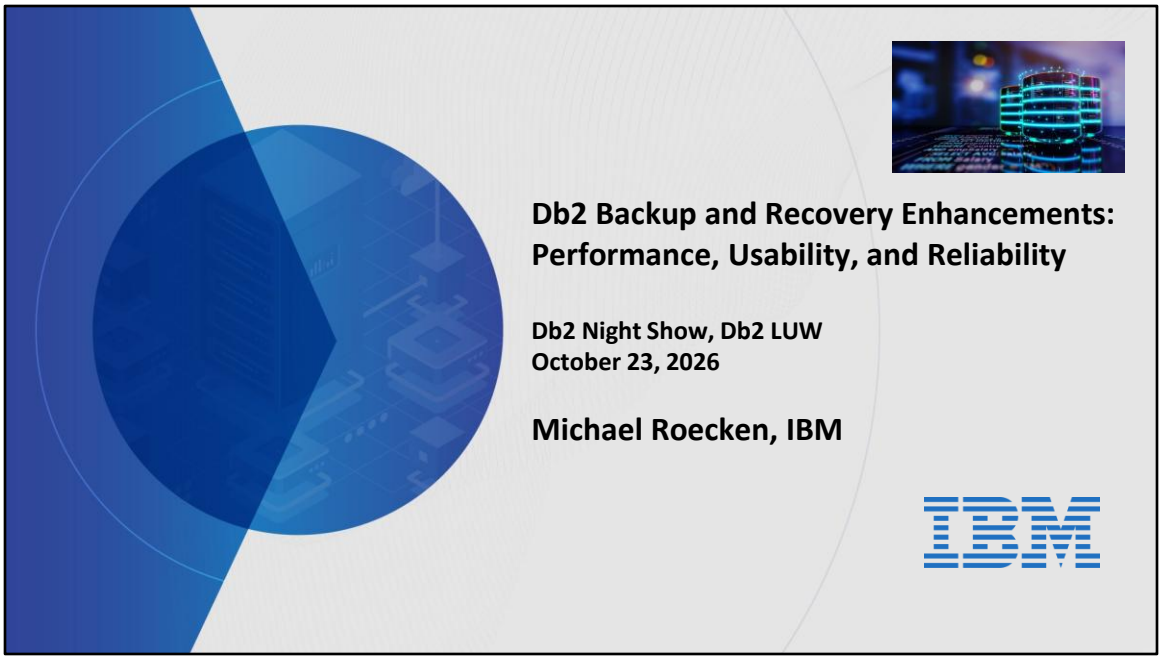


Since Db2 12.1, Backup and Recovery has undergone significant transformation addressing critical customer requirements and delivering measurable improvements in backup performance, operational flexibility, and data integrity. These enhancements span multiple deliverables since 12.1.0, introducing advanced parallelism capabilities, intelligent log management, extended metadata tracking, and flexible recovery object management. Collectively, these improvements reduce backup elapsed time, eliminate costly archive retrievals, enhance operational visibility, and provide greater control over backup and recovery operations.

1. **Performance Optimization** - Backup Intra-tablespace Parallelism (ITP), reduced log retrieval during backups, optimized flushing behavior, and early log inclusion in backup images to avoid costly archive retrievals.
2. **Enhanced Usability** - Extended history file comment fields, backup size tracking, encryption metadata storage, protected history entries, and improved vendor storage management.
3. **Data Integrity and Reliability** - DB2_BCKP_PAGE_VERIFICATION enabled by default, improved backup validation capabilities, and enhanced timeout handling for long-running operations.

Safe Harbor Statement

Copyright © IBM Corporation 2026. All rights reserved.

U.S. Government Users Restricted Rights - Use, duplication, or disclosure restricted by GSA ADP Schedule Contract with IBM Corporation

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.

IBM, the IBM logo, ibm.com and Db2 are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml

1. Performace

Enhancements aimed at reducing backup elapsed time, eliminating bottlenecks, and ensuring faster start and recovery times.

- **Backup intra-table-space parallelism (ITP)** ***(Release 12.1.0)***
  - **What it does:** Several backup media (BM) agents back up one large table space simultaneously by splitting DMS and automatic storage table spaces into page chunks.
  - **Registry variable / Interface:** `DB2_BACKUP_ITP` (default: `ON`)
- **Reduced flushing during backup** ***(Release 12.1.2)***
  - **What it does:** Backup flushes dirty pages only up to `lowtranlsn` (the oldest uncommitted change) rather than the maximum LSN, allowing backups to start much sooner.
  - **Registry variable / Interface:** `DB2_REDUCE_FLUSHING_DURING_BACKUP` (default: `ON`)

Db2 12.1 delivers comprehensive advancements across three pillars: dramatic performance gains through parallelism and smarter flushing, ironclad data integrity with inline verification and log retention, and modernized operations with decoupled pruning and cloud-native storage.

2. Reliability

Enhancements focused on early defect detection, data integrity, and standalone self-contained recovery.

- **Backup page verification** *(Release 12.1.0)*
  - **What it does:** Checksums pages read during backup; a corrupt page fails the backup immediately instead of silently ending up in the backup image.
  - **Registry variable / Interface:** `DB2_BCKP_PAGE_VERIFICATION` (default: `ON`)
- **Log retention for `BACKUP .. INCLUDE LOGS`** *(Release 12.1.3)*
  - **What it does:** Retains active logs needed by the backup locally so they are not archived, renamed, or reused before the backup finishes copying them.
  - **Registry variable / Interface:** `LOG_DISK_CAP` (Database Configuration parameter in `MB`)

3. Usability (and Operability)

Enhancements that simplify administration, integrate cloud and snapshot lifecycles, improve cluster recovery agility, and provide richer diagnostics.

