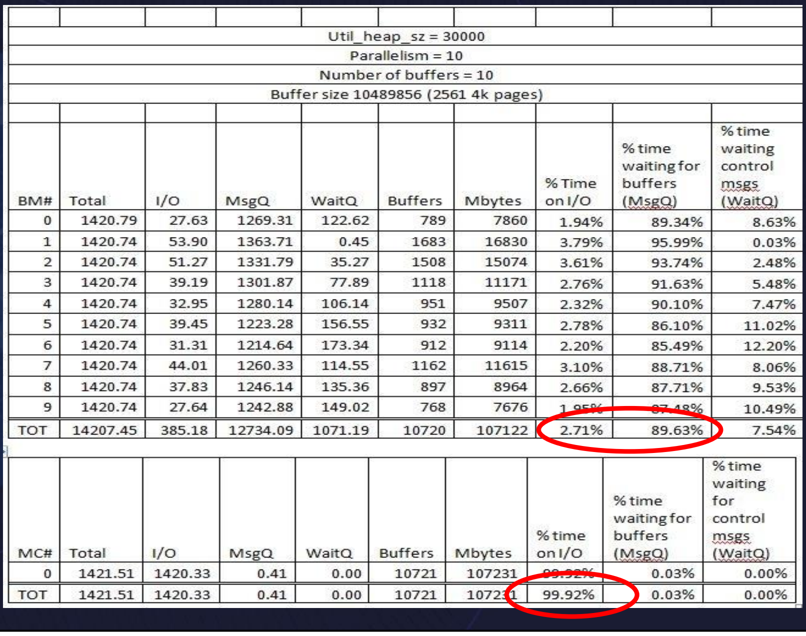


Backup & Restore – Performance (8|10)

BAR Stats Explanation:

- **BM# - T**he number we assigned to an individual Buffer Manipulator. BM's READ data from the database’s table spaces during a backup and place them into buffers.
- **MC# - T**he number assigned to an individual Media Controller. MC's WRITE buffers out to the target location.
- **Total -** The total amount of time spent by the process in seconds.
- **I/O -** The amount of time spent either reading or writing data. For the BM's this represents time reading data from table spaces and filling the buffer. For MC it is the time spent reading from a buffer and sending it to the target destination.
- **MsgQ -** This is the amount of time we spend waiting to get a buffer. For BM’s it is how long spent waiting to get an empty buffer for filling. For MC’s it is time spent waiting to get a full buffer in order to write out.
- **WaitQ -** Amount of time spent waiting on directives from the agent overseeing the whole backup.
- **Buffers -** The number of buffers processed by a particular BM or MC. A BM filled X number of buffers. An MC wrote out X number of buffers.
- **MBytes -** The amount of data handled by a particular BM or MC in megabytes (also kBytes, Gbytes, etc.).

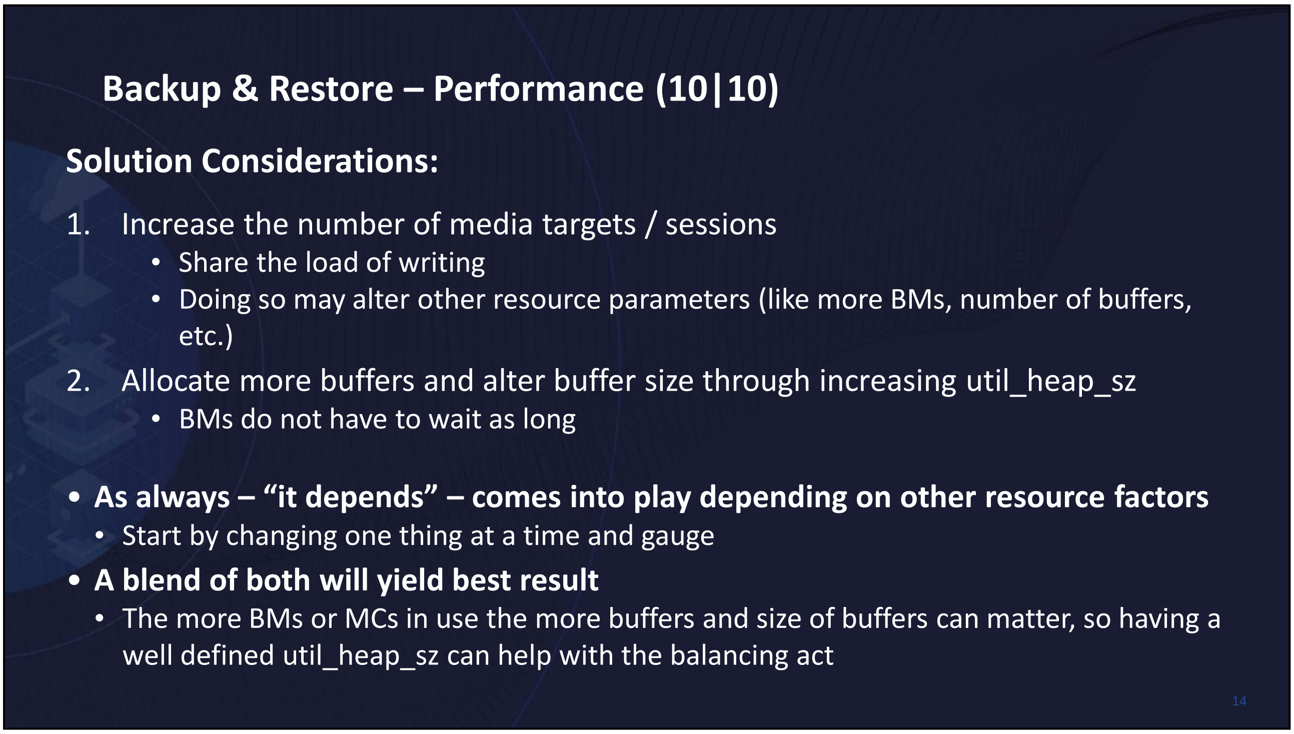
Backup & Restore – Performance (9|10)



Util_heap_sz = 30000									
Parallelism = 10									
Number of buffers = 10									
Buffer size 10489856 (2561 4k pages)									
BM#	Total	I/O	MsgQ	WaitQ	Buffers	Mbytes	% Time on I/O	% time waiting for buffers (MsgQ)	% time waiting control msgs (WaitQ)
0	1420.79	27.63	1269.31	122.62	789	7860	1.94%	89.34%	8.63%
1	1420.74	53.90	1363.71	0.45	1683	16830	3.79%	95.99%	0.03%
2	1420.74	51.27	1331.79	35.27	1508	15074	3.61%	93.74%	2.48%
3	1420.74	39.19	1301.87	77.89	1118	11171	2.76%	91.63%	5.48%
4	1420.74	32.95	1280.14	106.14	951	9507	2.32%	90.10%	7.47%
5	1420.74	39.45	1223.28	156.55	932	9311	2.78%	86.10%	11.02%
6	1420.74	31.31	1214.64	173.34	912	9114	2.20%	85.49%	12.20%
7	1420.74	44.01	1260.33	114.55	1162	11615	3.10%	88.71%	8.06%
8	1420.74	37.83	1246.14	135.36	897	8964	2.66%	87.71%	9.53%
9	1420.74	27.64	1242.88	149.02	768	7676	1.95%	87.48%	10.49%
TOT	14207.45	385.18	12734.09	1071.19	10720	107122	2.71%	89.63%	7.54%

MC#	Total	I/O	MsgQ	WaitQ	Buffers	Mbytes	% time on I/O	% time waiting for buffers (MsgQ)	% time waiting for control msgs (WaitQ)
0	1421.51	1420.33	0.41	0.00	10721	107231	99.92%	0.03%	0.00%
TOT	1421.51	1420.33	0.41	0.00	10721	107231	99.92%	0.03%	0.00%

- BMs:
  - On average spent 2.71% of time waiting to read data from the database
  - Spent 89.63% of the time waiting for buffers
- MCs:
  - On average spent 99.92% of the time waiting for I/O (in this case vendor)
- Conclusion:
  - MC cannot free up a buffer until the vendor confirms it has been written
  - BMs must wait for the MCs to free the buffers
  - Bottleneck is in writing to the target device

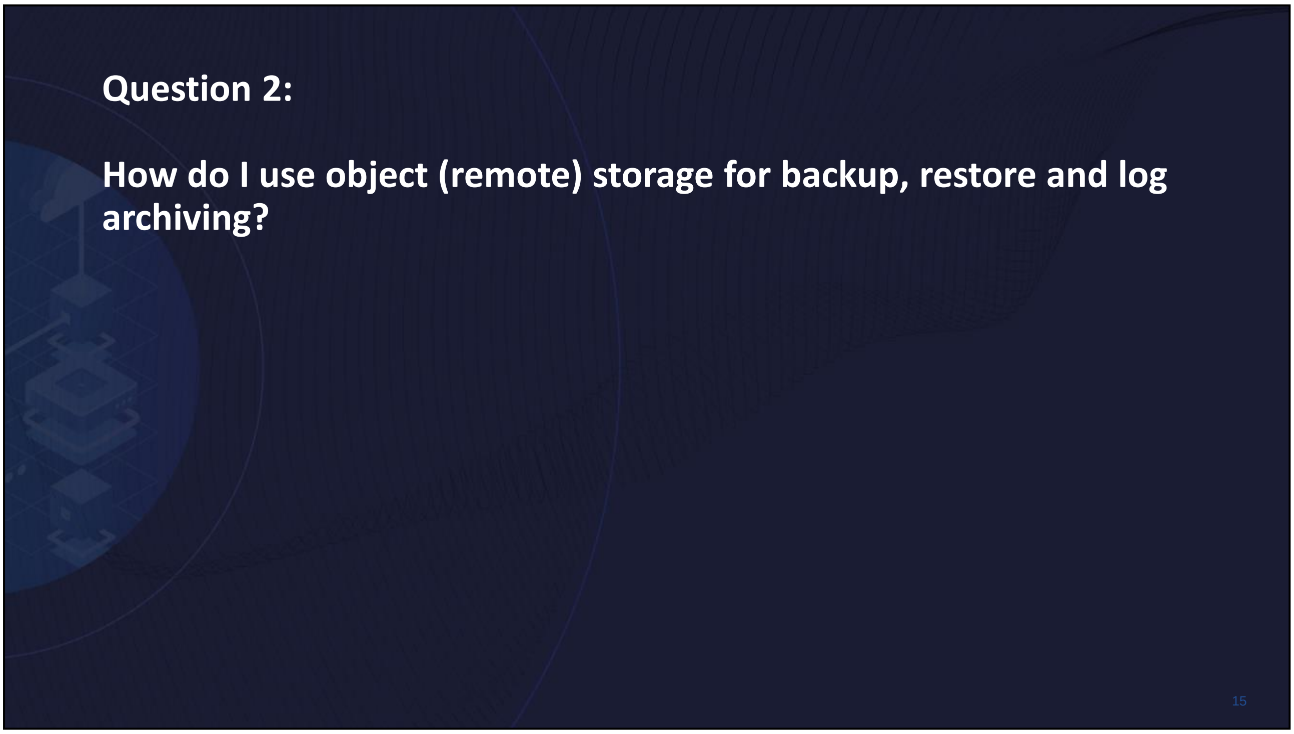


Backup & Restore – Performance (10|10)

Solution Considerations:

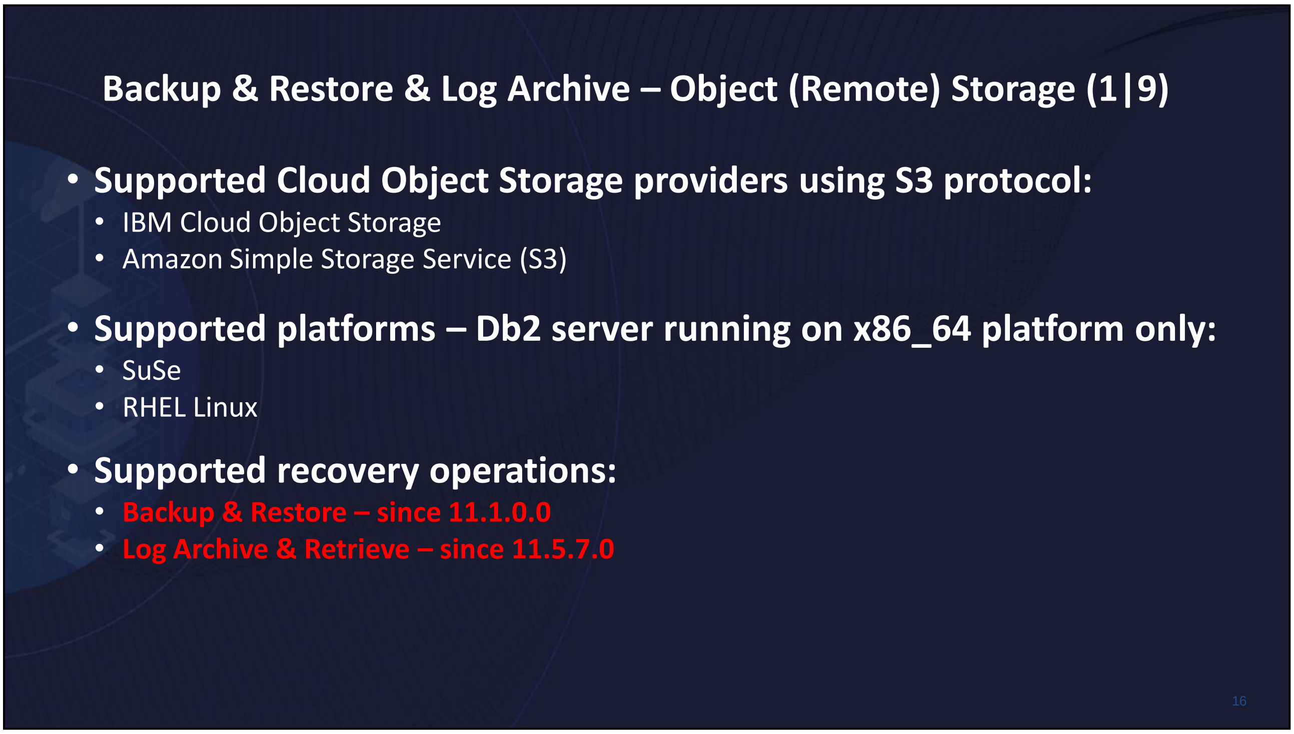
1. Increase the number of media targets / sessions
   - Share the load of writing
   - Doing so may alter other resource parameters (like more BMs, number of buffers, etc.)
2. Allocate more buffers and alter buffer size through increasing util_heap_sz
   - BMs do not have to wait as long

- **As always – “it depends” – comes into play depending on other resource factors**
  - Start by changing one thing at a time and gauge
- **A blend of both will yield best result**
  - The more BMs or MCs in use the more buffers and size of buffers can matter, so having a well defined util_heap_sz can help with the balancing act



Question 2:

How do I use object (remote) storage for backup, restore and log archiving?



Backup & Restore & Log Archive – Object (Remote) Storage (1|9)

- **Supported Cloud Object Storage providers using S3 protocol:**
  - IBM Cloud Object Storage
  - Amazon Simple Storage Service (S3)
- **Supported platforms – Db2 server running on x86_64 platform only:**
  - SuSe
  - RHEL Linux
- **Supported recovery operations:**
  - **Backup & Restore – since 11.1.0.0**
  - **Log Archive & Retrieve – since 11.5.7.0**

Remote storage requirements

- https://www.ibm.com/docs/en/db2/11.5?topic=reference-remote-storage-requirements

CATALOG STORAGE ACCESS command

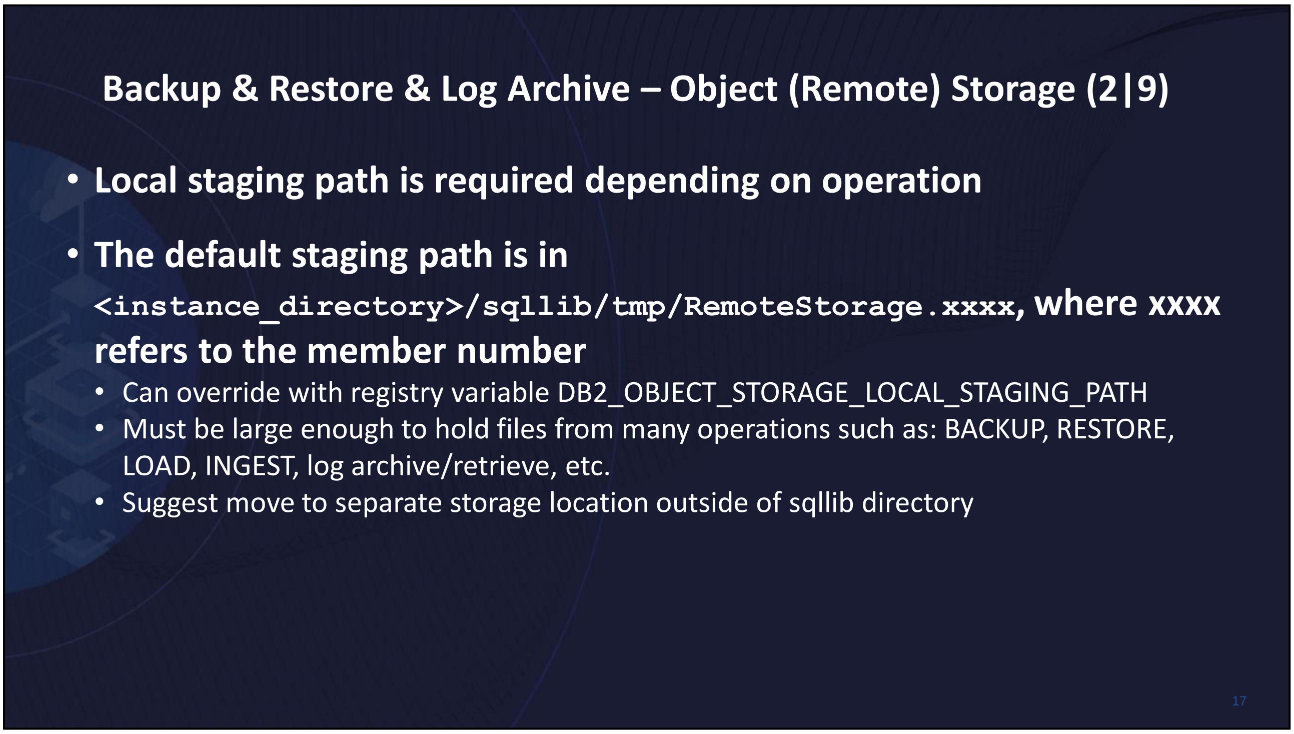
https://www.ibm.com/docs/en/db2/11.5?topic=commands-catalog-storage-access

DB2_ENABLE_COS_SDK

- https://www.ibm.com/docs/en/db2/11.5?topic=variables-miscellaneous#r0005669__M_DB2_ENABLE_COS_SDK

multipartsizemb – Remote storage multipart upload part size configuration parameter

- https://www.ibm.com/docs/en/db2/11.5?topic=parameters-multipartsizemb-remote-storage-multipart-upload-part-size



Backup & Restore & Log Archive – Object (Remote) Storage (2|9)

- **Local staging path is required depending on operation**
- **The default staging path is in `<instance_directory>/sqllib/tmp/RemoteStorage.xxxx`, where xxxx refers to the member number**
  - Can override with registry variable DB2_OBJECT_STORAGE_LOCAL_STAGING_PATH
  - Must be large enough to hold files from many operations such as: BACKUP, RESTORE, LOAD, INGEST, log archive/retrieve, etc.
  - Suggest move to separate storage location outside of sqllib directory

Remote storage requirements

- https://www.ibm.com/docs/en/db2/11.5?topic=reference-remote-storage-requirements

CATALOG STORAGE ACCESS command

https://www.ibm.com/docs/en/db2/11.5?topic=commands-catalog-storage-access

DB2_ENABLE_COS_SDK

- https://www.ibm.com/docs/en/db2/11.5?topic=variables-miscellaneous#r0005669__M_DB2_ENABLE_COS_SDK

multipartsizemb – Remote storage multipart upload part size configuration parameter

- https://www.ibm.com/docs/en/db2/11.5?topic=parameters-multipartsizemb-remote-storage-multipart-upload-part-size

Backup & Restore & Log Archive – Object (Remote) Storage (3|9)

- **DB2_ENABLE_COS_SDK = OFF**
  - Communication uses old legacy libcurl method
  - Requires local staging path to hold temporary files and has limitations on file sizes and lack of support for streaming multipart uploads
  - Should not set as registry variable is deprecated

  - **BACKUP**: each session mas maximum image size of 5GB up to a total of 5TB

18

Remote storage requirements

- https://www.ibm.com/docs/en/db2/11.5?topic=reference-remote-storage-requirements

CATALOG STORAGE ACCESS command

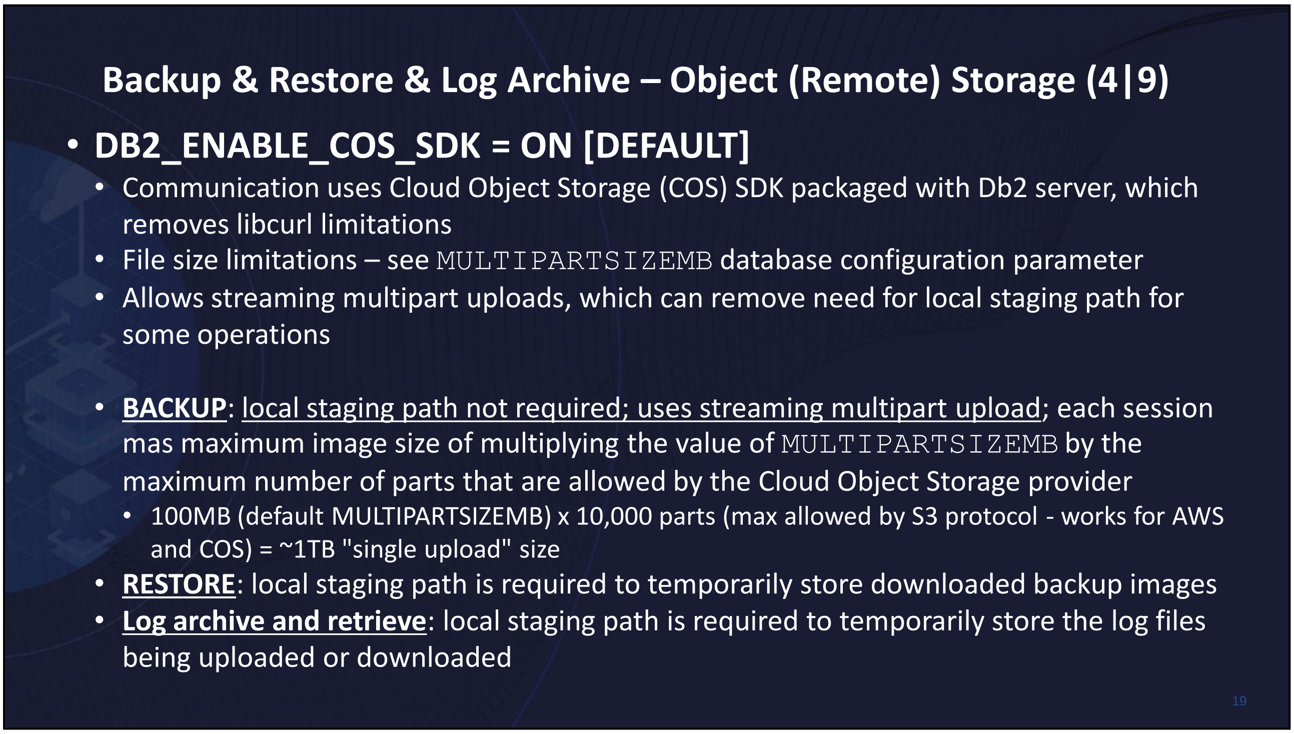
https://www.ibm.com/docs/en/db2/11.5?topic=commands-catalog-storage-access

DB2_ENABLE_COS_SDK

- https://www.ibm.com/docs/en/db2/11.5?topic=variables-miscellaneous#r0005669__M_DB2_ENABLE_COS_SDK

multipartsizemb – Remote storage multipart upload part size configuration parameter

- https://www.ibm.com/docs/en/db2/11.5?topic=parameters-multipartsizemb-remote-storage-multipart-upload-part-size



Backup & Restore & Log Archive – Object (Remote) Storage (4|9)

- **DB2_ENABLE_COS_SDK = ON [DEFAULT]**
  - Communication uses Cloud Object Storage (COS) SDK packaged with Db2 server, which removes libcurl limitations
  - File size limitations – see `MULTIPARTSIZEMB` database configuration parameter
  - Allows streaming multipart uploads, which can remove need for local staging path for some operations

  - **BACKUP**: local staging path not required; uses streaming multipart upload; each session mas maximum image size of multiplying the value of `MULTIPARTSIZEMB` by the maximum number of parts that are allowed by the Cloud Object Storage provider
    - 100MB (default MULTIPARTSIZEMB) x 10,000 parts (max allowed by S3 protocol - works for AWS and COS) = ~1TB "single upload" size
  - **RESTORE**: local staging path is required to temporarily store downloaded backup images
  - **Log archive and retrieve**: local staging path is required to temporarily store the log files being uploaded or downloaded

19

Remote storage requirements

- https://www.ibm.com/docs/en/db2/11.5?topic=reference-remote-storage-requirements

CATALOG STORAGE ACCESS command

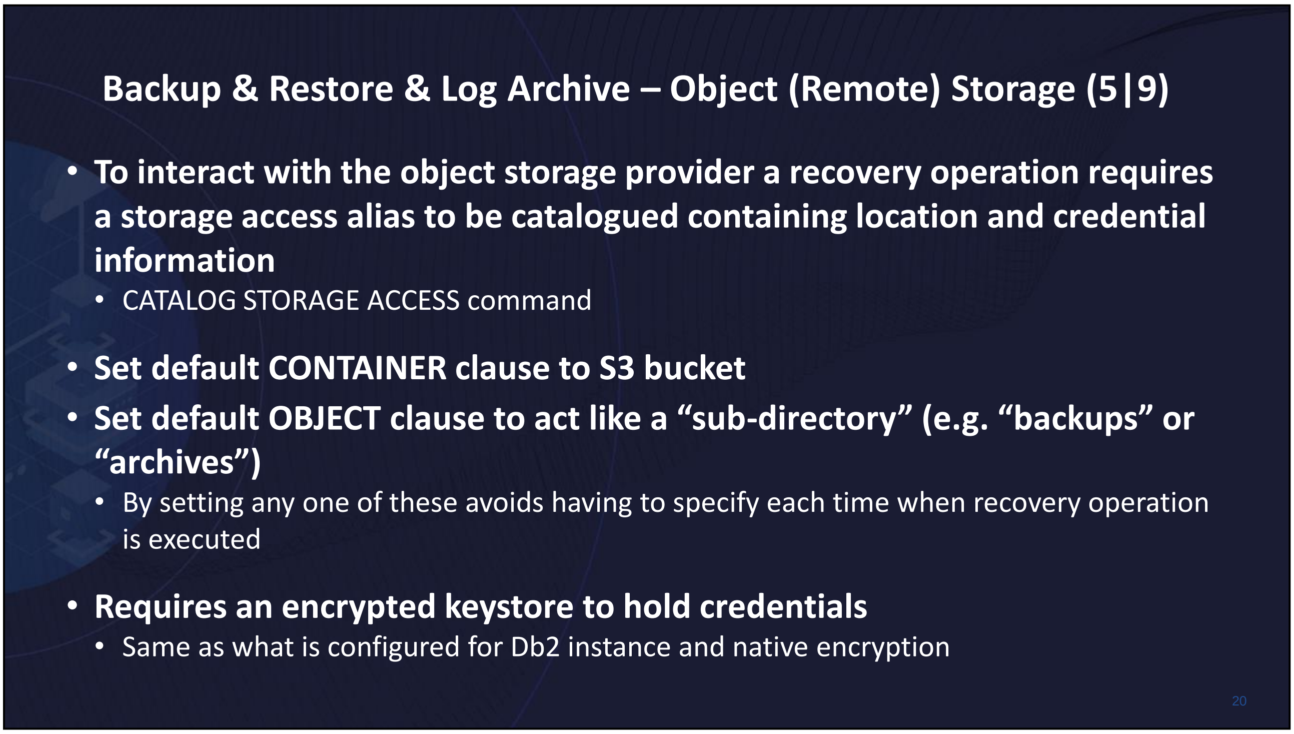
https://www.ibm.com/docs/en/db2/11.5?topic=commands-catalog-storage-access

DB2_ENABLE_COS_SDK

- https://www.ibm.com/docs/en/db2/11.5?topic=variables-miscellaneous#r0005669__M_DB2_ENABLE_COS_SDK

multipartsizemb – Remote storage multipart upload part size configuration parameter

- https://www.ibm.com/docs/en/db2/11.5?topic=parameters-multipartsizemb-remote-storage-multipart-upload-part-size



Backup & Restore & Log Archive – Object (Remote) Storage (5|9)

- **To interact with the object storage provider a recovery operation requires a storage access alias to be catalogued containing location and credential information**
  - CATALOG STORAGE ACCESS command
- **Set default CONTAINER clause to S3 bucket**
- **Set default OBJECT clause to act like a "sub-directory" (e.g. "backups" or "archives")**
  - By setting any one of these avoids having to specify each time when recovery operation is executed
- **Requires an encrypted keystore to hold credentials**
  - Same as what is configured for Db2 instance and native encryption

20

Remote storage requirements

- https://www.ibm.com/docs/en/db2/11.5?topic=reference-remote-storage-requirements

CATALOG STORAGE ACCESS command

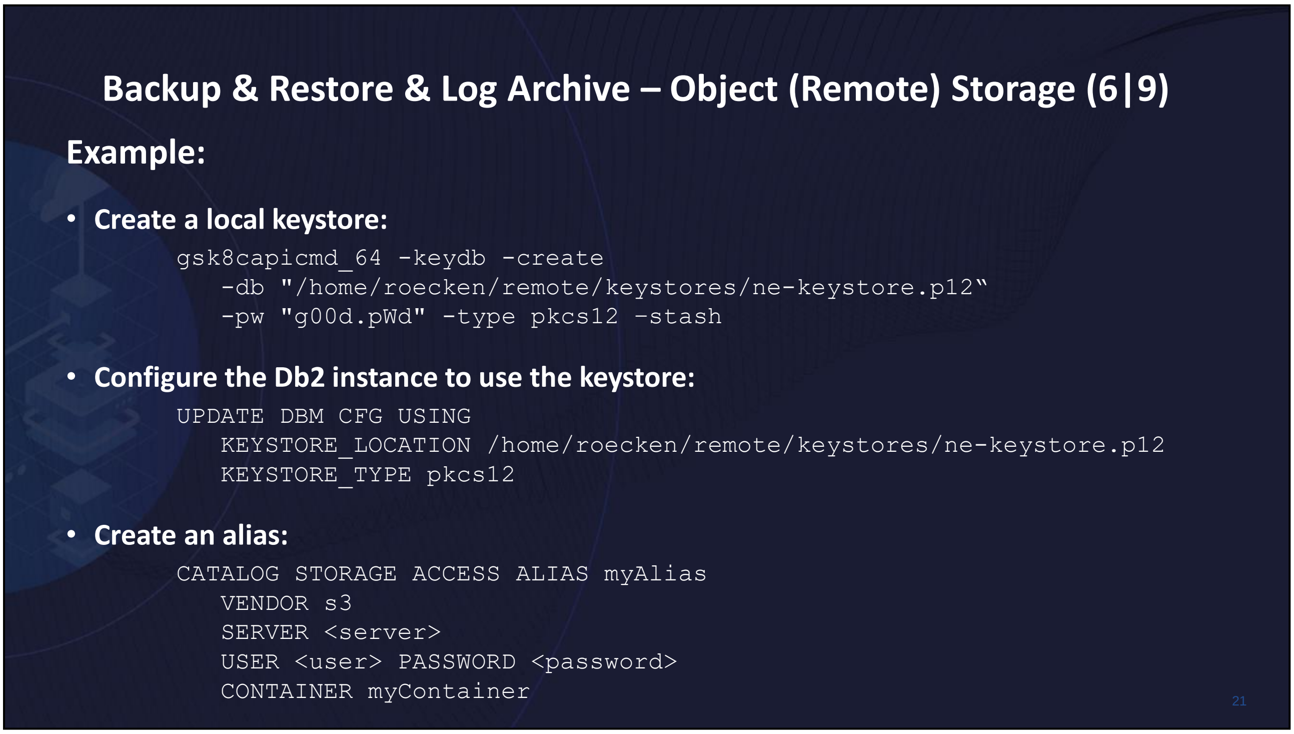
https://www.ibm.com/docs/en/db2/11.5?topic=commands-catalog-storage-access

DB2_ENABLE_COS_SDK

- https://www.ibm.com/docs/en/db2/11.5?topic=variables-miscellaneous#r0005669__M_DB2_ENABLE_COS_SDK

multipartsizemb – Remote storage multipart upload part size configuration parameter

- https://www.ibm.com/docs/en/db2/11.5?topic=parameters-multipartsizemb-remote-storage-multipart-upload-part-size



Backup & Restore & Log Archive – Object (Remote) Storage (6|9)

Example:

- **Create a local keystore:**

```
gsk8capicmd_64 -keydb -create
   -db "/home/roecken/remote/keystores/ne-keystore.p12‟
   -pw "g00d.pWd" -type pkcs12 -stash
```

- **Configure the Db2 instance to use the keystore:**

```
UPDATE DBM CFG USING
   KEYSTORE_LOCATION /home/roecken/remote/keystores/ne-keystore.p12
   KEYSTORE_TYPE pkcs12
```

- **Create an alias:**

```
CATALOG STORAGE ACCESS ALIAS myAlias
   VENDOR s3
   SERVER <server>
   USER <user> PASSWORD <password>
   CONTAINER myContainer
```

Remote storage requirements

- https://www.ibm.com/docs/en/db2/11.5?topic=reference-remote-storage-requirements

CATALOG STORAGE ACCESS command

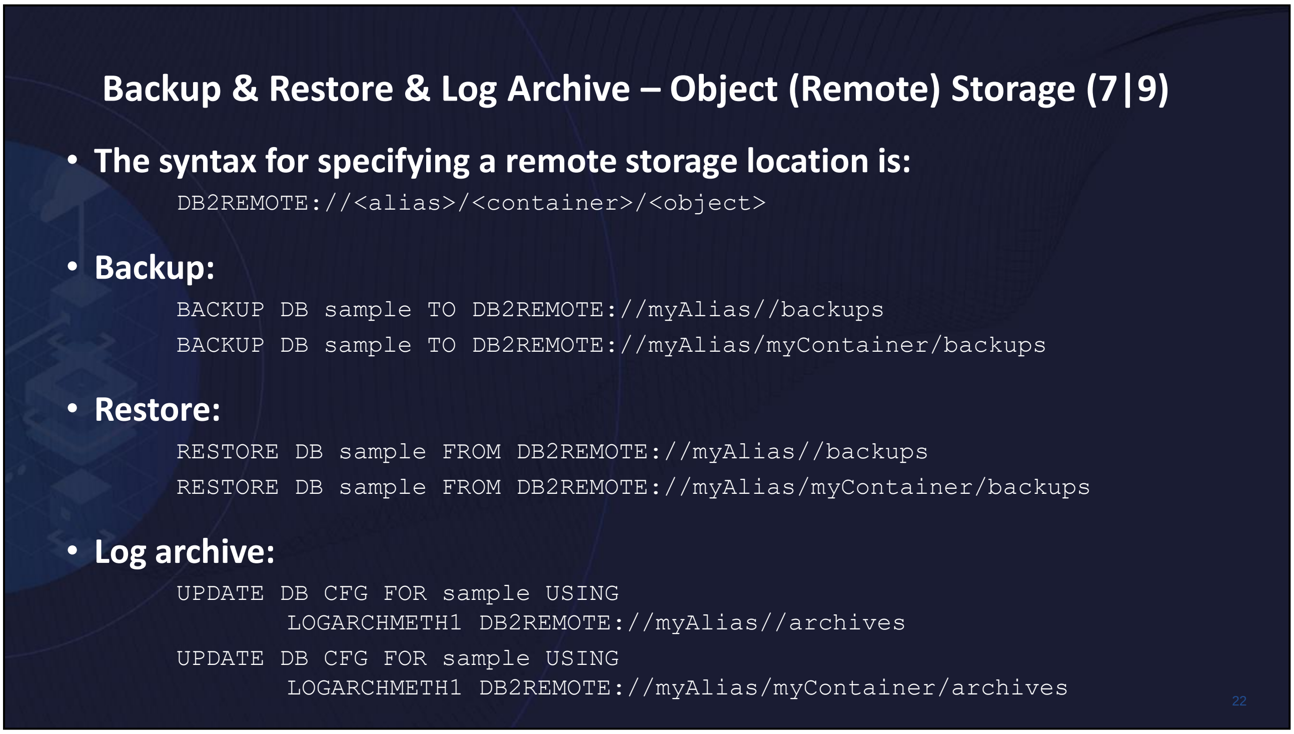
https://www.ibm.com/docs/en/db2/11.5?topic=commands-catalog-storage-access

DB2_ENABLE_COS_SDK

- https://www.ibm.com/docs/en/db2/11.5?topic=variables-miscellaneous#r0005669__M_DB2_ENABLE_COS_SDK

multipartsizemb – Remote storage multipart upload part size configuration parameter

- https://www.ibm.com/docs/en/db2/11.5?topic=parameters-multipartsizemb-remote-storage-multipart-upload-part-size



Backup & Restore & Log Archive – Object (Remote) Storage (7|9)

- **The syntax for specifying a remote storage location is:**

```
DB2REMOTE://<alias>/<container>/<object>
```

- **Backup:**

```
BACKUP DB sample TO DB2REMOTE://myAlias//backups
BACKUP DB sample TO DB2REMOTE://myAlias/myContainer/backups
```

- **Restore:**

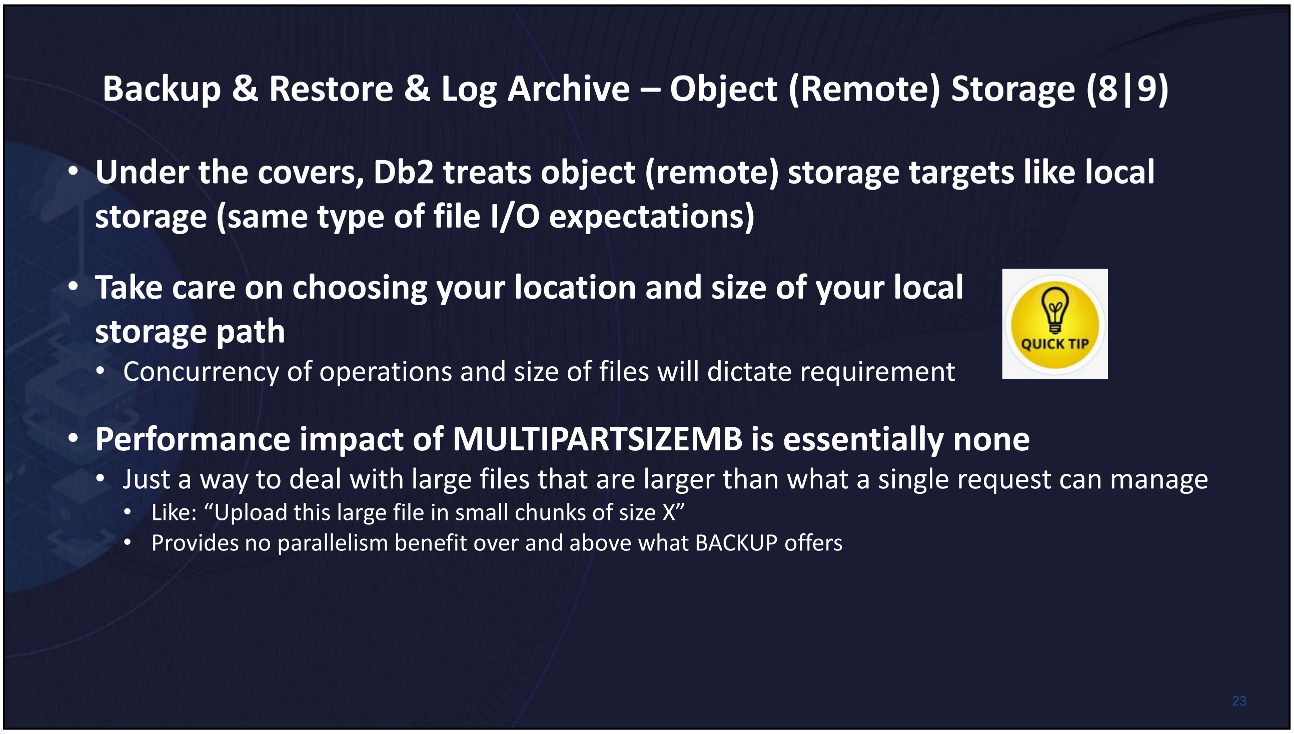
```
RESTORE DB sample FROM DB2REMOTE://myAlias//backups
RESTORE DB sample FROM DB2REMOTE://myAlias/myContainer/backups
```

- **Log archive:**

```
UPDATE DB CFG FOR sample USING
       LOGARCHMETH1 DB2REMOTE://myAlias//archives
UPDATE DB CFG FOR sample USING
       LOGARCHMETH1 DB2REMOTE://myAlias/myContainer/archives
```

22

Each operation will lookup the alias catalogued to find vendor, server, container, object and credential information. Then the file name such as: backup image file name or log archive chain target directory will be appended. Db2 will form the URL to communicate with the S3 Cloud Object Storage provider.



