Db2 Technical Migration Procedure – Power to PowerVS
Version 1.0

AIX version

August 5, 2024

Theresa Xu
Michael Wind
Samuel Matzek

Table of Contents

Document Information

Version Control

Release History	Date	Comments
1.0	August 5, 2024	Version 1 general release

Target Audience and Intent

This document presents four migration options, leveraging Db2 backup and restore procedures and Db2 HADR database synchronization to accomplish the migration of Db2 to Power Virtual Server from Power environments.

The target audience consists of solution and infrastructure architects and Db2 database administrators.

Options presented vary depending on Customer's tolerance for service disruption and desired end state configuration.

Scope and Coverage

Presented procedures assume availability of sufficient network connectivity to support data transfer and/or data replication from the source system to Power Virtual Server target systems.

- Discussion of alternative procedures using the Seagate Lyve Mobile Solution to physically transfer On-Premises backups/database files to Power Virtual Server infrastructure, along with links to execute the transfer procedures, is presented in Option 1.
- Links to detailed information about Seagate Lyve may be found in the document appendix.

Task steps vary per implementation. Refer to cited Db2 documentation within the document for details regarding the execution of specific commands.

Approach

- In order to minimize repetition, steps for the first option are treated as a baseline and are fully detailed. Options 2, 3, and 4 are treated as variations of Option 1, and only differentiating steps are covered in those sections.
- The term "PowerVS" is frequently used in this document as an abbreviation for "Power Virtual Server".

Disclaimer

Any attempt to execute these procedures will be performed in context with Customer's established procedures for operating and maintaining non-production and/or production systems. Customer will take customary actions to ensure system availability for maintenance and/or reconfiguration as required, and schedule downtime as required. Customer is responsible for reviewing these representative procedures in the context of their particular environment and adjusting as required.

The Db2 Database migration options described are not necessarily specific to PowerVS migrations. Db2 technical staff should recognize the procedures used and understand that not every technical detail or consideration have been explicitly identified. The person(s) executing the procedures are expected to understand the full scope of Db2 database backup and recovery methods – including those details not explicitly stated.

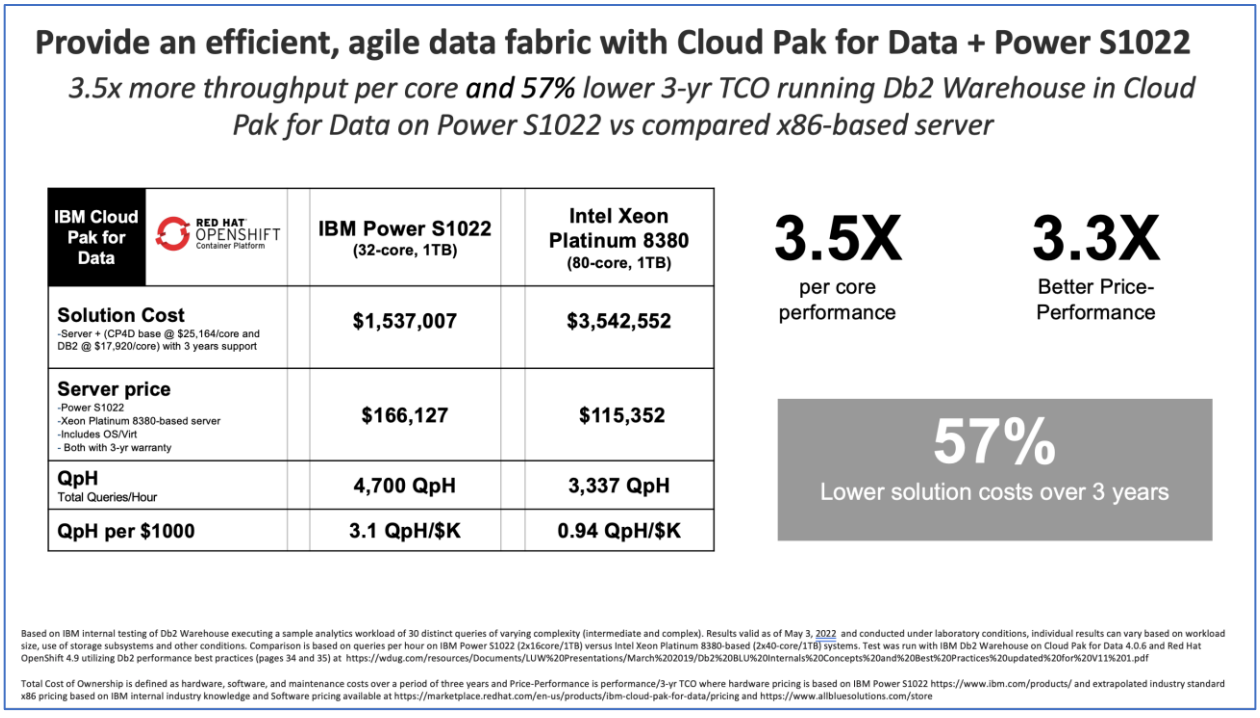
Why Migrate Db2 on AIX to Power Virtual Server