- **Decoupled history-file pruning** *(Release 12.1.4)*
  - **What it does:** Moves history pruning out of the backup execution path into an asynchronous `db2autoprune` thread.
  - **Registry variable / Interface:** `DB2_AUTOPRUNE_EDU` (default: `ON`)
- **Automatic backup to remote storage** *(Release 12.1.0)*
  - **What it does:** Allows automatic backup policies to write directly to cataloged storage access aliases (such as COS / S3), including image pruning.
  - **Registry variable / Interface:** Automatic backup policy XML
- **History file improvements** *(Release 12.1.0)*
  - **What it does:** Records total image size, sequence size, compression library, encryption status, and `INCLUDE LOGS` flag; comments expand from 30 to 254 characters.
  - **Registry variable / Interface:** `LIST HISTORY`, `SYSIBMADM.DB_HISTORY`
- **Pruning for archived logs linked to non-Db2 snapshots** *(Release 12.5.0)*
  - **What it does:** External snapshots taken during write suspend can be registered as backups to allow automatic pruning of archived logs.
  - **Registry variable / Interface:** `RECORD BACKUP` clause (`SET WRITE RESUME`, `RESTART DATABASE`, `db2inidb`)
- **Recovery through drop member (pureScale)** *(Release 12.1.0)*
  - **What it does:** Dropping a member preserves recovery metadata and logs; no topology life ID change, enabling restore and rollforward across dropped members. No new backup enforcement.
  - **Registry variable / Interface:** `db2iupdt -drop -m`
- **Restore/upgrade of downlevel images with topology changes (pureScale)** *(Release 12.1.3)*
  - **What it does:** `RESTORE DATABASE` and migration utilities handle pureScale downlevel backup images when the member topology differs from the target instance.
  - **Registry variable / Interface:** `CATALOG`, `UPGRADE`, `RESTORE DATABASE` commands
- **Object sizes in `db2adutl query`** *(Release 12.1.2)*
  - **What it does:** Displays backup image and archive log sizes directly in query output.
  - **Registry variable / Interface:** `db2adutl query`
- **BAR stats output improvements** *(Release 12.1.0)*
  - **What it does:** Adds `LogPhase` and `LogBytes` columns; formats BM elapsed times as human-readable `HH:MM:SS` in both `db2pd -barstats` and `db2diag.log`.
  - **Registry variable / Interface:** `db2pd -barstats`, `db2diag.log`
- **`db2pd -barstats` without an EDU ID** *(Release 12.1.4)*
  - **What it does:** Automatically discovers active backup/restore operations for a database without requiring an explicit agent EDU ID.
  - **Registry variable / Interface:** `db2pd -db <db> -barstats`

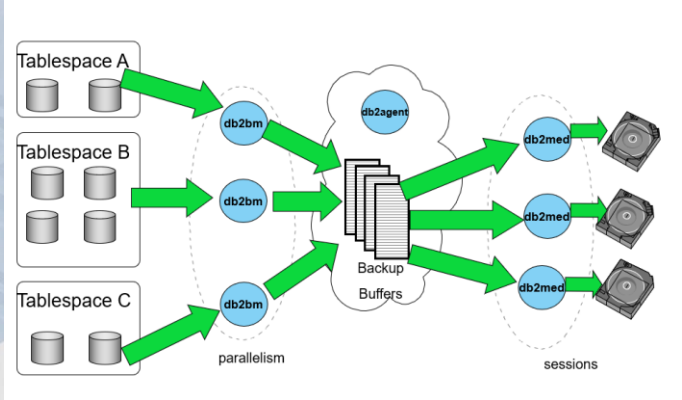
1. Performace – A Closer Look

Enhancements aimed at reducing backup elapsed time, eliminating bottlenecks, and ensuring faster start and recovery times.

- **Backup intra-table-space parallelism (ITP)** ***(Release 12.1.0)***
- **Reduced flushing during backup** ***(Release 12.1.2)***

Backup intra-table space parallelism (ITP) [1|4]

(Release 12.1.0)



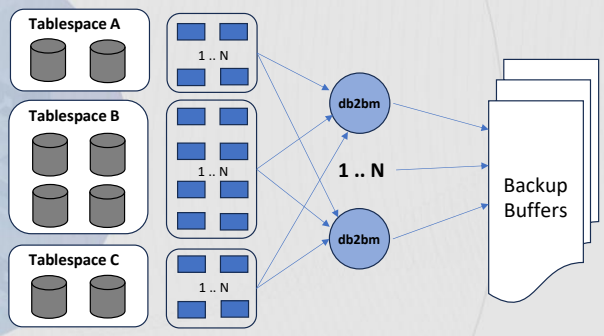
- **Today, backup assigns one db2bm EDU to a table space (tuned based on PARALLELISM)**
  - If a few large table spaces and many small table spaces, the small table spaces will be backed up quickly but then the large table spaces are read using a single EDU ➔ overall backup time is proportional to the size of the largest table space

7

ITP solves the classic bottleneck where one massive table space forced a single BM to do all the work while other BMs sat idle. DMS and automatic storage table spaces are partitioned into chunks dynamically, achieving up to 6x faster backups without altering the image format or restore path.

Backup intra-table space parallelism (ITP) [2|4]

(Release 12.1.0)



- **Solution is to break up table spaces into smaller segments (1 .. N) that can be assigned to BMs to be processed in parallel**
- **Split table space into an arbitrary number of chunks that 1 .. N db2bms will read**
  - N determined by resource factors, like CPU, memory and parallel IO
- **External registry variable:**
  - `DB2_BACKUP_ITP=ON/OFF`
  - Default is ON

8

Backup intra-table space parallelism (ITP) [3|4]

(Release 12.1.0)

- **In cases where the sizes of table spaces are highly skewed (e.g. one tablespace is >70% of the total database size) we are seeing performance improvements of 2x or more**
- **In cases where the table spaces are balanced, we see much smaller improvements**
- **The amount of improvement we can get is capped by the I/O capacity of the channel between the disk and the CPU**
- **ITP works by making unused I/O capacity in the system available. Once that capacity is filled, you hit the maximum improvement that ITP can provide**
- **We were able to greatly improve ITP performance by adding I/O capacity to our test system**
  - For example: install another fiber channel to your disk

9

Backup intra-table space parallelism (ITP) [4|4]

Backup DB SRC 468GB – Backup Parallelism 4 - Target 2 - Compress

db2 "backup database SRC to /backup/db2backup/SRC, /backup/db2backup/SRC with 8 buffers parallelism 4 compress"

