**IBM’s Power Private Cloud Rack for Db2 Warehouse Solution – Chapter 5: Backup & Restore**

A computer tower with a clock and graph

AI-generated content may be incorrect.

**By John Bell, Jana Wong, Peter Kokosielis**

1. **Introduction**

High-performance infrastructure is only as effective as its ability to recover from failure. In today’s analytics-driven environments—where data warehouses power both strategic insights and operational intelligence—backup and recovery processes must operate seamlessly alongside active workloads. A modern backup strategy should emphasize flexible, granular recovery capabilities that prioritize active data, reduce the need to continually back up inactive data and enable precise correction of failures with minimal operational impact.

This blog post explores **backup and restore strategies on IBM’s Power Private Cloud Rack (PCR),** which play a critical role in maintaining data integrity, minimizing downtime, and achieving defined recovery objectives. It outlines how to plan an effective recovery strategy, architect for resilience by establishing recovery time and recovery point objectives (RTO/RPO), enabling surgical restore capabilities, and performing live backups without disrupting ETL operations. By focusing on the most likely failure scenarios and backing up only active data, this approach supports scalable, efficient, and resilient recovery operations.

A real-world example—a 1.7 TB backup and multi-node restore on Power PCR—demonstrates the approach in action, complete with command references, performance metrics, and implementation best practices.

This is the fifth entry in the Power PCR blog series. For foundational context, refer to:

* [Introducing IBM’s Power10 Private Cloud Rack for Db2 Warehouse Solution: Architecture, Scalability, and Performance](https://community.ibm.com/community/user/blogs/jana-wong/2025/03/04/introducing-ibms-p10-pcr-for-db2wh)
* [Chapter 2: Db2 12.1 Performance Deep-Dive](https://community.ibm.com/community/user/blogs/jana-wong/2025/05/14/ibms-power10-private-cloud-rack-for-db2-warehouse)
* [Chapter 3: Step-By-Step Guide to Engage IBM Support](https://community.ibm.com/community/user/blogs/jana-wong/2025/05/22/ibms-power-10-private-cloud-rack-for-db2-warehouse)
* [Chapter 4: Enabling Q Replication](https://community.ibm.com/community/user/blogs/jana-wong/2025/05/28/ibms-power10-private-cloud-rack-for-db2-warehouse)

1. **Planning a recovery strategy**

In many environments, backup strategies tend to focus on operational elements such as scheduling, storage locations, archiving, and retention policies—often without fully addressing how recovery will be executed. However, it is the recovery strategy that should drive backup planning. This section explores how to design an effective recovery strategy and how early decisions—made during procurement, sizing, configuration, and data warehouse design—directly influence backup efficiency and recovery timelines.

Key considerations when planning a backup and recovery strategy include:

* **Recovery objectives**: Clearly define Recovery Point Objectives (RPO) and Recovery Time Objectives (RTO).
* **Physical data warehouse design**: Understand the relationship between tablespaces and tables to support targeted recovery.
* **Data ingest methods**: Consider how data is loaded (e.g., external tables, bulk insert techniques, or the LOAD utility) and how that impacts recovery.
* **Backup granularity**: Determine whether backups should be taken at the database or tablespace level.
* **Infrastructure capacity**: Account for the size of the database and the throughput capabilities of the underlying solution infrastructure.

Identifying the most likely failure scenarios and planning accordingly is critical to ensuring fast, targeted recovery with minimal disruption.

* 1. **Defining Your Recovery Objectives**

Establishing clear **Recovery Point Objectives (RPO)** and **Recovery Time Objectives (RTO)** is the first step to an effective backup and recovery strategy. In most enterprise environments, two sets of RPO and RTO are required: one for disaster recovery—typically measured in hours—and another for local recovery, which often targets a window of just a few minutes. These objectives should be defined early in the project lifecycle, ideally before hardware procurement, to ensure that infrastructure, database design, and replication strategies are aligned with business continuity requirements. During migrations, it is common to revisit and tighten existing RPO and RTO targets, making it essential to incorporate these considerations into planning and design phases. By locking in these service-level expectations up front, organizations can ensure that every architectural decision—from compute and storage specifications to network topology—supports the desired recovery performance.