Power Virtual Server (aka “PowerVS”) provides an excellent infrastructure and environment to relocate your Power on AIX assets to high performance virtual computing infrastructure.

To learn more about the PowerVS infrastructure and capabilities, please review “Getting Started with IBM Power Virtual Servers” documentation via the following URL:
https://cloud.ibm.com/docs/power-iaas?topic=power-iaas-getting-started

As described below in the graphic below, Power Virtual Server can offer significant price performance advantages over comparable x86-based platforms.

- Power9-based Power Virtual Server processors, available now, offer competitive price performance.
  - Since Db2 LUW licensing is per core based, the per core performance advantage of PowerVS over x86-based compute provides significant TCO advantages.
- Power10 processors (becoming generally available in more datacenters) offer > 2x per core performance advantage for both Db2WH (BIDAY) and Db2 OLTP (TPC-E) over Power9.
- In addition, modern AIX on PowerVS infrastructure can provide significant operational benefits.
  - An example: Aspera client/server infrastructure can be installed to facilitate extremely fast data transfers across networks.



The future of Db2 is bright, and performance, modernization and TCO benefits can be realized by migrating Db2 on AIX from Power to Power Virtual Server.

Migration Concepts

Requirements Discovery

This document assumes that a detailed discovery has previously collected for:

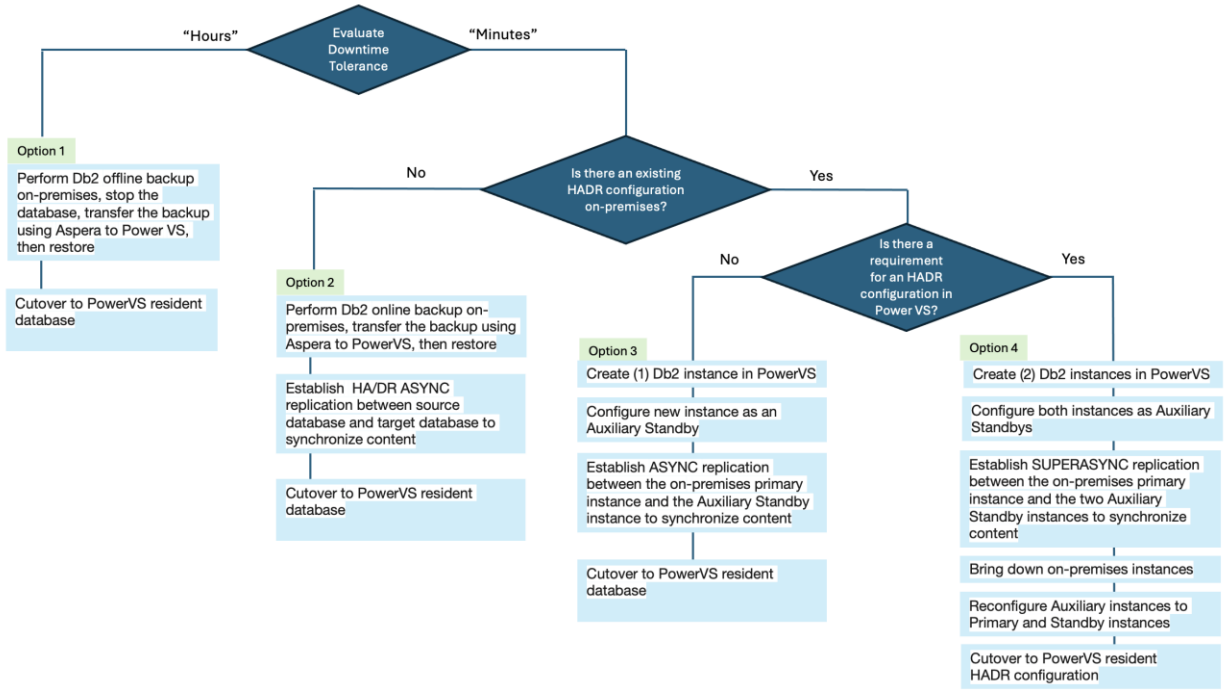
- business, technical, functional, non-functional requirements of the Db2 database being migrated (including access and availability requirements)
- installation configuration, HW/SW inventory and sizing information related to the source database and the underlying platform.
    - Collect relevant metrics related to sizing, change rate, service level requirements, tolerance for migration downtime, capacity of networks to support data transfer from source to target, etc.
- network connectivity and throughput available/feasible between source and target
- production and operations considerations, including clear understanding of service level requirements, tolerance for application and database downtime, etc.