Largest TS: 263GB
All TS: 468GB => Largest TBSP 56% of all tablespaces

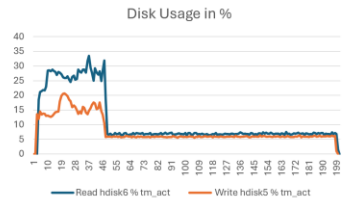
ITP OFF – 5879s – 100%

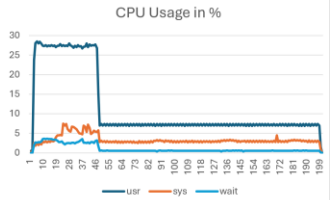
```
2024-07-08-15.23.02.164386+120 E9451776A2152    LEVEL: Info

Parallelism    = 4
Number of buffers = 8
Buffer size    = 16781312 (4097 4kB pages)

                                                                                                  Compr
BM#  Total    I/O      Compr    MsgQ     WaitQ    Throttle LogPhase  Buffers  MBytes  MBytes LogBytes
---  -------- -------- -------- -------- -------- -------- -------- -------- ------- ------- --------
000  1:37:59  0:07:03  1:30:12  0:00:00  0:00:00  0:00:00  0:00:00    16778  269690  228975        0
001  1:37:59  0:01:51  0:20:13  0:00:00  1:15:52  0:00:00  0:00:00     2263   65676   65276        0
002  1:37:59  0:02:00  0:20:04  0:00:00  1:15:52  0:00:00  0:00:00     1945   71132   70434        0
003  1:37:59  0:02:04  0:20:01  0:00:00  1:15:52  0:00:00  0:00:00     2132   71909   71620        0
---  -------- -------- -------- -------- -------- -------- -------- -------- ------- ------- --------
TOT    -        -        -        -        -        -        -       23118  478409  436306        0

MC#  Total    I/O              MsgQ     WaitQ    Buffers  MBytes
---  -------- --------         -------- -------- -------- --------
000  1:37:59  0:06:15           1:31:43  0:00:00    11357   181740
001  1:37:59  0:06:31           1:31:28  0:00:00    11763   188221
---  -------- --------         -------- -------- -------- --------
TOT    -        -                -        -       23120   369962
```





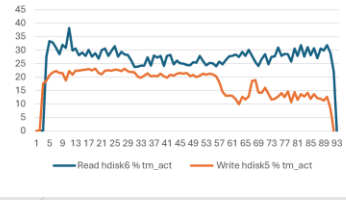
ITP ON, PARA 4 – 2654s – 45%

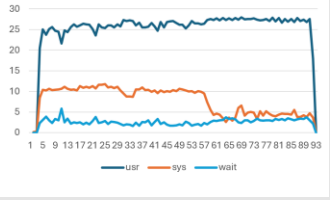
```
2024-07-08-16.10.10.539351+120 E9713896A2152    LEVEL: Info

Parallelism    = 4
Number of buffers = 8
Buffer size    = 16781312 (4097 4kB pages)

                                                                                                  Compr
BM#  Total    I/O      Compr    MsgQ     WaitQ    Throttle LogPhase  Buffers  MBytes  MBytes LogBytes
---  -------- -------- -------- -------- -------- -------- -------- -------- ------- ------- --------
000  0:44:14  0:04:22  0:39:37  0:00:00  0:00:00  0:00:00  0:00:00     5800  117628  109349        0
001  0:44:14  0:04:32  0:39:29  0:00:00  0:00:00  0:00:00  0:00:00     5792  121651  107956        0
002  0:44:14  0:04:26  0:39:33  0:00:00  0:00:00  0:00:00  0:00:00     5928  118155  104372        0
003  0:44:14  0:04:27  0:39:32  0:00:00  0:00:00  0:00:00  0:00:00     5598  120973  107964        0
---  -------- -------- -------- -------- -------- -------- -------- -------- ------- ------- --------
TOT    -        -        -        -        -        -        -       23118  478409  429644        0

MC#  Total    I/O              MsgQ     WaitQ    Buffers  MBytes
---  -------- --------         -------- -------- -------- --------
000  0:44:14  0:08:09           0:36:05  0:00:00    11598   185597
001  0:44:14  0:08:07           0:36:06  0:00:00    11522   184365
---  -------- --------         -------- -------- -------- --------
TOT    -        -                -        -       23120   369962
```





10

- POWER 8
- FC: 2x 8GBit ports – SDD and HDD on different ports
- AIX 7200-05-07-2346
- Backup target: HDD
- Backup source: SDD
- Db2 software compression used to slowdown I/O rate

Reduced flushing during backup
(Release 12.1.2)

- **Before:** An online backup began by flushing every dirty page up to the highest LSN
  - On a busy system, or one with a lot of dirty pages, the backup of user data could start much later than when the command was issued
  - Large bufferpools with concurrent workload can spend long time flushing
    - We have seen sometimes 30+ minutes
  - Long running online backups means more logs to retrieve and include into image
- **After:** Backup flushes only to lowtranlsn, the oldest uncommitted change
  - Pages dirtied by later transactions do not have to be written to disk before the backup of user data begins
  - Reduces the amount of bufferpool flushing during online backup
- **Benefit:**
  - Improved bufferpool flushing can see drastic performance gains for certain scenarios
  - Shorter running backups == less logs to include == quicker rollforward to end of backup
- **Registry variable / Interface:** `DB2_REDUCE_FLUSHING_DURING_BACKUP` (default: `ON`)
  - **Pre-12.1.2: Default is OFF**
    - SAP runs ON under DB2_WORKLOAD
  - **12.1.2: Default is now ON**
    - Removed from DB2_WORKLOAD=SAP

11

Prior to 12.1.2, flushing dirty buffer pool pages up to max LSN caused heavy delay for online backups. By flushing only up to lowtranlsn, backups start immediately and generate fewer logs, drastically shortening subsequent rollforward recovery time as well.

DB2_REDUCE_FLUSHING_DURING_BACKUP

```
- Operating system: All
- Default: ON, Values: ON | OFF
- At the beginning of an online backup operation, changed
pages within the buffer pool must be persisted to table
space storage. The persisting of these changed pages can
increase the duration of the backup operation in
configurations with very large buffer pools. This
variable specifies whether an online backup runs reduced
flushing of changed pages within the buffer pool.
- Changes to this variable do not require the database
instance to be restarted.
- Changes to this variable have no effect on an online
backup operation that is running at the time of the
```

variable change.

2. Reliability – A Closer Look

Enhancements focused on early defect detection, data integrity, and standalone self-contained recovery.