Backup & Restore & Log Archive – Object (Remote) Storage (8|9)

- **Under the covers, Db2 treats object (remote) storage targets like local storage (same type of file I/O expectations)**
- **Take care on choosing your location and size of your local storage path**
  - Concurrency of operations and size of files will dictate requirement

QUICK TIP

- **Performance impact of MULTIPARTSIZEMB is essentially none**
  - Just a way to deal with large files that are larger than what a single request can manage
    - Like: “Upload this large file in small chunks of size X”
    - Provides no parallelism benefit over and above what BACKUP offers

Remote storage requirements

- https://www.ibm.com/docs/en/db2/11.5?topic=reference-remote-storage-requirements

CATALOG STORAGE ACCESS command

https://www.ibm.com/docs/en/db2/11.5?topic=commands-catalog-storage-access

DB2_ENABLE_COS_SDK

- https://www.ibm.com/docs/en/db2/11.5?topic=variables-miscellaneous#r0005669__M_DB2_ENABLE_COS_SDK

multipartsizemb – Remote storage multipart upload part size configuration parameter

- https://www.ibm.com/docs/en/db2/11.5?topic=parameters-multipartsizemb-remote-storage-multipart-upload-part-size

Backup & Restore & Log Archive – Object (Remote) Storage (9|9)

- **Performance in general comes down to data volume and speed of the network**
  - Many customers have jumped on aboard and are surprised by performance
  - Target is likely slower since it is likely off-site and public-network connected
  - Consider what “zone” from your provider is being used for the remote SERVER
  - Try enabling compressed backups / log archives

Remote storage requirements

- https://www.ibm.com/docs/en/db2/11.5?topic=reference-remote-storage-requirements

CATALOG STORAGE ACCESS command

https://www.ibm.com/docs/en/db2/11.5?topic=commands-catalog-storage-access

DB2_ENABLE_COS_SDK

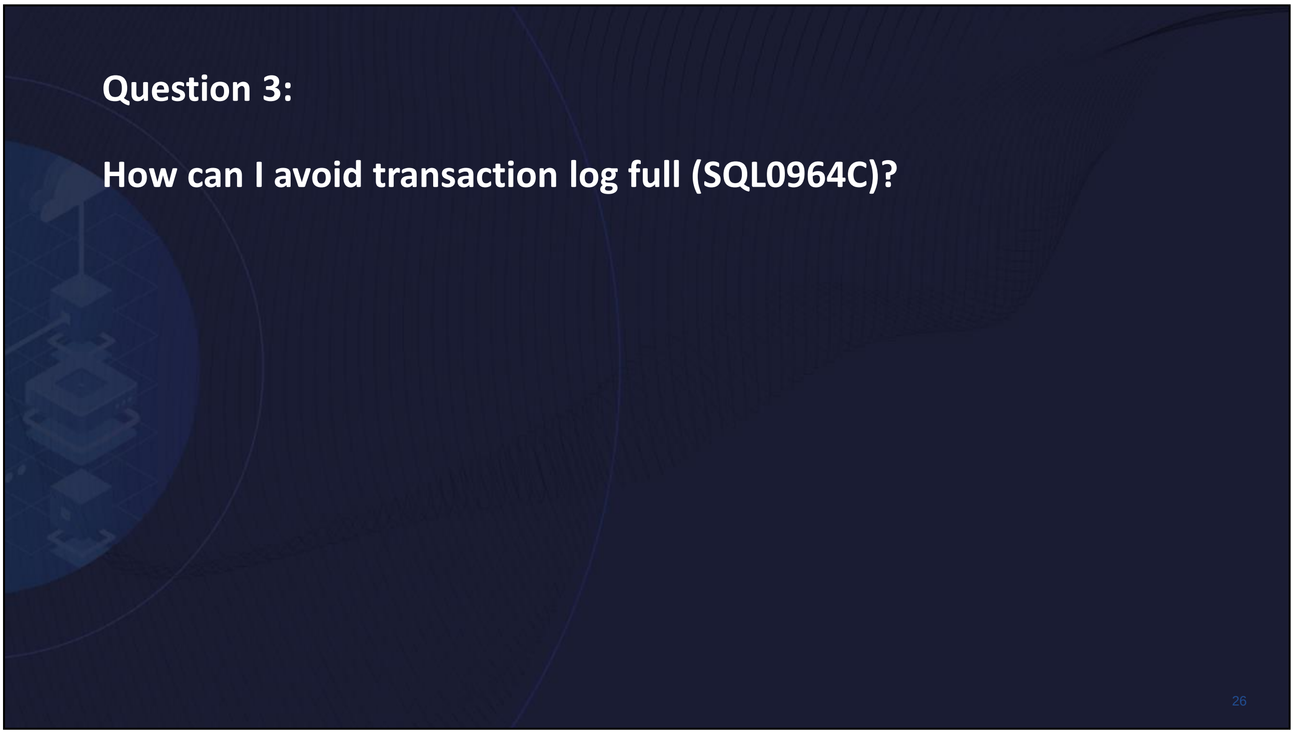
- https://www.ibm.com/docs/en/db2/11.5?topic=variables-miscellaneous#r0005669__M_DB2_ENABLE_COS_SDK

multipartsizemb – Remote storage multipart upload part size configuration parameter

- https://www.ibm.com/docs/en/db2/11.5?topic=parameters-multipartsizemb-remote-storage-multipart-upload-part-size

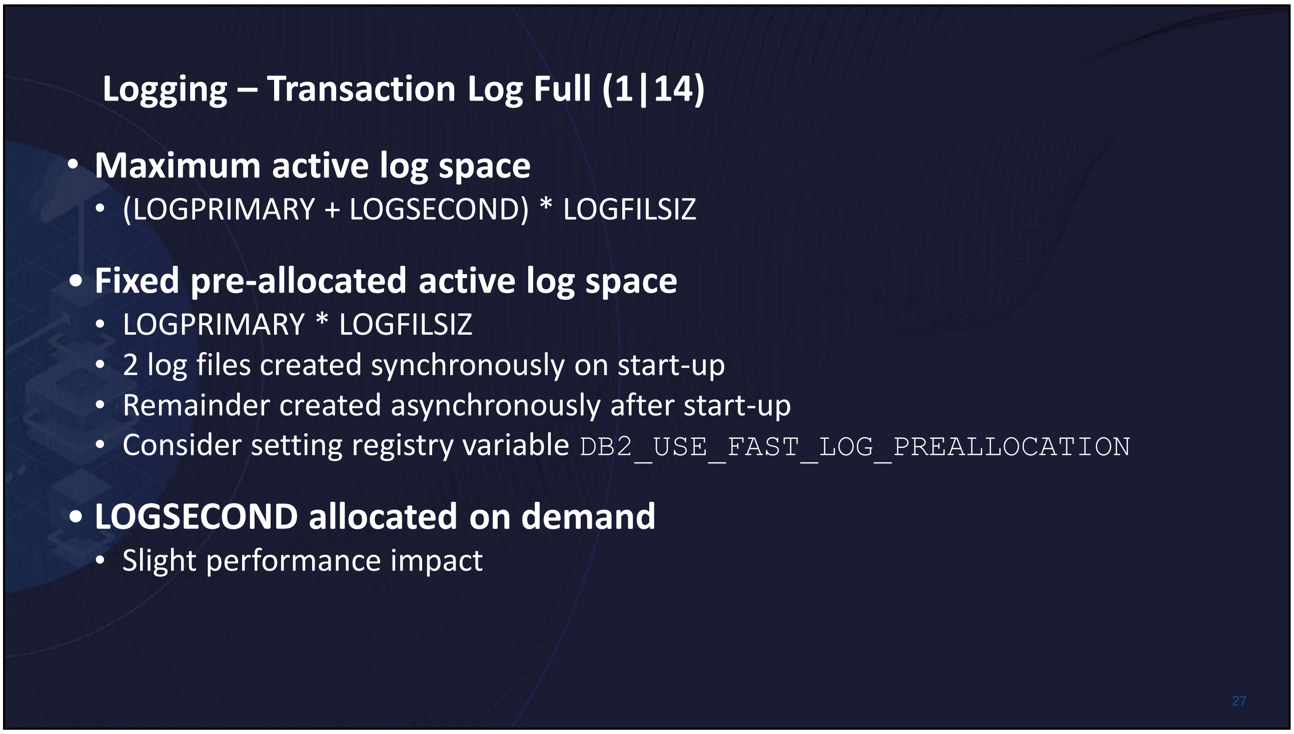
Agenda

- Backup & Restore
  - Performance
  - Object (Remote) Storage
- **Logging**
  - **Transaction Log Full**
  - **Disk Full in Archives**
- Recovery
  - Point-in-time Recovery
  - Inaccessible Table Spaces
  - Monitor Crash Recovery
- Upgrade



Question 3:

How can I avoid transaction log full (SQL0964C)?



Logging – Transaction Log Full (1|14)

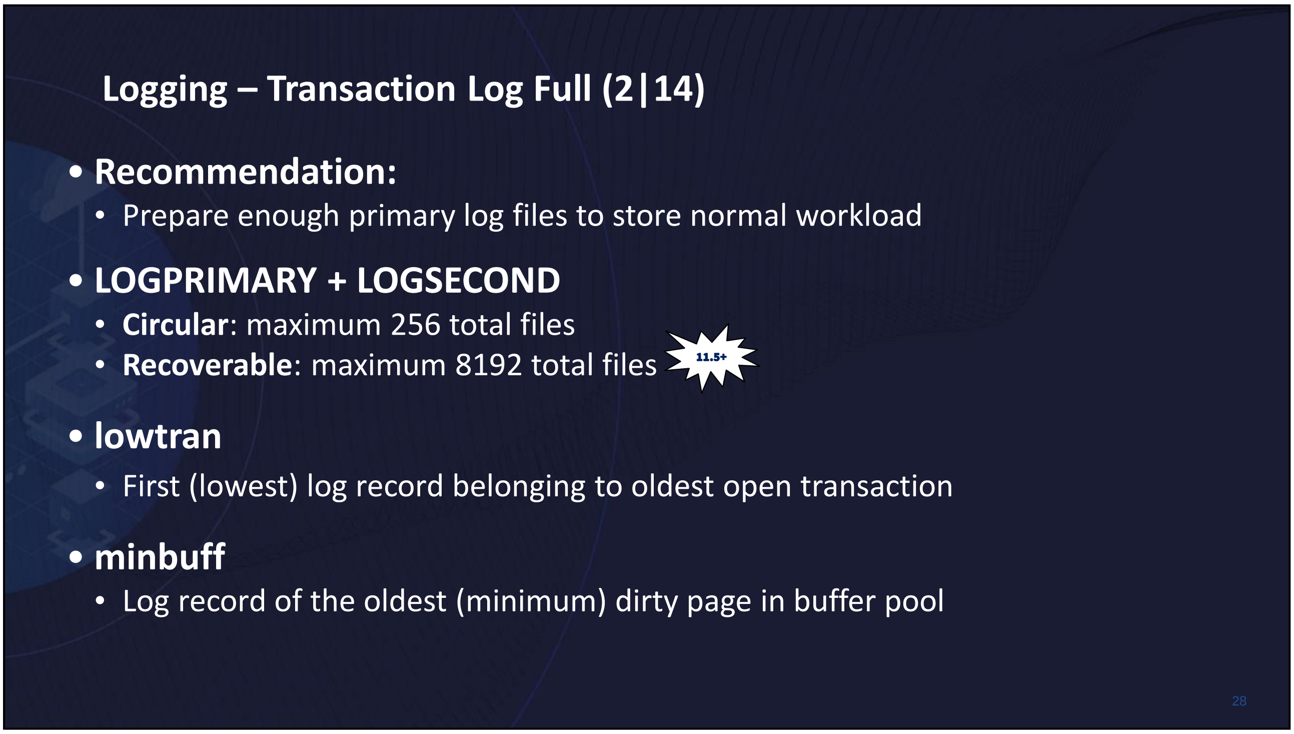
- **Maximum active log space**
  - (LOGPRIMARY + LOGSECOND) * LOGFILSIZ
- **Fixed pre-allocated active log space**
  - LOGPRIMARY * LOGFILSIZ
  - 2 log files created synchronously on start-up
  - Remainder created asynchronously after start-up
  - Consider setting registry variable `DB2_USE_FAST_LOG_PREALLOCATION`
- **LOGSECOND allocated on demand**
  - Slight performance impact

How to solve transaction log for the databases is full (SQL0964C)

- https://www.ibm.com/support/pages/db2-how-solve-transaction-log-databases-full-sql0964c

`DB2_USE_FAST_LOG_PREALLOCATION`

- Operating system: AIX and Linux on Veritas VxFS, JFS2, GPFS, ext4 (Linux only) and xFS (Linux only) file systems
- Default: `OFF`, `ON` under DB2_WORKLOAD=SAP, Values: `ON` or `OFF`
- Allows the fast preallocation file system feature to reserve space for log files, and speed up the process of creating or altering large log files, if the underlying file system supports this feature. This speed improvement is implemented at a small delta cost of performing actual space allocation during runtime when log records get written to such preallocated log files.

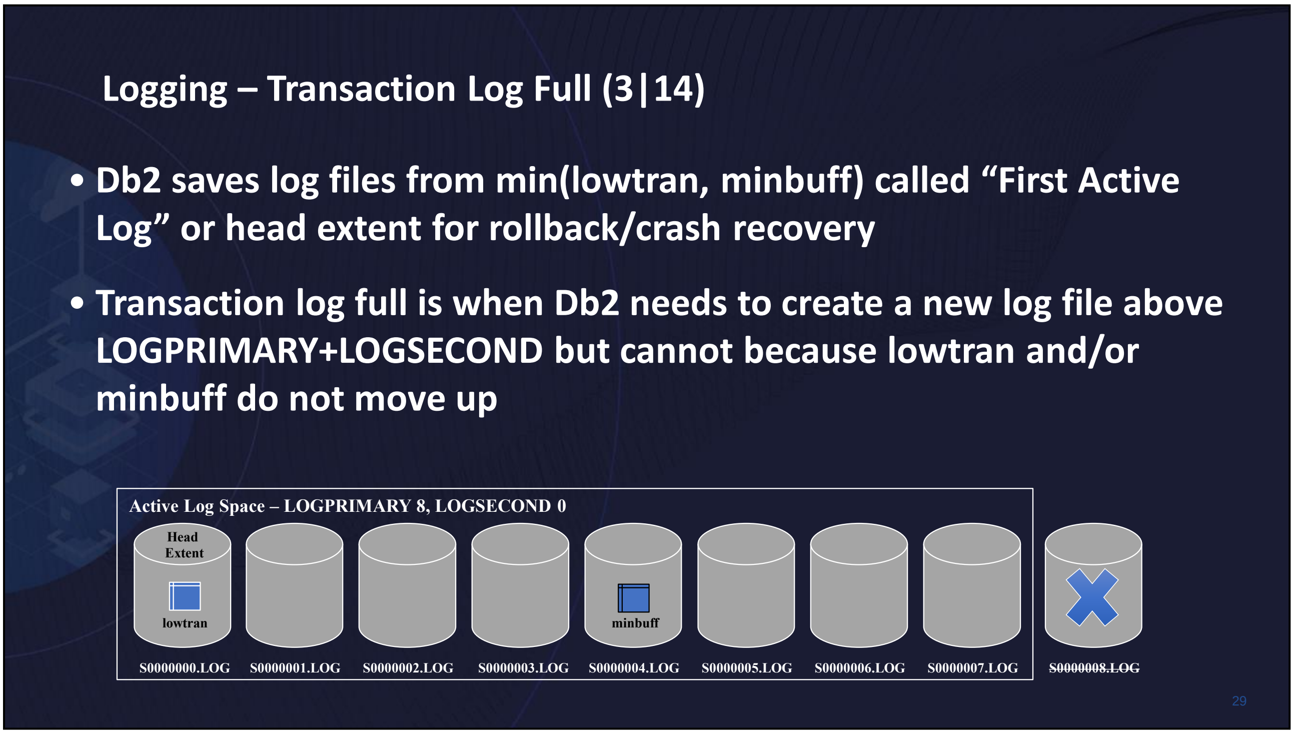


Logging – Transaction Log Full (2|14)

- **Recommendation:**
  - Prepare enough primary log files to store normal workload
- **LOGPRIMARY + LOGSECOND**
  - **Circular**: maximum 256 total files
  - **Recoverable**: maximum 8192 total files 11.5+
- **lowtran**
  - First (lowest) log record belonging to oldest open transaction
- **minbuff**
  - Log record of the oldest (minimum) dirty page in buffer pool

How to solve transaction log for the databases is full (SQL0964C)

- https://www.ibm.com/support/pages/db2-how-solve-transaction-log-databases-full-sql0964c



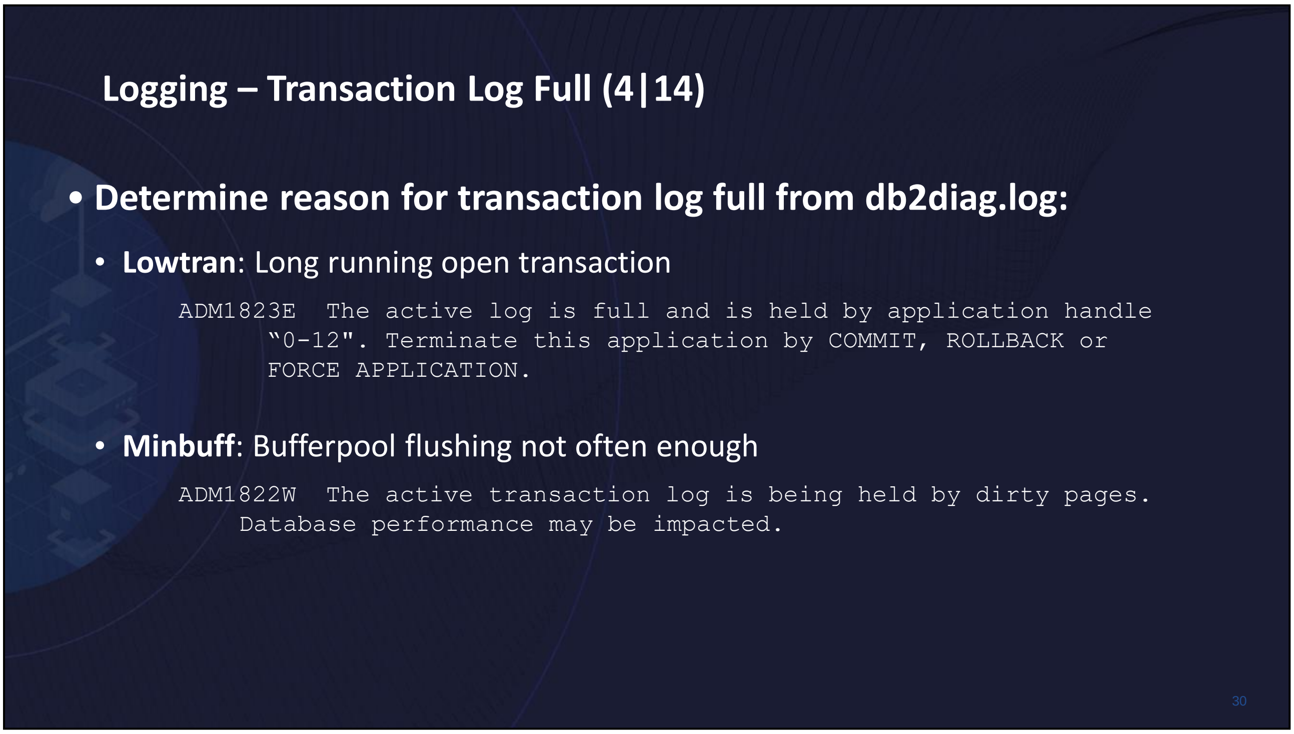
Logging – Transaction Log Full (3|14)

- Db2 saves log files from min(lowtran, minbuff) called “First Active Log” or head extent for rollback/crash recovery
- Transaction log full is when Db2 needs to create a new log file above LOGPRIMARY+LOGSECOND but cannot because lowtran and/or minbuff do not move up

Active Log Space – LOGPRIMARY 8, LOGSECOND 0
Head Extent
lowtran
minbuff
S0000000.LOG
S0000001.LOG
S0000002.LOG
S0000003.LOG
S0000004.LOG
S0000005.LOG
S0000006.LOG
S0000007.LOG
~~S0000008.LOG~~
29

How to solve transaction log for the databases is full (SQL0964C)

- https://www.ibm.com/support/pages/db2-how-solve-transaction-log-databases-full-sql0964c



Logging – Transaction Log Full (4|14)

- **Determine reason for transaction log full from db2diag.log:**
  - **Lowtran**: Long running open transaction

```
ADM1823E  The active log is full and is held by application handle
          “0-12". Terminate this application by COMMIT, ROLLBACK or
          FORCE APPLICATION.
```

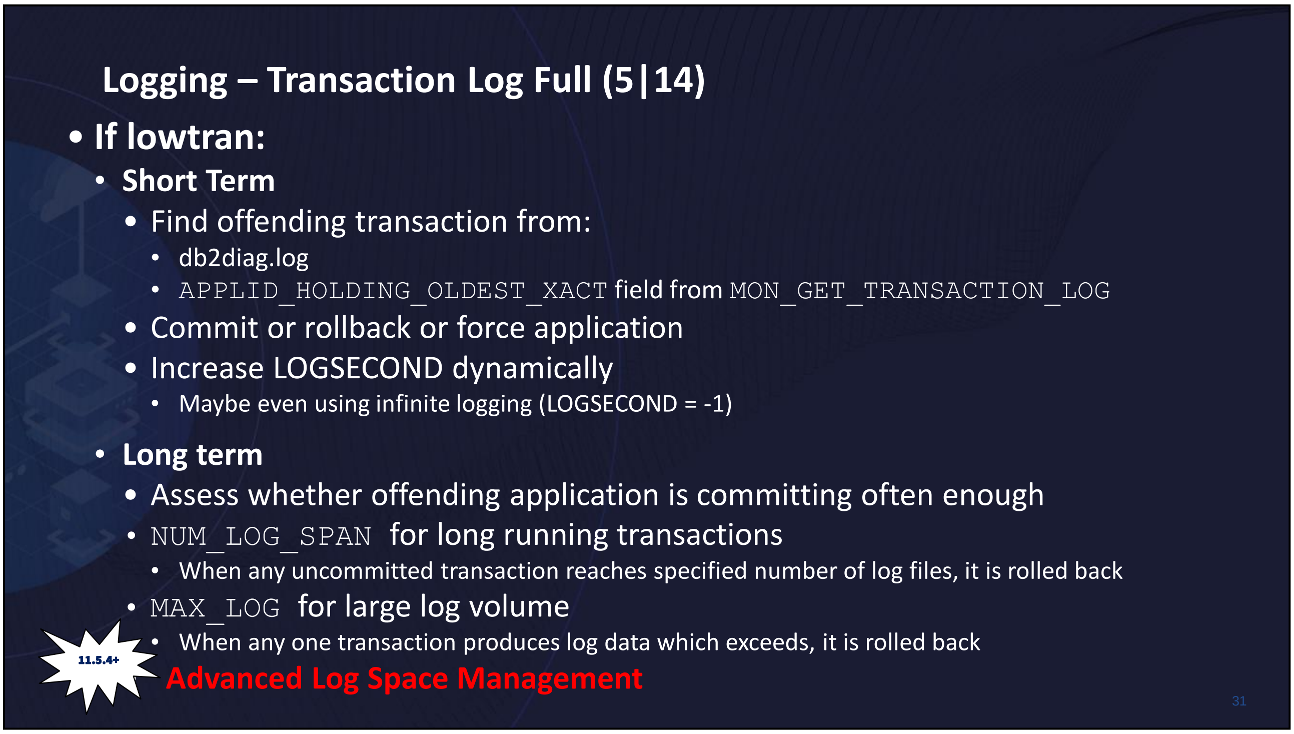
  - **Minbuff**: Bufferpool flushing not often enough

```
ADM1822W  The active transaction log is being held by dirty pages.
       Database performance may be impacted.
```

30

How to solve transaction log for the databases is full (SQL0964C)

- https://www.ibm.com/support/pages/db2-how-solve-transaction-log-databases-full-sql0964c



Logging – Transaction Log Full (5|14)

- **If lowtran:**
  - **Short Term**
    - Find offending transaction from:
      - db2diag.log
      - `APPLID_HOLDING_OLDEST_XACT` field from `MON_GET_TRANSACTION_LOG`
    - Commit or rollback or force application
    - Increase LOGSECOND dynamically
      - Maybe even using infinite logging (LOGSECOND = -1)
  - **Long term**
    - Assess whether offending application is committing often enough
    - `NUM_LOG_SPAN` for long running transactions
      - When any uncommitted transaction reaches specified number of log files, it is rolled back
    - `MAX_LOG` for large log volume
      - When any one transaction produces log data which exceeds, it is rolled back

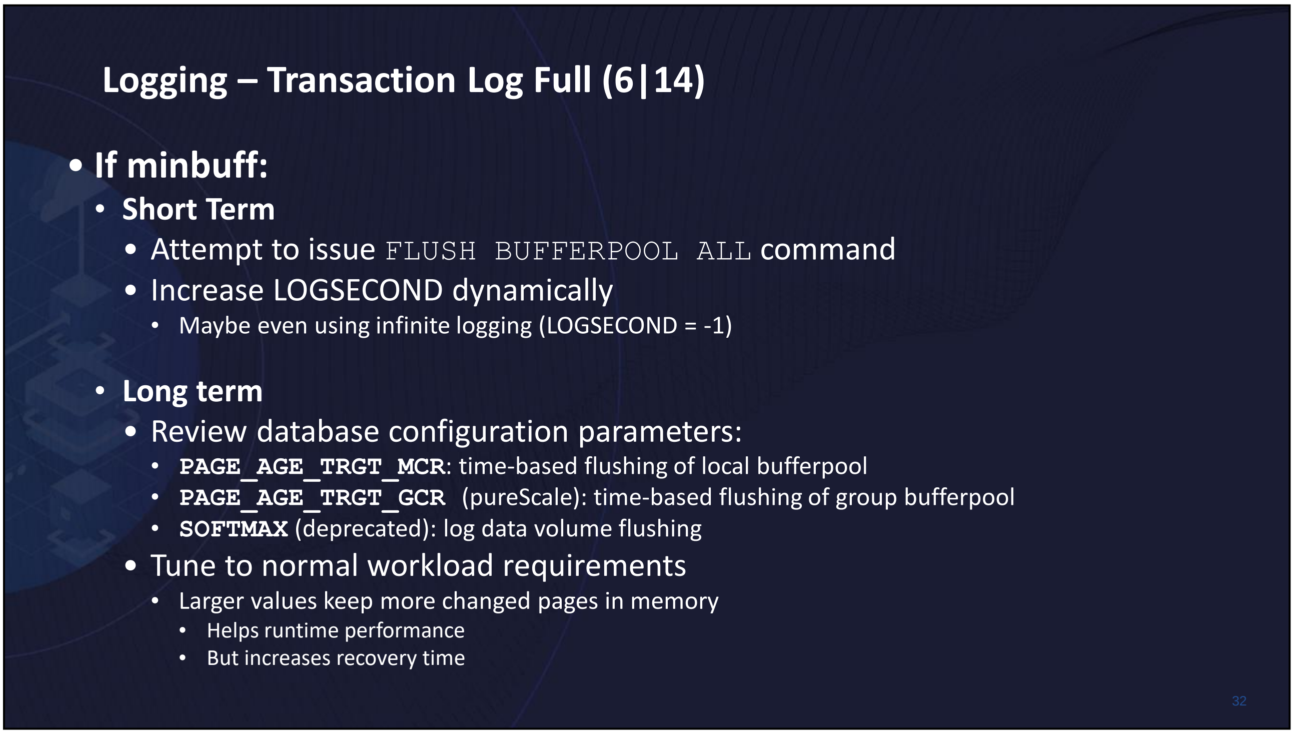
11.5.4+

Advanced Log Space Management

31

How to solve transaction log for the databases is full (SQL0964C)

- https://www.ibm.com/support/pages/db2-how-solve-transaction-log-databases-full-sql0964c

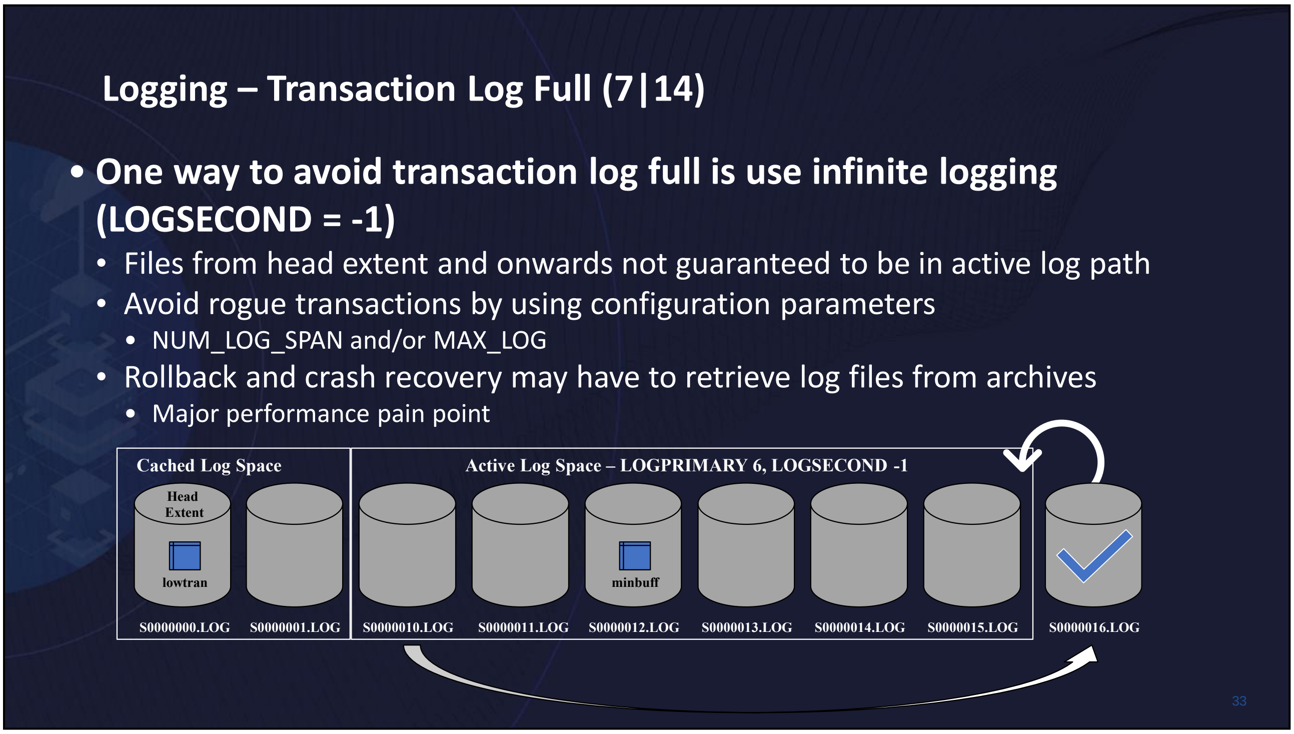


Logging – Transaction Log Full (6|14)

- **If minbuff:**
  - **Short Term**
    - Attempt to issue `FLUSH BUFFERPOOL ALL` command
    - Increase LOGSECOND dynamically
      - Maybe even using infinite logging (LOGSECOND = -1)
  - **Long term**
    - Review database configuration parameters:
      - **`PAGE_AGE_TRGT_MCR`**: time-based flushing of local bufferpool
      - **`PAGE_AGE_TRGT_GCR`** (pureScale): time-based flushing of group bufferpool
      - **`SOFTMAX`** (deprecated): log data volume flushing
    - Tune to normal workload requirements
      - Larger values keep more changed pages in memory
        - Helps runtime performance
        - But increases recovery time

How to solve transaction log for the databases is full (SQL0964C)

- https://www.ibm.com/support/pages/db2-how-solve-transaction-log-databases-full-sql0964c



Logging – Transaction Log Full (7|14)

- One way to avoid transaction log full is use infinite logging (LOGSECOND = -1)
  - Files from head extent and onwards not guaranteed to be in active log path
  - Avoid rogue transactions by using configuration parameters
    - NUM_LOG_SPAN and/or MAX_LOG
  - Rollback and crash recovery may have to retrieve log files from archives
    - Major performance pain point

Cached Log Space

Active Log Space – LOGPRIMARY 6, LOGSECOND -1

Head Extent

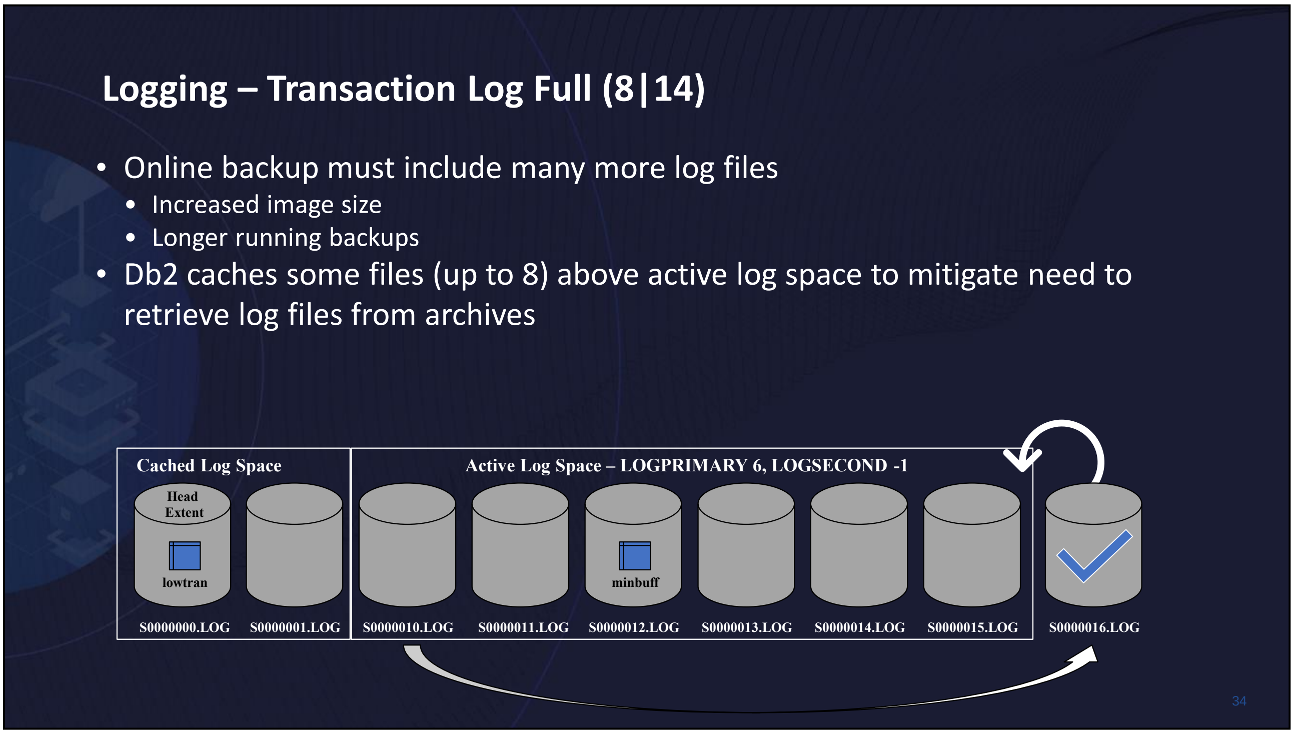
lowtran

minbuff

S0000000.LOG S0000001.LOG S0000010.LOG S0000011.LOG S0000012.LOG S0000013.LOG S0000014.LOG S0000015.LOG S0000016.LOG

How to solve transaction log for the databases is full (SQL0964C)

- https://www.ibm.com/support/pages/db2-how-solve-transaction-log-databases-full-sql0964c



Logging – Transaction Log Full (8|14)

- Online backup must include many more log files
  - Increased image size
  - Longer running backups
- Db2 caches some files (up to 8) above active log space to mitigate need to retrieve log files from archives