Based on the discovered information, configuration and sizing requirements for the target Db2 instance in the PowerVS environment should be identified before selecting a migration option.

Migration Options Covered By this Document

The following flowchart illustrates three options to covered by this document. The first two options facilitate database migrations from on-premises to Power Virtual Server.

- Option 1: Database backup and restore from AIX on Power to a PowerVS instance
- Option 2: Database backup and restore from AIX on Power to a PowerVS instance followed by temporary HADR enablement to synchronize source and target databases via HADR ASYNC mode. During cutover, perform database takeover on the PowerVS standby instance to make it as the new primary. Once the takeover completes, stop HADR on the PowerVS instance and remove the on-premises instance from the HADR cofiguration of the PowerVS instance. The PowerVS instance will be the only Db2 instance left after the migration (i.e. no HADR setup in PowerVS).



- Option 3: Use Power Virtual Server to establish an auxiliary standby instance via HADR SUPERASYNC mode to a preexisting Db2 HADR configuration that exists on-premises.During cutover, perform database takeover on the PowerVS auxiliary standby instance to make it as the new primary. Once the takeover completes, stop HADR on the PowerVS instance and remove the on-premises instances from the HADR cofiguration of the PowerVS instance. The PowerVS instance will be the only Db2 instance left after the migration (i.e. no HADR setup in PowerVS).
- Option 4: Use Power Virtual Server to establish (2) co-resident auxiliary standby instances via HADR SUPERASYNC mode to a preexisting Db2 HADR configuration that exists on-premises. During cutover, perform database takeover on one of the PowerVS auxiliary standby instances to make it as the new primary. Once the takeover completes, remove the on-premises instances from the HADR configuration of the PowerVS instances so the two PowerVS Db2 instances will be the only two instances (the primary and the principal standby) running in the HADR setup after the migration.

Selection Considerations

Customer Service Level Requirements and technical conditions influence the selection decision from the presented options.

1) Service Level Requirements, particularly when it comes to Restore Point Objectives and Restore Time Objectives.
    a. Option 1 assumes that the Customer is comfortable with a considerable service delay (RTO = "Days" or "Hours") before the source database is migrated to the target database. This expectation is often the case for non-production workloads but rarely for production databases.

b. **Options 2, 3, and 4** assume that the Customer is comfortable with a service interruption in terms of "Minutes". This expectation is often the case for production databases.

2) Database and AIX operating system version
    a. The migration procedures presented are applicable to the following versions:

Software	Versions
AIX	7.2, 7.3
Db2	11.0, 11.5

3) Data Characteristics:
    a. Integrity of the source database, use of standard or specialized character sets, amount/type of indexing applied.
4) Database Size, Network Throughput and Reliability
    a. Customer may have limited interim storage to handle backups.
    b. Transporting backups of a large database across a network will take time. Consider the following transfer example of 600 GB of backups across an end-to-end network connection where available protocols and bandwidth support 80 megabytes/second throughput. At this rate, it will take 125 minutes to transfer the data. If the database backups are much bigger or if the network connection throughput is less, more transfer time will be required. Unreliable networks may disrupt the transfer, requiring you to restart the transfer process, incurring delay.
        i. Customers can benefit from specialized transfer tools that compress data before/during transfer
    c. IBM's Aspera is a transfer tool that can significantly reduce data transfer times. More information can be found in the appendix of this document.
5) Availability of Skills to Execute.
    a. Options provided in this document require experienced DBA skills and the ability to work with infrastructure teams to migrate database content to a new target database on Power Virtual Server within a PowerVS workspace.
6) Downtime Tolerance
    a. Customer Service Level Requirements will dictate selection and execution of database migration techniques.