- **Backup page verification** *(Release 12.1.0)*
- **Log retention for `BACKUP .. INCLUDE LOGS`** *(Release 12.1.3)*

Backup page verification
(Release 12.1.0)

- **Before:** A backup would copy a corrupt page into the image without complaint
  - You would find out at restore or rollforward time, which is the worst time
- **After:** Backup verifies the page checksum of each page it reads
  - If a page fails, the backup fails
- **Benefit:**
  - You learn about corruption on the day it is backed up and while you still have the copy of the data, rather than when you need the image
  - Have a chance to take a new backup if issue is transient or after issue is resolved
- **Registry variable / Interface:** `DB2_BCKP_PAGE_VERIFICATION` (default: `ON`)
  - **Pre-12.1: Default is OFF**
    - SAP runs ON under DB2_WORKLOAD
    - pureScale runs ON
  - **12.1: Default is now ON**
    - Removed from DB2_WORKLOAD=SAP
    - Performance impact: 2% degradation

13

Silent data corruption is a DBA's worst nightmare when discovered during a disaster recovery restore. With Backup Page Verification enabled by default, Db2 performs hardware-accelerated checksum verification with only ~2.5% overhead, catching corrupted pages immediately.

DB2_BCKP_PAGE_VERIFICATION

•Operating system: All

•Default: ON, Values: ON， OFF

•Specifies whether DMS and AS page validation occurs during a backup. This registry variable is dynamic, that is, it can be set or unset without having to stop and start the instance.

Log retention for BACKUP .. INCLUDE LOGS
(Release 12.1.3)

- **Previous Problem:**
  - Transaction logs generated during online backup must be included in the backup image (unless EXCLUDE LOG is specified) and is done at the end of backup
  - Logs in the active path can be deleted/reused at any time once archived
  - When backup needs to include logs, logs not found locally need to be retrieved
  - Log retrievals could be costly and impact backup performance
- **Solution:**
  - When `LOG_DISK_CAP` db cfg parm is set to a non-zero value, logs required by online backup are no longer deleted/reused while backup in progress
    - No retrieval from the archives, so faster and more reliable online backups
    - `UPDATE DB CFG FOR sample USING LOG_DISK_CAP 200` (in MB)
    - Use a size large enough to comfortably hold the logs that accumulate during your typical backup window.
  - NOTE: Retention is strictly **best-effort**
    - If disk space runs low or LOG_DISK_CAP is reached, logs will be reused and Db2 falls back to archive retrieval; it will never fill the filesystem



14

INCLUDE LOGS backups ensure standalone recoverable images. With LOG_DISK_CAP in 12.1.3, Db2 temporarily retains necessary logs on disk during the backup window, eliminating expensive round-trips to the archive while safeguarding against filesystem exhaustion.

log_disk_cap - Active log space disk capacity configuration parameter https://www.ibm.com/docs/en/db2/12.1.x?topic=dcp-log-disk-cap-active-log-space-disk-capacity

NOTE: Databases in single and multi-partitioned (DPF) instances are fully covered. In pureScale, only the member running the backup keeps its logs; remote members may still need to retrieve logs from the archive.

3. Usability (and Operability) – A Closer Look

Enhancements that simplify administration, integrate cloud and snapshot lifecycles, improve cluster recovery agility, and provide richer diagnostics.

- **Automatic backup to remote storage** (Release 12.1.0)
- **History file improvements** (Release 12.1.0)
- **Decoupled history-file pruning from backup** (Release 12.1.4)
- **Pruning for archived logs linked to non-Db2 snapshots** (Release 12.5.0)
- **Recovery through drop member (pureScale)** (Release 12.1.0)
- **Restore/upgrade of downlevel images with topology changes (pureScale)** (Release 12.1.3)
- **Object sizes in db2adutl query** (Release 12.1.2)
- **BAR stats output improvements** (Release 12.1.0)
- **db2pd -barstats without an EDU ID** (Release 12.1.4)

Automatic backup to remote storage
(Release 12.1.0)

- **BACKUP command currently supports remote/object storage as a media type:**

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

- **There also exists automatic backup through AUTO_MAINT / AUTO_DB_BACKUP db cfg parms**
- **This is a policy-based backup using XML syntax – it does not have syntax to support remote storage as a media type**
- **In 11.5.7/8, we enhanced our object storage support, this extends that to automatic backup**
- **Example policy file would contain:**

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

16

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

Automatic history file pruning for automatic Db2 backups
(Release 12.1.0)

- **Applies to automatic backup through `AUTO_MAINT` / `AUTO_DB_BACKUP` db cfg parms**
- **Currently has its own automatic pruning policy of Db2 backup images (keeps most recent)**
- **This contradicts Db2's automatic pruning feature controlled by `AUTO_DEL_REC_OBJ` db cfg parm**
- **Request is to have automatic backup and automatic pruning behave together**
- **Solution:**
  - If `AUTO_DEL_REC_OBJ` is `ON` then use Db2's automatic pruning feature
  - Otherwise continue to use current automatic backup policy of keeping most recent

17

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

History file improvements [1|5]

(Release 12.1.0)

Improvements:

1. **Provide Backup Size Info In History File**
2. **Store encrypted backup information in history file**
3. **Increase size of COMMENT field**
   - Increased from 30 to 254 characters

Motivation:

- **More customers want better reporting or to script around inventory of objects, like images and log files:**
  - Object sizes
  - Compressed? Encrypted? Backup includes logs?
  - Build restore estimations based on backup image size

18

Richer History Metadata (12.1.0): LIST HISTORY and SYSIBMADM.DB_HISTORY record Total Size, Seq Size, Compression Library (C, U, N, Z), Encryption status, and IncludeLogs.

Expanded Comments: History comments expanded from 30 to 254 characters.