Cached Log Space

Active Log Space – LOGPRIMARY 6, LOGSECOND -1

Head Extent

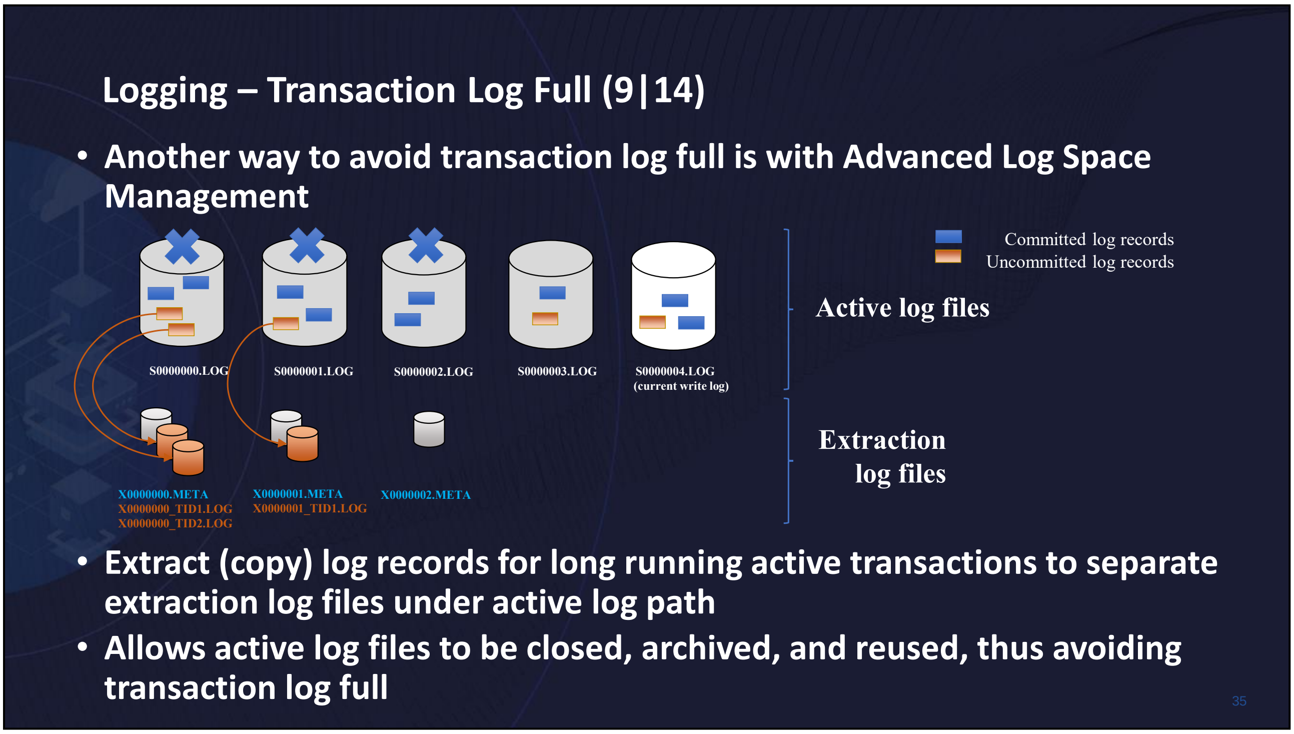
lowtran

minbuff

S0000000.LOG S0000001.LOG S0000010.LOG S0000011.LOG S0000012.LOG S0000013.LOG S0000014.LOG S0000015.LOG S0000016.LOG

How to solve transaction log for the databases is full (SQL0964C)

- https://www.ibm.com/support/pages/db2-how-solve-transaction-log-databases-full-sql0964c

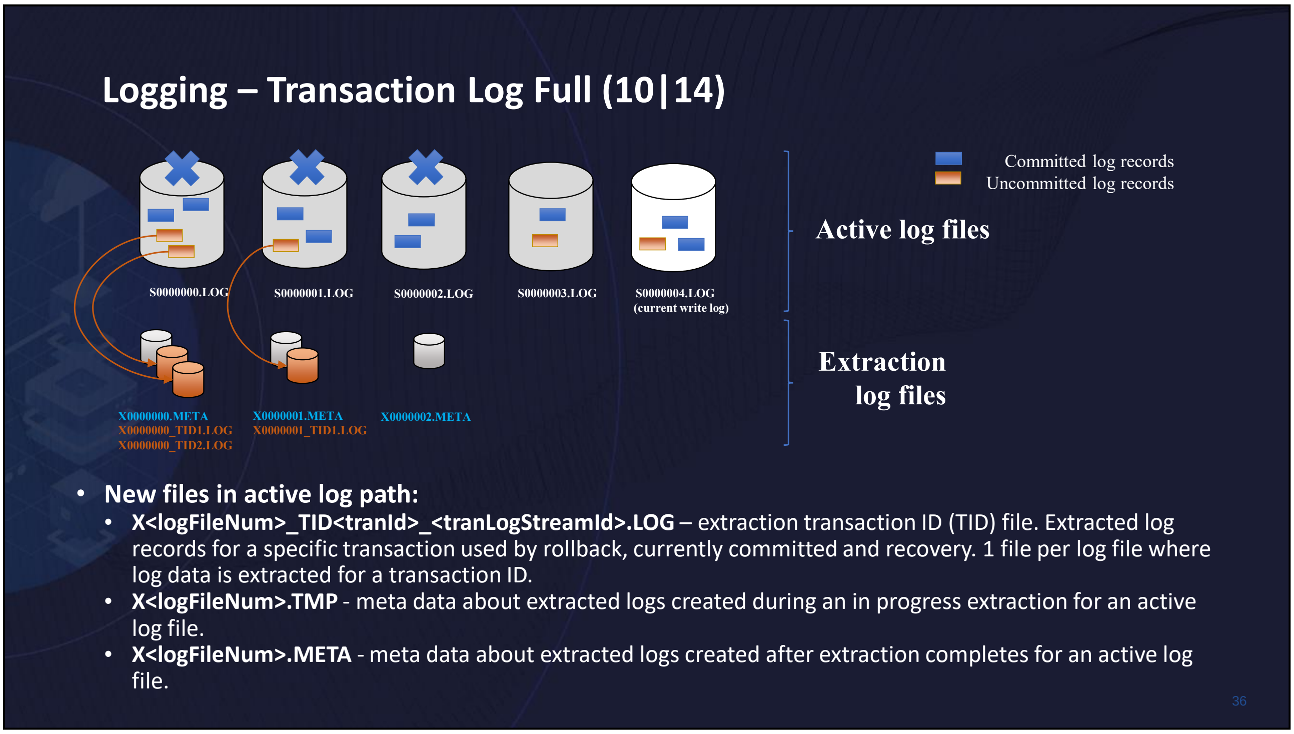


Logging – Transaction Log Full (9|14)

- Another way to avoid transaction log full is with Advanced Log Space Management

- Extract (copy) log records for long running active transactions to separate extraction log files under active log path
- Allows active log files to be closed, archived, and reused, thus avoiding transaction log full

Extraction will kick in when a certain percentage of log space has been consumed and will extract log records starting from head extent ID into separate extraction log files stored in the active log path.



Logging – Transaction Log Full (10|14)

🟦 Committed log records
🟧 Uncommitted log records

S0000000.LOG
S0000001.LOG
S0000002.LOG
S0000003.LOG
S0000004.LOG
(current write log)

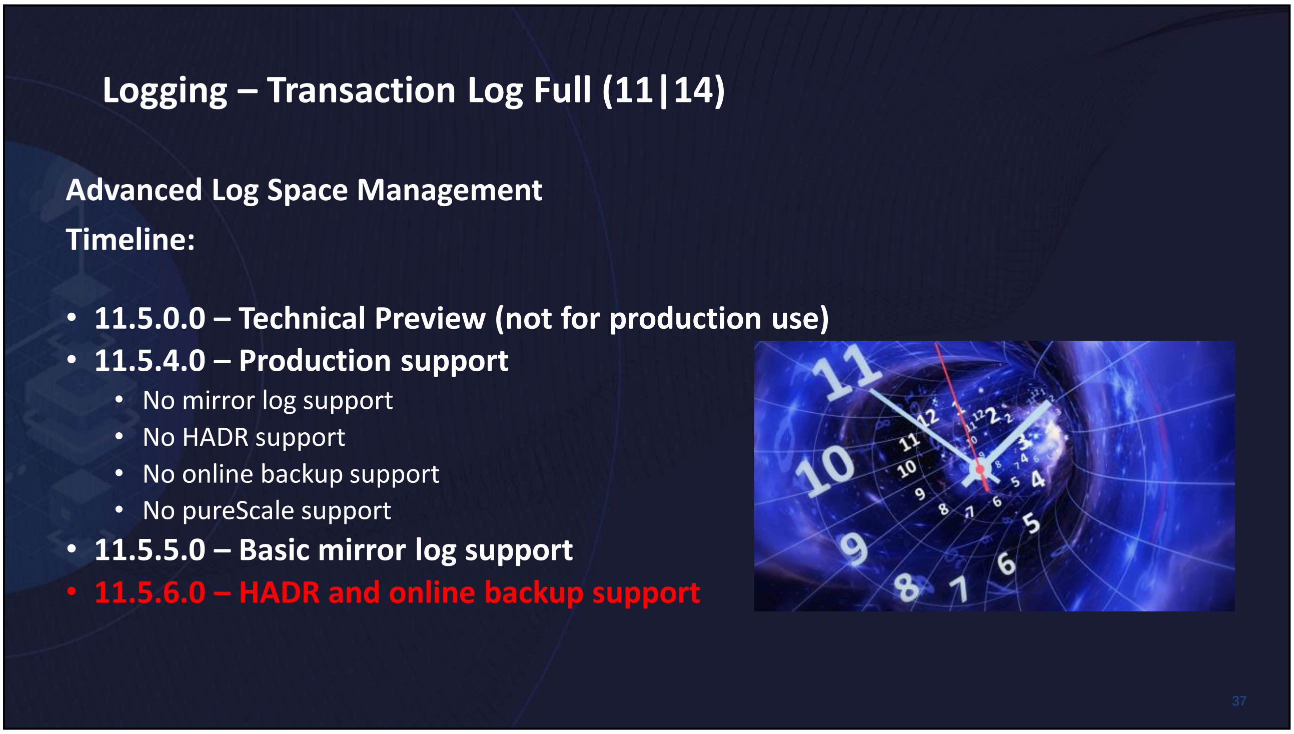
Active log files

X0000000.META
X0000000_TID1.LOG
X0000000_TID2.LOG
X0000001.META
X0000001_TID1.LOG
X0000002.META

Extraction
log files

- **New files in active log path:**
  - **X<logFileNum>_TID<tranId>_<tranLogStreamId>.LOG** – extraction transaction ID (TID) file. Extracted log records for a specific transaction used by rollback, currently committed and recovery. 1 file per log file where log data is extracted for a transaction ID.
  - **X<logFileNum>.TMP** - meta data about extracted logs created during an in progress extraction for an active log file.
  - **X<logFileNum>.META** - meta data about extracted logs created after extraction completes for an active log file.

36

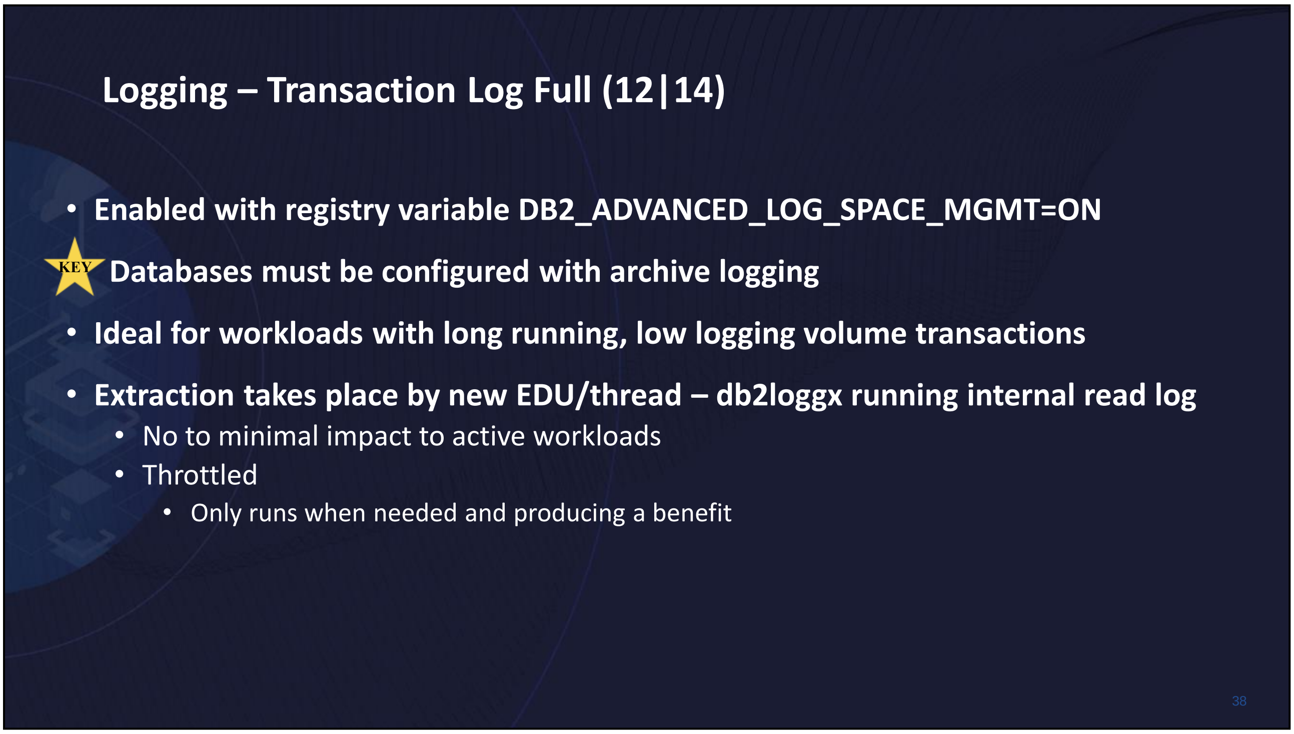


Logging – Transaction Log Full (11|14)

Advanced Log Space Management

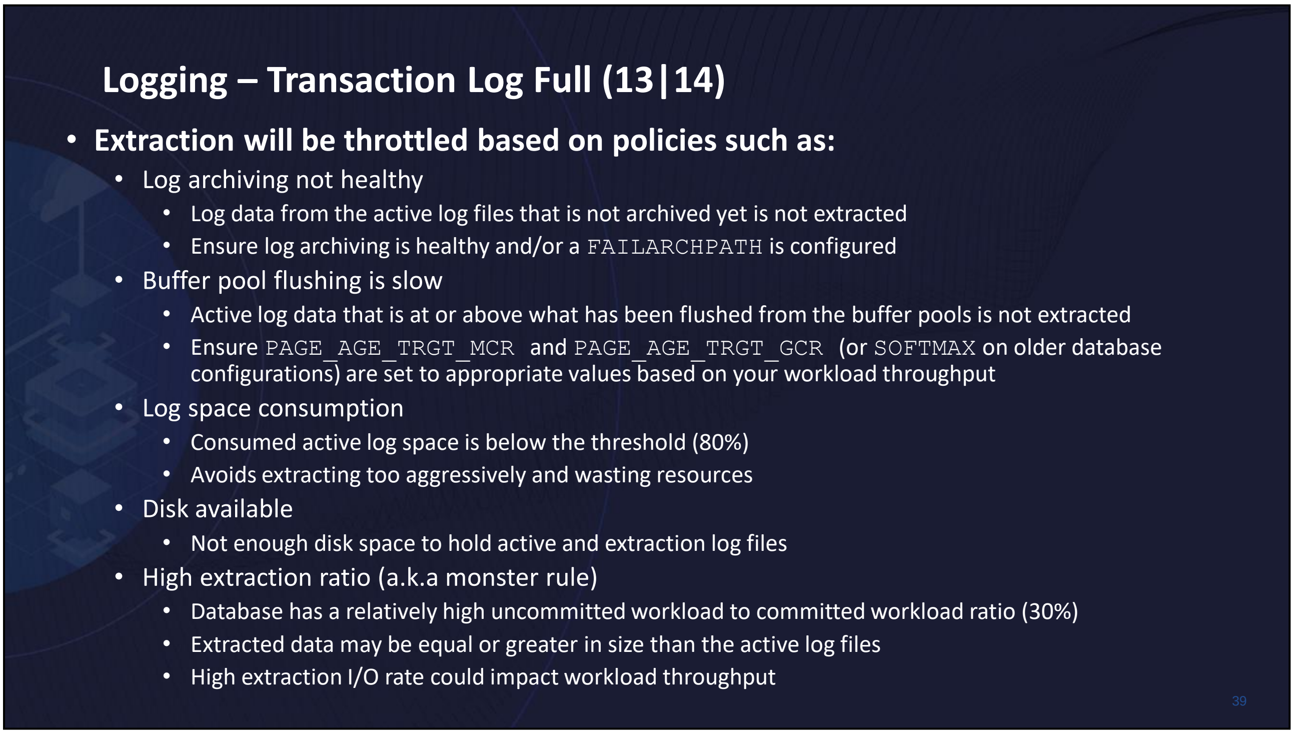
Timeline:

- **11.5.0.0 – Technical Preview (not for production use)**
- **11.5.4.0 – Production support**
  - No mirror log support
  - No HADR support
  - No online backup support
  - No pureScale support
- **11.5.5.0 – Basic mirror log support**
- **11.5.6.0 – HADR and online backup support**



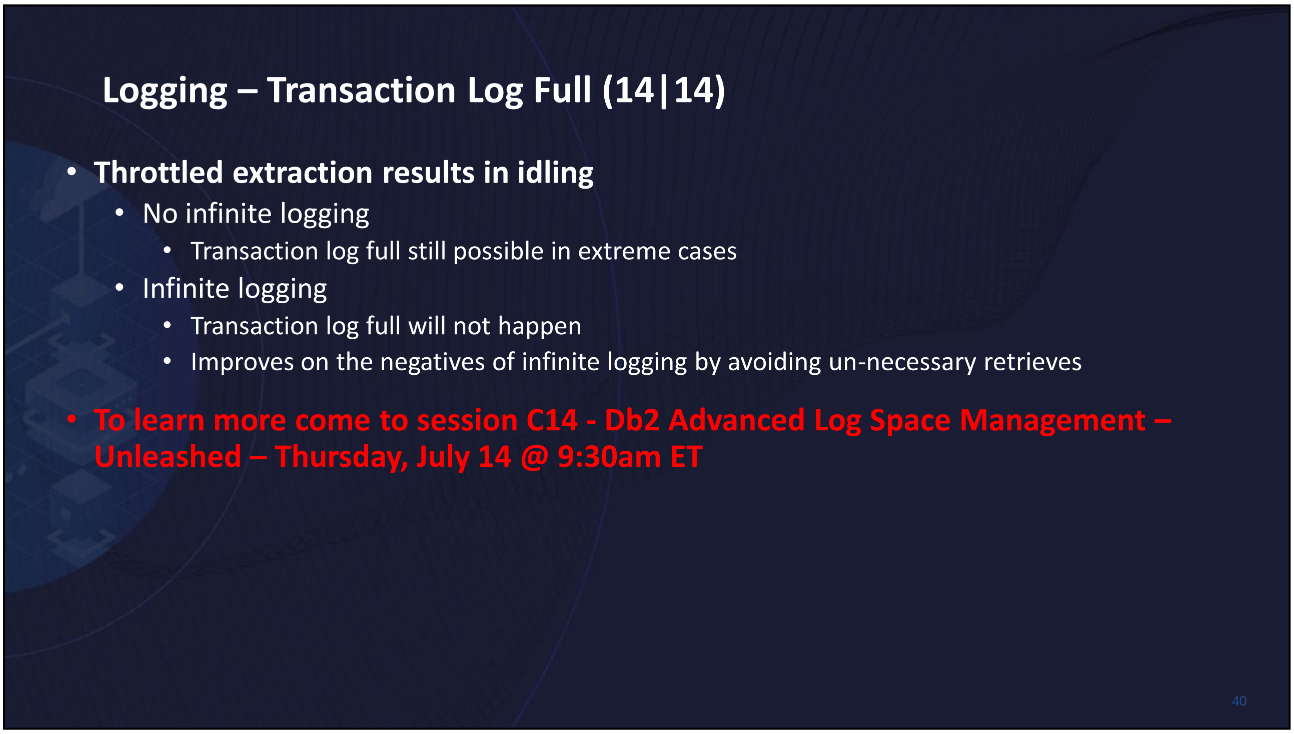
Logging – Transaction Log Full (12|14)

- Enabled with registry variable DB2_ADVANCED_LOG_SPACE_MGMT=ON
- KEY Databases must be configured with archive logging
- Ideal for workloads with long running, low logging volume transactions
- Extraction takes place by new EDU/thread – db2loggx running internal read log
  - No to minimal impact to active workloads
  - Throttled
    - Only runs when needed and producing a benefit



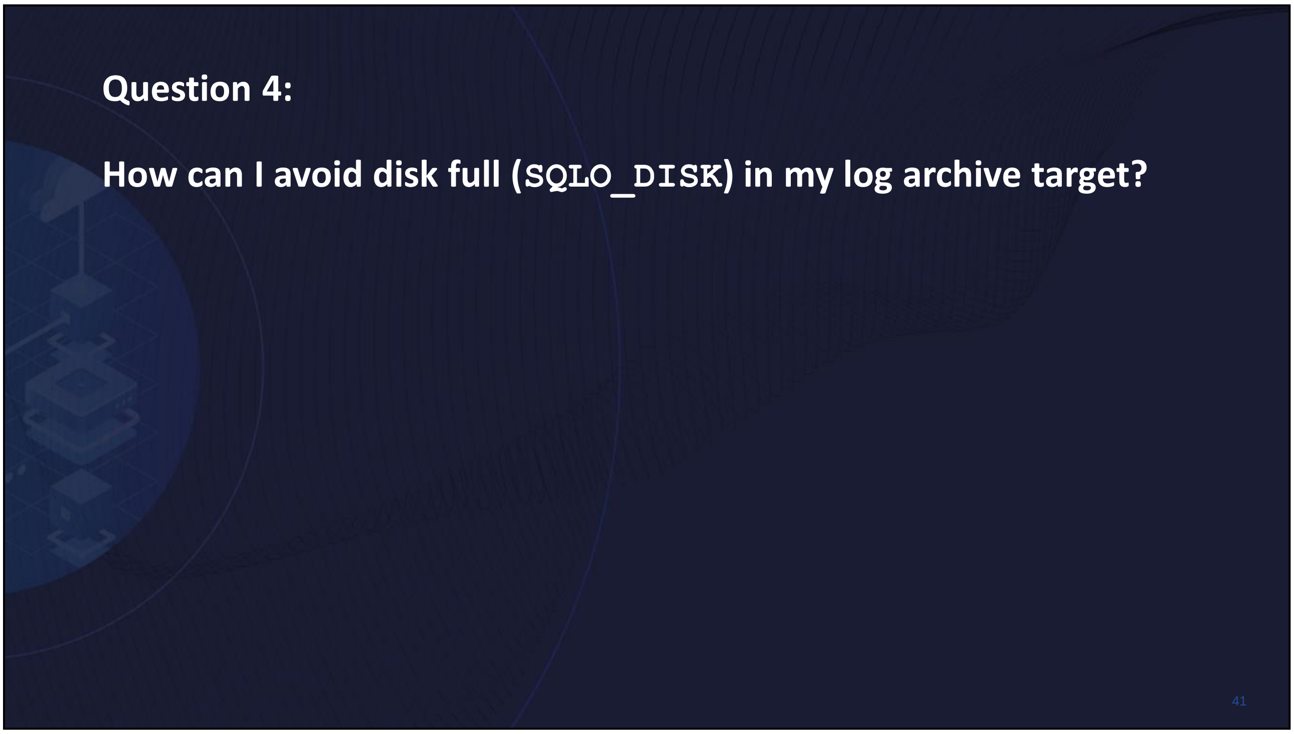
Logging – Transaction Log Full (13|14)

- **Extraction will be throttled based on policies such as:**
  - Log archiving not healthy
    - Log data from the active log files that is not archived yet is not extracted
    - Ensure log archiving is healthy and/or a `FAILARCHPATH` is configured
  - Buffer pool flushing is slow
    - Active log data that is at or above what has been flushed from the buffer pools is not extracted
    - Ensure `PAGE_AGE_TRGT_MCR` and `PAGE_AGE_TRGT_GCR` (or `SOFTMAX` on older database configurations) are set to appropriate values based on your workload throughput
  - Log space consumption
    - Consumed active log space is below the threshold (80%)
    - Avoids extracting too aggressively and wasting resources
  - Disk available
    - Not enough disk space to hold active and extraction log files
  - High extraction ratio (a.k.a monster rule)
    - Database has a relatively high uncommitted workload to committed workload ratio (30%)
    - Extracted data may be equal or greater in size than the active log files
    - High extraction I/O rate could impact workload throughput



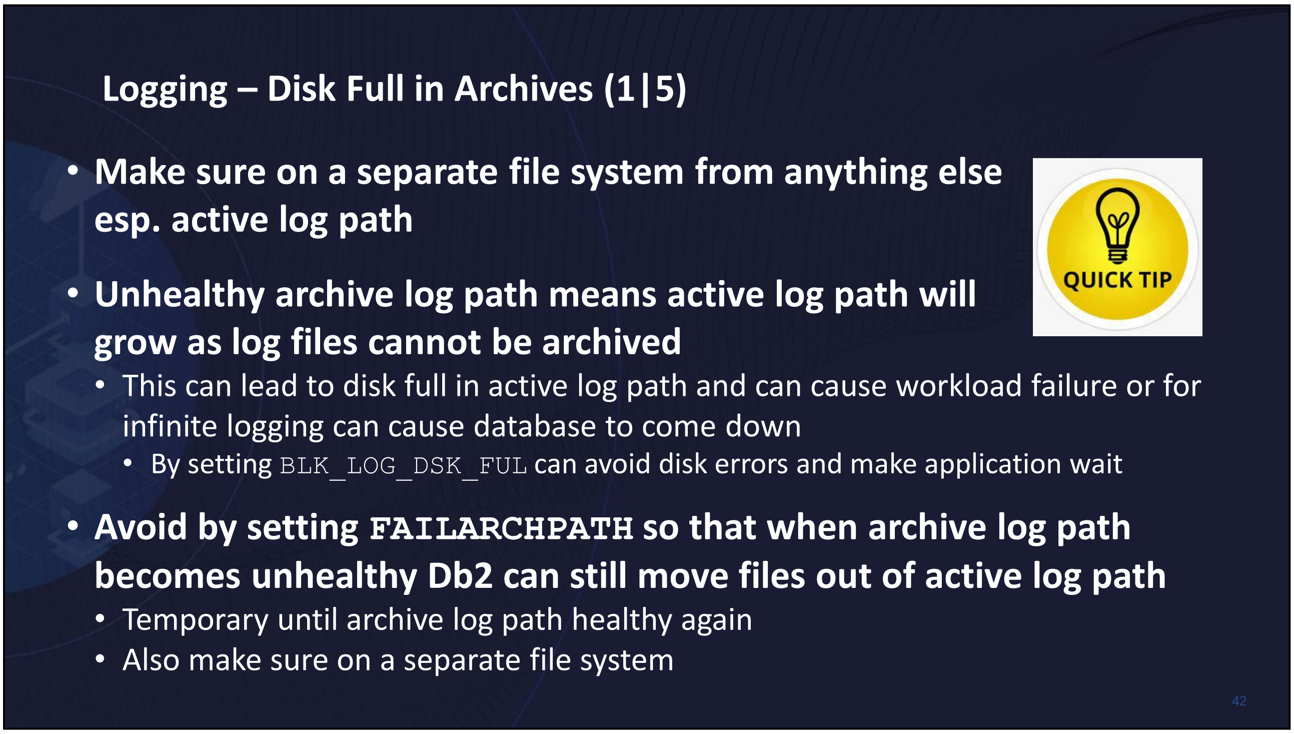
Logging – Transaction Log Full (14|14)

- **Throttled extraction results in idling**
  - No infinite logging
    - Transaction log full still possible in extreme cases
  - Infinite logging
    - Transaction log full will not happen
    - Improves on the negatives of infinite logging by avoiding un-necessary retrieves
- **To learn more come to session C14 - Db2 Advanced Log Space Management – Unleashed – Thursday, July 14 @ 9:30am ET**



Question 4:

How can I avoid disk full (`SQLO_DISK`) in my log archive target?



Logging – Disk Full in Archives (1|5)

QUICK TIP

- **Make sure on a separate file system from anything else esp. active log path**
- **Unhealthy archive log path means active log path will grow as log files cannot be archived**
  - This can lead to disk full in active log path and can cause workload failure or for infinite logging can cause database to come down
    - By setting `BLK_LOG_DSK_FUL` can avoid disk errors and make application wait
- **Avoid by setting `FAILARCHPATH` so that when archive log path becomes unhealthy Db2 can still move files out of active log path**
  - Temporary until archive log path healthy again
  - Also make sure on a separate file system

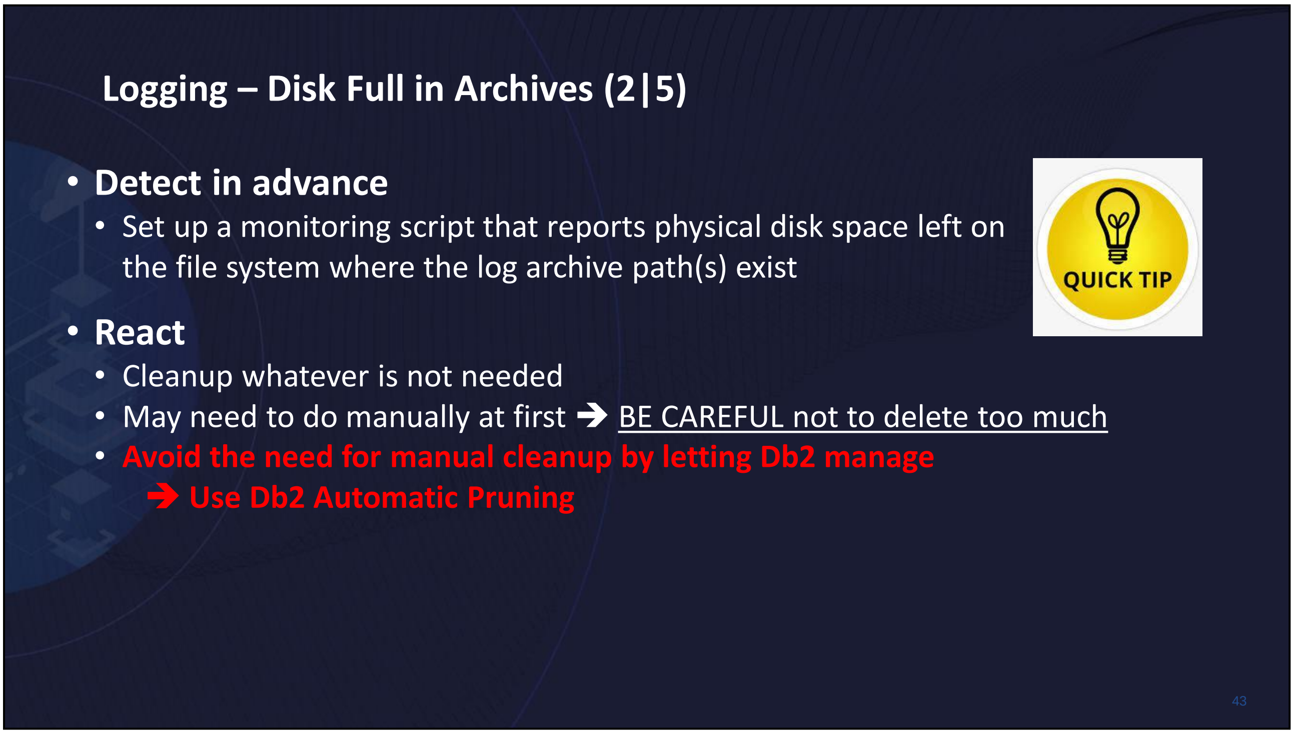
42

db2diag.log would show something like this:

```
EDUID   : 397                  EDUNAME: db2logmgr (SAMPLE)
FUNCTION: DB2 UDB, data protection services, sqlpgArchiveLogDisk, probe:2530
MESSAGE : ZRC=0x850F000C=-2062614516=SQLO_DISK "Disk full."
          DIA8312C Disk was full.
DATA #1 : String, 66 bytes
/db2archives/db2inst/NODE0000/SQL00001/LOGSTREAM0000/S0000031.LOG

EDUID   : 397                  EDUNAME: db2logmgr (SAMPLE)
FUNCTION: DB2 UDB, data protection services, sqlpgArchiveLogFile, probe:3150
MESSAGE : ADM1848W  Failed to archive log file "S0000031.LOG" to
          "/db2archives/db2inst/SAMPLE/NODE0000/LOGSTREAM0000/C00
          00000/" from
          "/home/db2inst/db2inst/NODE0000/SQL00001/LOGSTREAM0000/".
```

Configuration parameters for database logging

- https://www.ibm.com/docs/en/db2/11.5?topic=logging-configuration-parameters



Logging – Disk Full in Archives (2|5)

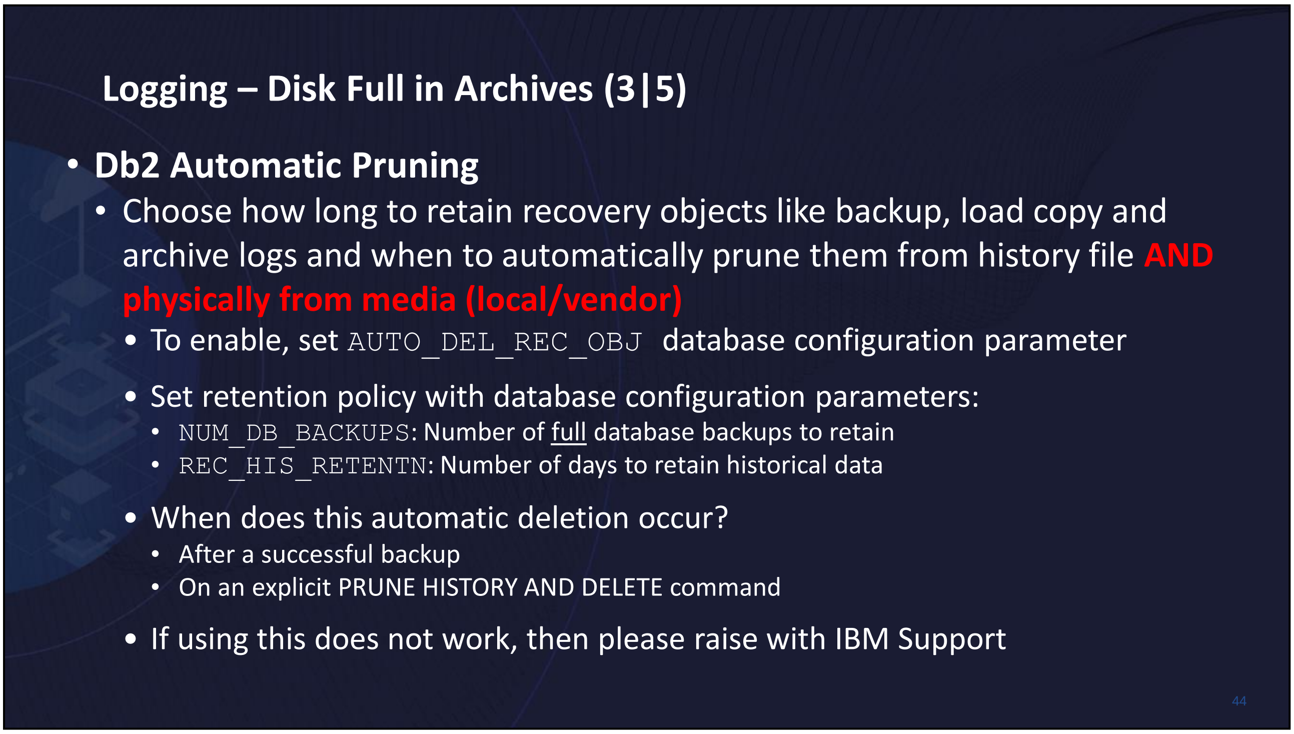
- **Detect in advance**
  - Set up a monitoring script that reports physical disk space left on the file system where the log archive path(s) exist
- **React**
  - Cleanup whatever is not needed
  - May need to do manually at first ➔ BE CAREFUL not to delete too much
  - **Avoid the need for manual cleanup by letting Db2 manage**
    ➔ Use Db2 Automatic Pruning

Configuration parameters for database logging

- https://www.ibm.com/docs/en/db2/11.5?topic=logging-configuration-parameters

Automating recovery history file pruning

- https://www.ibm.com/docs/en/db2/11.5?topic=file-automating-recovery-history-pruning

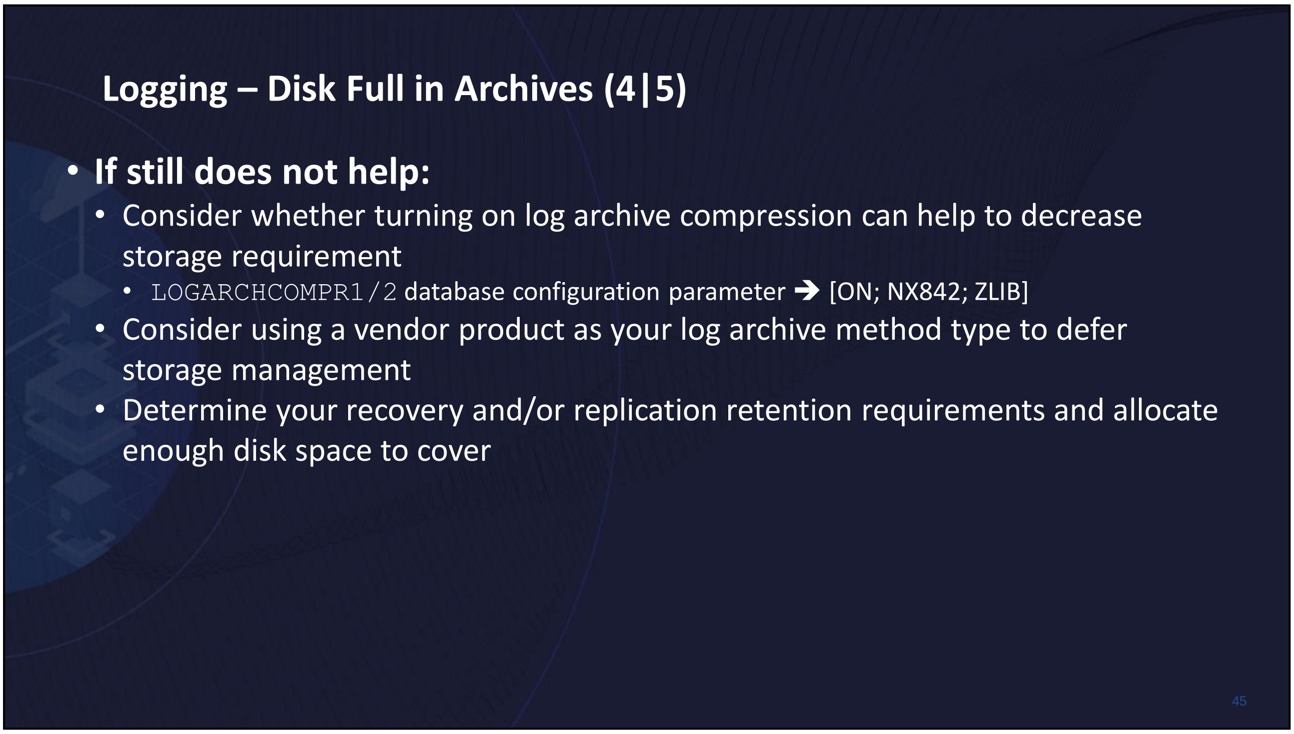


Logging – Disk Full in Archives (3|5)