Any migration procedure a Customer elects to use should be executed in the context of a detailed, well-rehearsed transfer and cutover plan.

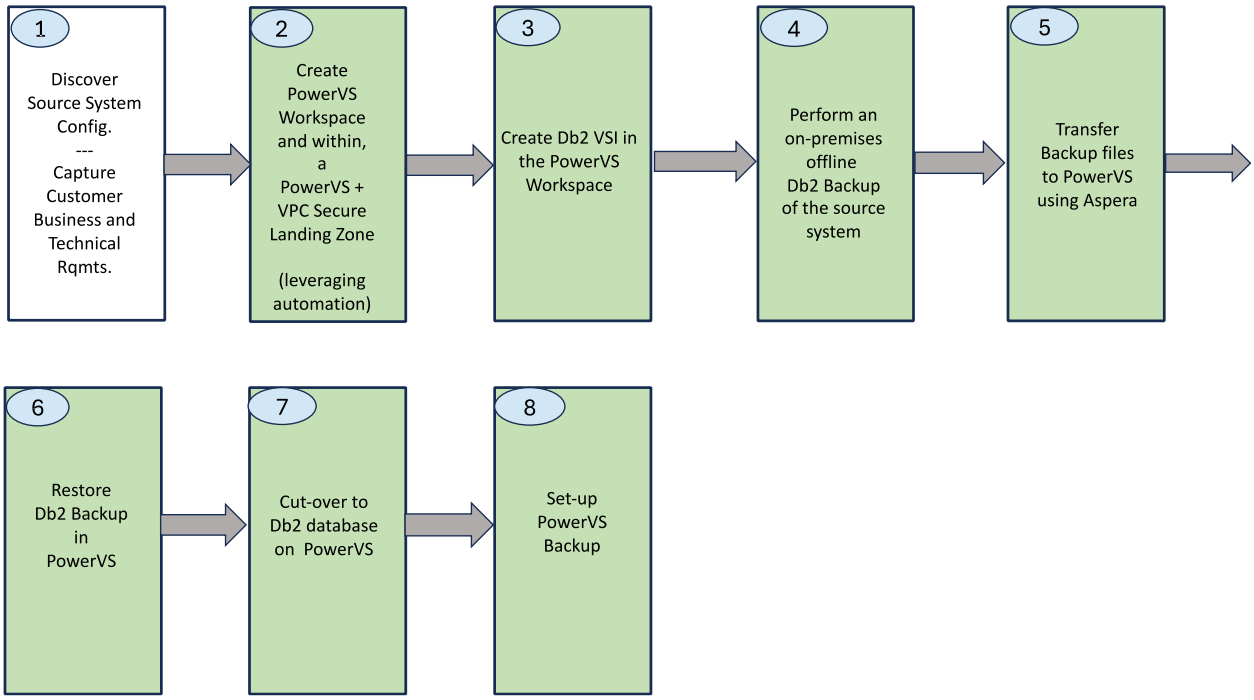
Migration Options

Option 1: Database Backup and Restore

This option presents a procedure that uses **db2 backup** to take a point in time backup of a Db2 database resident on Power, transfers the backup file contents to PowerVS local file storage, then applies **db2 restore** to restore the database contents to the PowerVS-resident Db2 database.