History file improvements [2|5]
(Release 12.1.0)

- History file entry for backup:
  - Total Size: total size of all backup image sequences in bytes
  - Sequence Size: size of an individual backup image sequence in bytes
  - Compression Lib: ' ' (Not Compressed), 'C' (Default Compr Lib), 'U' (User Defined Custom Lib), 'N' (NX842), 'Z' (ZLIB)
  - Encrypted: Yes/No for 'LIST HISTORY', 1/0 for 'SYSIBMADM.DB_HISTORY'
  - Include Logs: Yes/No for 'LIST HISTORY', 1/0 for 'SYSIBMADM.DB_HISTORY'

- History file entry for load copy:
  - Total Size: total size of all load copy image sequences in bytes
  - Sequence Size: size of an individual load copy image sequence in bytes
  - Encrypted: Yes/No for 'LIST HISTORY', 1/0 for 'SYSIBMADM.DB_HISTORY'

- History file entry for log:
  - Total Size: actual size of the archived log in bytes
  - Compression Lib: ' ' (Not Compressed), 'C' (Default Compr Lib), 'N' (NX842), 'Z' (ZLIB)
  - Encrypted: Yes/No for 'LIST HISTORY', 1/0 for 'SYSIBMADM.DB_HISTORY'

19

History file improvements [3|5]
(Release 12.1.0)

- SYSIBMADM.DB_HISTORY displays Total_Size, Seq_Size, Compression_Library, Encrypted, and IncludeLogs for an online backup image made up of three sequences with compression(ZLIB) and encryption enabled:

```
$ db2 "select EID,TOTAL_SIZE, SEQ_SIZE, COMPRESSION_LIBRARY, ENCRYPTED, INCLUDE_LOGS from SYSIBMADM.DB_HISTORY where OPERATION='B'"

EID                 TOTAL_SIZE          SEQ_SIZE            COMPRESSION_LIBRARY ENCRYPTED INCLUDE_LOGS
------------------- ------------------- ------------------- ------------------- --------- ------------
                  4            48308224            24166400 Z                           1            1
                  5            48308224            12070912 Z                           1            1
                  6            48308224            12070912 Z                           1            1

  3 record(s) selected.
~
```

History file improvements [4|5]
(Release 12.1.0)

- **SYSIBMADM.DB_HISTORY displays Total Size, Compression Library and Encrypted for a log archive with both compression and encryption enabled:**

```
$ db2 "select EID, TOTAL_SIZE, COMPRESSION_LIBRARY, ENCRYPTED from SYSIBMADM.DB_HISTORY where OPERATION='X'"

EID                 TOTAL_SIZE          COMPRESSION_LIBRARY ENCRYPTED
------------------- ------------------- ------------------- ---------
                  2              356352 C                           1

  1 record(s) selected.
```

History file improvements [5|5]

(Release 12.1.0)

- **LIST HISTORY displays Total Size, Compression Library and Encrypted for a log archive with both compression and encryption enabled:**

```
 $ db2 list history archive log all for db sample

Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log  Backup ID
-- --- ------------------ ---- --- ------------ ------------ --------------
 X  D  20240711195435       1    D  S0000000.LOG C0000000
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
   Comment:
Start Time: 20240711195435
  End Time: 20240711195436
    Status: A
Total Size: 356352 (bytes)
 Compr Lib: C
 Encrypted: Yes
--------------------------------------------------------------------------------
 EID: 2 Location: /home/wlarry/logs/wlarry/SAMPLE/NODE0000/LOGSTREAM0000/C0000000/S0000000.LOG
```

Decoupled history-file pruning from backup [1|2]

(Release 12.1.4)

- **Most-Voted Availability RFE/Idea**
  - Previously, automatic history pruning (and `AUTO_DEL_REC_OBJ` file deletion) ran synchronously inside the backup job, inflating backup windows esp. when dealing with large history files or vendor/remote storage
  - Utility monitoring would show backup stuck at 99%
- **Solution:**
  - Automatic history pruning is decoupled from backup and done asynchronously by new db2autoprune EDU (thread) spawned on database activation
  - Can be configured through `DB2_AUTOPRUNE_EDU` [Default `ON`]
  - Trigger for async autoprune:
    - Periodically: every 24 hours
    - Successful Backup:
      - Online backup sends signal to db2autoprune to execute an immediate auto prune
      - Offline backup registers to kick of auto prune on next database activation
- **Benefit:** Predictable and faster backups

23

Decoupling history pruning resolves a top customer request. Pruning of history records and deleted recovery objects now occurs asynchronously via the db2autoprune EDU every 24 hours or post-backup, stabilizing backup execution times.

Decoupled history-file pruning from backup [2|2]

(Release 12.1.4)

- **Check db2diag.log for completion message**
  - After every successful auto prune operation, pruning statistics are dumped in the db2diag.log by db2autoprune:

```
2026-07-29-11.29.41.035916-420 E1049939E855          LEVEL: Info
PID     : 126766               TID : 139759577982528 KTID : 128676<
PROC    : db2sysc
INSTANCE: aprilico             NODE : 000            DB   : SAMPLE
APPHDL  : 0-94                 APPID: *LOCAL.DB2.260729182942
AUTHID  : APRILICO             HOSTNAME: db2i-aprillico1-xneww-x86.de
EDUID   : 547                  EDUNAME: db2autoprune (SAMPLE)
FUNCTION: DB2 UDB, database utilities, sqluxLogDataStats, probe:2050
MESSAGE : Database Recovery History File Pruning Statistics
DATA #1 : String, 272 bytes

 Auto deletion of recovery objects (AUTO_DEL_REC_OBJ): ON
Number of database backups to retain (NUM_DB_BACKUPS): 1
  Recovery history retention (days) (REC_HIS_RETENTN): 0

  Elapsed processing time: 0.02 seconds
 Number of entries pruned: 0
Number of objects deleted: 0
```

- **PRUNE HISTORY command can now be issued without timestamp**
  - Like auto prune; based on `NUM_DB_BACKUPS`, `REC_HIS_RETENTN` and `AUTO_DEL_REC_OBJ`

24

Decoupling history pruning resolves a top customer request. Pruning of history records and deleted recovery objects now occurs asynchronously via the db2autoprune EDU every 24 hours or post-backup, stabilizing backup execution times.