- **Db2 Automatic Pruning**
  - Choose how long to retain recovery objects like backup, load copy and archive logs and when to automatically prune them from history file **AND physically from media (local/vendor)**
    - To enable, set `AUTO_DEL_REC_OBJ` database configuration parameter
    - Set retention policy with database configuration parameters:
      - `NUM_DB_BACKUPS`: Number of full database backups to retain
      - `REC_HIS_RETENTN`: Number of days to retain historical data
    - When does this automatic deletion occur?
      - After a successful backup
      - On an explicit PRUNE HISTORY AND DELETE command
    - If using this does not work, then please raise with IBM Support

Automating recovery history file pruning

- https://www.ibm.com/docs/en/db2/11.5?topic=file-automating-recovery-history-pruning

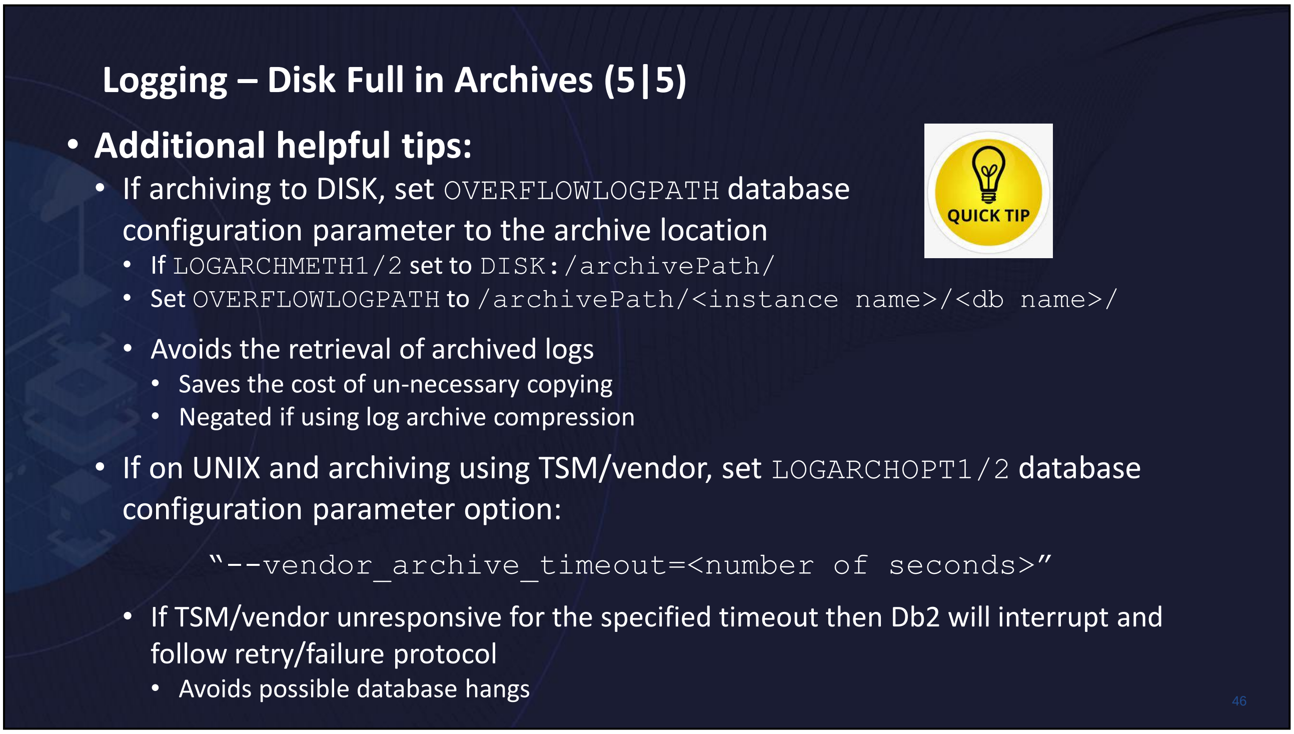


Logging – Disk Full in Archives (4|5)

- **If still does not help:**
  - Consider whether turning on log archive compression can help to decrease storage requirement
    - `LOGARCHCOMPR1/2` database configuration parameter ➔ [ON; NX842; ZLIB]
  - Consider using a vendor product as your log archive method type to defer storage management
  - Determine your recovery and/or replication retention requirements and allocate enough disk space to cover

Configuration parameters for database logging

- https://www.ibm.com/docs/en/db2/11.5?topic=logging-configuration-parameters



Logging – Disk Full in Archives (5|5)

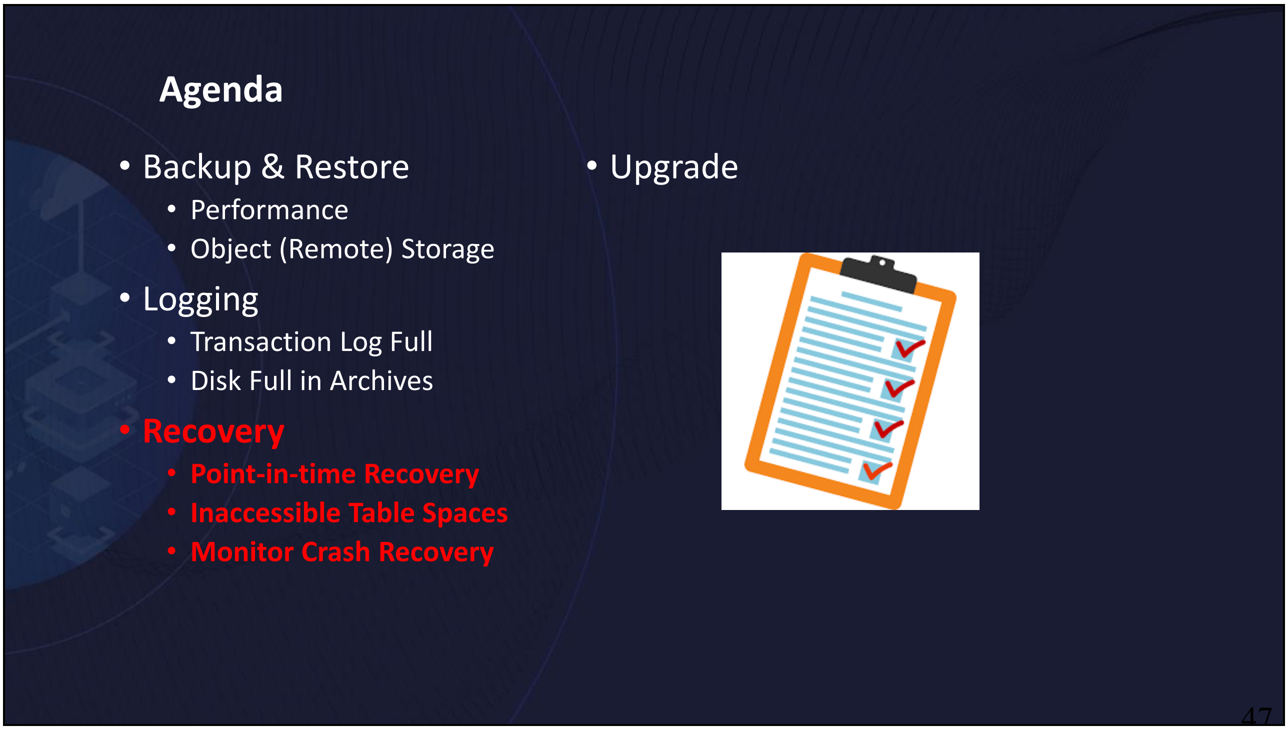
- **Additional helpful tips:**
  - If archiving to DISK, set `OVERFLOWLOGPATH` database configuration parameter to the archive location
    - If `LOGARCHMETH1/2` set to `DISK:/archivePath/`
    - Set `OVERFLOWLOGPATH` to `/archivePath/<instance name>/<db name>/`
    - Avoids the retrieval of archived logs
      - Saves the cost of un-necessary copying
      - Negated if using log archive compression
  - If on UNIX and archiving using TSM/vendor, set `LOGARCHOPT1/2` database configuration parameter option:

```
“--vendor_archive_timeout=<number of seconds>”
```

    - If TSM/vendor unresponsive for the specified timeout then Db2 will interrupt and follow retry/failure protocol
      - Avoids possible database hangs

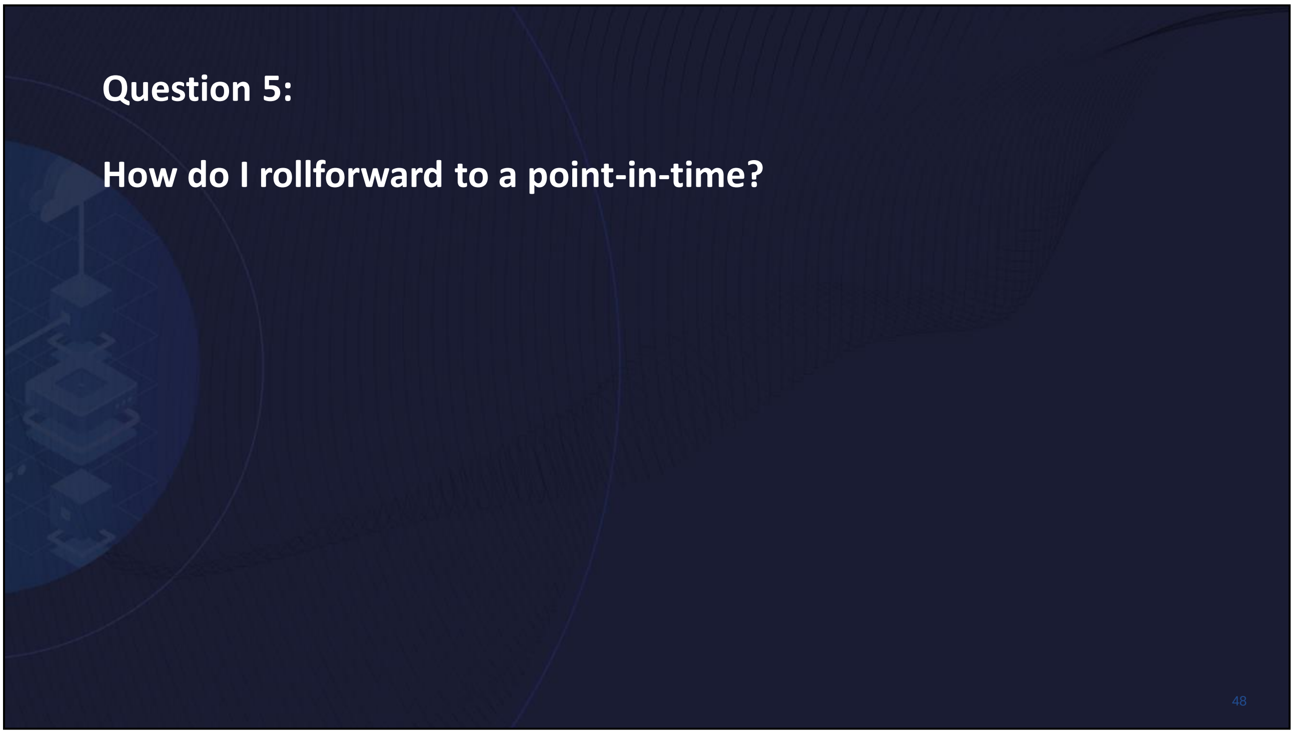
Using an overflow log path

- https://www.ibm.com/docs/en/db2/11.5?topic=parameters-using-overflow-log-path



Agenda

- Backup & Restore
  - Performance
  - Object (Remote) Storage
- Logging
  - Transaction Log Full
  - Disk Full in Archives
- **Recovery**
  - **Point-in-time Recovery**
  - **Inaccessible Table Spaces**
  - **Monitor Crash Recovery**
- Upgrade

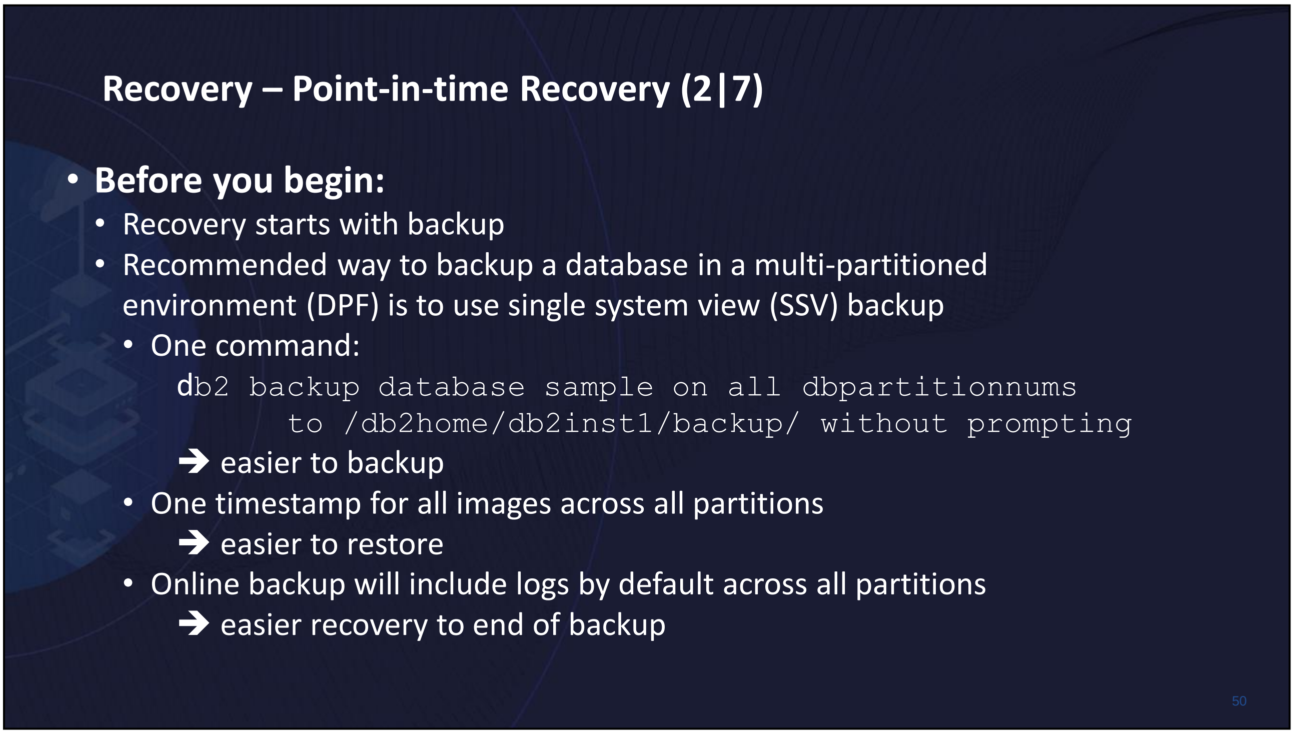


Question 5:

How do I rollforward to a point-in-time?

Recovery – Point-in-time Recovery (1|7)

- **Complications:**
  - Determining what point in time to use, esp. in a multi-partition environment (DPF)
    - Want to be precise to limit any data loss
    - Minimum recovery time (MRT)
  - Making all log files available
    - Archives or overflow?
    - Handling location in a multi-partition environment
  - If a multi-partition environment, coordinating command across partitions



Recovery – Point-in-time Recovery (2|7)

- **Before you begin:**
  - Recovery starts with backup
  - Recommended way to backup a database in a multi-partitioned environment (DPF) is to use single system view (SSV) backup
    - One command:

```
db2 backup database sample on all dbpartitionnums
        to /db2home/db2inst1/backup/ without prompting
```

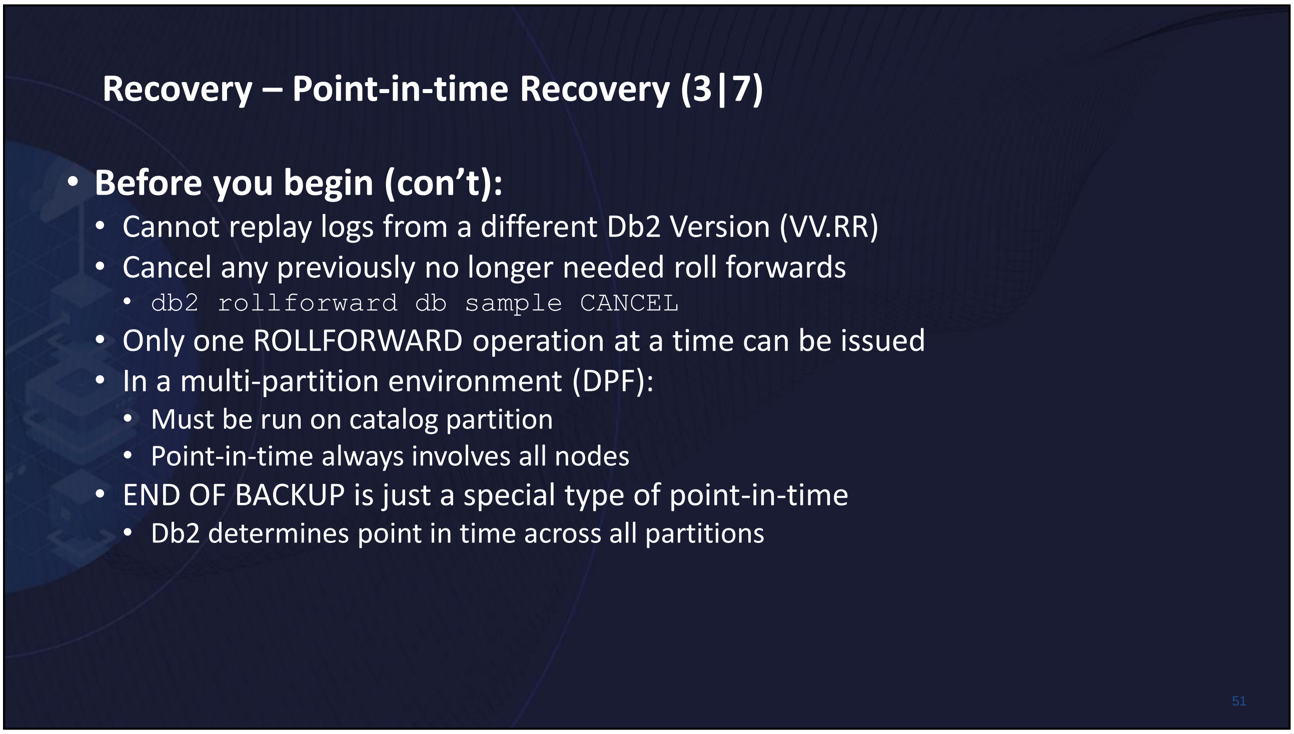
➔ easier to backup

- One timestamp for all images across all partitions

➔ easier to restore

- Online backup will include logs by default across all partitions

➔ easier recovery to end of backup

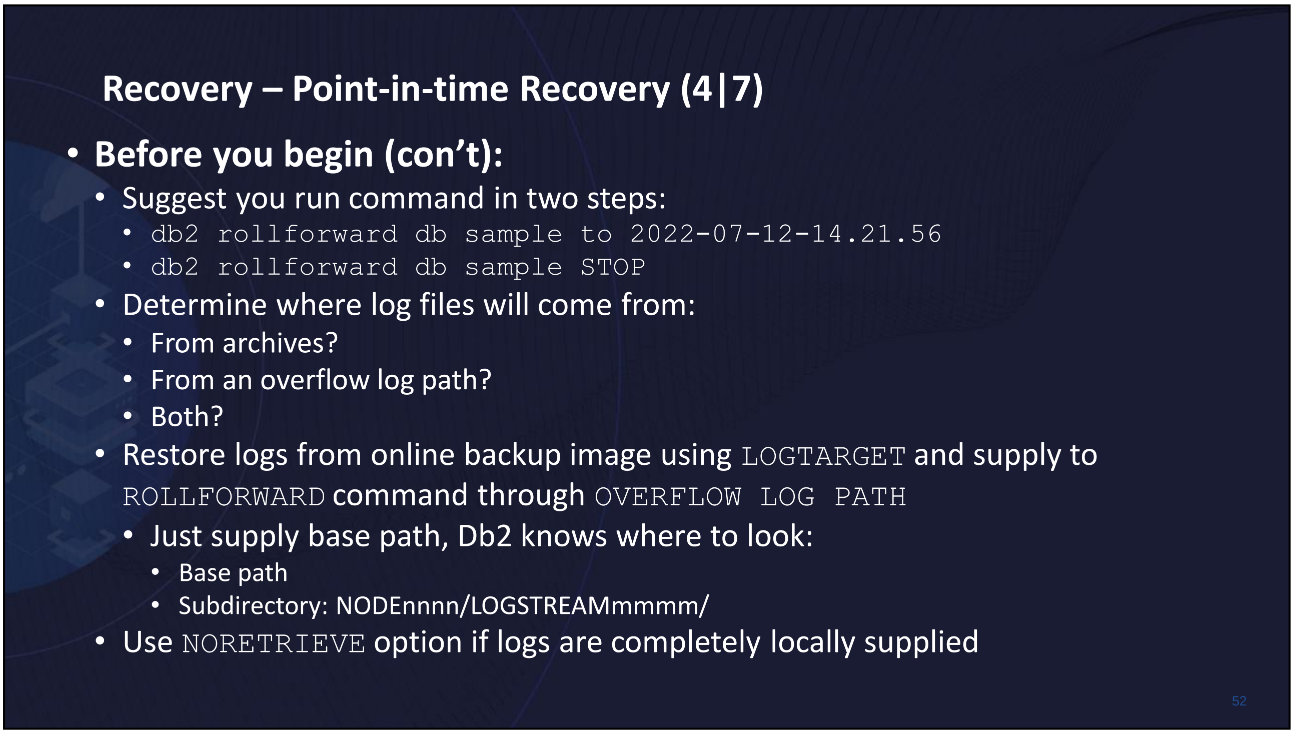


Recovery – Point-in-time Recovery (3|7)

- **Before you begin (con't):**
  - Cannot replay logs from a different Db2 Version (VV.RR)
  - Cancel any previously no longer needed roll forwards
    - `db2 rollforward db sample CANCEL`
  - Only one ROLLFORWARD operation at a time can be issued
  - In a multi-partition environment (DPF):
    - Must be run on catalog partition
    - Point-in-time always involves all nodes
  - END OF BACKUP is just a special type of point-in-time
    - Db2 determines point in time across all partitions

ROLLFORWARD DATABASE command

- https://www.ibm.com/docs/en/db2/11.5?topic=commands-rollforward-database

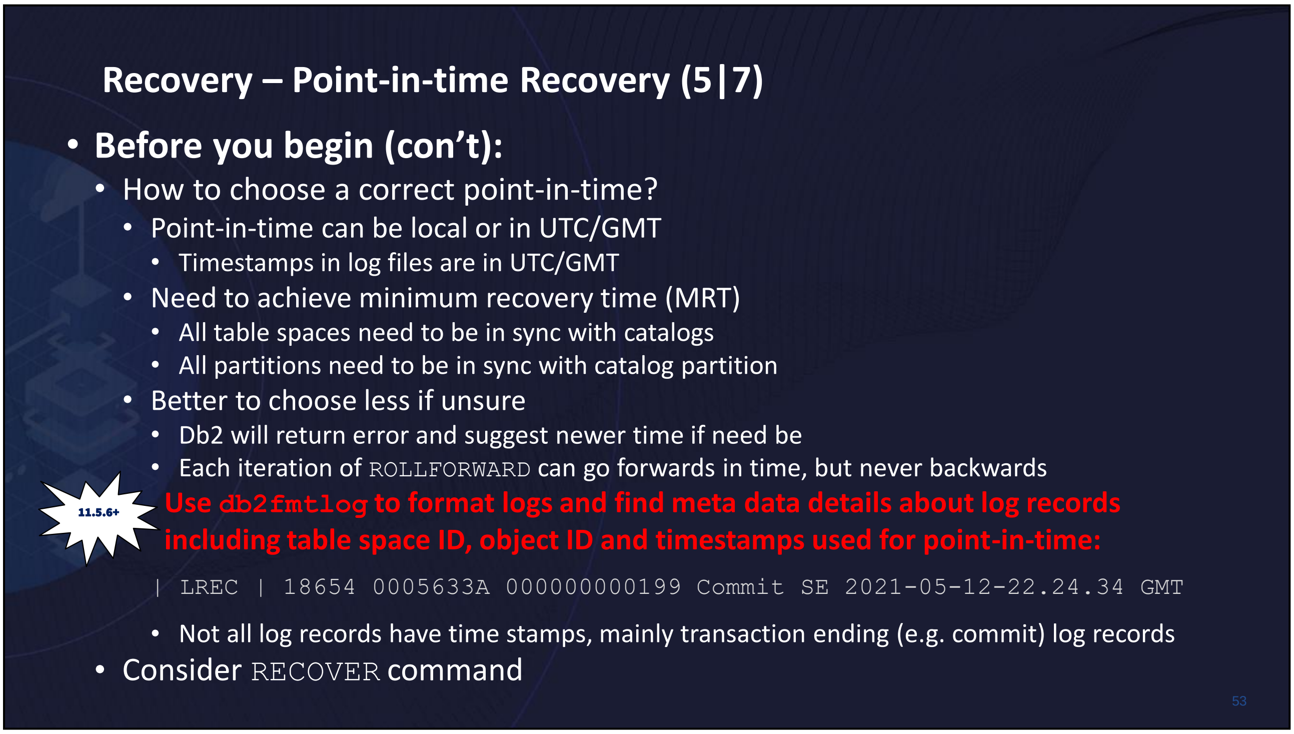


Recovery – Point-in-time Recovery (4|7)

- **Before you begin (con't):**
  - Suggest you run command in two steps:
    - `db2 rollforward db sample to 2022-07-12-14.21.56`
    - `db2 rollforward db sample STOP`
  - Determine where log files will come from:
    - From archives?
    - From an overflow log path?
    - Both?
  - Restore logs from online backup image using `LOGTARGET` and supply to `ROLLFORWARD` command through `OVERFLOW LOG PATH`
    - Just supply base path, Db2 knows where to look:
      - Base path
      - Subdirectory: NODEnnnn/LOGSTREAMmmmm/
  - Use `NORETRIEVE` option if logs are completely locally supplied

ROLLFORWARD DATABASE command

- https://www.ibm.com/docs/en/db2/11.5?topic=commands-rollforward-database



Recovery – Point-in-time Recovery (5|7)

- **Before you begin (con't):**
  - How to choose a correct point-in-time?
    - Point-in-time can be local or in UTC/GMT
      - Timestamps in log files are in UTC/GMT
    - Need to achieve minimum recovery time (MRT)
      - All table spaces need to be in sync with catalogs
      - All partitions need to be in sync with catalog partition
    - Better to choose less if unsure
      - Db2 will return error and suggest newer time if need be
      - Each iteration of `ROLLFORWARD` can go forwards in time, but never backwards

11.5.6+

Use `db2fmtlog` to format logs and find meta data details about log records including table space ID, object ID and timestamps used for point-in-time:

```
| LREC | 18654 0005633A 000000000199 Commit SE 2021-05-12-22.24.34 GMT
```

- Not all log records have time stamps, mainly transaction ending (e.g. commit) log records
- Consider `RECOVER` command

ROLLFORWARD DATABASE command

- https://www.ibm.com/docs/en/db2/11.5?topic=commands-rollforward-database

How to determine the minimum recovery time for a rollforward operation

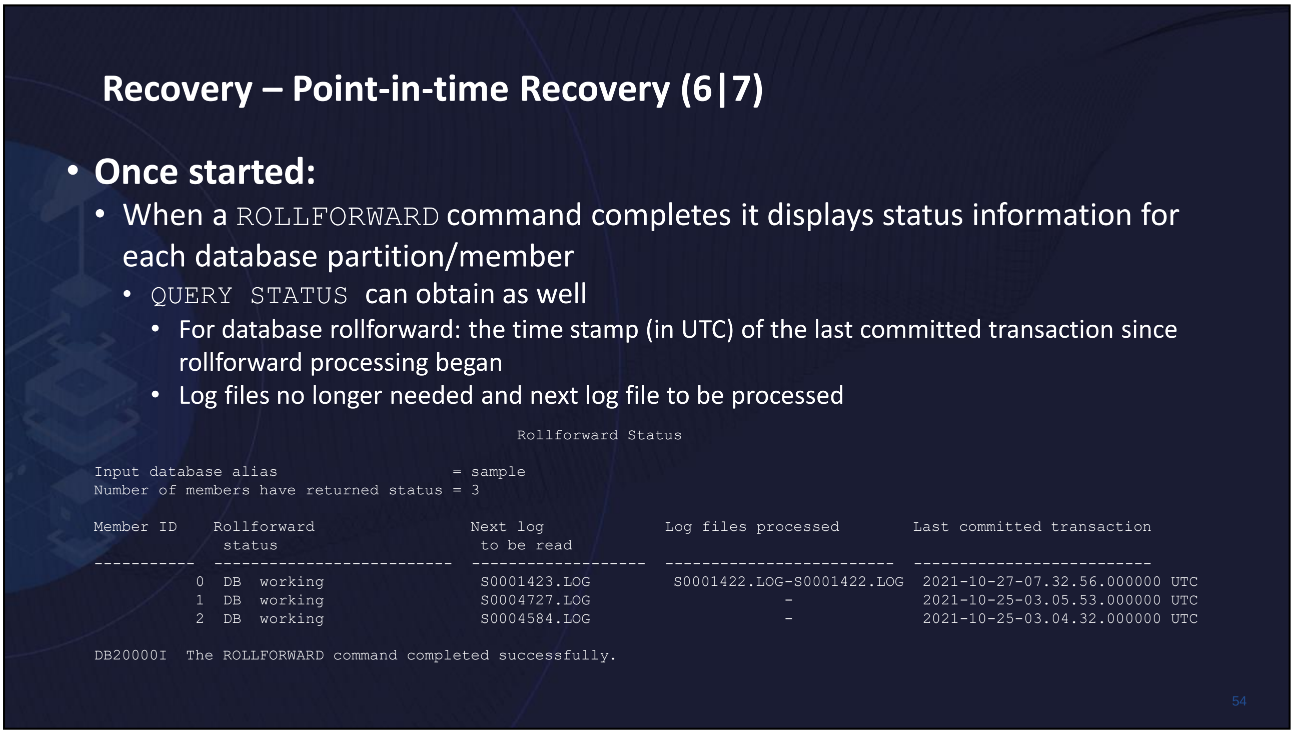
- https://www.ibm.com/support/pages/how-determine-minimum-recovery-time-rollforward-operation

db2fmtlog – Format and display log file information command

- https://www.ibm.com/docs/en/db2/11.5?topic=commands-db2fmtlog-format-display-log-file-information-tool

RECOVER DATABASE command

- https://www.ibm.com/docs/en/db2/11.5?topic=commands-recover-database



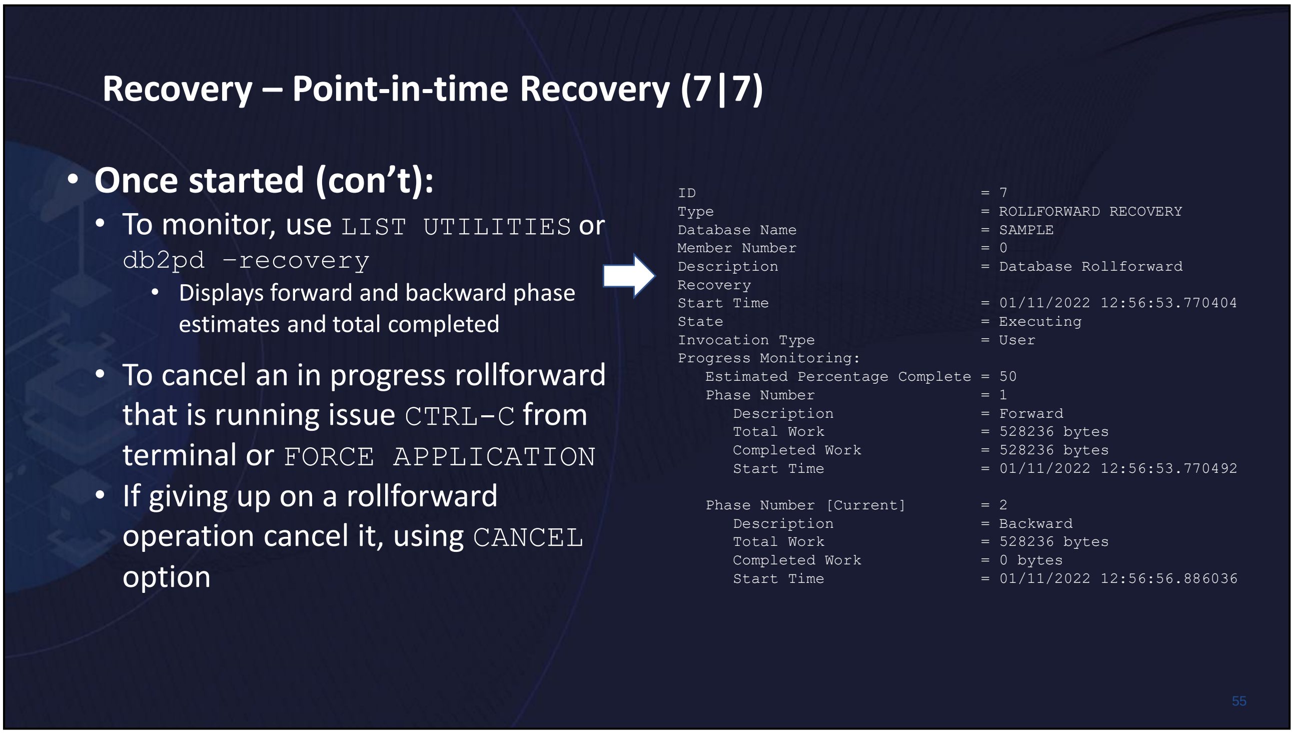
Recovery – Point-in-time Recovery (6|7)

- **Once started:**
  - When a `ROLLFORWARD` command completes it displays status information for each database partition/member
    - `QUERY STATUS` can obtain as well
      - For database rollforward: the time stamp (in UTC) of the last committed transaction since rollforward processing began
      - Log files no longer needed and next log file to be processed

```
                                              Rollforward Status

Input database alias                   = sample
Number of members have returned status = 3

Member ID    Rollforward                 Next log            Log files processed        Last committed transaction
              status                      to be read
-----------  --------------------------  -------------------  -------------------------  --------------------------
          0  DB  working                  S0001423.LOG         S0001422.LOG-S0001422.LOG  2021-10-27-07.32.56.000000 UTC
          1  DB  working                  S0004727.LOG                     -               2021-10-25-03.05.53.000000 UTC
          2  DB  working                  S0004584.LOG                     -               2021-10-25-03.04.32.000000 UTC

DB20000I  The ROLLFORWARD command completed successfully.
```

ROLLFORWARD DATABASE command

- https://www.ibm.com/docs/en/db2/11.5?topic=commands-rollforward-database



Recovery – Point-in-time Recovery (7|7)

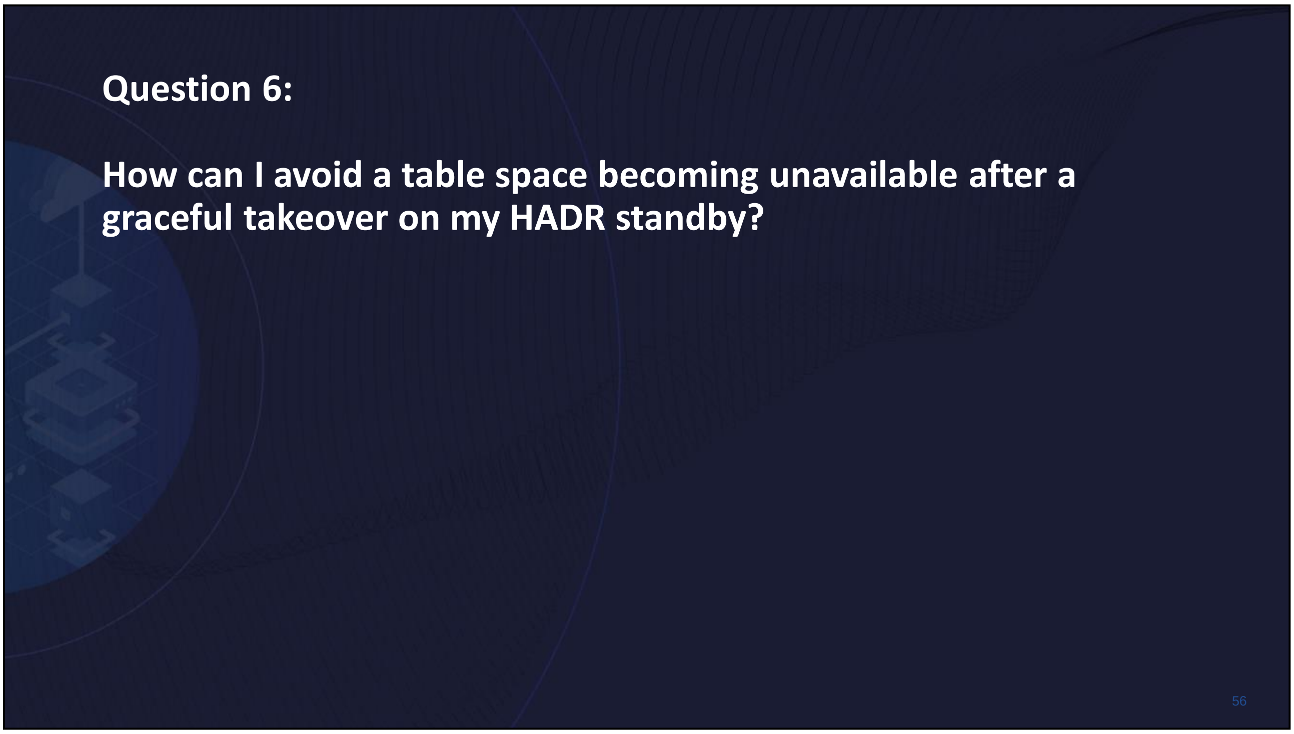
- **Once started (con't):**
  - To monitor, use `LIST UTILITIES` or `db2pd -recovery`
    - Displays forward and backward phase estimates and total completed
  - To cancel an in progress rollforward that is running issue `CTRL-C` from terminal or `FORCE APPLICATION`
  - If giving up on a rollforward operation cancel it, using `CANCEL` option

```
ID                                 = 7
Type                               = ROLLFORWARD RECOVERY
Database Name                      = SAMPLE
Member Number                      = 0
Description                        = Database Rollforward
Recovery
Start Time                         = 01/11/2022 12:56:53.770404
State                              = Executing
Invocation Type                    = User
Progress Monitoring:
   Estimated Percentage Complete = 50
   Phase Number                    = 1
      Description                  = Forward
      Total Work                   = 528236 bytes
      Completed Work               = 528236 bytes
      Start Time                   = 01/11/2022 12:56:53.770492

   Phase Number [Current]          = 2
      Description                  = Backward
      Total Work                   = 528236 bytes
      Completed Work               = 0 bytes
      Start Time                   = 01/11/2022 12:56:56.886036
```

55

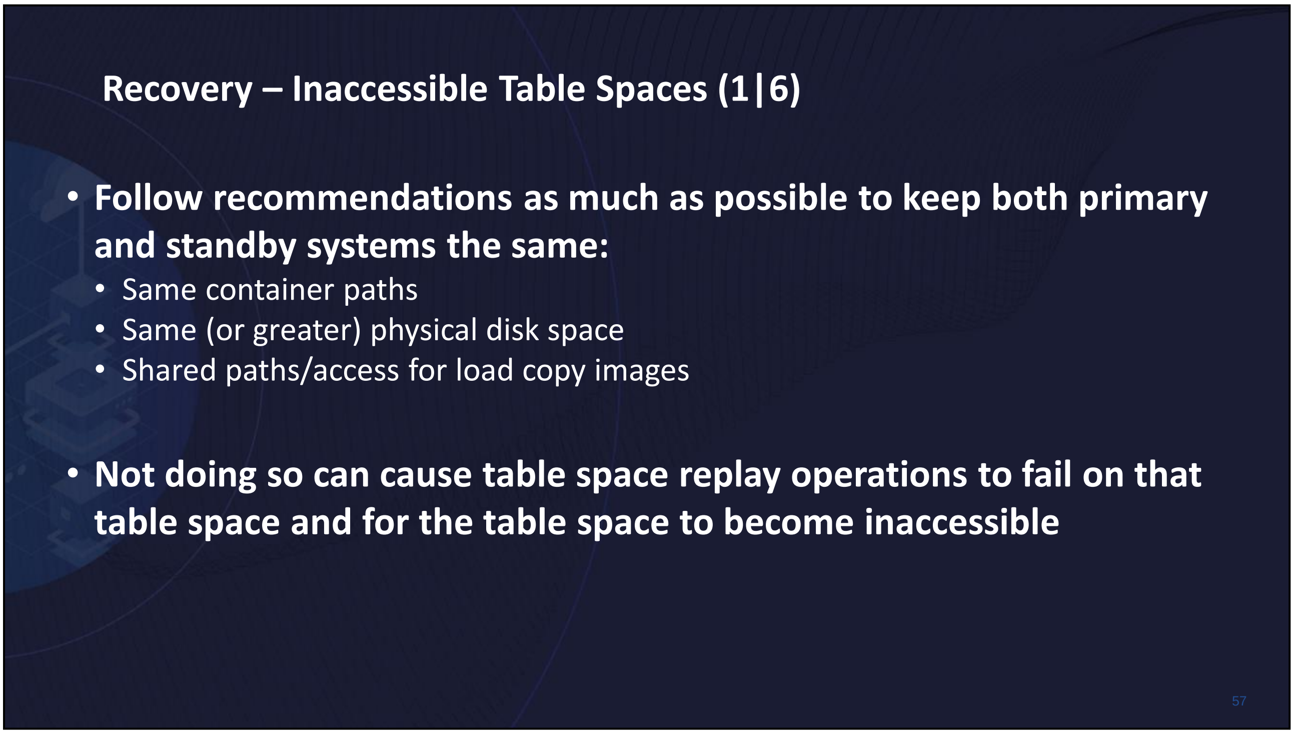
ROLLFORWARD DATABASE command

- https://www.ibm.com/docs/en/db2/11.5?topic=commands-rollforward-database



Question 6:

How can I avoid a table space becoming unavailable after a graceful takeover on my HADR standby?