|  |  |  |
| --- | --- | --- |
| Scope | RPO Target | RTO Target |
| Local Recovery | Minutes | Minutes |
| Disaster Recovery | Hours | Hours |

* 1. **Physical Data Warehouse Design**

The physical architecture of a data warehouse must be intentionally designed to support an effective backup and recovery strategy. For example, a data warehouse that places all tables and indexes in just a few table spaces would not be conducive to a best practice backup and recovery strategy due to the following factors:

* Poor separation of subject areas or schemas
* Increased backup duration
* Slower recovery times

Imagine instead of reloading an entire 100 TB database, one could identify the exact tablespace or partition that went corrupt, backup only active data, and recover just that slice in minutes. To make this possible and avoid the limitations above a well-structured tablespace-to-table relationship strategy that separates subject areas (sales, finance, logs, etc.) into distinct tablespaces should be implemented. Key tables should be mapped into those tablespaces so that, when a failure strikes, the problematic objects can be targeted. This not only supports performance and scalability but also enhances the flexibility and efficiency of backup and recovery operations. Benefits of this approach include:

* Physical data separation for improved performance
* Data isolation to support security and compliance
* Simplified disaster recovery population
* Optimized ingest performance
* Support for point-of-failure restore scenarios

From a backup and recovery perspective, this design enables parallel processing and targeted restores - both vertically (within a schema) and horizontally (across partitions) - which are essential to reduce downtime and ensure continuity. Db2 on IBM’s Power PCR runs one backup stream per MLN - parallelizing the process into 120 surgical jobs across 120 database partitions on a Power PCR BRL+ERL system.

* 1. **Data Warehouse Ingest method**

An effective backup and recovery strategy should prioritize restore readiness over backup execution - ensuring that backups can run quietly in the background without disrupting business-critical operations. The method used to ingest data into the warehouse plays a significant role in achieving this objective. The best practice for bringing external data into Db2 data warehousing tables is the INSERT from EXTERNAL TABLE operation which offers the best performance and flexibility with other concurrent database operations. The LOAD utility is an additional option for data ingest in environments where continuous backup operations (using the backup utility) are not being performed. Choosing the right ingest method not only supports uninterrupted backup operations but also benefits broader aspects of data warehousing.

* 1. **Backup granularity**

In IBM’s P10 PCR Db2 Warehouse environments, three primary backup methods are available: **full-database backups**, **tablespace-level backups**, and **schema-level (logical) exports**. Each method offers distinct advantages and trade-offs in terms of simplicity, precision, and performance.

|  |  |  |  |
| --- | --- | --- | --- |
| ****Backup Method**** | ****Description**** | ****Advantages**** | ****Considerations**** |
| ****Database-Level Backup**** | Full physical backup of the entire database using a single BACKUP DATABASE <dbname>  command. | * Simple to configure and execute * Supports full and selective TBSP restores | * Longer restore times * Over-reads during partial restores * Rarely used for local recovery due to low probability of needing full database restore |
| ****Tablespace-Level Backup**** | Physical backup of individual tablespaces, requiring multiple commands. | * Enables highly targeted, fast restores * Offers flexibility * Restore is pointed: Reads only necessary backup images | * Requires individual backup commands (automation can help) per TBSP – potentially hundreds * Reduced Redo Logging needs to be disabled for columnar workloads. |
| ****Schema-Level (Logical) Backup**** | Logical export of all tables within a schema using external tables and compression. | * Allows selective table or schema restores * Useful for preserving critical reference or lookup tables | * Slower than physical backups * May have compatibility limitations |

In practice, a hybrid approach often delivers the best results—combining weekly full-database backups, daily incremental or tablespace-level backups, and periodic schema exports. This strategy balances simplicity, precision, and performance.

* 1. **Infrastructure Capacity**

The success of any backup and recovery strategy is closely tied to the performance of the underlying infrastructure—particularly storage and network throughput. Backup and restore operations rely on reading from and writing to storage systems external to the database itself. As such, the size of the database and the throughput capacity of the surrounding infrastructure—such as storage bandwidth, network speed, and I/O performance—directly impact the ability to meet defined RPO and RTO targets. IBM’s Power PCR leverages the IBM **Scale Storage Solution 6000** (SSS6000), a high-throughput, low-latency storage platform that enables continuous online backups without impacting active workloads. To meet disaster recovery SLAs, backup images should be replicated off-site, while archive logs must be written directly to an external SSS6000 by configuring LOGARCHMETH1 appropriately. Without this level of optimization, even well-designed backup strategies can degrade into slow, full-database restores, undermining the benefits of granular, point-of-failure recovery.