Pruning for archived logs linked to non-Db2 snapshots [1|2]

(Release 12.5.0)

- **Before:**
  - A storage snapshot while the database was suspended (`SET WRITE SUSPEND`) was not recorded in the history file
    - This means automatic history file pruning was not triggered and so `AUTO_DEL_REC_OBJ` was not honored leaving archive logs in place
    - Users would need to do manual clean up or run into the risk of hitting disk full in the archive target
- **After:**
  - Users can now specify to record a non-Db2 snapshot backup in the history file:
    - `RECORD BACKUP` syntax:
      - `SET WRITE RESUME FOR DATABASE` **`RECORD BACKUP`**
      - `RESTART DB sample WRITE RESUME` **`RECORD BACKUP`**
      - `db2inidb sample AS STANDBY` **`RECORD BACKUP`**

25

Example: non-Db2 snapshot backup:

1. *db2 SET WRITE SUSPEND FOR DB*
2. *__copy database and log files using non-Db2 tool__*
3. *db2 SET WRITE RESUME FOR DB*

Pruning for archived logs linked to non-Db2 snapshots [2|2]
(Release 12.5.0)

- **Example:**
  - `db2 SET WRITE SUSPEND FOR DB`
  - <copy database/log paths using non-Db2 tool>
  - `db2 SET WRITE RESUME FOR DB RECORD BACKUP`
- **History entry: Op value "S" for non-Db2 snapshot backup**
  - Treated as a BACKUP entry and used to manage archive log files by auto and manual prune
  - Entry represents SET WRITE SUSPEND so no deletion of "image" created by non-Db2 snapshot

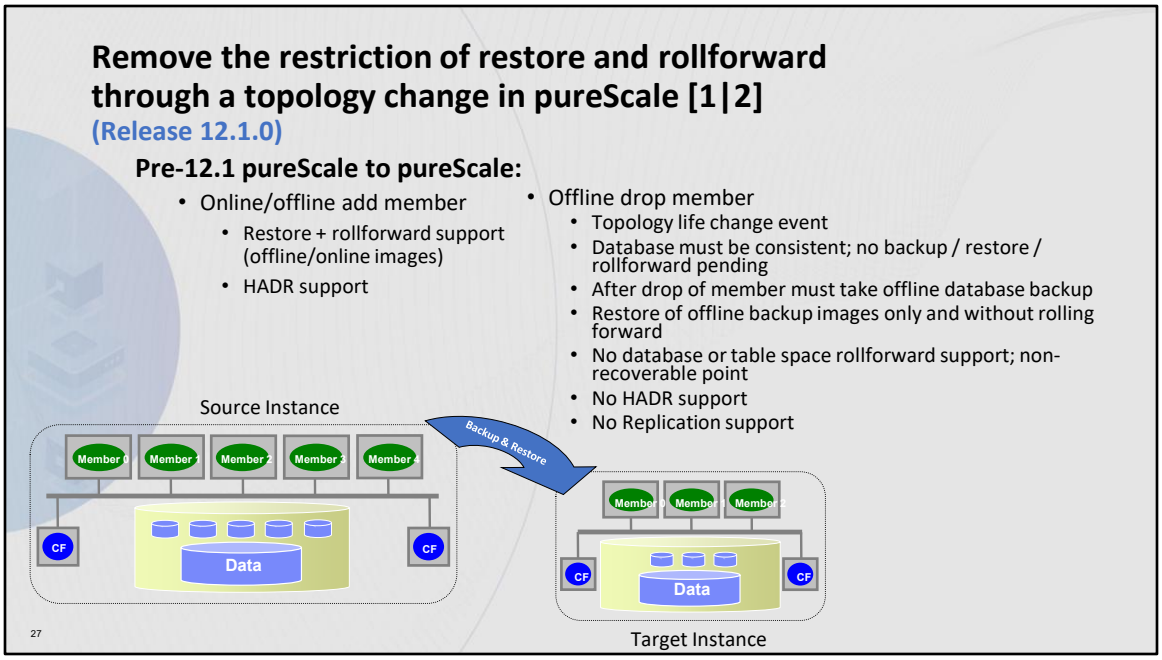
```
Op Obj Timestamp+Sequence Type Dev Earliest Log Current Log  Backup ID
-- --- ------------------ ---- --- ------------ ------------ --------------
S  D   20260508142954      N       S0000032.LOG S0000032.LOG
--------------------------------------------------------------------------------
 Contains 3 tablespace(s):

00001 SYSCATSPACE
00002 TEMPSPACE1
00003 USERSPACE1
--------------------------------------------------------------------------------
    Comment: SET WRITE SUSPEND INCLUDE LOGS
Start Time: 20260508142954
  End Time: 20260508142954
    Status: A
--------------------------------------------------------------------------------
 EID: 22
```

26

- Limitation:
  - Unlike integrated backup, history file entry does not have image location so backup is not deleted when entry is pruned
  - Cannot be used as part of RECOVER/REBUILD/RESTORE/INCREMENTAL AUTOMATIC command

- History entry is tied to SWS command so non-integrated backup taken offline is NOT recorded



Db2 10.5 came out with online add member and restore+rollforward support. Drop member is offline with these restrictions:

The last member cannot be dropped.

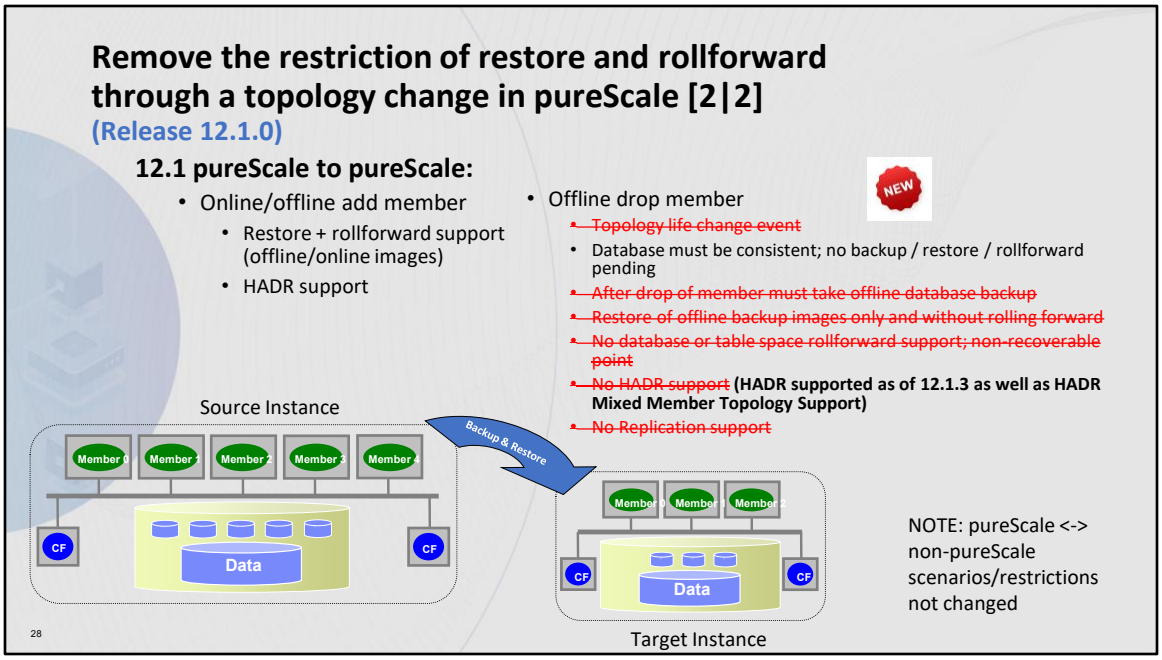
The catalogued databases must satisfy the following criteria:

- Consistent database.
- Have at least one reconciled member remaining after the drop.
- HADR role standard.
- Not backup pending.
- Not topology life change pending or in progress.
- Not database rollforward pending.
- All table spaces must be in normal state (for example, not rollforward or restore pending).
- Not restore pending.

A drop member is considered a topology life change event: in addition to writing a topology life change log record, this generates a new topology life identifier.

The user is required to take a full offline database backup after one or more drop member operations.

Any restore that results in members dropped from the topology is considered a topology life change event and must have a full offline backup taken after the restore to guarantee recoverability.



Db2 10.5 came out with online add member and restore+rollforward support.
Db2 12.1 came out with offline drop member and restore+rollforward support.

Offline drop member has these restrictions:

The last member cannot be dropped.

The catalogued databases must satisfy the following criteria:

- Consistent database.
- Have at least one reconciled member remaining after the drop.
- HADR role standard.
- Not backup pending.
- Not topology life change pending or in progress.
- Not database rollforward pending.
- Not restore pending.

Restore/upgrade of downlevel images with topology changes
(Release 12.1.3)

- **Pre-12.1.3, following commands did not work on a downlevel database that was topology change pending**
  - CATALOG DATABASE
  - UPGRADE DATABASE
  - RESTORE DATABASE
  - db2ckupgrade
- **Examples:**
  - pureScale 3 members to pureScale 1 member (drop member pending)
  - pureScale 1 member to pureScale 3 members (add member pending)
  - pureScale any member to non-pureScale single partition (config type mismatch)
- **Expectation:**
  - UPGRADE or RESTORE+UPGRADE and use database with no issues
  - If explicit drop member pending or pureScale to non-pureScale involving a pre-12.1 database, post UPGRADE database will be BACKUP PENDING

29

Flexible Migration: UPGRADE DATABASE and RESTORE DATABASE handle downlevel images (11.1+) when target number of members differ.

Object sizes in db2adutl query
(Release 12.1.2)

- **Tool to help manage database recovery objects stored in IBM Storage Protect (formerly IBM Spectrum Protect / Tivoli Storage Manager (TSM))**
- **Backup images, load copy images and log archives now have size, in bytes, available**
  - Assist with recovery estimation
  - Inventory tracking
  - The new default for 12.1.2

```
  Query for database TESTDB

Retrieving FULL DATABASE BACKUP information.
    1 Time: 20250119181811, Size: 302096384 (bytes), Oldest log: S0000003.LOG, DB Partition Number: 0, Sessions: 1
    2 Time: 20250119174807, Size: 268533760 (bytes), Oldest log: S0000000.LOG, DB Partition Number: 0, Sessions: 1

...

Retrieving LOAD COPY information.
    1 Time: 20250119180412, Size: 278528 (bytes), DB Partition Number: 0, Sessions: 1

Retrieving LOG ARCHIVE information.
   Log file: S0000000.LOG, Size: 12288 (bytes), Chain Num: 0, Log stream: 0, Taken at: 2025-01-19-21.00.25
   Log file: S0000001.LOG, Size: 57344 (bytes), Chain Num: 0, Log stream: 0, Taken at: 2025-01-19-21.00.28
   Log file: S0000002.LOG, Size: 573440 (bytes), Chain Num: 0, Log stream: 0, Taken at: 2025-01-19-21.06.08
```

30

db2adutl query (12.1.2): Displays backup and log image sizes directly in output.

New columns added for BAR stats diag entry

(Release 12.1.0)

- **Three new fields by default:**
  - Throttle (added in 11.5.8)
  - LogPhase
  - LogBytes

```
FUNCTION: DB2 UDB, database utilities, sqluxLogDataStats, probe:2051
MESSAGE : Performance statistics
DATA #1 : String, 1864 bytes

Parallelism       = 10
Number of buffers = 20
Buffer size       = 16781312 (4097 4kB pages)

BM#    Total      I/O      MsgQ      WaitQ   Throttle  LogPhase     Buffers   kBytes   LogBytes
---  --------  --------  --------  --------  --------  --------    --------  --------  --------
000      0.59      0.35      0.00      0.23      0.00      0.00           4     16400         0
...
---  --------  --------  --------  --------  --------  --------    --------  --------  --------
TOT         -         -         -         -         -         -          15    128432         0
```

- **Open: How to account for time using remote storage media?**

31

Enhanced BAR Stats Output (12.1.0):
- Added LogPhase and LogBytes columns.
- Formatted BM elapsed times as human-readable HH:MM:SS (e.g. 25:47:53 instead of seconds).
- Emitted in both db2pd -barstats and db2diag.log.