Recovery – Inaccessible Table Spaces (1|6)

- **Follow recommendations as much as possible to keep both primary and standby systems the same:**
  - Same container paths
  - Same (or greater) physical disk space
  - Shared paths/access for load copy images

- **Not doing so can cause table space replay operations to fail on that table space and for the table space to become inaccessible**

57

High availability disaster recovery (HADR)

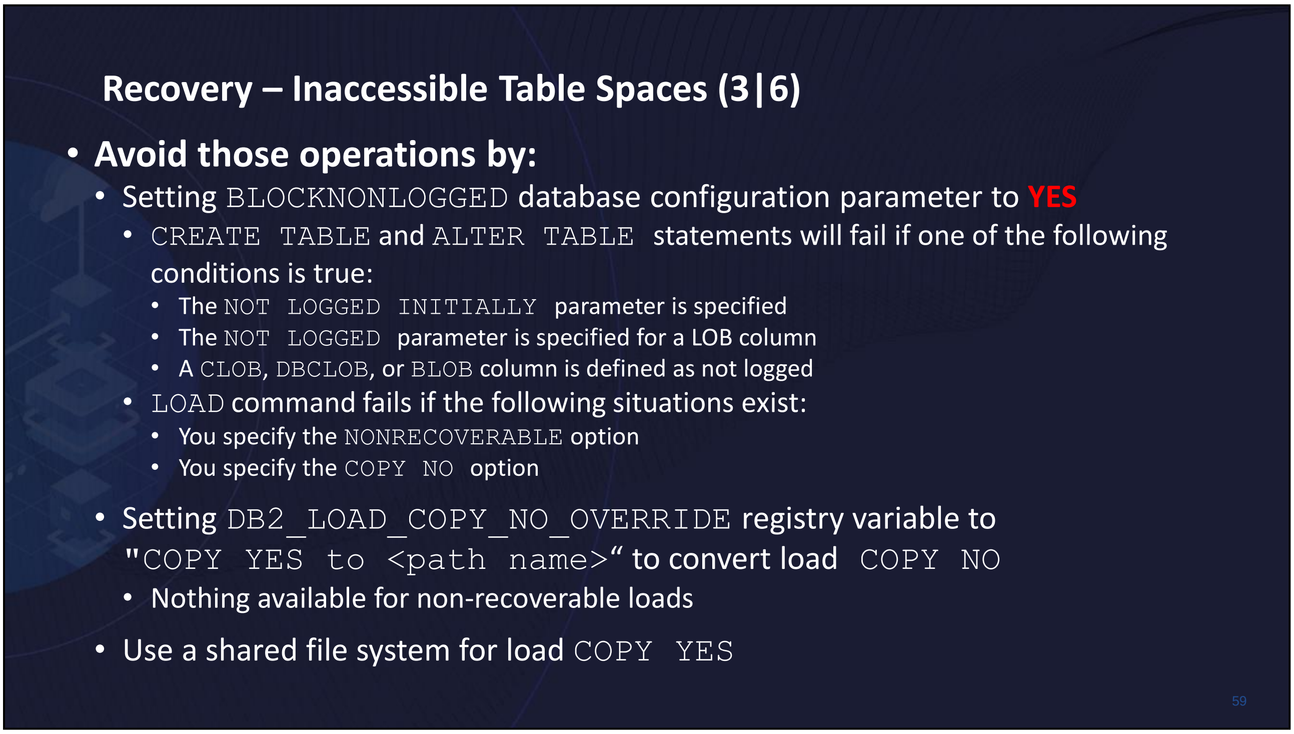
https://www.ibm.com/docs/en/db2/11.5?topic=server-high-availability-disaster-recovery-hadr

Recovery – Inaccessible Table Spaces (2|6)

- **Certain operations can place a table space and/or a table into an inaccessible state on the standby:**
  - Not logged operations
  - Non-recoverable load / load `COPY NO`
  - Load `COPY YES` image not available for replay

High availability disaster recovery (HADR)

https://www.ibm.com/docs/en/db2/11.5?topic=server-high-availability-disaster-recovery-hadr



Recovery – Inaccessible Table Spaces (3|6)

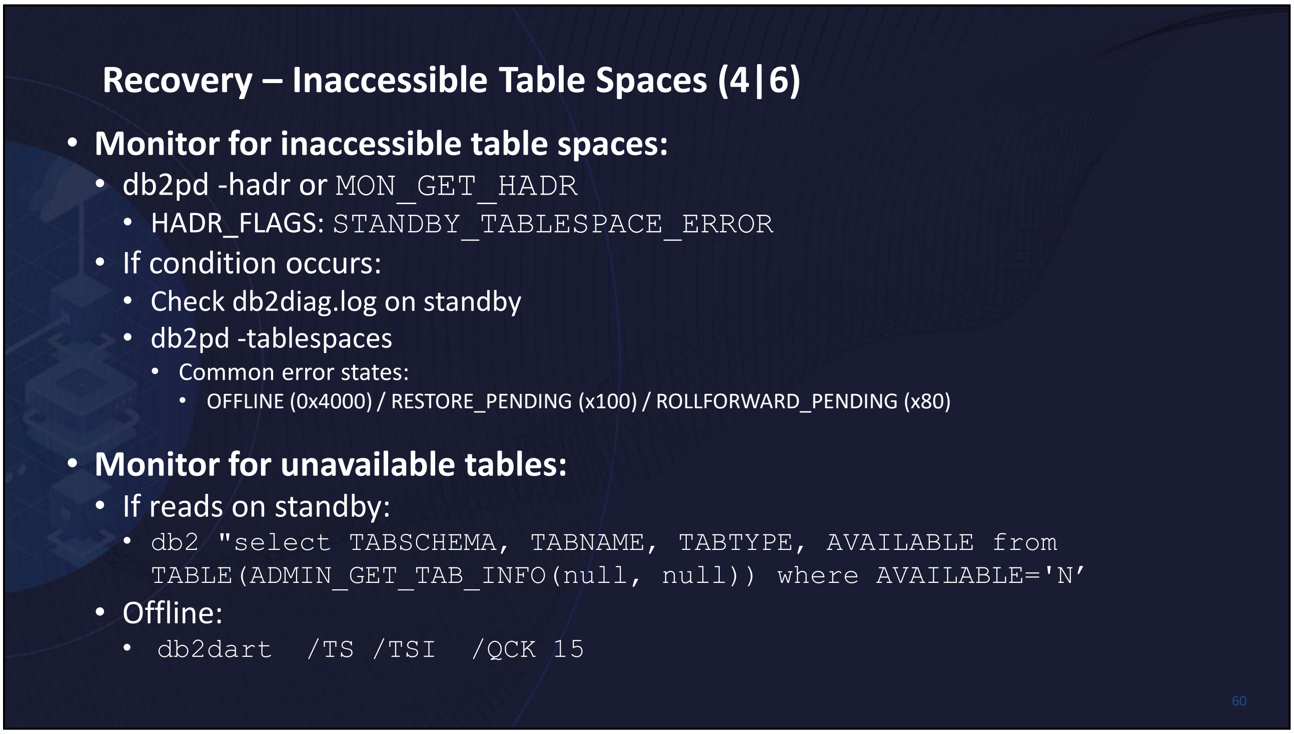
- **Avoid those operations by:**
  - Setting `BLOCKNONLOGGED` database configuration parameter to **YES**
    - `CREATE TABLE` and `ALTER TABLE` statements will fail if one of the following conditions is true:
      - The `NOT LOGGED INITIALLY` parameter is specified
      - The `NOT LOGGED` parameter is specified for a LOB column
      - A `CLOB`, `DBCLOB`, or `BLOB` column is defined as not logged
    - `LOAD` command fails if the following situations exist:
      - You specify the `NONRECOVERABLE` option
      - You specify the `COPY NO` option
  - Setting `DB2_LOAD_COPY_NO_OVERRIDE` registry variable to `"COPY YES to <path name>“` to convert load `COPY NO`
    - Nothing available for non-recoverable loads
  - Use a shared file system for load `COPY YES`

blocknonlogged - Block creation of tables that allow non-logged activity configuration parameter

https://www.ibm.com/docs/en/db2/11.5?topic=dcp-blocknonlogged-block-creation-tables-that-allow-non-logged-activity

DB2_LOAD_COPY_NO_OVERRIDE and controlling non-recoverable load operations

https://www.ibm.com/support/pages/db2loadcopynooverride-and-controlling-non-recoverable-load-operations



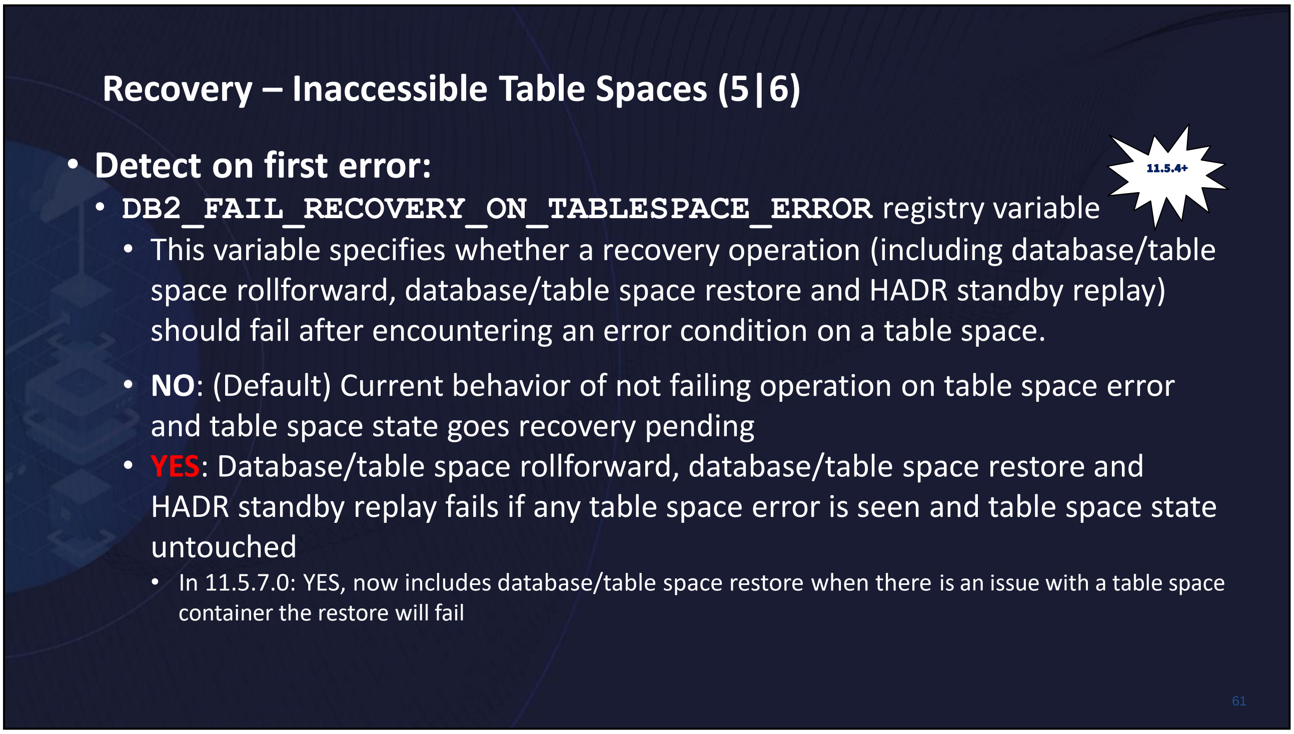
Recovery – Inaccessible Table Spaces (4|6)

- **Monitor for inaccessible table spaces:**
  - db2pd -hadr or `MON_GET_HADR`
    - HADR_FLAGS: `STANDBY_TABLESPACE_ERROR`
  - If condition occurs:
    - Check db2diag.log on standby
    - db2pd -tablespaces
      - Common error states:
        - OFFLINE (0x4000) / RESTORE_PENDING (x100) / ROLLFORWARD_PENDING (x80)
- **Monitor for unavailable tables:**
  - If reads on standby:
    - `db2 "select TABSCHEMA, TABNAME, TABTYPE, AVAILABLE from TABLE(ADMIN_GET_TAB_INFO(null, null)) where AVAILABLE='N’`
  - Offline:
    - `db2dart  /TS /TSI  /QCK 15`

60

Monitoring and identifying tablespaces in invalid or error state, or tables in Inoperative state on the HADR Standby database

- https://www.ibm.com/support/pages/monitoring-and-identifying-tablespaces-invalid-or-error-state-or-tables-inoperative-state-hadr-standby-database



Recovery – Inaccessible Table Spaces (5|6)

11.5.4+

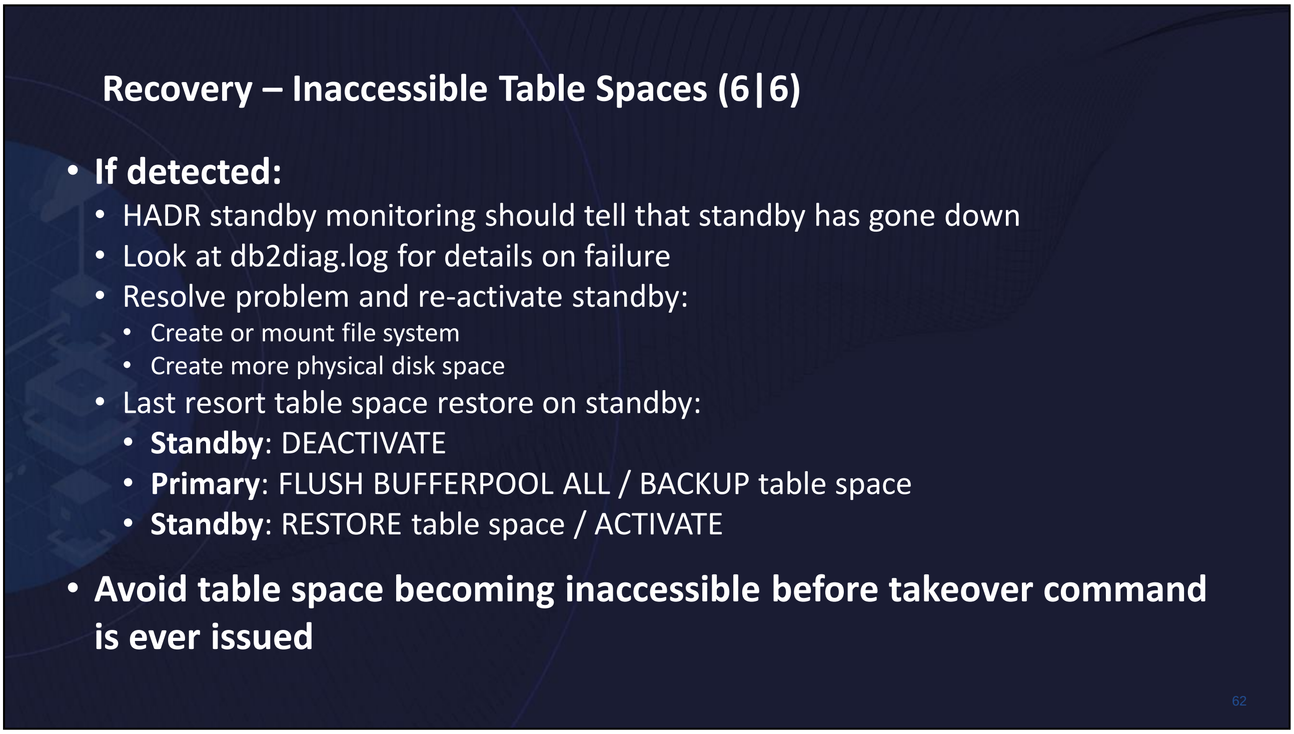
- **Detect on first error:**
  - `DB2_FAIL_RECOVERY_ON_TABLESPACE_ERROR` registry variable
    - This variable specifies whether a recovery operation (including database/table space rollforward, database/table space restore and HADR standby replay) should fail after encountering an error condition on a table space.
    - **NO**: (Default) Current behavior of not failing operation on table space error and table space state goes recovery pending
    - **YES**: Database/table space rollforward, database/table space restore and HADR standby replay fails if any table space error is seen and table space state untouched
      - In 11.5.7.0: YES, now includes database/table space restore when there is an issue with a table space container the restore will fail

Problem:

- Historically, Db2 recovery (restore, rollforward, HADR standby replay, crash recovery) tries to make available as much data as possible; any error in one table space does not stop the recovery operation from failing
- In recent years, a set of customers argue that all table spaces are important, so any failure in one (e.g. container issue) should cause recovery operation to fail; by not failing makes database un-useable post-recovery and increases outage due to having to figure out issue and then doing recovery again

DB2_FAIL_RECOVERY_ON_TABLESPACE_ERROR

- https://www.ibm.com/docs/en/db2/11.5?topic=variables-miscellaneous#r0005669__M_DB2_FAIL_RECOVERY_ON_TABLESPACE_ERROR



Recovery – Inaccessible Table Spaces (6|6)

- **If detected:**
  - HADR standby monitoring should tell that standby has gone down
  - Look at db2diag.log for details on failure
  - Resolve problem and re-activate standby:
    - Create or mount file system
    - Create more physical disk space
  - Last resort table space restore on standby:
    - **Standby**: DEACTIVATE
    - **Primary**: FLUSH BUFFERPOOL ALL / BACKUP table space
    - **Standby**: RESTORE table space / ACTIVATE
- **Avoid table space becoming inaccessible before takeover command is ever issued**

Problem:

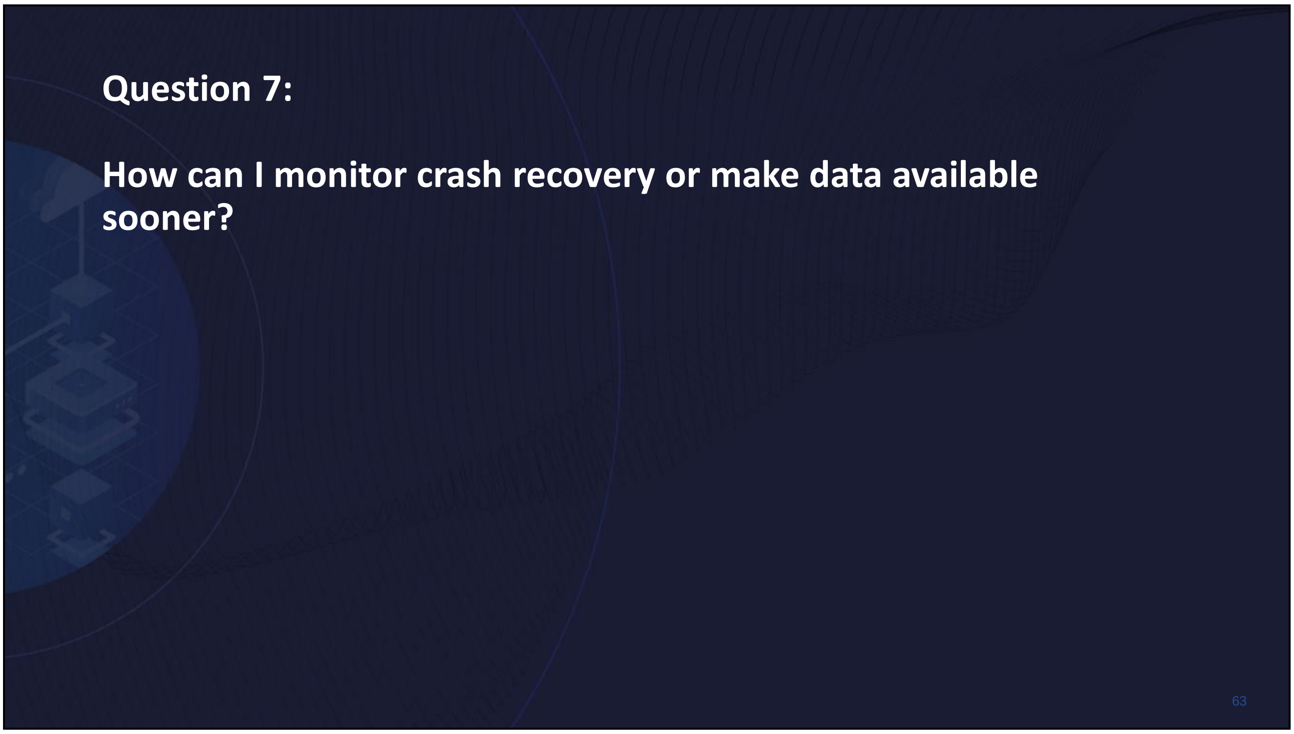
- Historically, Db2 recovery (restore, rollforward, HADR standby replay, crash recovery) tries to make available as much data as possible; any error in one table space does not stop the recovery operation from failing
- In recent years, a set of customers argue that all table spaces are important, so any failure in one (e.g. container issue) should cause recovery operation to fail; by not failing makes database un-useable post-recovery and increases outage due to having to figure out issue and then doing recovery again

DB2_FAIL_RECOVERY_ON_TABLESPACE_ERROR

- https://www.ibm.com/docs/en/db2/11.5?topic=variables-miscellaneous#r0005669__M_DB2_FAIL_RECOVERY_ON_TABLESPACE_ERROR

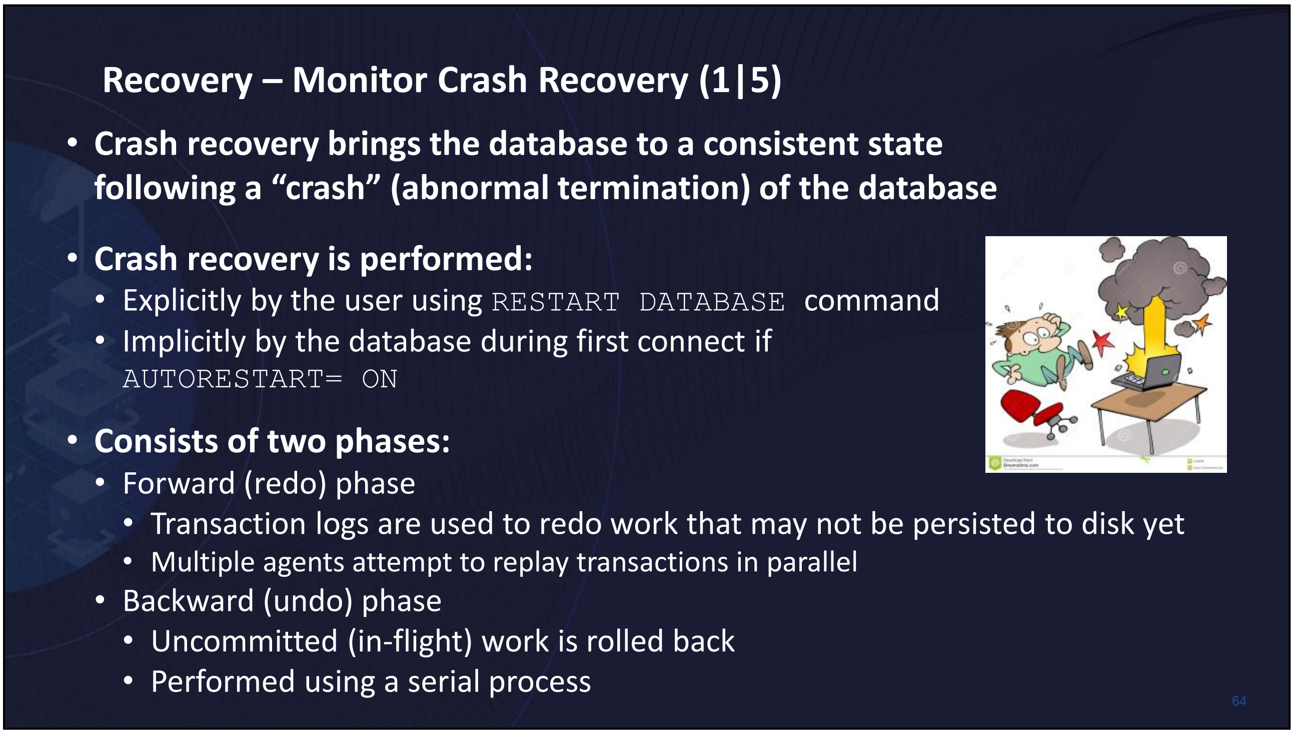
Recovering table space errors on an HADR standby database

- https://www.ibm.com/docs/en/db2/11.5?topic=databases-recovering-table-space-errors-hadr-standby-database



Question 7:

How can I monitor crash recovery or make data available sooner?



Recovery – Monitor Crash Recovery (1|5)

- **Crash recovery brings the database to a consistent state following a "crash" (abnormal termination) of the database**
- **Crash recovery is performed:**
  - Explicitly by the user using `RESTART DATABASE` command
  - Implicitly by the database during first connect if `AUTORESTART= ON`
- **Consists of two phases:**
  - Forward (redo) phase
    - Transaction logs are used to redo work that may not be persisted to disk yet
    - Multiple agents attempt to replay transactions in parallel
  - Backward (undo) phase
    - Uncommitted (in-flight) work is rolled back
    - Performed using a serial process

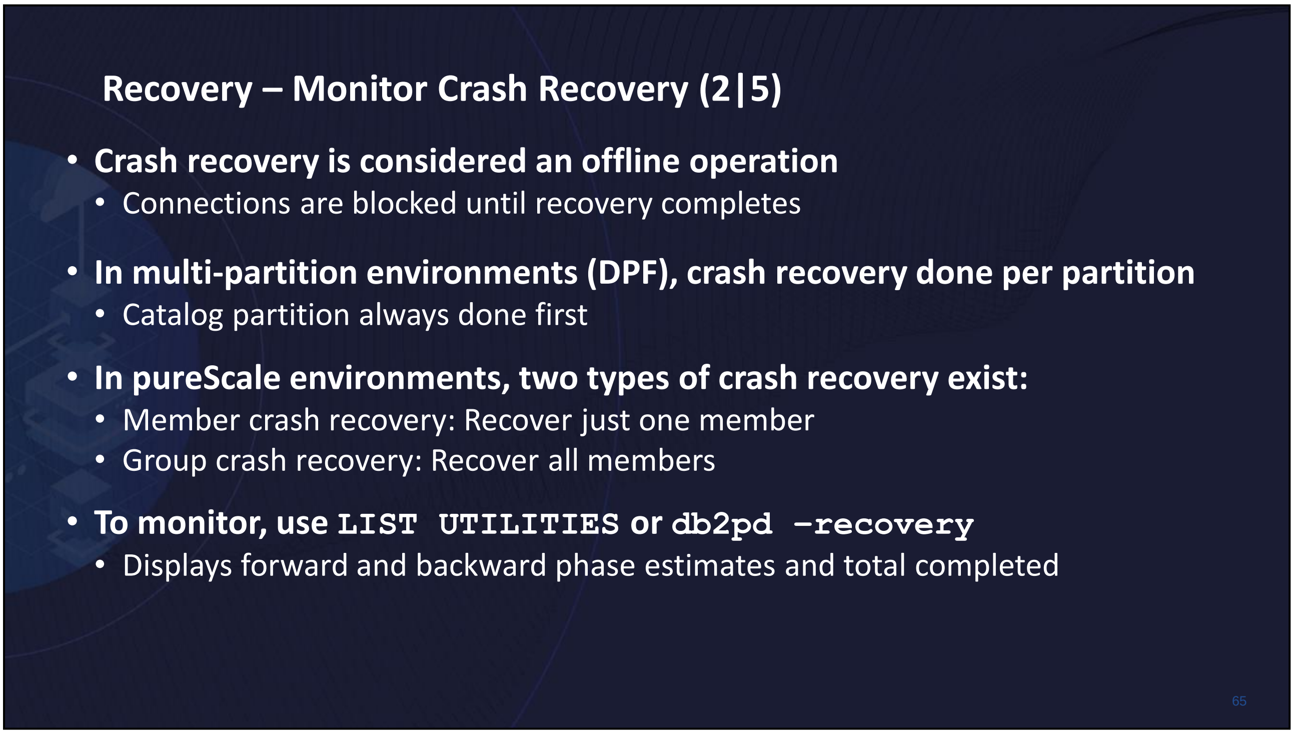
64

At the time of a database failure resulting from an event such as a power failure, there might have been changes to the database which:

- Have not been committed, but updated the data in the buffer pool
- Have been committed, but have not been written from the buffer pool to the disk
- Have been committed and written from the buffer pool to the disk.

When a database is restarted, the log files will be used to perform a crash recovery of the database which ensures that the database is left in a consistent state (that is, all committed transactions are applied to the database and all uncommitted transactions are not applied to the database). To determine which records from the log file need to be applied to the database, the database manager uses information recorded in a log control file. (The database manager actually maintains two copies of the log control file, `SQLOGCTL.LFH.1` and `SQLOGCTL.LFH.2`, so that if one copy is damaged, the database manager can still use the other copy.) These log control files are periodically written to disk, and, depending on the frequency of this event, the database manager might be applying log records of committed transactions or applying log records that describe changes that have already been written from the buffer pool to disk. These log records have no impact on the database, but applying them introduces some additional processing time into the database restart process.

The log control files are always written to disk when a log file is full, and during soft checkpoints. You can use this configuration parameter to control the frequency of soft checkpoints.

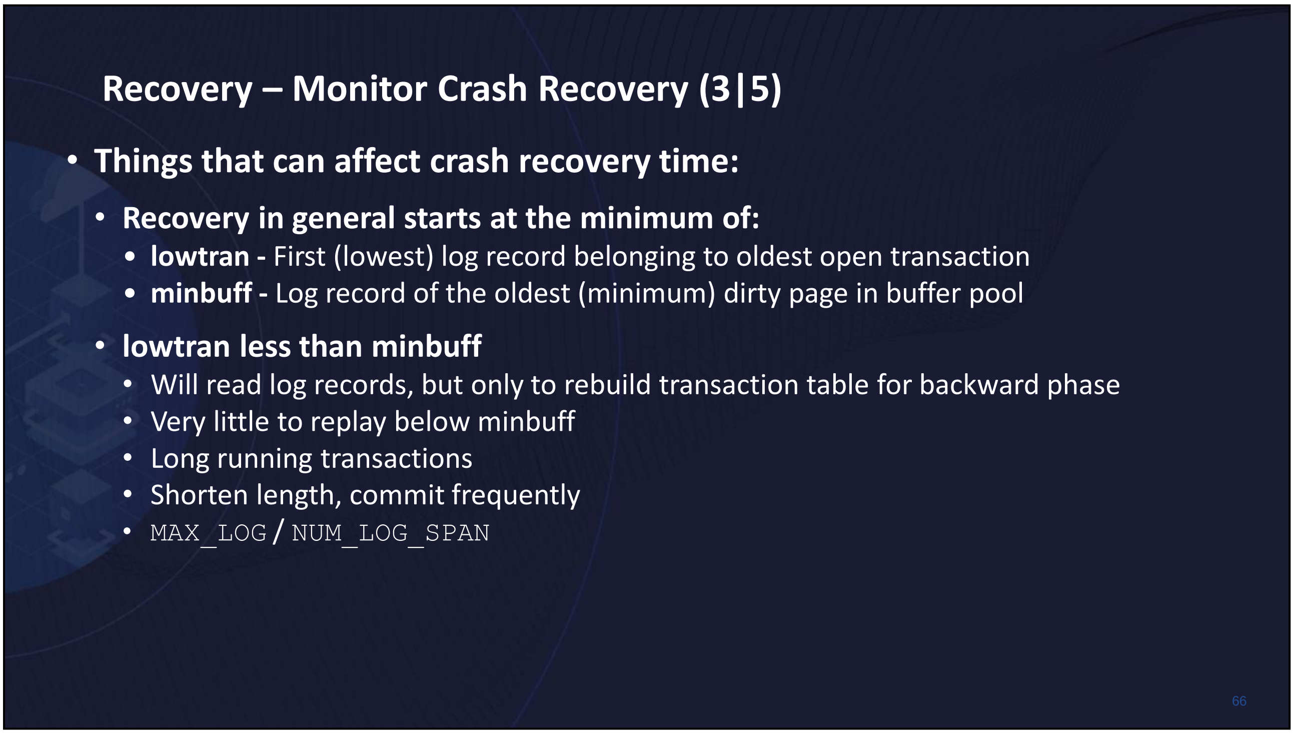


Recovery – Monitor Crash Recovery (2|5)

- **Crash recovery is considered an offline operation**
  - Connections are blocked until recovery completes
- **In multi-partition environments (DPF), crash recovery done per partition**
  - Catalog partition always done first
- **In pureScale environments, two types of crash recovery exist:**
  - Member crash recovery: Recover just one member
  - Group crash recovery: Recover all members
- **To monitor, use `LIST UTILITIES` or `db2pd -recovery`**
  - Displays forward and backward phase estimates and total completed

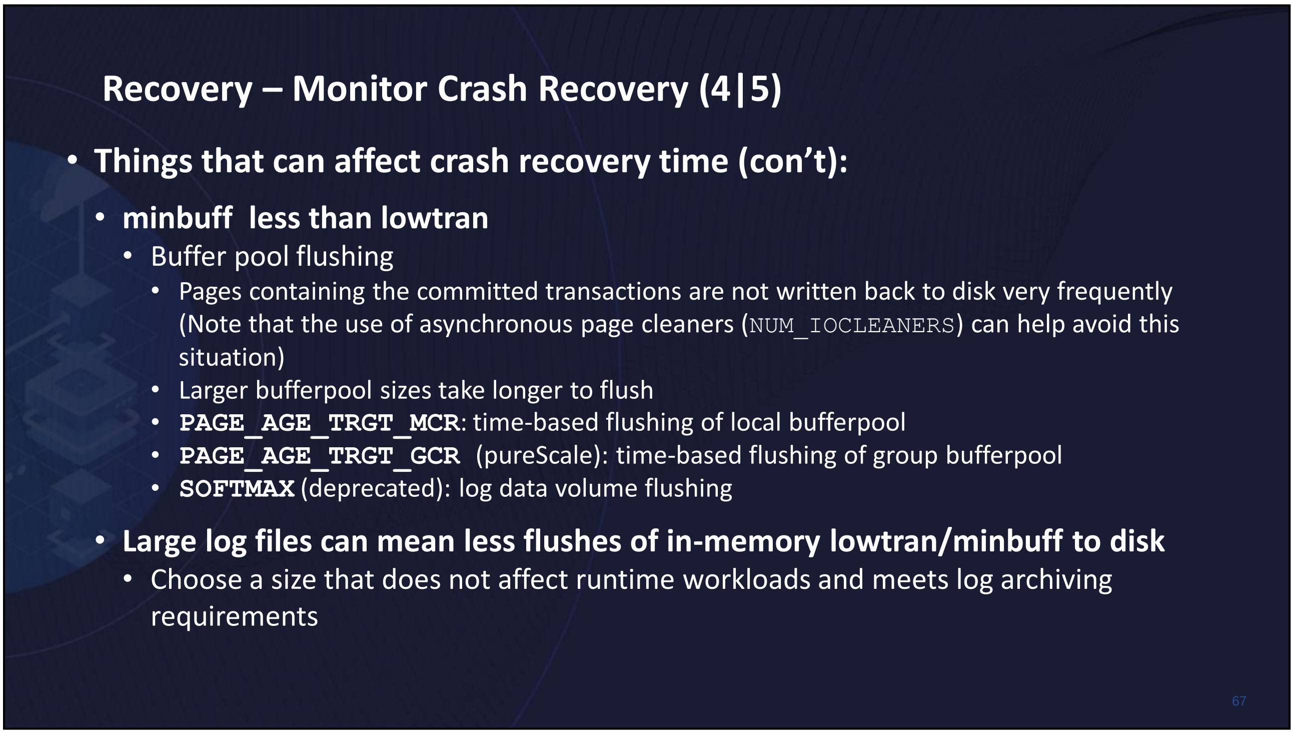
65

```
 LIST UTILITIES SHOW DETAIL

ID                                     = 1
Type                                   = CRASH RECOVERY
Database Name                          = SAMPLE
Member Number                          = 0
Description                            = Crash Recovery (or Rollforward Recovery)
Start Time                             = 04/29/2021 13:20:05.646020
State                                  = Executing
Invocation Type                        = User
Progress Monitoring:
          Estimated Percentage Complete = 75
          Phase Number [Current]        = 1
                    Description           = Forward
                    Total Work            = 163834 bytes
                    Completed Work        = 124145 bytes
                    Start Time            = 04/29/2021 13:20:05.646121
          Phase Number                  = 2
                    Description           = Backward
                    Total Work            = 163834 bytes
                    Completed Work        = 0 bytes
                    Start Time            = Not Started
```