Step Sequence

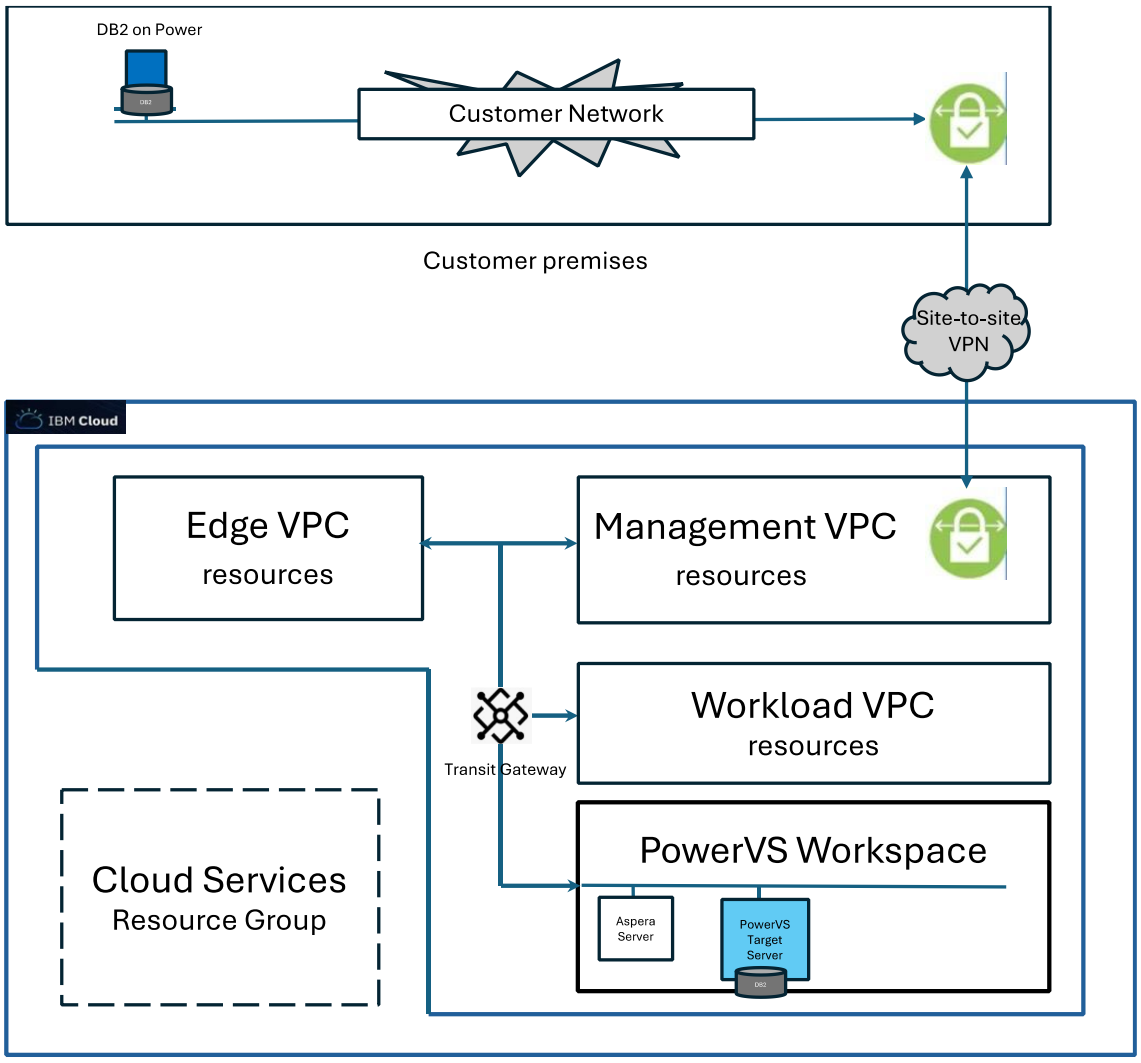
The following illustrates Option 1 activity flow:



For Option 1, we will assume discovery and requirements capture has been completed, and cover activities required in steps 2-8.

Migration Scenario Diagram

Sample migration scenario is illustrated below.



The IBM Cloud environment as depicted above is a very high-level representation of what can be deployed using automation. It does not depict the various servers and subnets deployed in various deployment areas.

Here a Site-to-Site VPN is used to connect the on-premises environment to IBM Cloud. Customer have other connectivity options to deploy dedicated, high performance bandwidth. IBM DirectLink is typically provisioned.

An Aspera server has been deployed in the PowerVS environment to facilitate high speed transfers of backups across the network.

Creating the Migration Target Environment

The target migration environment consists of a Power Virtual Server workspace, the target DB2 server and associated storage, and the networking to allow communication between the source and target DB2 servers.