New formatting for BAR stats diag entry
(Release 12.1.0)

- **Representing times as seconds makes it difficult to understand the scale of the time**

```
BM#    Total       I/O       Compr     MsgQ      WaitQ     Buffers   GBytes    GBytes
---  --------  --------  --------  --------  --------  --------  --------  --------
000  92873.78  43700.49  9262.30   11644.32  26728.88    406173      6334     11189
001  92862.32  26121.07  6196.32    9588.78  49743.70    274118      4271      6344
...
```

- **How long is 92000 seconds? These values are for human consumption so they should be human-readable**
- **We now represent the time in HH:MM:SS format**

```
BM#    Total       I/O.      Compr     MsgQ      WaitQ     Buffers   Gbytes    Gbytes
---  --------  --------  --------  --------  --------  --------  --------  --------
000  25:47:53  12:08:20  2:34:22   3:14:04   7:25:28     406173      6334     11189
001  25:47:42   7:15:21  1:43:16   2:39:48  13:49:03     274118      4271      6344
...
```

32

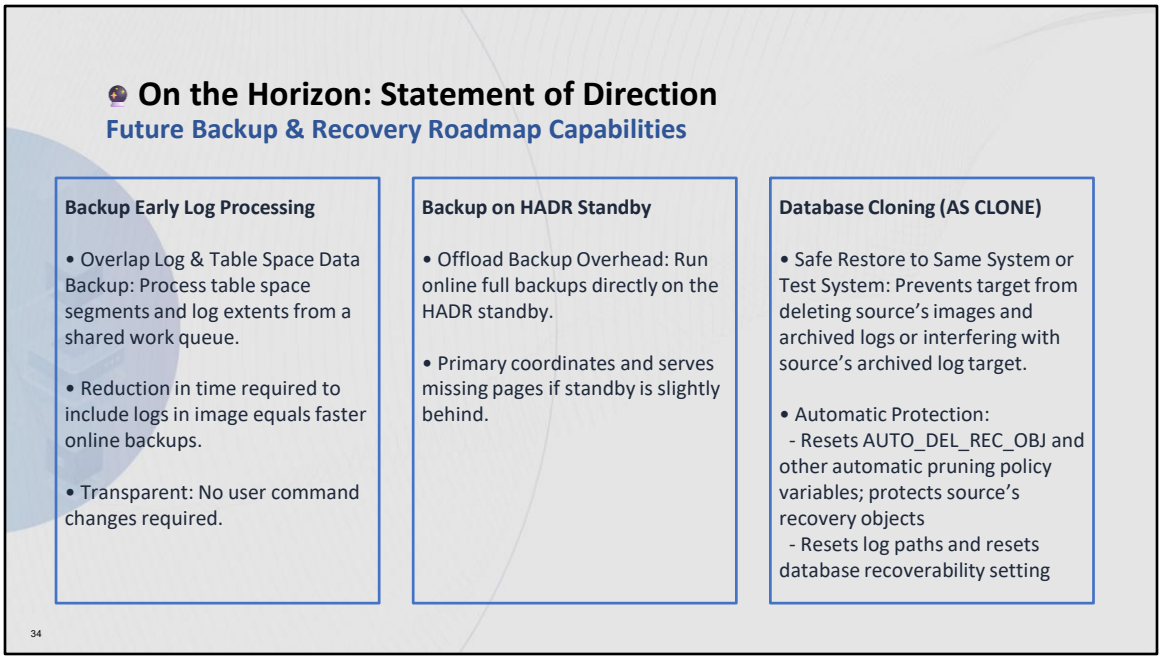
db2pd -barstats without an EDU ID

(Release 12.1.4)

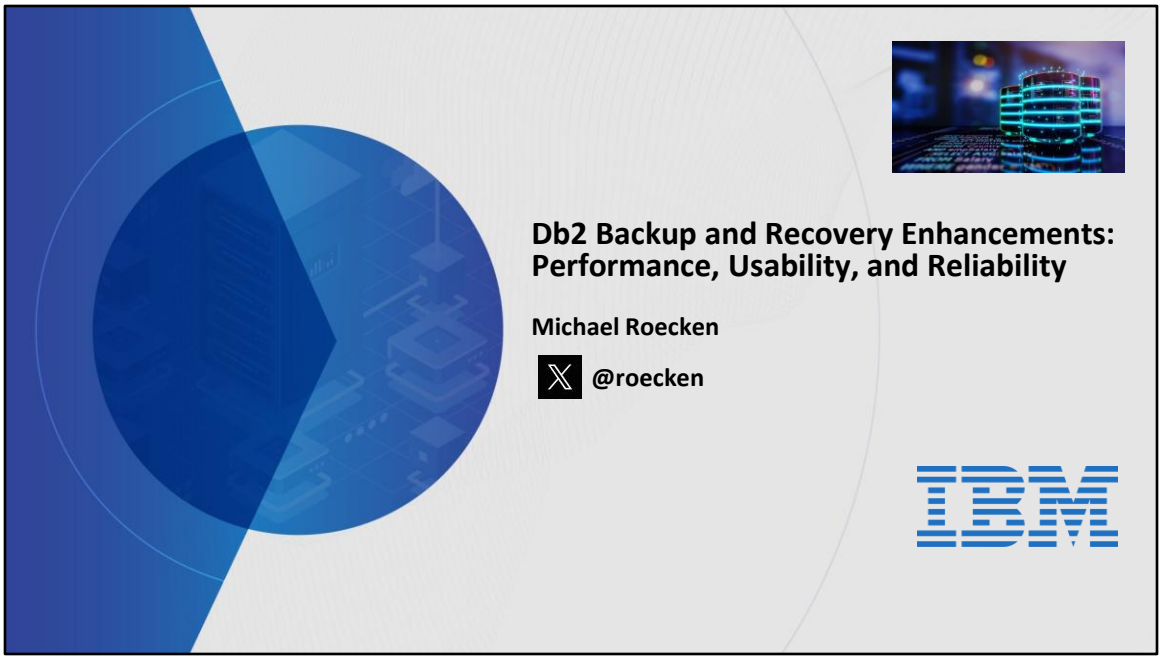
- **Automatically discovers active backup/restore operations for a database without requiring an explicit agent EDU ID:**
  - `db2pd -db sample -barstats`
- **When no EDU ID is supplied, Db2 scans the agent list and finds the backup or restore operation running on that database**
- **If more than one backup or restore is running on the same database at the same time, Db2 reports the ambiguity and prints all the active EDU IDs so you can supply the right one:**
  - `db2pd -db sample -barstats <agent_eduid>`

33

Simplified db2pd -barstats (12.1.4):
- Runs with just db2pd -db sample -barstats without requiring an agent EDU ID.



Looking ahead, Db2 is focusing on early log processing for faster INCLUDE LOGS backups, enabling online backups on HADR standby databases to offload production I/O, and the AS CLONE modifier for safe database duplication.



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, logging/transaction management and upgrade/migration.

Please connect with Michael on X (formerly known as Twitter) at @roecken and LinkedIn.