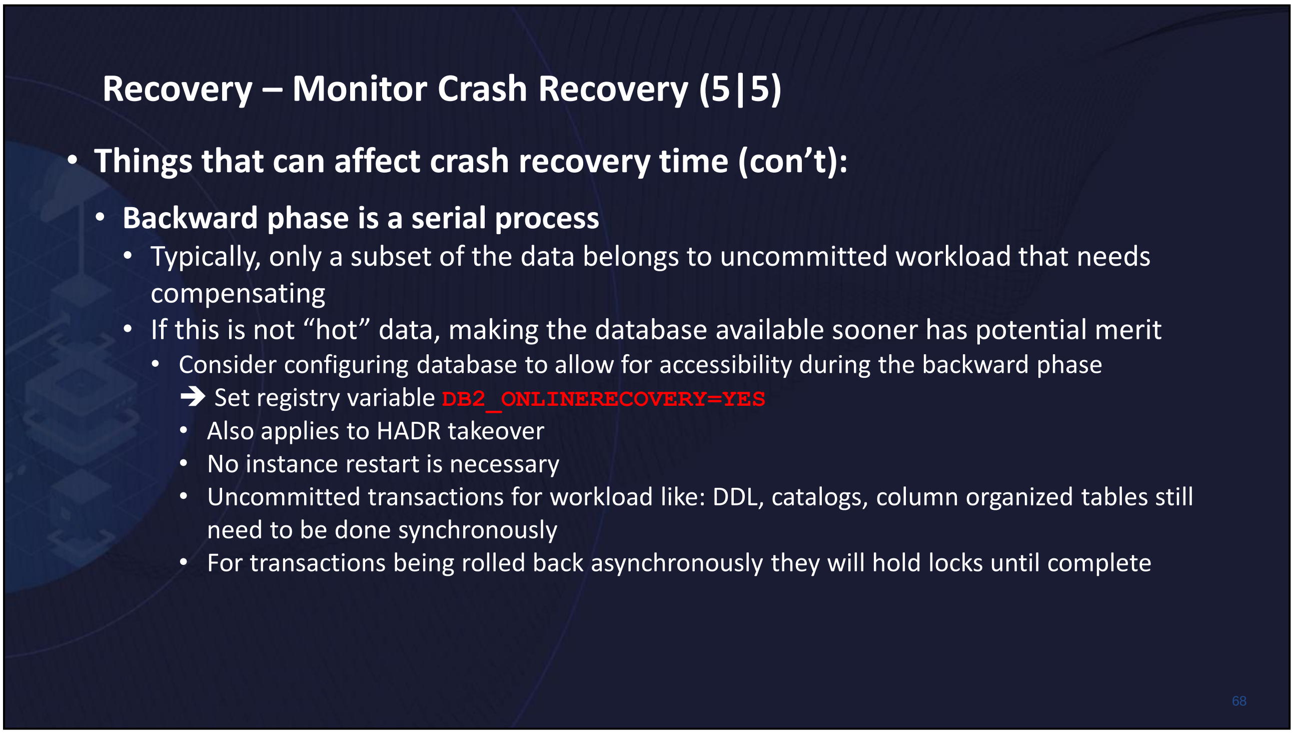
Recovery – Monitor Crash Recovery (3|5)

- **Things that can affect crash recovery time:**
  - **Recovery in general starts at the minimum of:**
    - **lowtran -** First (lowest) log record belonging to oldest open transaction
    - **minbuff -** Log record of the oldest (minimum) dirty page in buffer pool
  - **lowtran less than minbuff**
    - Will read log records, but only to rebuild transaction table for backward phase
    - Very little to replay below minbuff
    - Long running transactions
    - Shorten length, commit frequently
    - `MAX_LOG` / `NUM_LOG_SPAN`



Recovery – Monitor Crash Recovery (4|5)

- **Things that can affect crash recovery time (con't):**
  - **minbuff less than lowtran**
    - Buffer pool flushing
      - Pages containing the committed transactions are not written back to disk very frequently (Note that the use of asynchronous page cleaners (`NUM_IOCLEANERS`) can help avoid this situation)
      - Larger bufferpool sizes take longer to flush
      - **`PAGE_AGE_TRGT_MCR`**: time-based flushing of local bufferpool
      - **`PAGE_AGE_TRGT_GCR`** (pureScale): time-based flushing of group bufferpool
      - **`SOFTMAX`** (deprecated): log data volume flushing
  - **Large log files can mean less flushes of in-memory lowtran/minbuff to disk**
    - Choose a size that does not affect runtime workloads and meets log archiving requirements

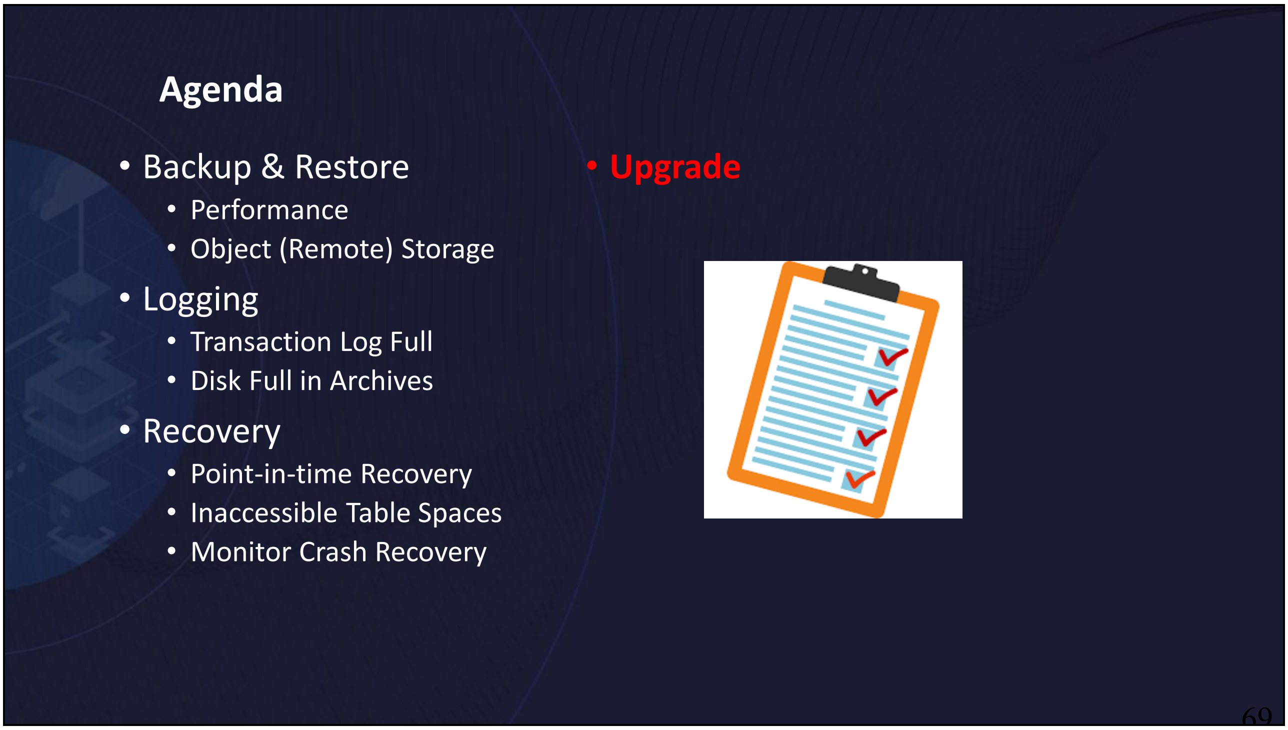


Recovery – Monitor Crash Recovery (5|5)

- **Things that can affect crash recovery time (con't):**
  - **Backward phase is a serial process**
    - Typically, only a subset of the data belongs to uncommitted workload that needs compensating
    - If this is not "hot" data, making the database available sooner has potential merit
      - Consider configuring database to allow for accessibility during the backward phase
        ➔ Set registry variable `DB2_ONLINERECOVERY=YES`
        - Also applies to HADR takeover
        - No instance restart is necessary
        - Uncommitted transactions for workload like: DDL, catalogs, column organized tables still need to be done synchronously
        - For transactions being rolled back asynchronously they will hold locks until complete

Database accessibility during Backward phase of crash recovery or HADR takeover

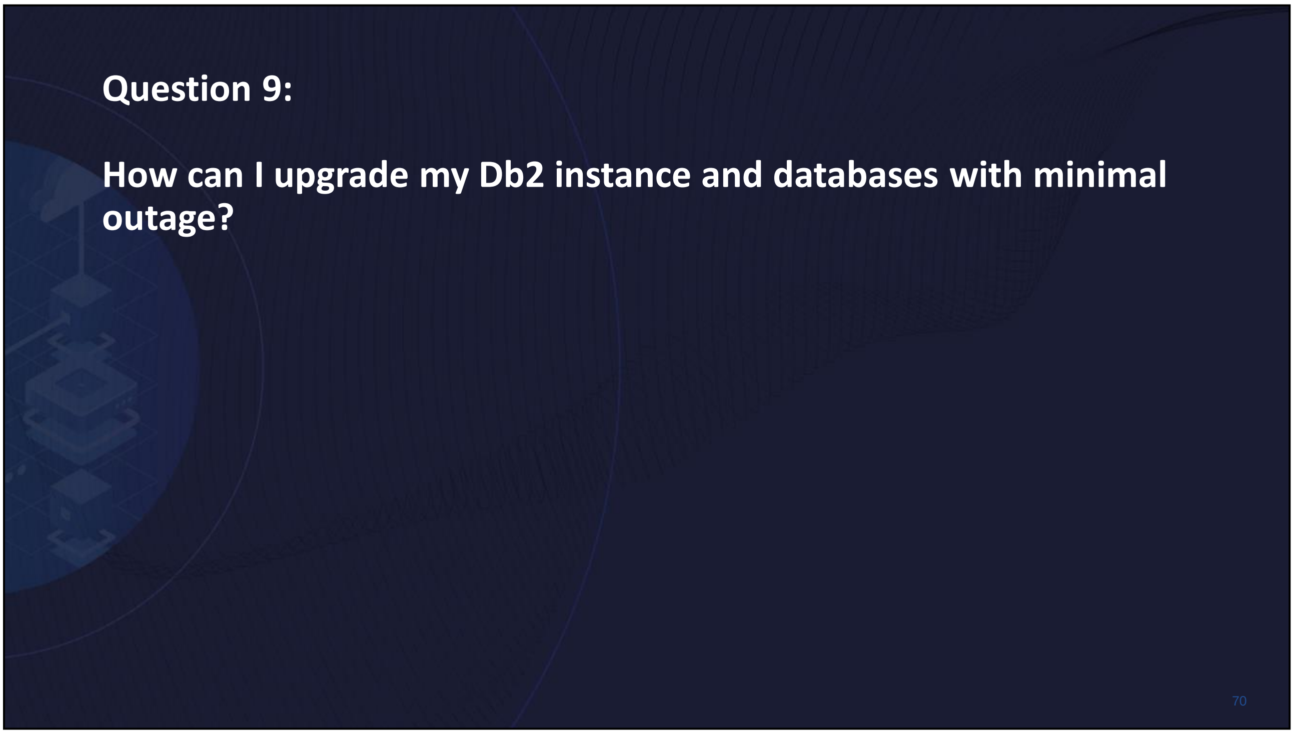
https://www.ibm.com/docs/en/db2/11.5?topic=cr-database-accessibility-during-backward-phase-crash-recovery-hadr-takeover



Agenda

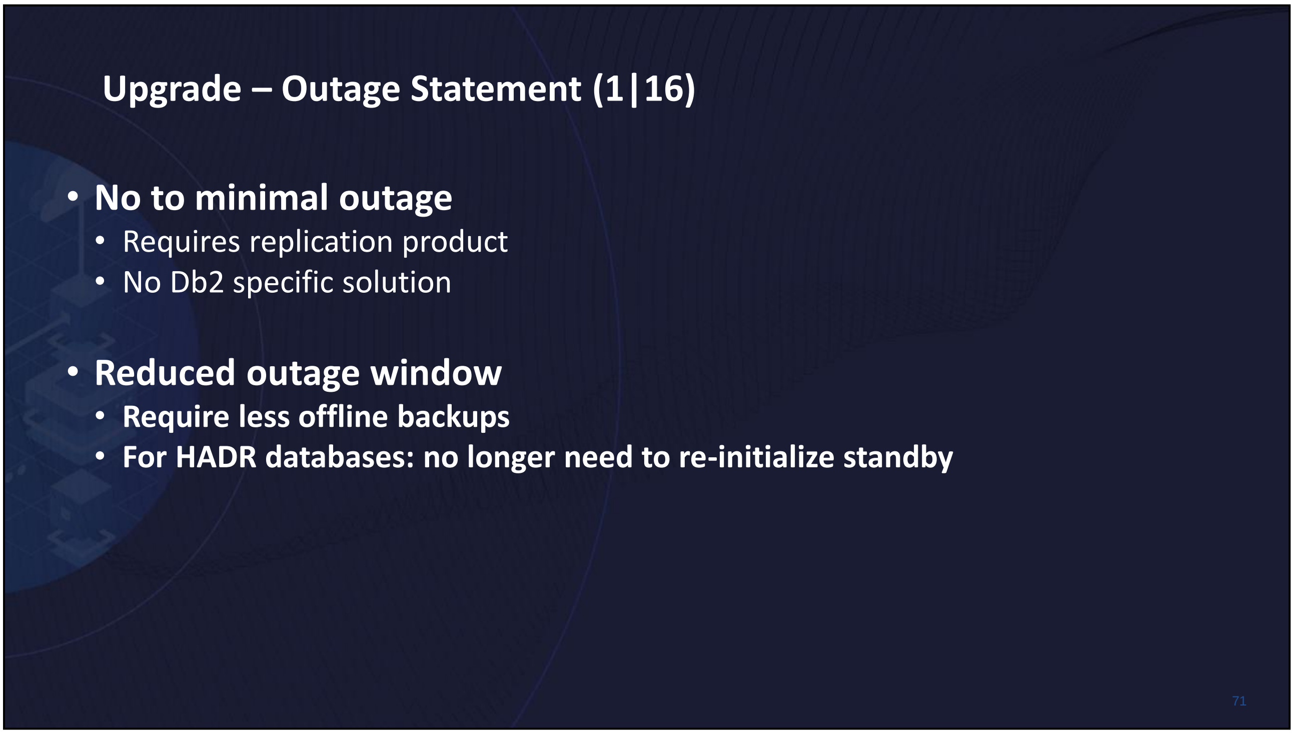
- Backup & Restore
  - Performance
  - Object (Remote) Storage
- Logging
  - Transaction Log Full
  - Disk Full in Archives
- Recovery
  - Point-in-time Recovery
  - Inaccessible Table Spaces
  - Monitor Crash Recovery

- **Upgrade**



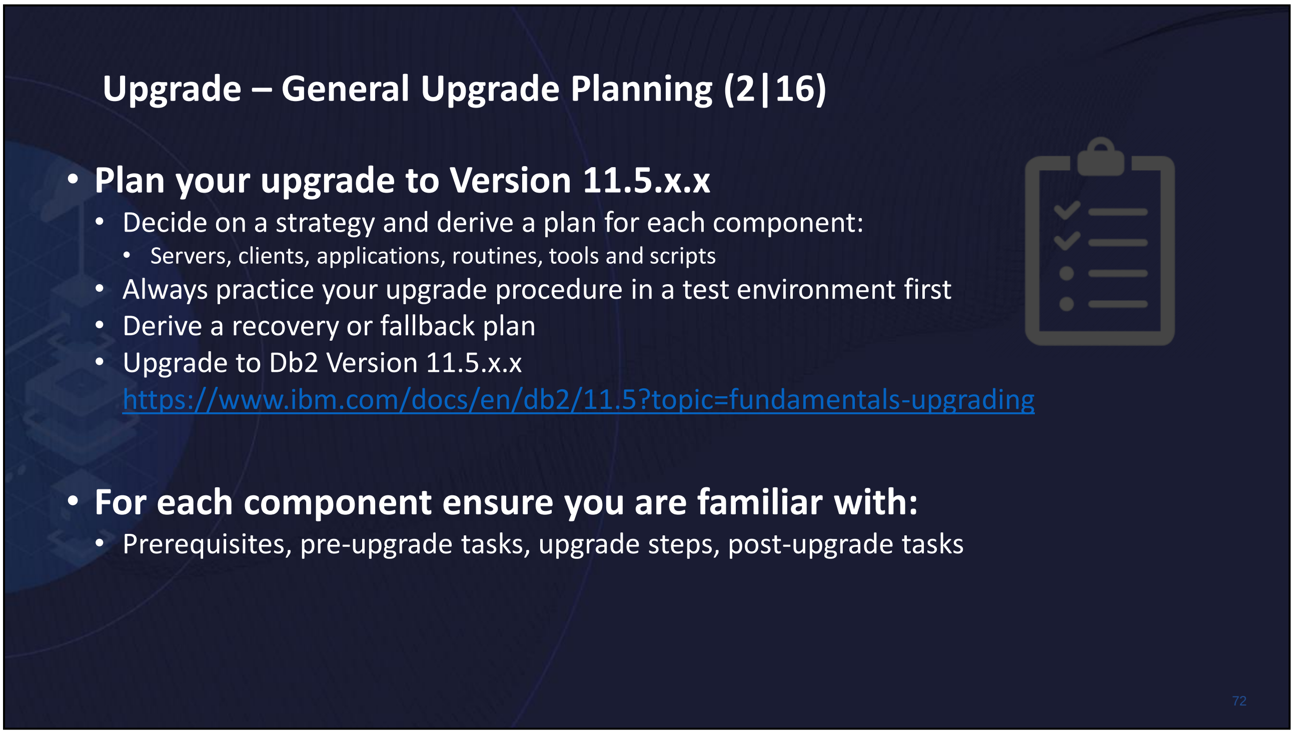
Question 9:

How can I upgrade my Db2 instance and databases with minimal outage?



Upgrade – Outage Statement (1|16)

- **No to minimal outage**
  - Requires replication product
  - No Db2 specific solution
- **Reduced outage window**
  - **Require less offline backups**
  - **For HADR databases: no longer need to re-initialize standby**



Upgrade – General Upgrade Planning (2|16)

- **Plan your upgrade to Version 11.5.x.x**
  - Decide on a strategy and derive a plan for each component:
    - Servers, clients, applications, routines, tools and scripts
  - Always practice your upgrade procedure in a test environment first
  - Derive a recovery or fallback plan
  - Upgrade to Db2 Version 11.5.x.x
    https://www.ibm.com/docs/en/db2/11.5?topic=fundamentals-upgrading

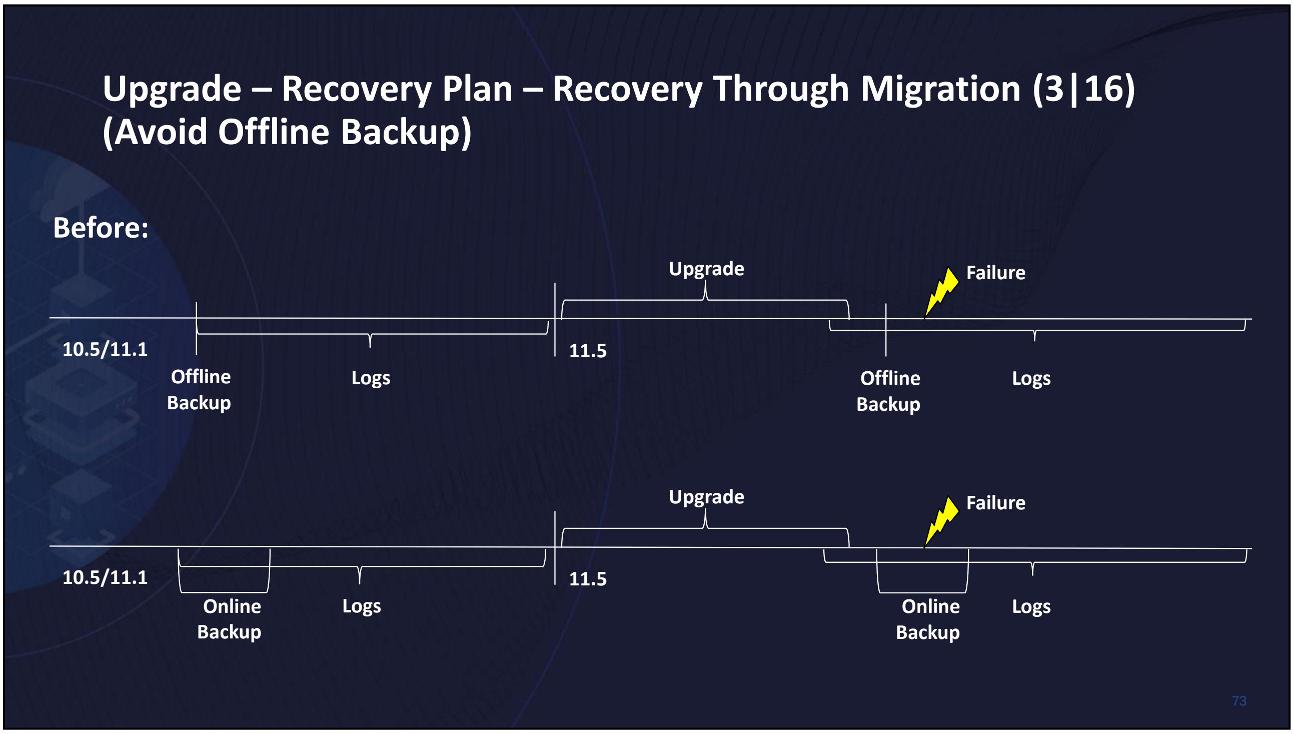
- **For each component ensure you are familiar with:**
  - Prerequisites, pre-upgrade tasks, upgrade steps, post-upgrade tasks

Upgrading a Db2 server (Windows)

- https://www.ibm.com/docs/en/db2/11.5?topic=servers-upgrading-db2-server-windows

Upgrading a Db2 server (Linux and UNIX)

- https://www.ibm.com/docs/en/db2/11.5?topic=servers-upgrading-db2-server-linux-unix



Upgrade – Recovery Plan – Recovery Through Migration (3|16) (Avoid Offline Backup)

Before:

Upgrade

Failure

10.5/11.1

Offline Backup

Logs

11.5

Offline Backup

Logs

Upgrade

Failure

10.5/11.1

Online Backup

Logs

11.5

Online Backup

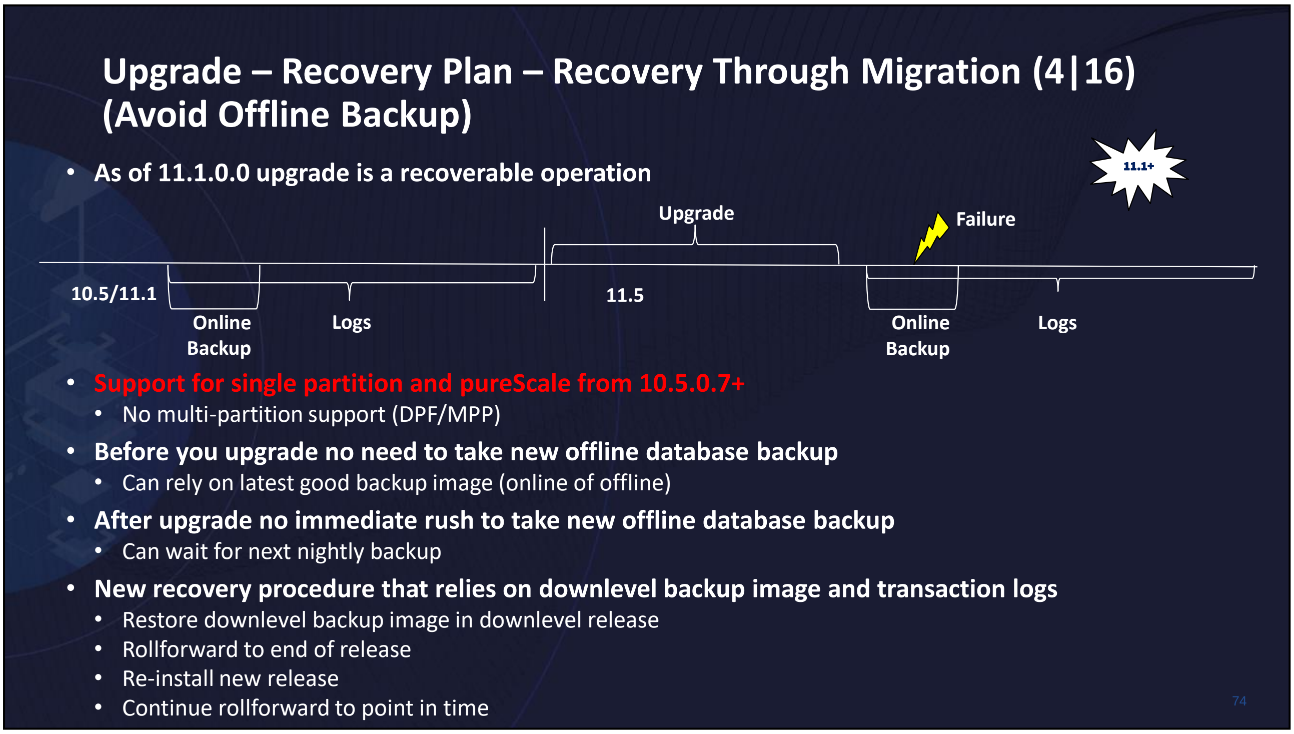
Logs

Upgrading a Db2 server (Windows)

- https://www.ibm.com/docs/en/db2/11.5?topic=servers-upgrading-db2-server-windows

Upgrading a Db2 server (Linux and UNIX)

- https://www.ibm.com/docs/en/db2/11.5?topic=servers-upgrading-db2-server-linux-unix



Upgrade – Recovery Plan – Recovery Through Migration (4|16) (Avoid Offline Backup)

- As of 11.1.0.0 upgrade is a recoverable operation

11.1+

Upgrade

Failure

10.5/11.1

Online Backup

Logs

11.5

Online Backup

Logs

- Support for single partition and pureScale from 10.5.0.7+
  - No multi-partition support (DPF/MPP)
- Before you upgrade no need to take new offline database backup
  - Can rely on latest good backup image (online of offline)
- After upgrade no immediate rush to take new offline database backup
  - Can wait for next nightly backup
- New recovery procedure that relies on downlevel backup image and transaction logs
  - Restore downlevel backup image in downlevel release
  - Rollforward to end of release
  - Re-install new release
  - Continue rollforward to point in time

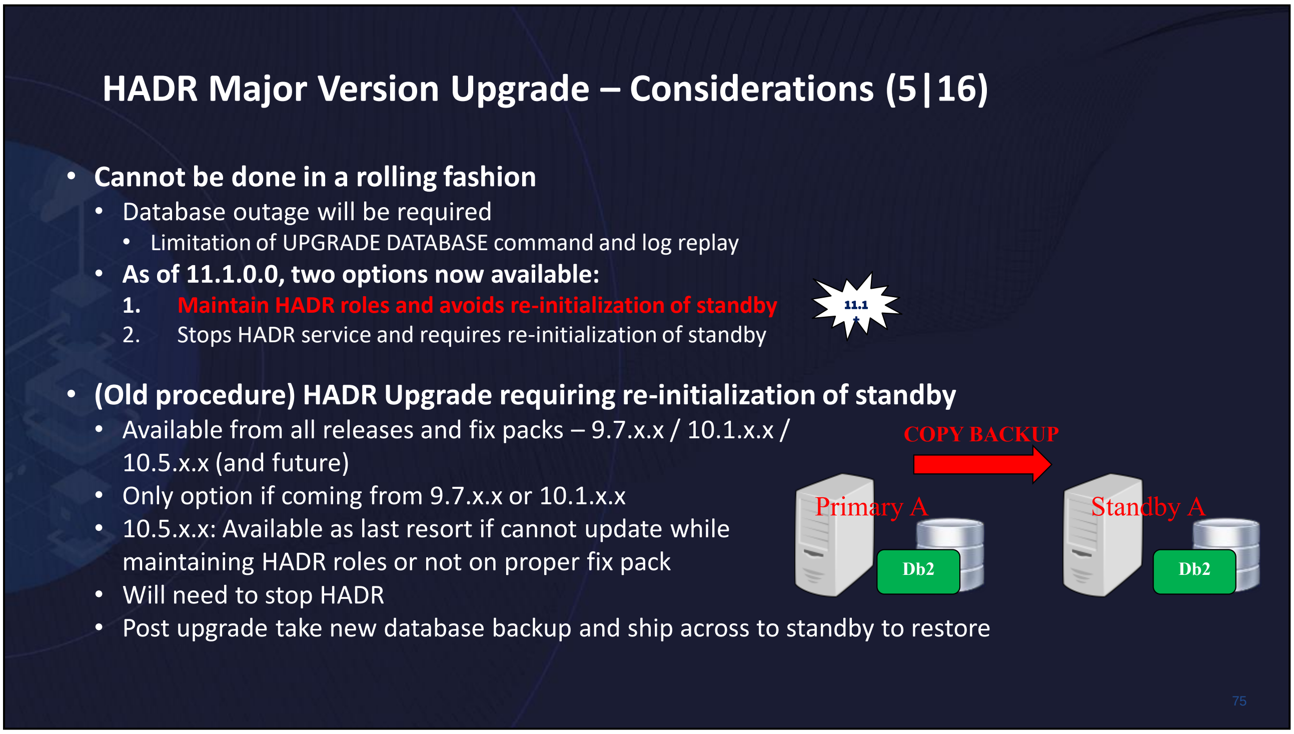
74

Upgrading a Db2 server (Windows)

- https://www.ibm.com/docs/en/db2/11.5?topic=servers-upgrading-db2-server-windows

Upgrading a Db2 server (Linux and UNIX)

- https://www.ibm.com/docs/en/db2/11.5?topic=servers-upgrading-db2-server-linux-unix



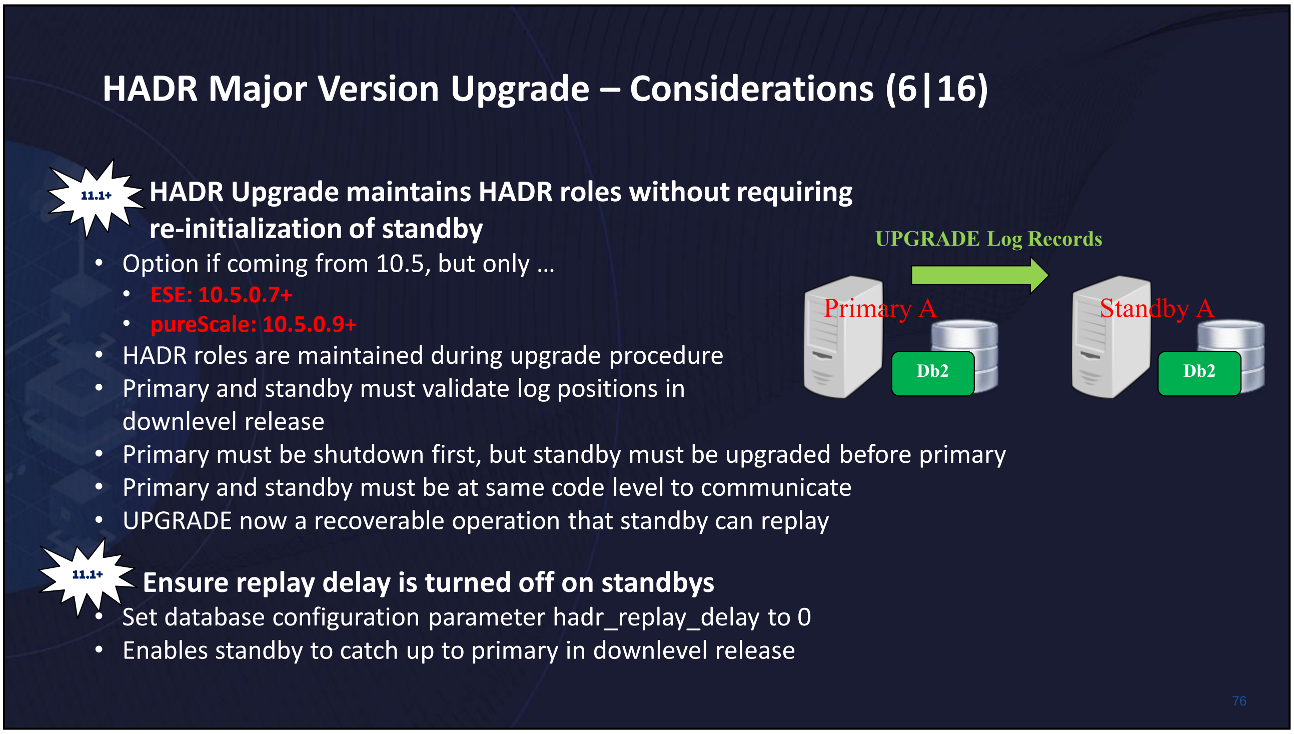
HADR Major Version Upgrade – Considerations (5|16)

- **Cannot be done in a rolling fashion**
  - Database outage will be required
    - Limitation of UPGRADE DATABASE command and log replay
  - **As of 11.1.0.0, two options now available:**

    1. **Maintain HADR roles and avoids re-initialization of standby**

    2. Stops HADR service and requires re-initialization of standby

- **(Old procedure) HADR Upgrade requiring re-initialization of standby**
  - Available from all releases and fix packs – 9.7.x.x / 10.1.x.x / 10.5.x.x (and future)
  - Only option if coming from 9.7.x.x or 10.1.x.x
  - 10.5.x.x: Available as last resort if cannot update while maintaining HADR roles or not on proper fix pack
  - Will need to stop HADR
  - Post upgrade take new database backup and ship across to standby to restore



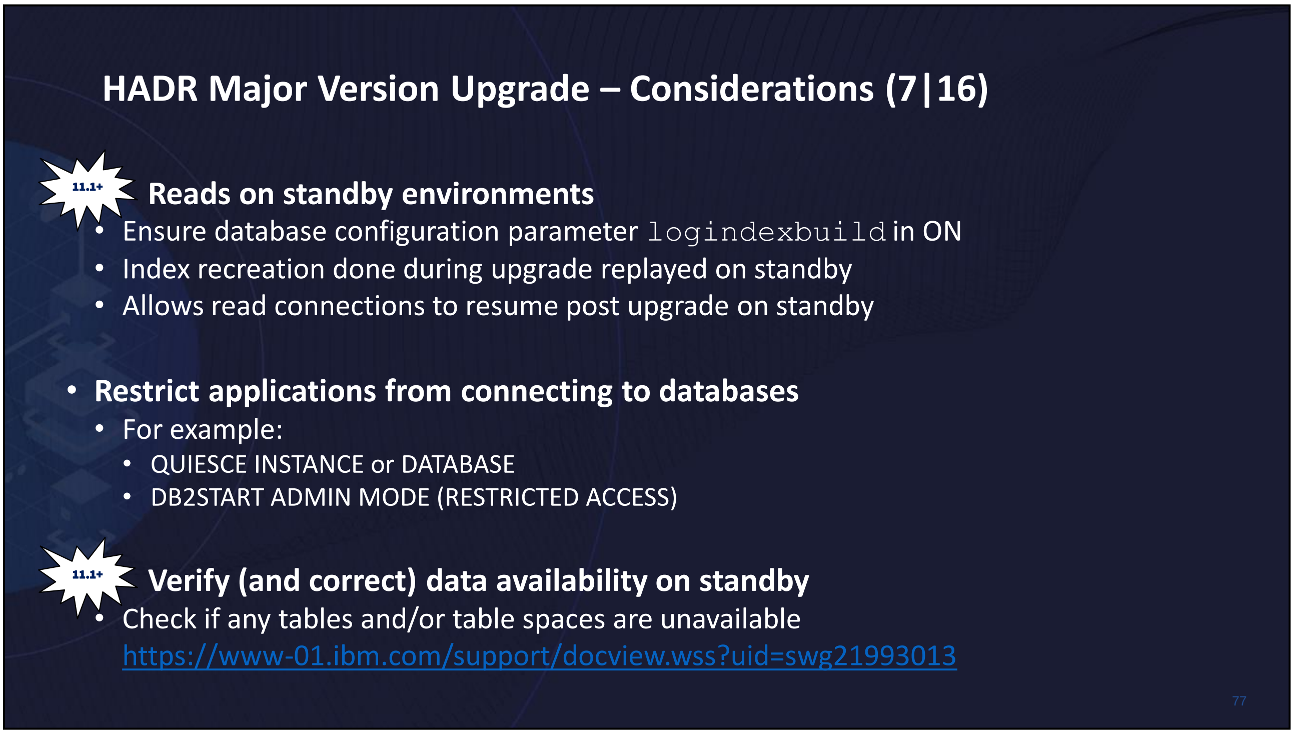
HADR Major Version Upgrade – Considerations (6|16)

11.1+ **HADR Upgrade maintains HADR roles without requiring re-initialization of standby**

- Option if coming from 10.5, but only …
  - **ESE: 10.5.0.7+**
  - **pureScale: 10.5.0.9+**
- HADR roles are maintained during upgrade procedure
- Primary and standby must validate log positions in downlevel release
- Primary must be shutdown first, but standby must be upgraded before primary
- Primary and standby must be at same code level to communicate
- UPGRADE now a recoverable operation that standby can replay

11.1+ **Ensure replay delay is turned off on standbys**

- Set database configuration parameter hadr_replay_delay to 0
- Enables standby to catch up to primary in downlevel release



HADR Major Version Upgrade – Considerations (7|16)

11.1+ **Reads on standby environments**

- Ensure database configuration parameter `logindexbuild` in ON
- Index recreation done during upgrade replayed on standby
- Allows read connections to resume post upgrade on standby

- **Restrict applications from connecting to databases**
  - For example:
    - QUIESCE INSTANCE or DATABASE
    - DB2START ADMIN MODE (RESTRICTED ACCESS)