**3. Data Solution Professionals Best Practices on Power PCR**

Building on the foundational concepts of recovery objectives, physical design, and ingest methods, this section outlines recommended backup and restore practices for Data Solution Professionals (DSPs) working with Db2 Warehouse on IBM’s Power PCR. These recommendations are aligned with a Point-of-Failure Restore Strategy, which emphasizes targeted, efficient recovery over broad, full-database restores. While final decisions will be made by DSPs in collaboration with IBM, the following guidance assumes the use of database-level backups for consistency. If tablespace-level backups are preferred, the same principles apply with appropriate substitutions. To ensure robust protection against local database and data corruption on the primary Db2 database, it is recommended to implement the following best practices:

1. **Local Backup Storage**  
   Store backup image files locally within the same data center on the **Scale Storage System 6000 (SSS6000)** using the **ESS file system**. This ensures fast access and recovery in the event of local failures.

In addition, for each **MLN (Member Logical Node)**, configure the **database transaction logs** to be archived to a path on the **External Storage** by setting the database configuration parameter LOGARCHMETH1accordingly.

**Current Setting Example:**  
On IIAS, LOGARCHMETH1 is currently set to:  
DISK:/mnt/db2archive/archive\_log/  
This should be updated to a valid path on the External Storage.

1. **External Backup Storage**

To ensure resilience and recoverability, backup image files should be stored externally. This approach not only safeguards data against local failures but also supports enterprise-grade backup and restore strategies. Two common methods for external storage are:

1. **Using File Management Software**

IBM Power PCR and all Db2 Warehouse deployments on Kubernetes are designed to integrate seamlessly with leading enterprise backup solutions. While not mandatory, it is **strongly recommended** to use a **file management solution** to streamline the handling of backup image files. Supported options include:

* **IBM Storage Protect** (formerly Spectrum Protect and TSM)
* **Veritas NetBackup**
* **EMC NetWorker**

These tools manage both the backup image files and the underlying storage infrastructure, offering robust scheduling, retention, and recovery capabilities. Integration with these platforms ensures that backups are consistent, secure, and aligned with enterprise data governance policies.

1. **Attaching an External Cluster File System**

For organizations preferring direct external storage integration, the Power PCR supports two primary methods:

#### **a. NFS-Attached External Storage**

Network File System (NFS) is the simplest and most commonly used method. Since the Power PCR is already connected to the customer’s corporate network, NFS storage can be made available with minimal configuration.

**Steps to configure NFS storage:**

* **Provision NFS Storage** on the corporate network.
* **Create an OpenShift StorageClass**:

kind: StorageClass

metadata:

  name: nfs-external1-rwx-sc

parameters:

  archiveOnDelete: "false"

  provisioner: k8s-sigs.io/nfs-subdir-external-provisioner

reclaimPolicy: Delete

volumeBindingMode: Immediate

* **Add the StorageClass to the Db2u Custom Resource**:

- name: external1

  spec:

    accessModes:

      - ReadWriteMany

    resources:

      requests:

        storage: <storage size>Gi

    storageClassName: nfs-external1-rwx-sc

  type: create

* **Scale Down and Scale Up Db2uInstance** to apply changes:
  + 1. Set replicas to 0, wait for pods to terminate.
    2. Restore replicas to original count (e.g., 2), wait for pods to reach 1/1 ready state.
* **Verify Mount Point:**

oc rsh <db2u-pod> bash -l

df -h

Look for /mnt/external/path1.

* **Run Backup Command**:

BACKUP DATABASE BLUDB TO /mnt/external/path1

#### **b. External SAN-Attached Storage**

For performance-intensive environments, SAN-attached storage offers significantly higher throughput—often 5 to 10 times faster than NFS. This method requires customer-supplied infrastructure:

* Dual SAN switches for high availability
* Storage controller and media
* Cabling from controller to switches and switches to each worker node

Once the SAN is powered, configured, and connected, the setup process mirrors the NFS method starting from the creation of the OpenShift StorageClass.