11.1+ **Verify (and correct) data availability on standby**

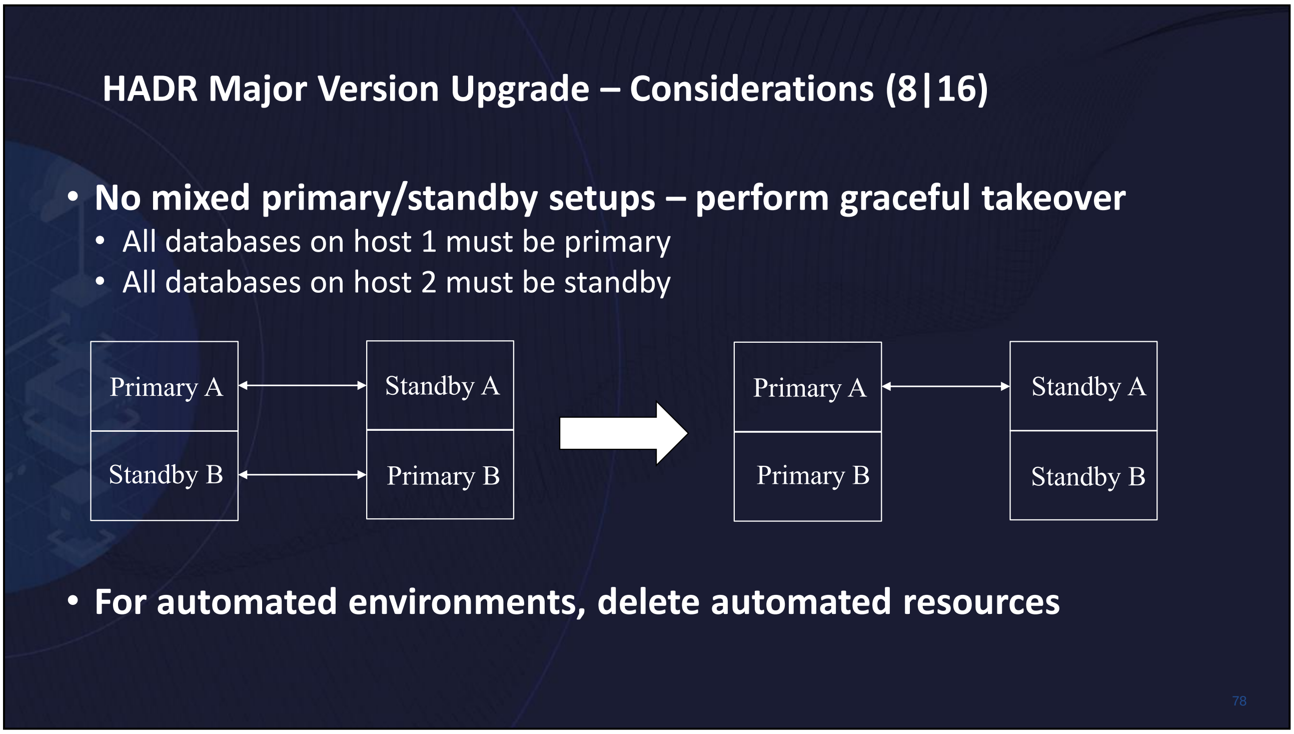
- Check if any tables and/or table spaces are unavailable
  https://www-01.ibm.com/support/docview.wss?uid=swg21993013

QUIESCE command

- https://www.ibm.com/docs/en/db2/11.5?topic=commands-quiesce

db2start - Start Db2 command

https://www.ibm.com/docs/en/db2/11.5?topic=commands-db2start-start-db2



HADR Major Version Upgrade – Considerations (8|16)

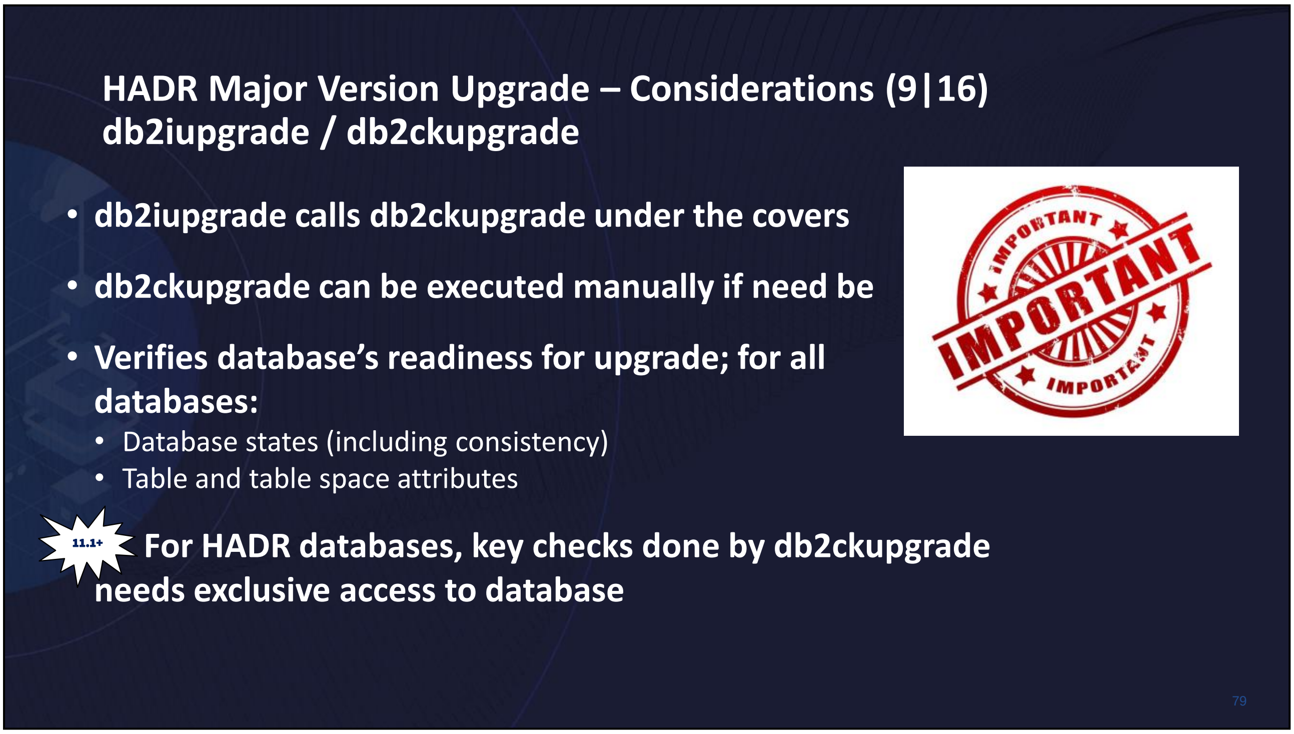
- **No mixed primary/standby setups – perform graceful takeover**
  - All databases on host 1 must be primary
  - All databases on host 2 must be standby

- **For automated environments, delete automated resources**

78

Db2 high availability instance configuration utility (db2haicu)

https://www.ibm.com/docs/en/db2/11.5?topic=db2haicu-tool



HADR Major Version Upgrade – Considerations (9|16)
db2iupgrade / db2ckupgrade

- db2iupgrade calls db2ckupgrade under the covers
- db2ckupgrade can be executed manually if need be
- Verifies database's readiness for upgrade; for all databases:
  - Database states (including consistency)
  - Table and table space attributes

11.1+ For HADR databases, key checks done by db2ckupgrade needs exclusive access to database

79

db2iupgrade - Upgrade instance command

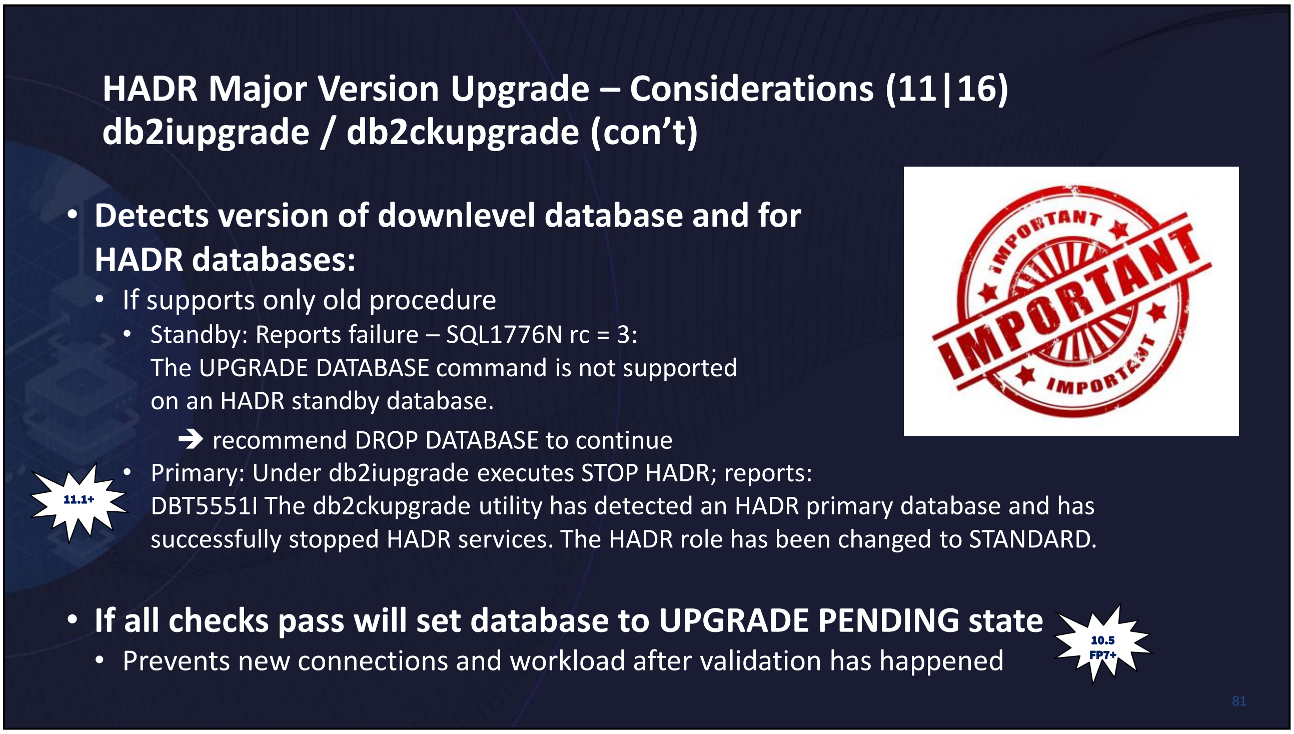
https://www.ibm.com/docs/en/db2/11.5?topic=commands-db2iupgrade-upgrade-instance

db2ckupgrade - Check database for upgrade command

https://www.ibm.com/docs/en/db2/11.5?topic=commands-db2ckupgrade-check-database-upgrade

HADR Major Version Upgrade – Considerations (10|16) db2iupgrade / db2ckupgrade (con't)

- **Detects version of downlevel database and for HADR databases:**
  - If supports **new** procedure
    - Standby: Will skip database
    - Primary: Verifies database; on top of normal checking, specific for primary:
      - Checks for table space availability; any table space in abnormal state reports warning:
        `DBT5552W The db2ckupgrade utility has detected that a table space is in an invalid state on the HADR standby database and needs attention.`
      - **Validates primary log shipping position matches with each standby's log replay position**
        - Waits approx. hadr_timeout / 2 seconds; minimum 5 seconds
        - Purpose is to avoid moving to new release and need to replay downlevel log records
        - On error:
          `DBT5535N The db2ckupgrade utility failed because the HADR primary's log shipping position does not match the HADR standby's log replay position.`



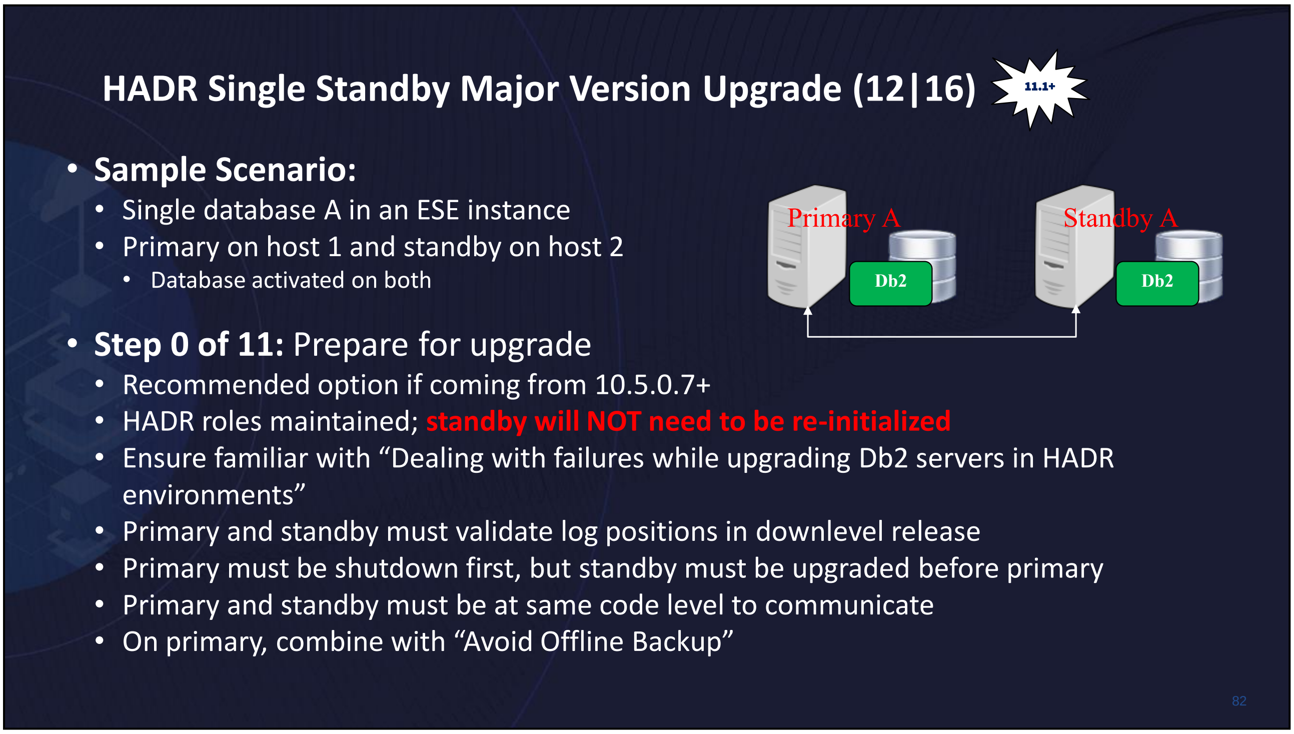
HADR Major Version Upgrade – Considerations (11|16) db2iupgrade / db2ckupgrade (con't)

- **Detects version of downlevel database and for HADR databases:**
  - If supports only old procedure
    - Standby: Reports failure – SQL1776N rc = 3:
      The UPGRADE DATABASE command is not supported
      on an HADR standby database.
      ➔ recommend DROP DATABASE to continue
    - Primary: Under db2iupgrade executes STOP HADR; reports:
      DBT5551I The db2ckupgrade utility has detected an HADR primary database and has successfully stopped HADR services. The HADR role has been changed to STANDARD.

11.1+

- **If all checks pass will set database to UPGRADE PENDING state**
  - Prevents new connections and workload after validation has happened

10.5 FP7+



HADR Single Standby Major Version Upgrade (12|16)

11.1+
Primary A
Db2
Standby A
Db2

- **Sample Scenario:**
  - Single database A in an ESE instance
  - Primary on host 1 and standby on host 2
    - Database activated on both
- **Step 0 of 11:** Prepare for upgrade
  - Recommended option if coming from 10.5.0.7+
  - HADR roles maintained; **standby will NOT need to be re-initialized**
  - Ensure familiar with “Dealing with failures while upgrading Db2 servers in HADR environments”
  - Primary and standby must validate log positions in downlevel release
  - Primary must be shutdown first, but standby must be upgraded before primary
  - Primary and standby must be at same code level to communicate
  - On primary, combine with “Avoid Offline Backup”

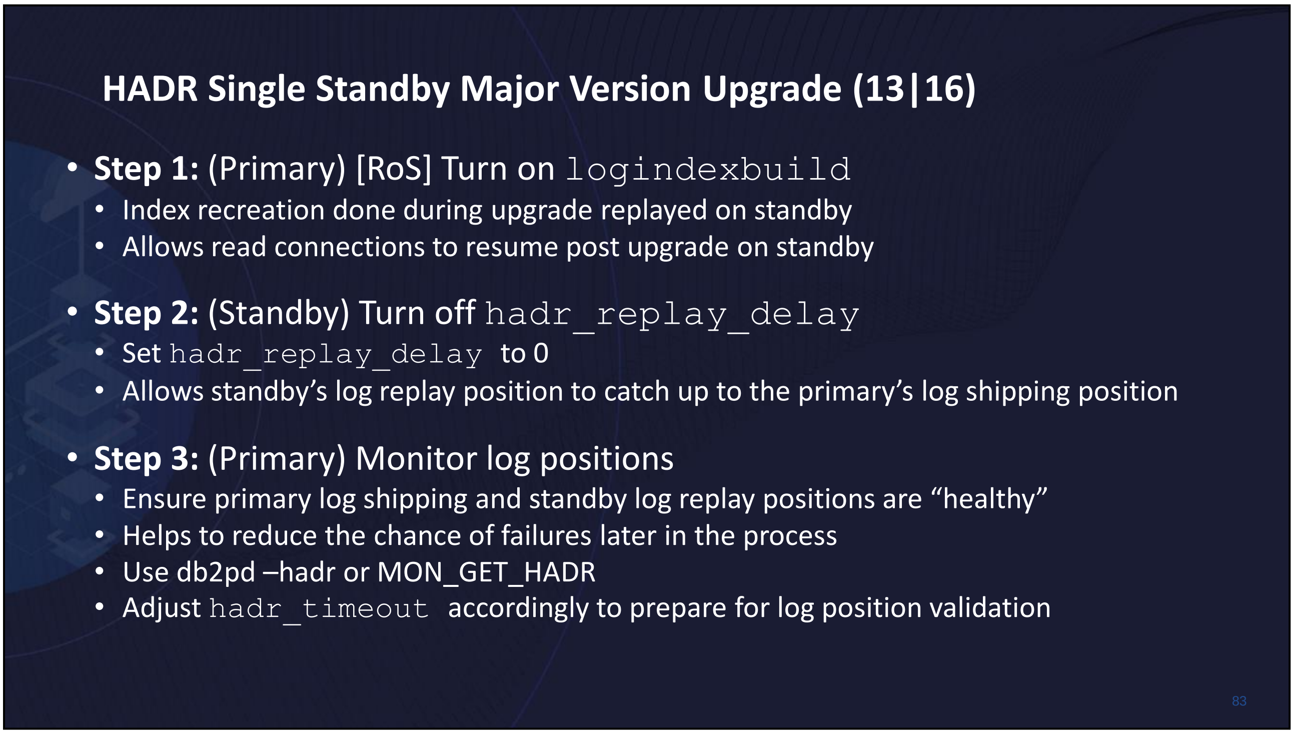
82

Upgrading Db2 servers in HADR environments (without standby reinitialization)

https://www.ibm.com/docs/en/db2/11.5?topic=udhadrhe-upgrading-db2-servers-in-hadr-environments-without-standby-reinitialization

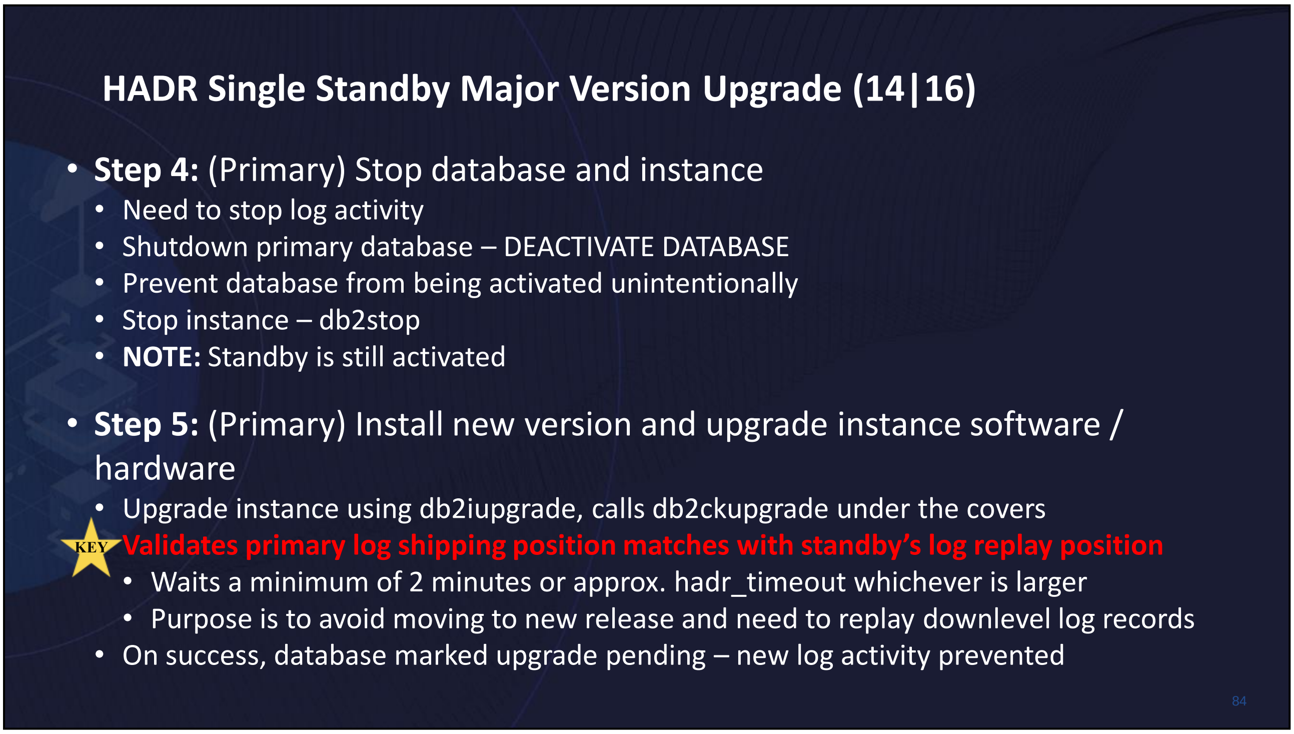
Recovering through a Db2 server upgrade

https://www.ibm.com/docs/en/db2/11.5?topic=servers-recovering-through-db2-server-upgrade



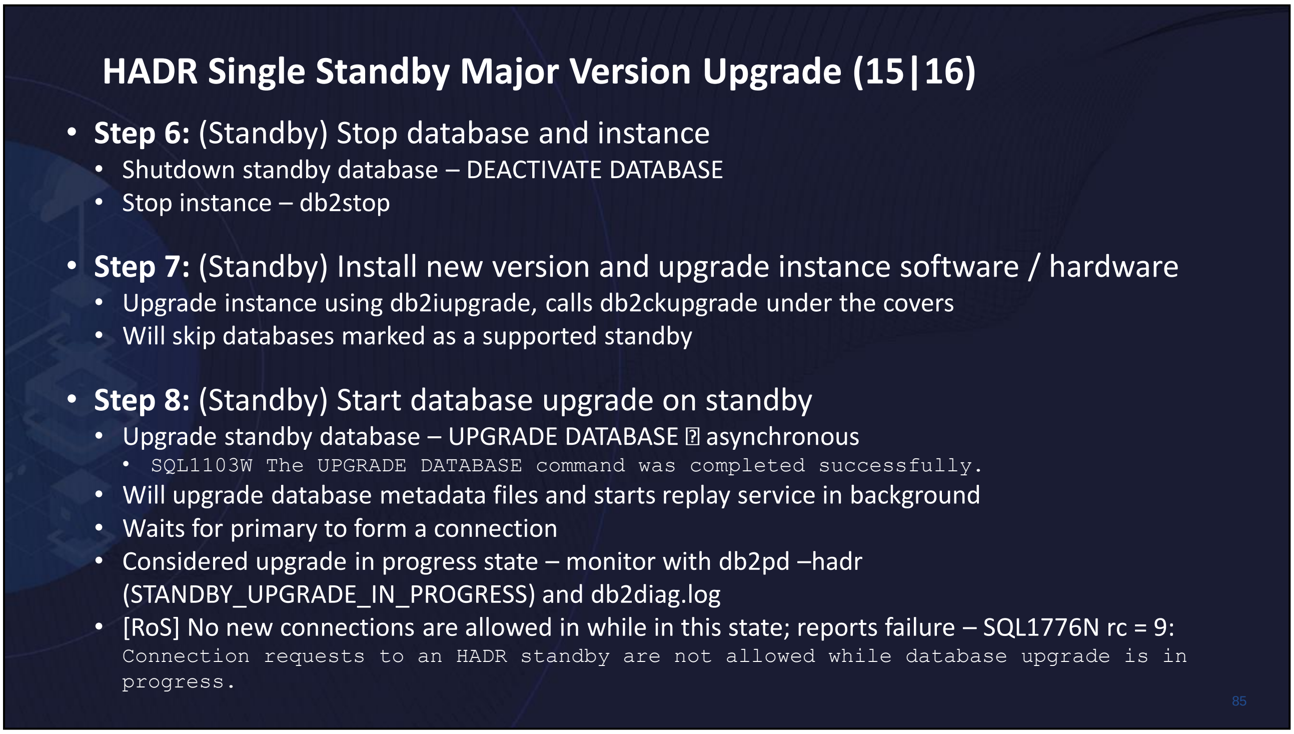
HADR Single Standby Major Version Upgrade (13|16)

- **Step 1:** (Primary) [RoS] Turn on `logindexbuild`
  - Index recreation done during upgrade replayed on standby
  - Allows read connections to resume post upgrade on standby
- **Step 2:** (Standby) Turn off `hadr_replay_delay`
  - Set `hadr_replay_delay` to 0
  - Allows standby's log replay position to catch up to the primary's log shipping position
- **Step 3:** (Primary) Monitor log positions
  - Ensure primary log shipping and standby log replay positions are "healthy"
  - Helps to reduce the chance of failures later in the process
  - Use db2pd –hadr or MON_GET_HADR
  - Adjust `hadr_timeout` accordingly to prepare for log position validation



HADR Single Standby Major Version Upgrade (14|16)

- **Step 4:** (Primary) Stop database and instance
  - Need to stop log activity
  - Shutdown primary database – DEACTIVATE DATABASE
  - Prevent database from being activated unintentionally
  - Stop instance – db2stop
  - **NOTE:** Standby is still activated
- **Step 5:** (Primary) Install new version and upgrade instance software / hardware
  - Upgrade instance using db2iupgrade, calls db2ckupgrade under the covers
  - KEY **Validates primary log shipping position matches with standby's log replay position**
    - Waits a minimum of 2 minutes or approx. hadr_timeout whichever is larger
    - Purpose is to avoid moving to new release and need to replay downlevel log records
  - On success, database marked upgrade pending – new log activity prevented

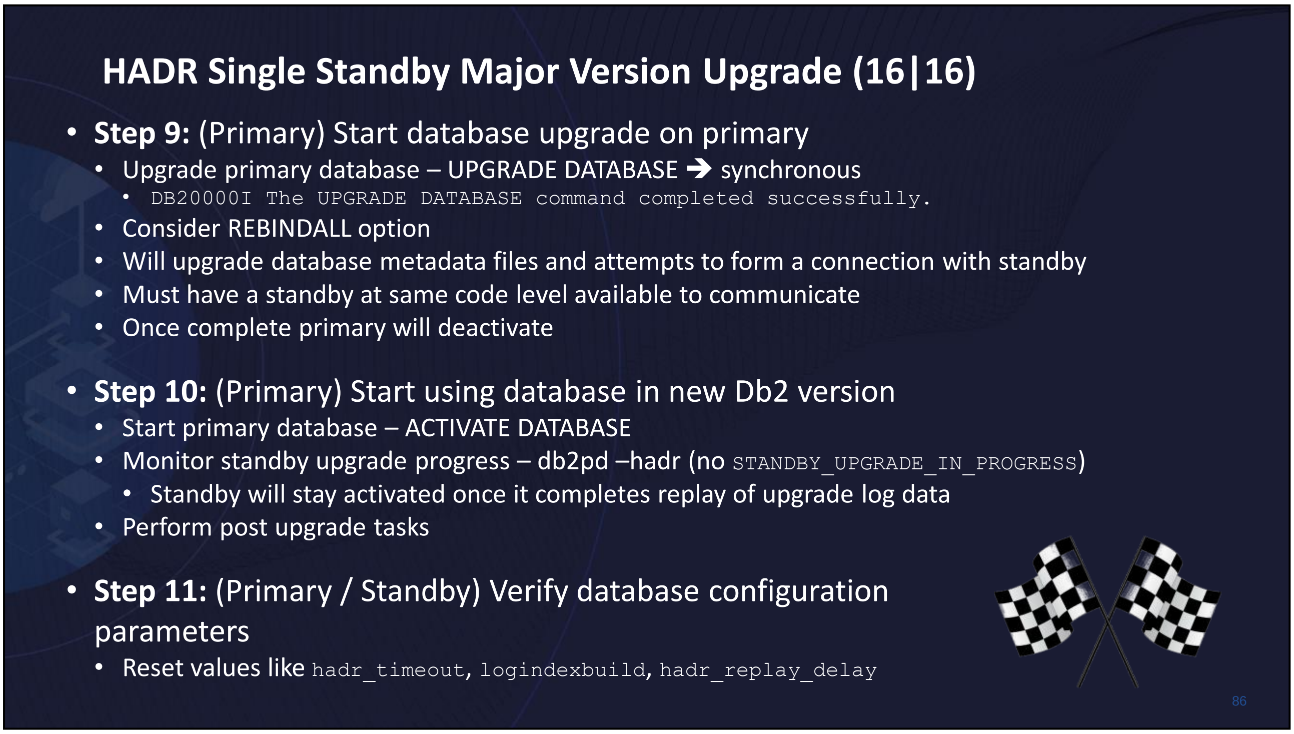


HADR Single Standby Major Version Upgrade (15|16)

- **Step 6:** (Standby) Stop database and instance
  - Shutdown standby database – DEACTIVATE DATABASE
  - Stop instance – db2stop

- **Step 7:** (Standby) Install new version and upgrade instance software / hardware
  - Upgrade instance using db2iupgrade, calls db2ckupgrade under the covers
  - Will skip databases marked as a supported standby

- **Step 8:** (Standby) Start database upgrade on standby
  - Upgrade standby database – UPGRADE DATABASE ⍰ asynchronous
    - `SQL1103W The UPGRADE DATABASE command was completed successfully.`
  - Will upgrade database metadata files and starts replay service in background
  - Waits for primary to form a connection
  - Considered upgrade in progress state – monitor with db2pd –hadr (STANDBY_UPGRADE_IN_PROGRESS) and db2diag.log
  - [RoS] No new connections are allowed in while in this state; reports failure – SQL1776N rc = 9: `Connection requests to an HADR standby are not allowed while database upgrade is in progress.`



HADR Single Standby Major Version Upgrade (16|16)

- **Step 9:** (Primary) Start database upgrade on primary
  - Upgrade primary database – UPGRADE DATABASE ➔ synchronous
    - `DB20000I The UPGRADE DATABASE command completed successfully.`
  - Consider REBINDALL option
  - Will upgrade database metadata files and attempts to form a connection with standby
  - Must have a standby at same code level available to communicate
  - Once complete primary will deactivate
- **Step 10:** (Primary) Start using database in new Db2 version
  - Start primary database – ACTIVATE DATABASE
  - Monitor standby upgrade progress – db2pd –hadr (no `STANDBY_UPGRADE_IN_PROGRESS`)
    - Standby will stay activated once it completes replay of upgrade log data
  - Perform post upgrade tasks
- **Step 11:** (Primary / Standby) Verify database configuration parameters
  - Reset values like `hadr_timeout`, `logindexbuild`, `hadr_replay_delay`

Resources

- **IBM Documentation**
  - https://www.ibm.com/docs/en/db2/11.5
- **HADR Wiki (incl. Best Practices)**
  - https://ibm.github.io/db2-hadr-wiki/

Questions ???

Thank You

Speaker: Michael Roecken

Company: IBM

@roecken

Session Code: C11

Please fill out your session evaluation before leaving!

Michael Roecken is a senior software developer with Db2 for Linux, UNIX, and Windows platforms at the IBM Toronto Lab. Michael has worked since 2000 designing, implementing and supporting various features and capabilities in the areas of: backup/restore, crash/rollforward recovery, high availability/disaster recovery, and logging/transaction management.

Please connect with Michael on Twitter at @roecken and LinkedIn.

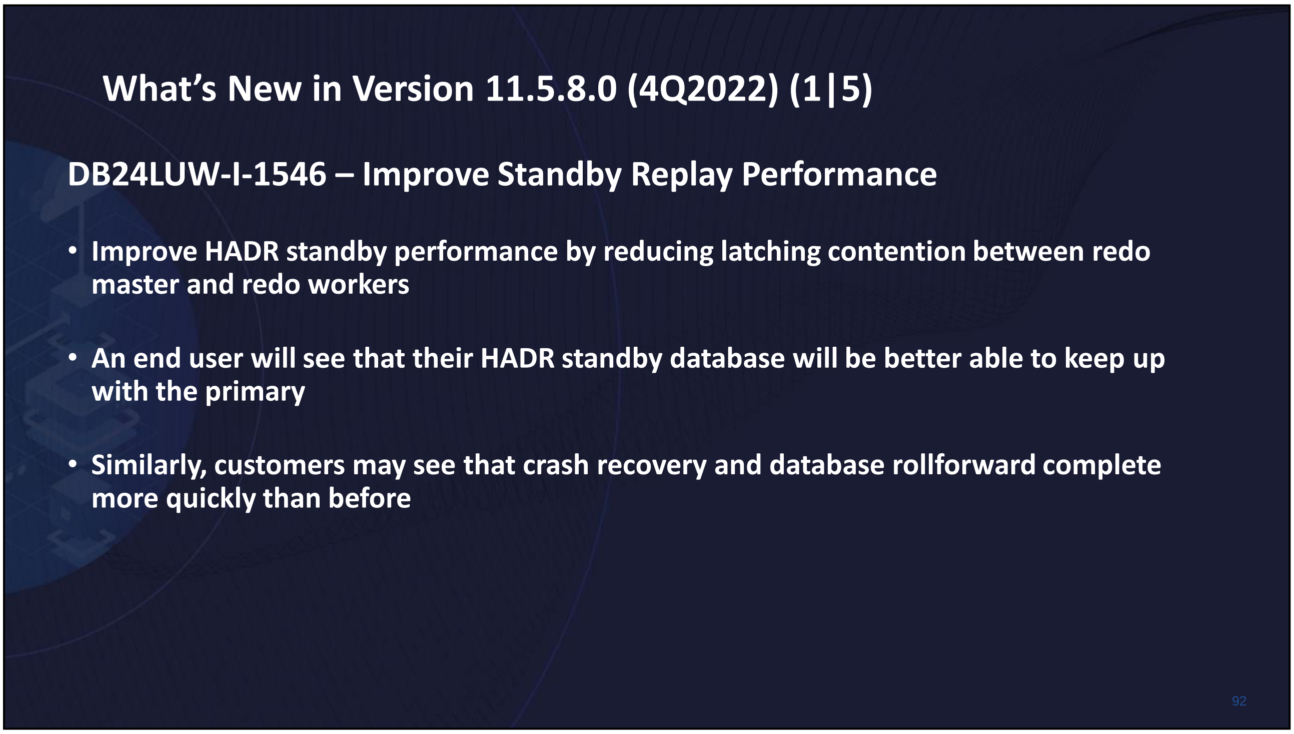
Submit Questions for the Db2 LUW Expert Panel

Wednesday 4:30 – 5:30pm
in Back Bay D

POLLEV.COM / TURTLES

Agenda

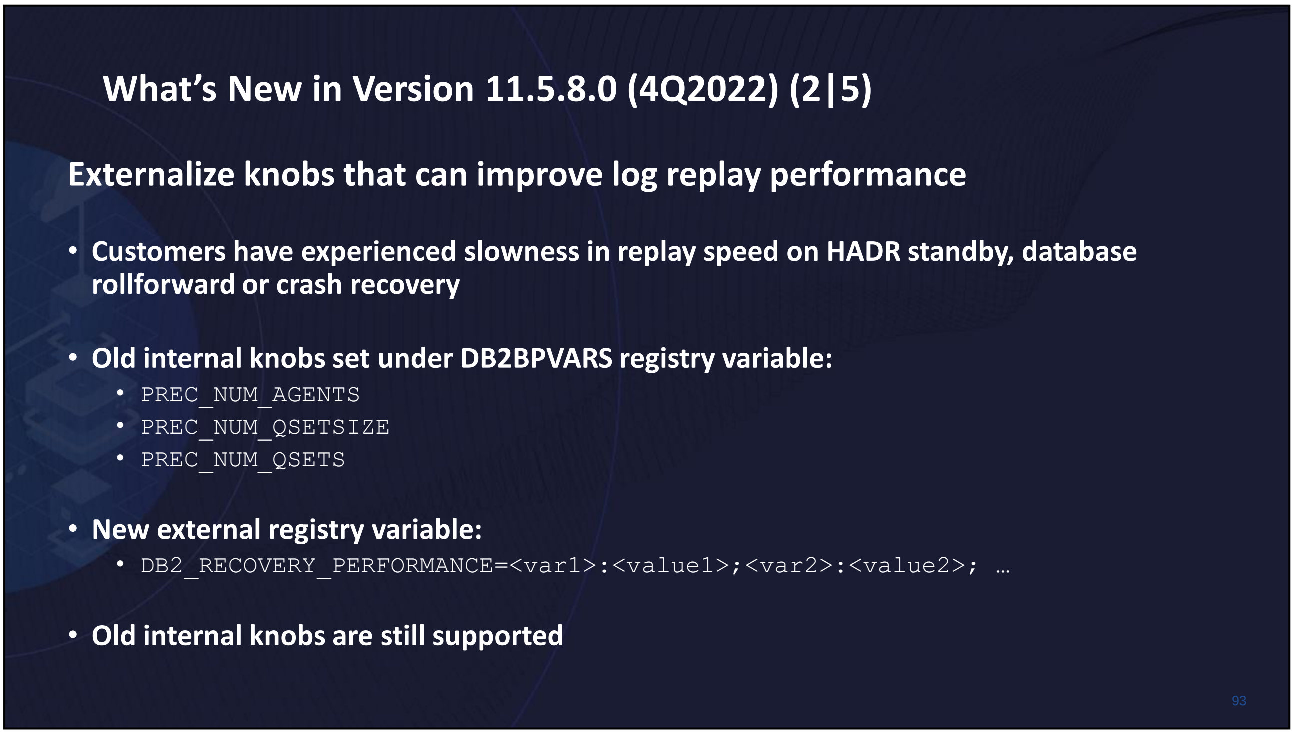
- Backup & Restore
  - Performance
  - Object (Remote) Storage
- Logging
  - Transaction Log Full
  - Disk Full in Archives
- Recovery
  - Point-in-time Recovery
  - Inaccessible Table Spaces
  - Monitor Crash Recovery
- Upgrade
- **Bonus:**
  - **What's New in Version 11.5.8.0 (4Q2022)**



What’s New in Version 11.5.8.0 (4Q2022) (1|5)

DB24LUW-I-1546 – Improve Standby Replay Performance

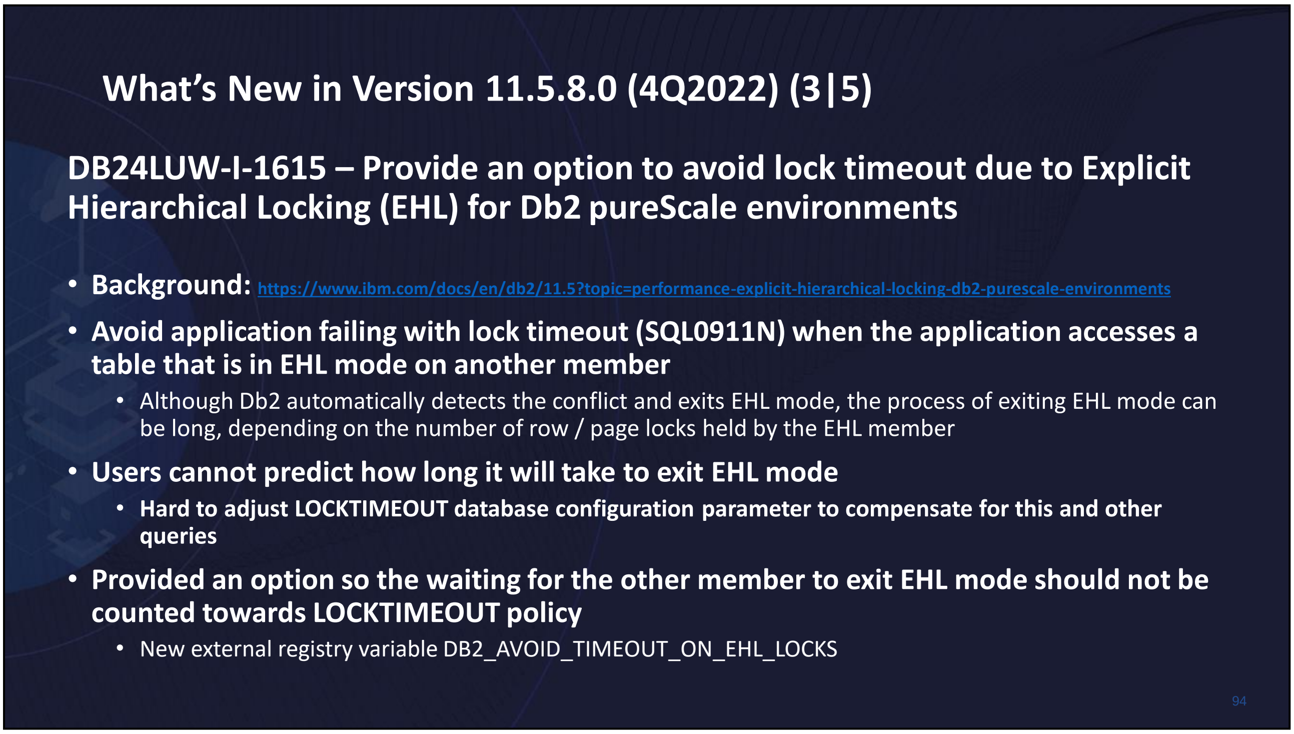
- Improve HADR standby performance by reducing latching contention between redo master and redo workers
- An end user will see that their HADR standby database will be better able to keep up with the primary
- Similarly, customers may see that crash recovery and database rollforward complete more quickly than before



What’s New in Version 11.5.8.0 (4Q2022) (2|5)

Externalize knobs that can improve log replay performance

- Customers have experienced slowness in replay speed on HADR standby, database rollforward or crash recovery
- Old internal knobs set under DB2BPVARS registry variable:
  - `PREC_NUM_AGENTS`
  - `PREC_NUM_QSETSIZE`
  - `PREC_NUM_QSETS`
- New external registry variable:
  - `DB2_RECOVERY_PERFORMANCE=<var1>:<value1>;<var2>:<value2>; …`
- Old internal knobs are still supported

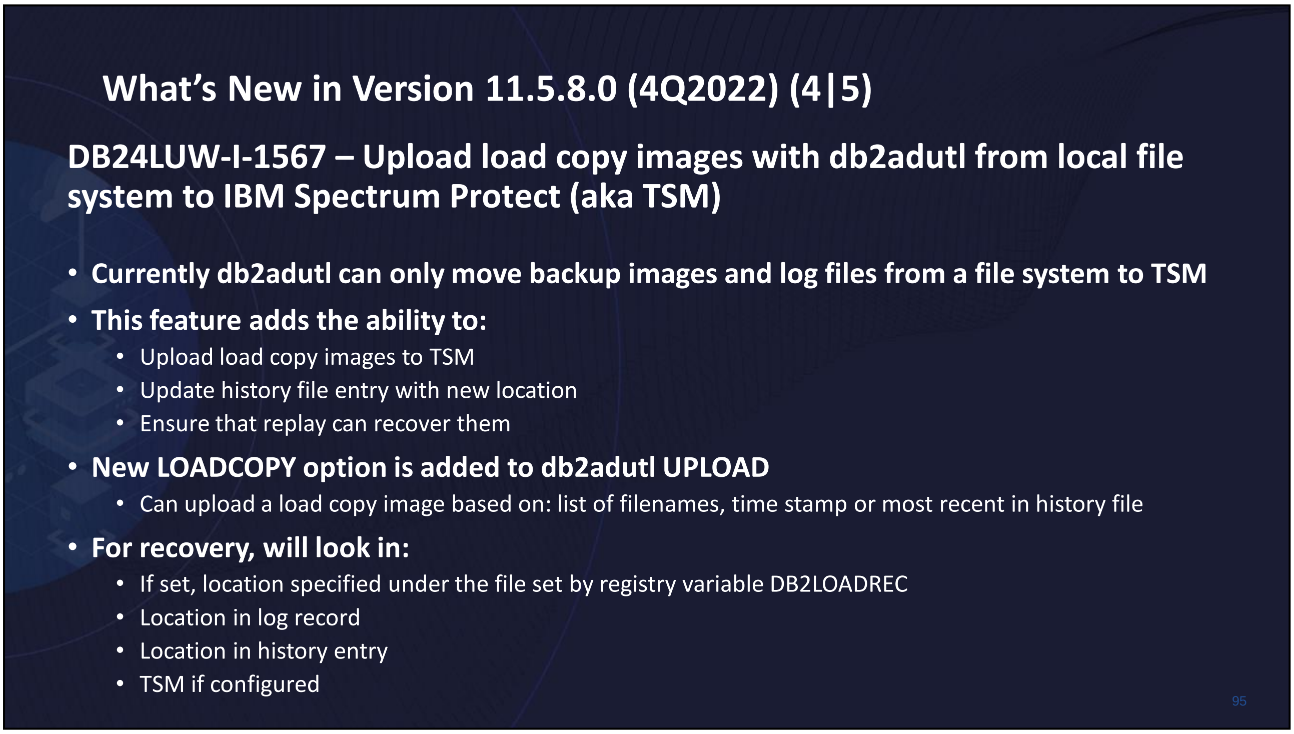


What’s New in Version 11.5.8.0 (4Q2022) (3|5)

DB24LUW-I-1615 – Provide an option to avoid lock timeout due to Explicit Hierarchical Locking (EHL) for Db2 pureScale environments

- **Background:** https://www.ibm.com/docs/en/db2/11.5?topic=performance-explicit-hierarchical-locking-db2-purescale-environments
- **Avoid application failing with lock timeout (SQL0911N) when the application accesses a table that is in EHL mode on another member**
  - Although Db2 automatically detects the conflict and exits EHL mode, the process of exiting EHL mode can be long, depending on the number of row / page locks held by the EHL member
- **Users cannot predict how long it will take to exit EHL mode**
  - **Hard to adjust LOCKTIMEOUT database configuration parameter to compensate for this and other queries**
- **Provided an option so the waiting for the other member to exit EHL mode should not be counted towards LOCKTIMEOUT policy**
  - New external registry variable DB2_AVOID_TIMEOUT_ON_EHL_LOCKS

Change to DB2_AVOID_TIMEOUT_ON_EHL_LOCKS can take effect immediately without recycling the instance or the database.



What’s New in Version 11.5.8.0 (4Q2022) (4|5)

DB24LUW-I-1567 – Upload load copy images with db2adutl from local file system to IBM Spectrum Protect (aka TSM)

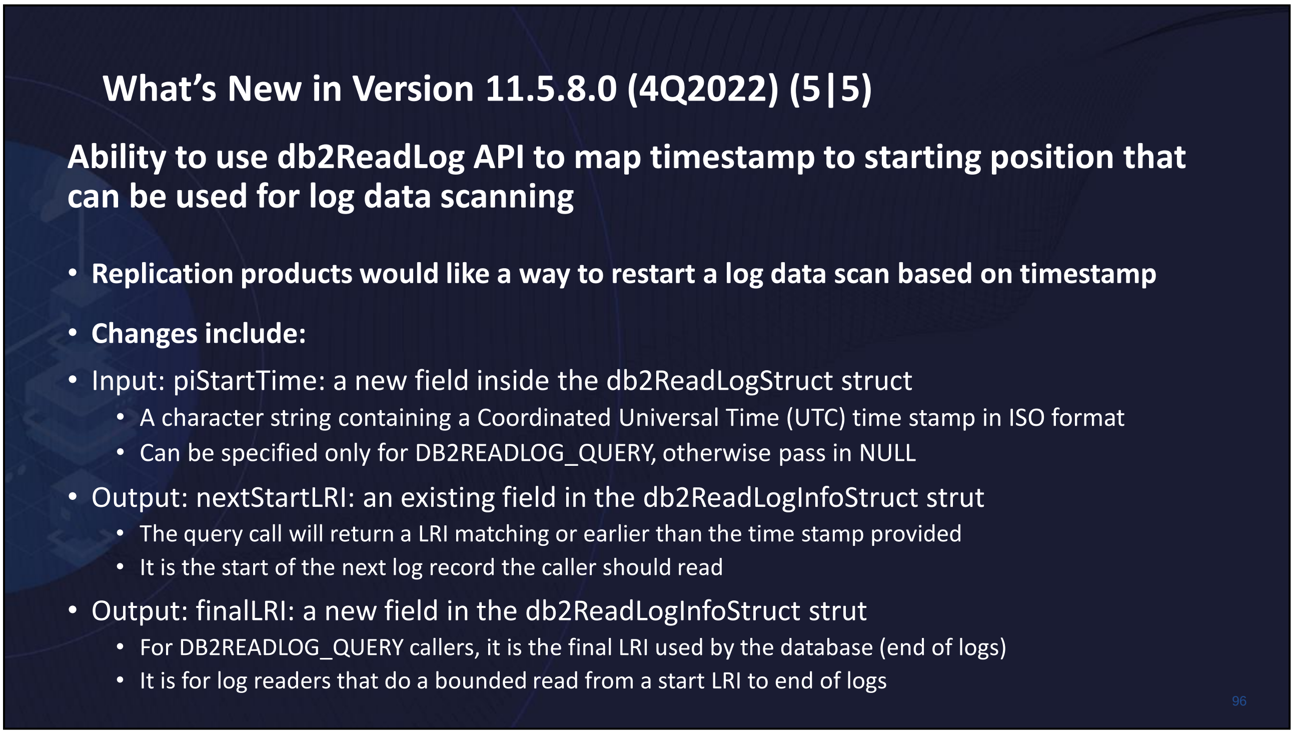
- **Currently db2adutl can only move backup images and log files from a file system to TSM**
- **This feature adds the ability to:**
  - Upload load copy images to TSM
  - Update history file entry with new location
  - Ensure that replay can recover them
- **New LOADCOPY option is added to db2adutl UPLOAD**
  - Can upload a load copy image based on: list of filenames, time stamp or most recent in history file
- **For recovery, will look in:**
  - If set, location specified under the file set by registry variable DB2LOADREC
  - Location in log record
  - Location in history entry
  - TSM if configured

95

```
To help with resiliency, users do load copy to local media and then later upload to TSM (due to network
issues, TSM server maintenance, reliability issues, etc.).
```

```
UPLOAD [ [ AND REMOVE ]
                 [ LOADCOPY [ [ TAKEN AT <timestamp> ] | [ <filename> ] ] ]
                 [ BACKUP [ [ TAKEN AT <timestamp> ] | [ WITH LOGS ] ] | [ <filename> ] ]
                 [ LOGS [ [ BETWEEN <Sn1> AND <Sn2> ] | [ OLDER THAN <Sn1> ] ] [ CHAIN <n> ] [ FROM [
LOGARCHMETH1 | LOGARCHMETH2 ] ] ]
                 [ MGMTCLASS <mgmtclass> ]
                 [ DB2USER <db2 username> ] [ DB2PASSWORD <db2 password> ] ]
```

- If the AND REMOVE option has been supplied, db2adutl would upload and then physically delete the local object
- The new syntax diagram no longer specifically mentions the option IMAGES, but for backward compatibility it will still be supported.



What’s New in Version 11.5.8.0 (4Q2022) (5|5)

Ability to use db2ReadLog API to map timestamp to starting position that can be used for log data scanning

- **Replication products would like a way to restart a log data scan based on timestamp**
- **Changes include:**
- Input: piStartTime: a new field inside the db2ReadLogStruct struct
  - A character string containing a Coordinated Universal Time (UTC) time stamp in ISO format
  - Can be specified only for DB2READLOG_QUERY, otherwise pass in NULL
- Output: nextStartLRI: an existing field in the db2ReadLogInfoStruct strut
  - The query call will return a LRI matching or earlier than the time stamp provided
  - It is the start of the next log record the caller should read
- Output: finalLRI: a new field in the db2ReadLogInfoStruct strut
  - For DB2READLOG_QUERY callers, it is the final LRI used by the database (end of logs)
  - It is for log readers that do a bounded read from a start LRI to end of logs

- In order to provide this new functionality, the db2ReadLog API will be extended to support a start time stamp at query time that will map to a LRI

- All new fields are uplevel and downlevel client/server aware

Backup Slides

- **What’s New in Version 11.5.6.0**
- What’s New in Version 11.5.7.0

What’s New 11.5.6.0

- https://www.ibm.com/docs/en/db2/11.5?topic=new-1156

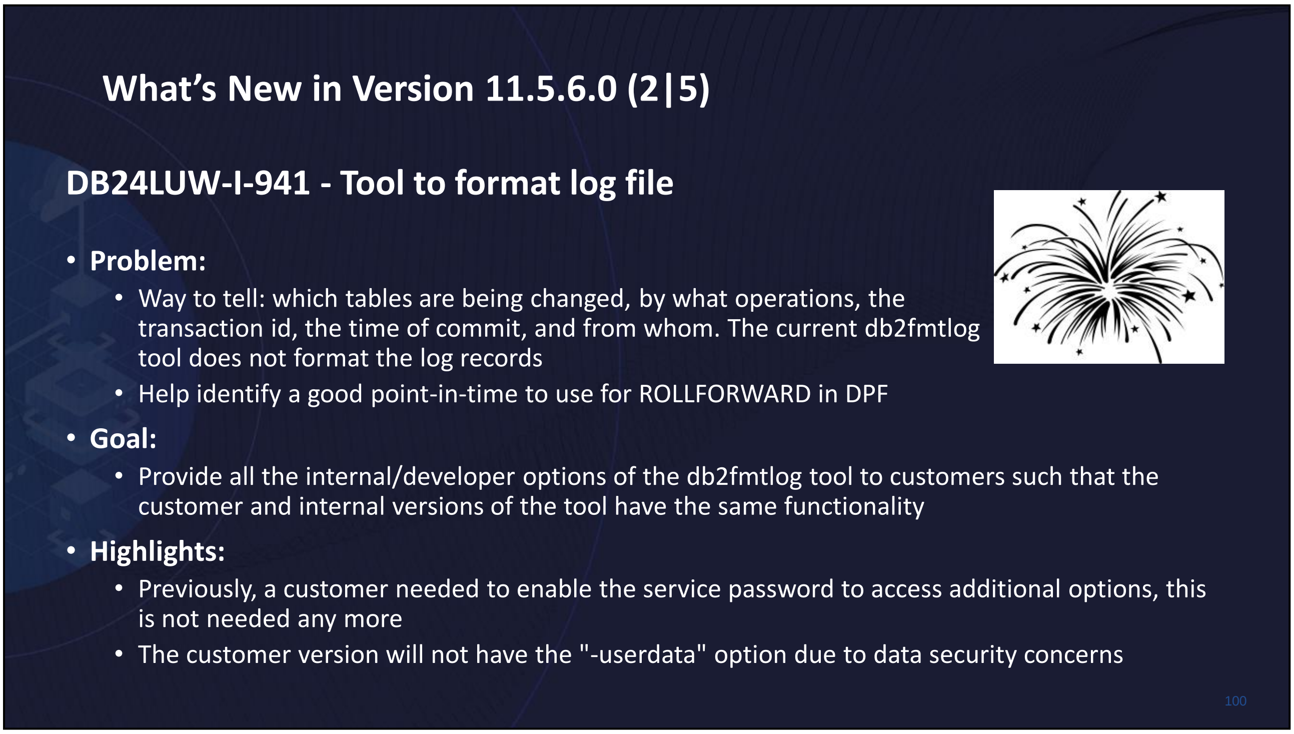
What’s New 11.5.7.0

- https://www.ibm.com/docs/en/db2/11.5?topic=new-1157

What’s New in Version 11.5.6.0 (1|5)

DB24LUW-I-1038 - db2adutl performance

- **db2adult delete takes long**
  - Deletes objects one by one using TSM delete API
  - High load on TSM server interferes with new workload
- **Re-wrote to batch as many objects (the TSM server allows) into one TSM delete API call**
- **Internal tests showed 3x improvement**

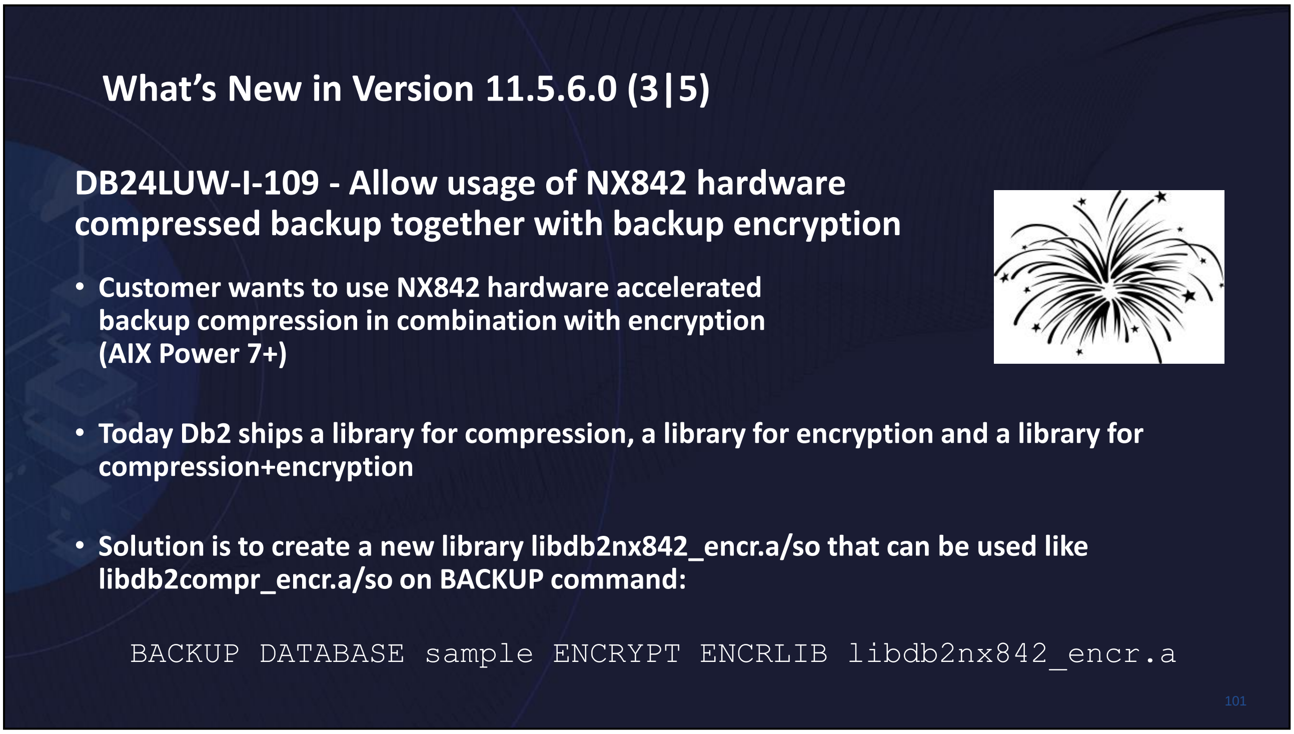


What’s New in Version 11.5.6.0 (2|5)

DB24LUW-I-941 - Tool to format log file

- **Problem:**
  - Way to tell: which tables are being changed, by what operations, the transaction id, the time of commit, and from whom. The current db2fmtlog tool does not format the log records
  - Help identify a good point-in-time to use for ROLLFORWARD in DPF
- **Goal:**
  - Provide all the internal/developer options of the db2fmtlog tool to customers such that the customer and internal versions of the tool have the same functionality
- **Highlights:**
  - Previously, a customer needed to enable the service password to access additional options, this is not needed any more
  - The customer version will not have the "-userdata" option due to data security concerns

100

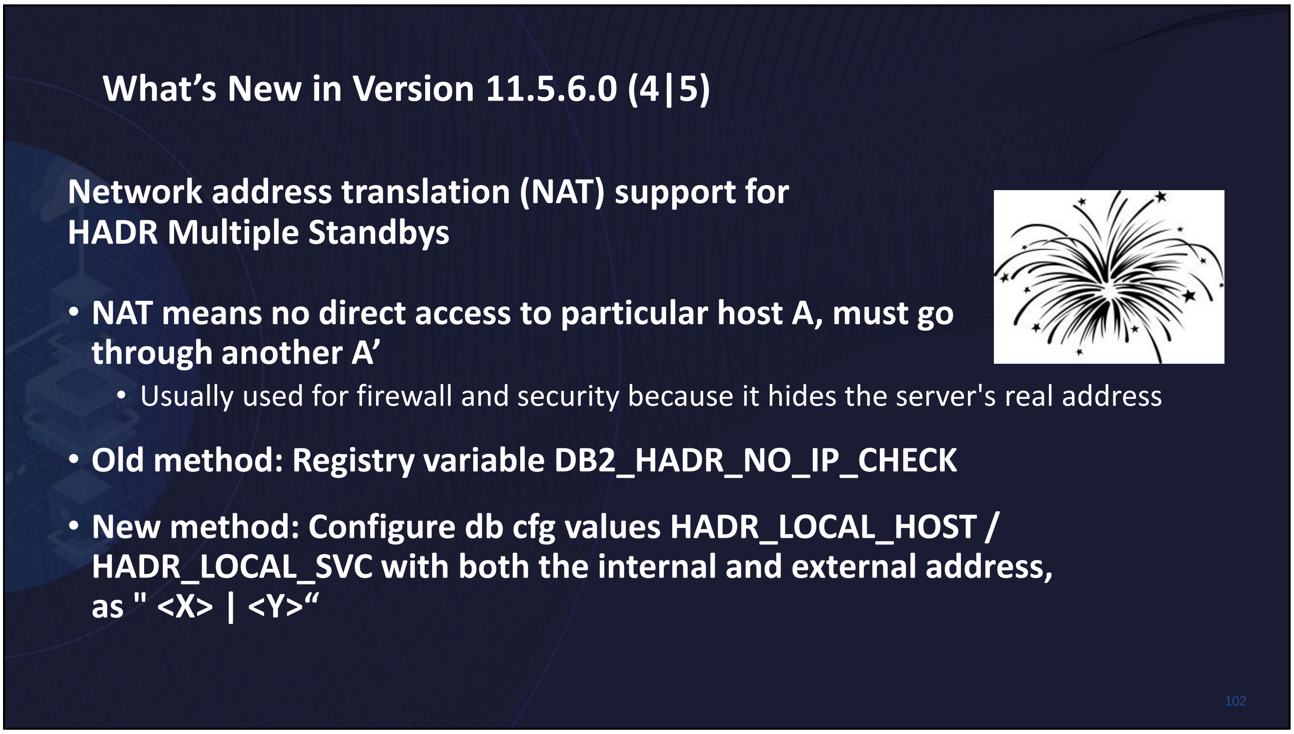


What’s New in Version 11.5.6.0 (3|5)

DB24LUW-I-109 - Allow usage of NX842 hardware compressed backup together with backup encryption

- Customer wants to use NX842 hardware accelerated backup compression in combination with encryption (AIX Power 7+)
- Today Db2 ships a library for compression, a library for encryption and a library for compression+encryption
- Solution is to create a new library libdb2nx842_encr.a/so that can be used like libdb2compr_encr.a/so on BACKUP command:

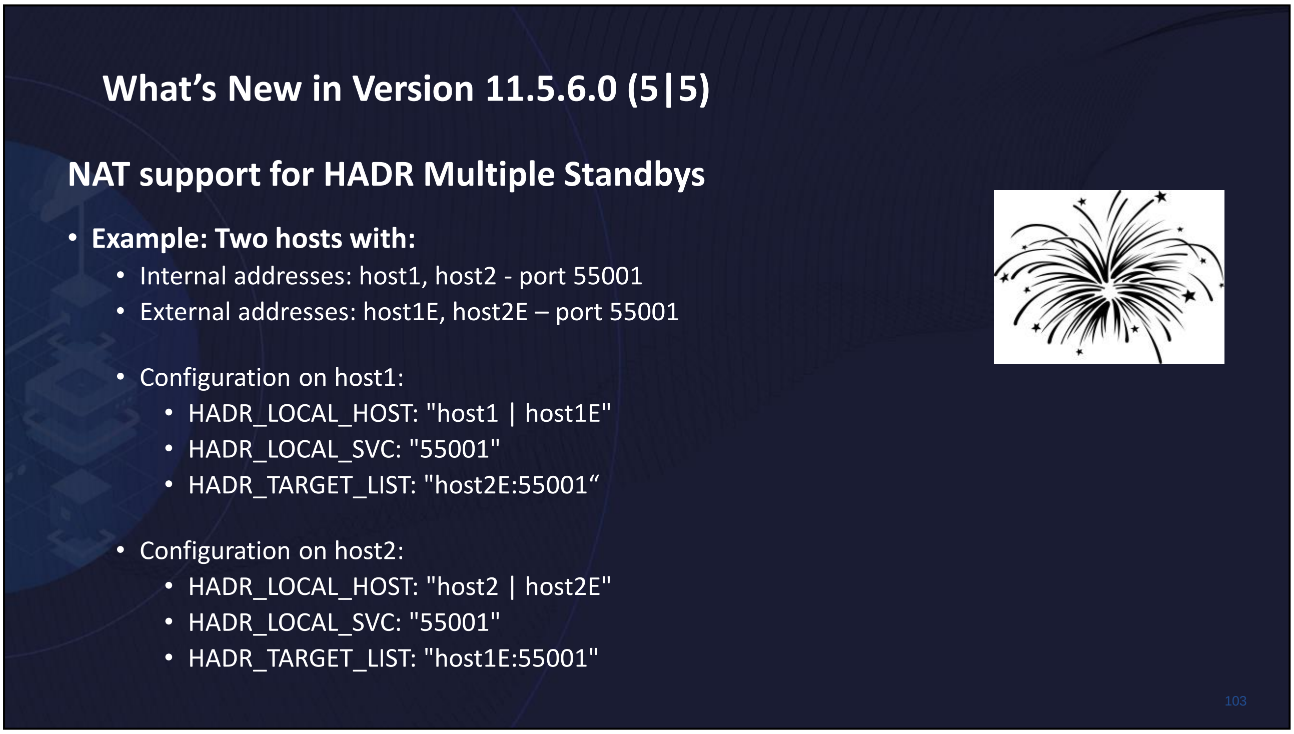
```
BACKUP DATABASE sample ENCRYPT ENCRLIB libdb2nx842_encr.a
```



What’s New in Version 11.5.6.0 (4|5)

Network address translation (NAT) support for HADR Multiple Standbys

- **NAT means no direct access to particular host A, must go through another A’**
  - Usually used for firewall and security because it hides the server's real address
- **Old method: Registry variable DB2_HADR_NO_IP_CHECK**
- **New method: Configure db cfg values HADR_LOCAL_HOST / HADR_LOCAL_SVC with both the internal and external address, as " <X> | <Y>“**



What’s New in Version 11.5.6.0 (5|5)

NAT support for HADR Multiple Standbys

- **Example: Two hosts with:**
  - Internal addresses: host1, host2 - port 55001
  - External addresses: host1E, host2E – port 55001

  - Configuration on host1:
    - HADR_LOCAL_HOST: "host1 | host1E"
    - HADR_LOCAL_SVC: "55001"
    - HADR_TARGET_LIST: "host2E:55001“

  - Configuration on host2:
    - HADR_LOCAL_HOST: "host2 | host2E"
    - HADR_LOCAL_SVC: "55001"
    - HADR_TARGET_LIST: "host1E:55001"

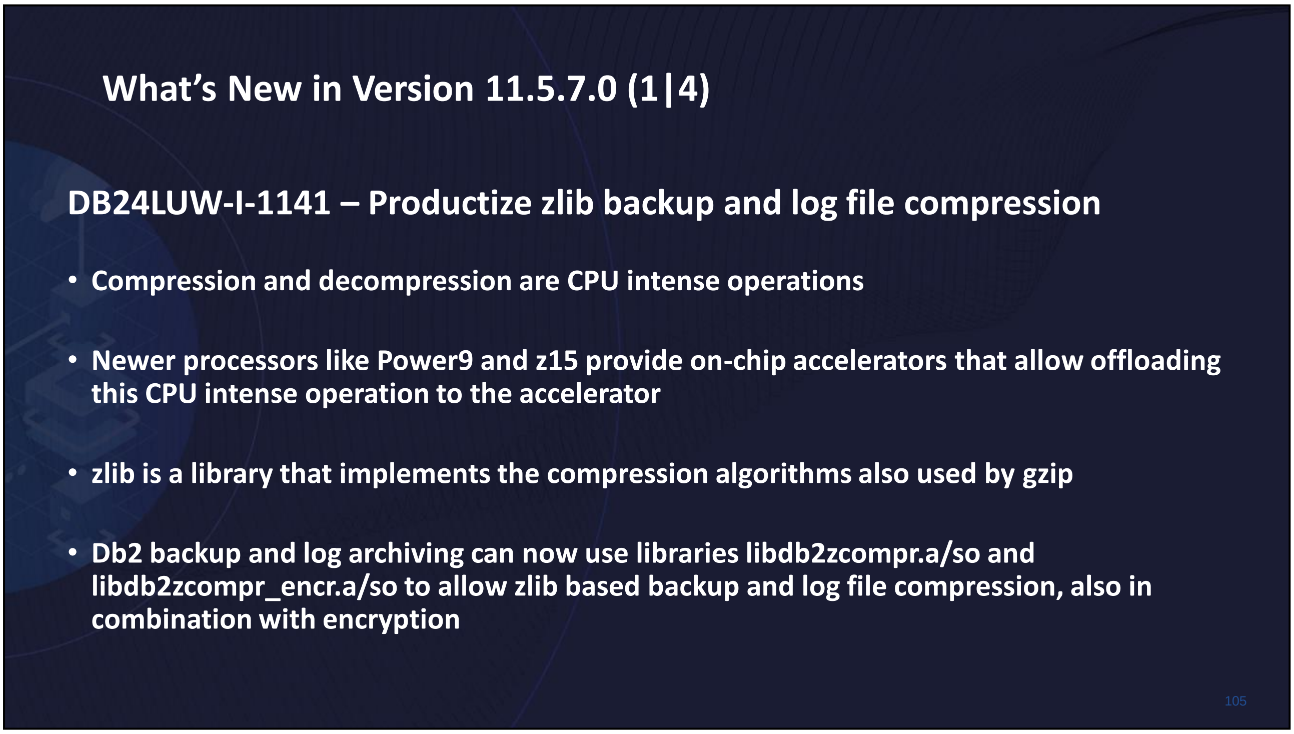
- What’s New in Version 11.5.6.0
- **What’s New in Version 11.5.7.0**

What’s New 11.5.6.0

- https://www.ibm.com/docs/en/db2/11.5?topic=new-1156

What’s New 11.5.7.0

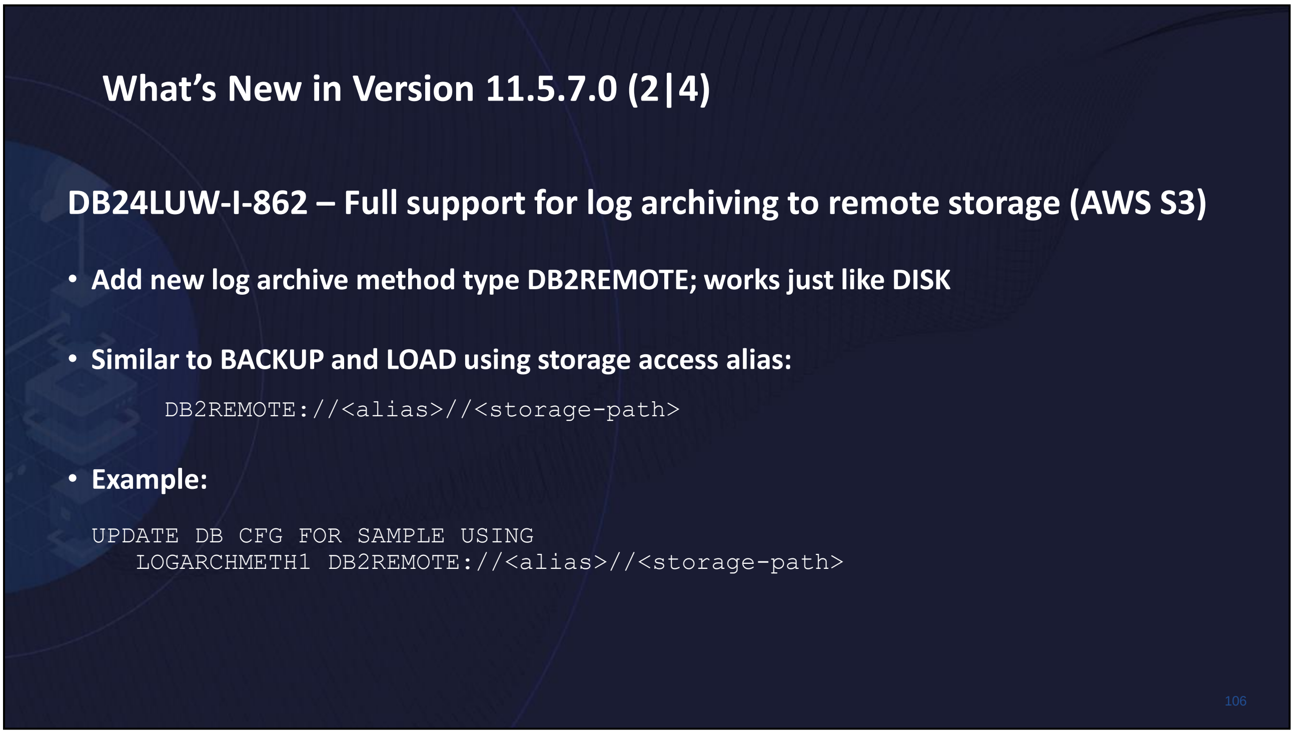
- https://www.ibm.com/docs/en/db2/11.5?topic=new-1157



What’s New in Version 11.5.7.0 (1|4)

DB24LUW-I-1141 – Productize zlib backup and log file compression

- Compression and decompression are CPU intense operations
- Newer processors like Power9 and z15 provide on-chip accelerators that allow offloading this CPU intense operation to the accelerator
- zlib is a library that implements the compression algorithms also used by gzip
- Db2 backup and log archiving can now use libraries libdb2zcompr.a/so and libdb2zcompr_encr.a/so to allow zlib based backup and log file compression, also in combination with encryption



What’s New in Version 11.5.7.0 (2|4)

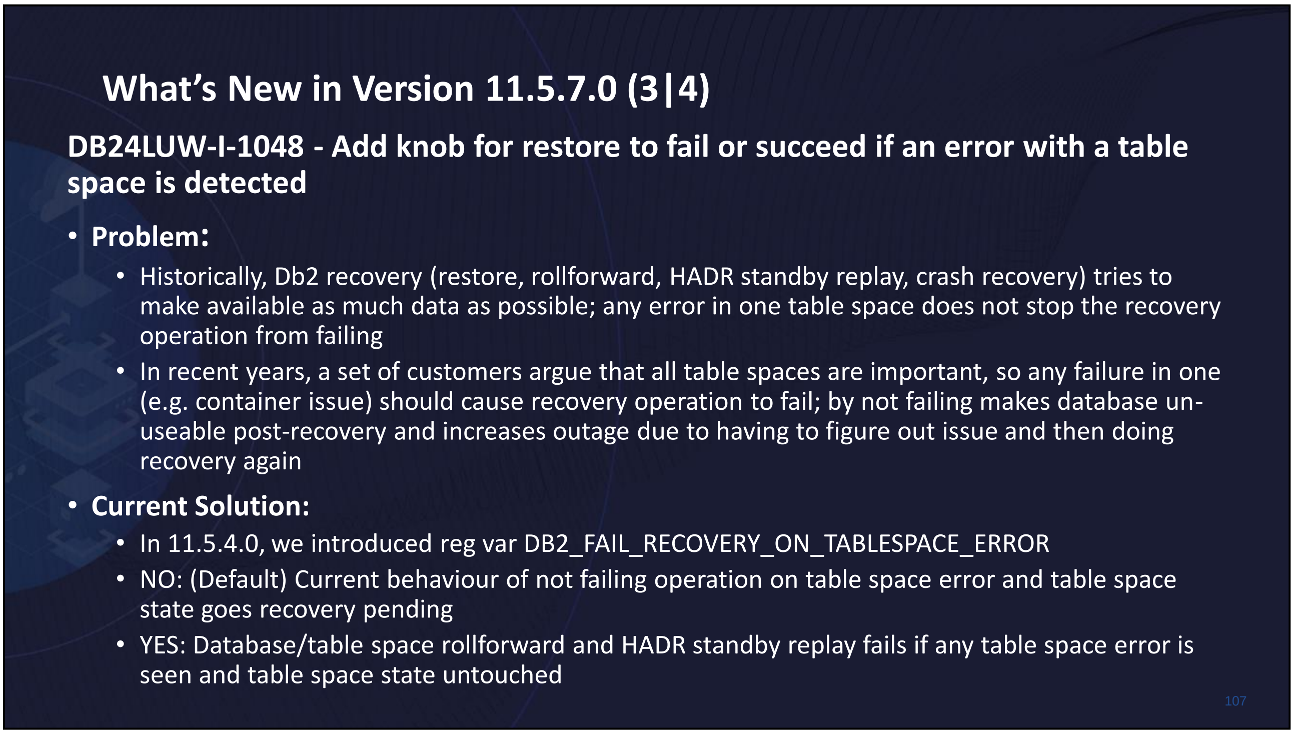
DB24LUW-I-862 – Full support for log archiving to remote storage (AWS S3)

- **Add new log archive method type DB2REMOTE; works just like DISK**
- **Similar to BACKUP and LOAD using storage access alias:**

```
DB2REMOTE://<alias>//<storage-path>
```

- **Example:**

```
UPDATE DB CFG FOR SAMPLE USING
   LOGARCHMETH1 DB2REMOTE://<alias>//<storage-path>
```



What’s New in Version 11.5.7.0 (3|4)

DB24LUW-I-1048 - Add knob for restore to fail or succeed if an error with a table space is detected

- **Problem:**
  - Historically, Db2 recovery (restore, rollforward, HADR standby replay, crash recovery) tries to make available as much data as possible; any error in one table space does not stop the recovery operation from failing
  - In recent years, a set of customers argue that all table spaces are important, so any failure in one (e.g. container issue) should cause recovery operation to fail; by not failing makes database un-useable post-recovery and increases outage due to having to figure out issue and then doing recovery again
- **Current Solution:**
  - In 11.5.4.0, we introduced reg var DB2_FAIL_RECOVERY_ON_TABLESPACE_ERROR
  - NO: (Default) Current behaviour of not failing operation on table space error and table space state goes recovery pending
  - YES: Database/table space rollforward and HADR standby replay fails if any table space error is seen and table space state untouched

What’s New in Version 11.5.7.0 (4|4)

DB24LUW-I-1048 - Add knob for restore to fail or succeed if an error with a table space is detected (cont’d)

- **In 11.5.7.0:**
  - Add restore to DB2_FAIL_RECOVERY_ON_TABLESPACE_ERROR
  - If set to YES, if restore has an issue with a table space container the restore will fail
    - Will consider any error when restoring the table space definition