1. **Backup Frequency**

To ensure data integrity and minimize recovery time, implement a structured backup schedule combining **weekly full backups** with **daily incremental backups**:

**Weekly Full Online Database Backup**  
Perform a full online backup of each Db2 database once per week. Db2 will back up tablespace by tablespace, with the ability to parallelize the process across multiple tablespaces and within tablespaces themselves.  
In environments like Power PCR BRL+ERL, which contain 120 MLNs (Db2 engines), each MLN will be backed up in parallel, generating at least one backup image file per MLN.

**Daily Incremental Online Database Backup**  
Between weekly full backups, schedule daily incremental backups. These backups capture only the changes since the last full or incremental backup, reducing backup time and storage usage.  
Like full backups, incremental backups are performed tablespace by tablespace and support parallel processing across MLNs.

1. **Disaster Recovery Replication**  
   Replicate these backup image files to a **secondary SSS6000** located in the **Disaster Recovery (DR) data center**.
2. **Local Read/Write Operations**  
   Configure the Db2 database to **always read from and write to the local SSS6000** for backup and recovery operations. This minimizes latency and dependency on remote systems during routine operations.
3. **Fallback Option**  
   If no file management software is used, direct read/write access to the **attached SSS6000 (referred to as External Storage)** is still supported.

**4. Real-World Example Walk-Through**

To evaluate the performance of backup and restore operations on IBM’s P10 PCR system, we deployed an **100TB** internal data warehouse workload called **Big Data Insights (BDI)** on a Base Rack Large Cloud Rack environment. The 100TB refers to the uncompressed Flatfile data size that was used to populate the tables. The database occupied **30.9TB size on disk**. This setup provided a realistic scenario to observe how the system handles large-scale, parallelized Db2 backups an restores across multiple MLNs.

**4.1 Schema Level Backup**  
The schema level backup is done via the following stored procedure call and was tested with both a 10TB BDI schema and 100TB BDI schema to determine if backup time and resulting backup image size scale linearly. The backup image is stored on the local disk.

db2 -v "call sysproc.logical\_backup('-type full -schema ${schema} -path /mnt/backup/DBBackup/${schema}')"

**Results:**

The following graph shows that the schema level backup time and image size scale linearly from 9min 23 sec for the 10TB setup to **1h 30min and 30.6TB for the 100TB Setup**.

**A graph of a schema

AI-generated content may be incorrect.**

**4.2 Full Online Database Backup**

In a similar fashion to the schema level backup we performed a **full online database backup** with only the 100TB BDI schema still in place on the database resulting in the following overall database size:

BLUDB = 30905369 MB [30.9 TB]

SUM(SCHEMA) = 30903301 MB

OTHER METADATA/CACHE/CONFIG FILES = 2068 MB

**Backup command:**

db2 "BACKUP DATABASE BLUDB ON ALL DBPARTITIONNUMS ONLINE TO ${BackupDir} INCLUDE LOGS WITHOUT PROMPTING"

The backup progress can be monitored using:

db2 list utilities show detail

**Results:**

**Backup Time:** 1:02:13h (32% faster than equivalent schema-level backup)  
**Backup Location:** /mnt/backup/DBBackup/FullOnlineDBBackup100TB  
**Backup Size:** 30TB

**4.3 Full Restore Procedure**

1. **Step 1: Prepare the Database for the Restore Process**  
   On each node, we disabled high-availability, terminated active connections, and restarted the database in restricted mode to ensure that only authorized administrative operations—such as restore—can be performed, preventing access by other users or applications during the process.

# Disable Stayalive Probe. This probe checks every 15 minutes to see if wolverine and / or Db2 is running. If down the probe triggers a pod restart after 15 minutes (something we want to avoid during restore).  
rah 'touch /db2u/tmp/.pause\_probe'

sudo wvcli system disable -m 'Disable HA before Db2 maintenance'

wait\_interval

wvcli system ds

db2 terminate

db2 force application all

db2 deactivate database bludb

db2stop

ipclean -a

db2set -null DB2COMM

db2start admin mode restricted access

**Duration: < 2 minutes**

1. **Step 2:** **Restore the Database on the Catalog Node (MLN 0)**

db2\_all '<<+0<db2 RESTORE DATABASE BLUDB \

FROM /mnt/backup/DBBackup/FullOnlineDBBackup100TB \

TAKEN AT 20250725184659 \

INTO BLUDB LOGTARGET /mnt/backup/logs \

REPLACE EXISTING WITHOUT PROMPTING'

The restore process can be monitored via:

[db2inst1@c-db2u-cr-db2u-0]$ db2 list utilities show detail

ID = 1

Type = RESTORE

Database Name = BLUDB

Member Number = 0

Description = db

Start Time = 07/28/2025 18:30:21.354600

State = Executing

Invocation Type = User

Progress Monitoring:

Completed Work = 3686903808 bytes

Start Time = 07/28/2025 18:30:21.354606

**Duration (catalog):** ~ 30 minutes

1. **Step 3: Restore the Database on All Other Data MLNs (1..47) in parallel**

db2\_all '<<-0<||db2 RESTORE DATABASE BLUDB \

FROM /mnt/backup/DBBackup/FullOnlineDBBackup100TB \

TAKEN AT 20250725184659 \

INTO BLUDB LOGTARGET /mnt/backup/logs \

REPLACE EXISTING WITHOUT PROMPTING'

**Duration:** ~1 hour 18 minutes

1. **Step 4: Roll forward to End of Backup**

db2 -v "ROLLFORWARD DATABASE BLUDB TO END OF BACKUP AND COMPLETE"

**Duration:** 10 seconds

1. **Step 5: Restart database and resume Normal Operations**

db2stop force

ipclean -a

db2set DB2COMM=TCPIP,SSL

db2start

db2 activate database <DBNAME>  
  
#activate HA

sudo wvcli system enable -m "Enable HA after Db2 maintenance"  
  
# Enable Stayalive Probe Trigger.

rah 'rm /db2u/tmp/.pause\_probe'

#Connect to database

db2 connect to bludb

**4.4 System Utilization during Backup & Restore**

The following NMON Visualization graphs show the system utilization on the Power PCR BRL system during the full online database backup and subsequent restore operation. The system is healthy with disk utilization averaging around 50-60% and very low CPU utilization of under 5%.

**A close-up of a graph

AI-generated content may be incorrect.**

**5.Conclusion**

Effective backup and restore strategies are essential for maintaining data resilience in modern analytics environments. On IBM’s Power Private Cloud Rack (PCR), these strategies must be designed not only for performance but also for precision—enabling fast, targeted recovery with minimal disruption to ongoing operations. By aligning backup planning with recovery objectives, leveraging local and replicated storage, and implementing structured backup schedules, Data Solution Professionals can ensure that critical data remains protected and recoverable. The real-world example presented in this post demonstrates how these best practices translate into measurable performance and operational confidence. As data volumes and complexity continue to grow, adopting a recovery-first mindset will be key to sustaining availability, integrity, and trust in enterprise data platforms.

**About the Authors**

**John Bell** is a Distinguished Engineer and Data Warehouse Architect at IBM, with over 25 years of experience in data warehousing and analytics. He has played a pivotal role in developing IBM's data warehouse solutions, including IBM’s Power10 Private Cloud Rack reference architecture. He can be reached at john.bell@ibm.com.

**Jana Wong** is the principal performance focal for Data Warehouse on-premise solutions at the IBM Silicon Valley Lab, with over 15 years of experience in Databases, SQL, QA, and Project Management. She holds a Master’s in Computer Science from the University of Rostock. Recently, she led the development and automation of a benchmark kit for validating IBM's Power10 Private Cloud Rack and played a key role in evaluating the performance of reference architectures such as IIAS/Sailfish and P10 PCR. Jana can be reached at [jfitzge@us.ibm.com](mailto:jfitzge@us.ibm.com).

**Peter Kokosielis** is the manager of Db2 Performance Quality Assurance, Db2 Warehouse on Power Private Cloud Rack QA and Big Data and Data Virtualization QA. He has extensive experience at IBM in Db2 LUW database performance both in OLTP and Data Warehouse settings along with deep experience in platform exploitation on Power and Intel based processing architectures, hardware accelerators, virtualization and operating systems.