Various network architectures can be used to provide private communication between the two DB2 servers. The Power Virtual Servers documentation highlights some of the network architectures that can be used and documents VPN connectivity options. The choice of network architecture is dependent on the organization's requirements for considerations such as firewalls, encryption, network bandwidth, network latency, and routing.

Automation can be used to simplify and accelerate the creation of the Power Virtual server workspace, DB2 server, and associated networking.

- The Power Virtual Server with VPC landing zone deployable architecture , located in the IBM Cloud Catalog, uses a menu-driven approach to creating a Power Virtual Server workspace that can support database and application server deployments on PowerVS for various t-shirt sizes, with capability to deploy a fully integrated PowerVS + VPC integrated layout, compliant with deployment best practices, with options to select stock operating system images.

- The Cloud Resource and Infrastructure-as-Code Generator (CRAIG) provides multiple, fully customizable templates such as the Power VS Proof of Concept template. These templates can be customized with AIX servers and the exact CPU, memory, and disk requirements needed for DB2 along with the associated cloud networking to connect the migration target server to the on-premises source server. CRAIG is a terraform generation tool that provides a granular approach to sizing and defining deployments.

The operating system of the target DB2 server can be a deployed from a stock image in the Power Virtual Server image catalog, a generated custom image of the source server, or restored from a mksysb backup of the source server. Custom images can be generated using IBM PowerVC or the create_ova AIX command. See the Power Virtual Servers documentation for more information on deploying custom images and restoring mksysb images.

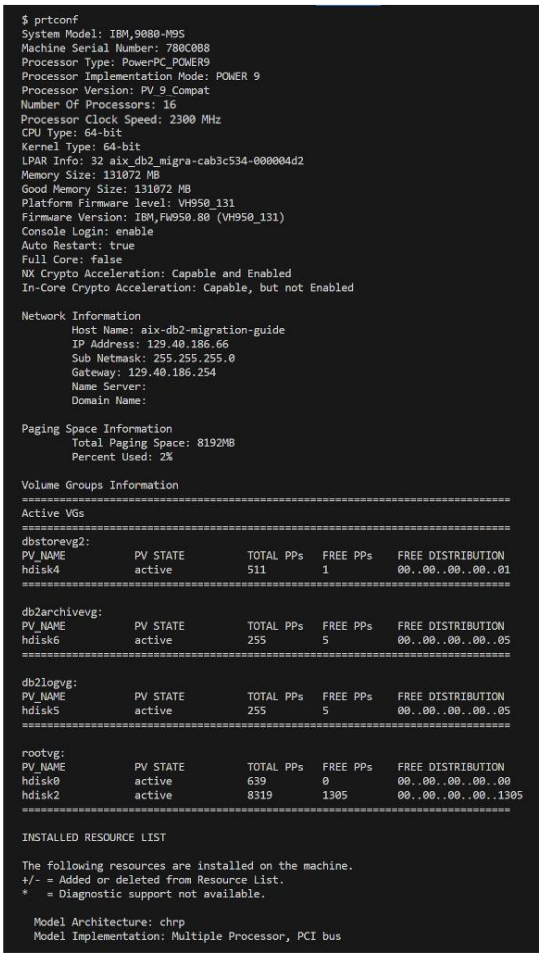
Sizing and Configuring the Target System

A Db2 specialist should work with a Power systems specialist to develop a sizing and configuration specification for the target Power Virtual Server and DB2 instance.

A full treatment of this subject is outside the scope of this document. The following summarizes considerations that should be taken into account:

1. Source Power system characteristics:
    a. LPAR specification:
        i. Computer system model
        ii. Number of allocated cores, memory, storage, storage configuration
        iii. Versions of software, software agents in use
            1. Operating system (and associated customizations)
            2. Backup
            3. Monitoring
            4. Integration interfaces
        iv. Utilization statistics (standard load, peak capacity)
    b. Database specifications
        i. DB version/patch levels
        ii. Storage capacity and configuration; storage in use, data growth rates
        iii. Licensing model
2. Available target system capacities
    a. Power Virtual Server specification:
        i. Currently based on Power9 or Power10 systems, depending on selection, will most likely provide equivalent or superior per-core performance to the source system.
        ii. Size the PowerVS instance using rperf core ratio for the server type
            1. Very important for a Power systems specialist to work with the Database SME to specify the right number of cores to meet database requirements.
        iii. Versions of software, software agents to be used
            1. Can a stock operating system image be selected, or must the operating system image be copied over from the source system?
            2. What are backup requirements for the target system? Can standard IBM Cloud Catalog selections for backup software be applied, or will customer provide their own solution?
            3. Who will monitor the resulting LPAR and how will that be done? Will standard or custom agents be installed at the OS level?
            4. What integrations will the resulting Power Virtual Server require?
        iv. Expected utilization for the target system
    b. Database Specifications
        i. How well do target DB versions and patch levels match the source? If there are different, what adjustments are required before/during/after migration of data?
        ii. How well does capacity and configuration meet capacity and performance requirements?
        iii. What licensing will support establishment of Db2 on the target environment?

The following is a comparative example of source and target server configurations, and how they can be reviewed using **prtconf**:

	On-premises Db2 Primary	New PowerVS Db2 Instance
Compute		
Network		
Storage		

Ensuring that the firmware levels of the target system are as current or newer than the source system is a good precautionary step.

When it is time to perform an installation of the target Db2 database, Customer should have secured installation media and licensing to support installation of Db2 on the Db2 instance.

Perform an On-Premises Db2 Backup to Local Storage

For this option an offline Db2 database backup image should be taken.

The following are representative steps to back up a source database called ECOM:

1. Review the database configuration.

```
$ db2 get db cfg for ecom
```

2. Confirm the source host name.

```
$ hostname
aix-db2-migration-guide
```

3. Make directories on source system to receive backups.

```
$ mkdir - p /db2backups/db2inst1/ecom_backup_offline/dir1
$ mkdir - p /db2backups/db2inst1/ecom_backup_offline/dir2
```

4. Backup the source database.

```
$ time db2 "backup database ECOM on all dbpartitionnums online to
/db2backups/db2inst1/ecom_backup_offline/dir1,/db2backups/db2inst1/ecom_backup_offline/dir2
COMPRESS COMPRLIB libdb2compr.a"
Part Result
---- --------------------------------------------------------------------
0000 DB20000I  The BACKUP DATABASE command completed successfully.

Backup successful.  The time stamp for this backup image is 20240320122901

real   14m39.70s
user   0m0.01s
sys.   0m0.00s
```

5. Review contents and note size of backup directories.

```
$ ls -lart /db2backups/db2inst1/ecom_backup_offline/*
/db2backups/db2inst1/ecom_backup_offline/dir1:
total 17830176
drwxr-xr-x   4 db2inst1 db2iadm1            256
drwxr-xr-x   2 db2inst1 db2iadm1            256
-rw-------   1 db2inst1 db2iadm1    91290337824 ECOM.0.db2inst1.DBPART000.20240320122901.001

/db2backups/db2inst1/ecom_backup_offline/dir2:
total 19534584
drwxr-xr-x   4 db2inst1 db2iadm1            256
drwxr-xr-x   2 db2inst1 db2iadm1            256
-rw-------   1 db2inst1 db2iadm1    10010694720 ECOM.0.db2inst1.DBPART000.20240320122901.002
```

6. Copy files from source to target.

   We recommend using Aspera to transfer the files from source to target. This involves
   - provisioning an Aspera server in the Power Virtual Server workspace and also configuring it as an NFS server for migrated contents and sharing contents to the target Db2 database
   - installing the Aspera client on the source system.

Note: The following .pdf provides a more detailed description of Aspera setup: https://cloud.ibm.com/media/docs/downloads/power-iaas/accelerated_migration.pdf

A sample command sequence is shown below:

From source system, transmit files to the allocated Aspera server. The following assumptions have been made:

- Aspera client has been installed on the source system.
- Aspera server has been installed on a server in the Power Virtual Server workspace (here assumed located at 192.168.102.179) and directory contents of the Aspera server are NFS accessible by the target Power Virtual Server.

```
$ time ascp -r -p /db2backups/db2inst1/ root@192.168.102.170: /db2backups/db2inst1
```

7. Access the allocated Aspera server as root to review what was transmitted.

```
# hostname
hmcs-aspera-server
[root@hmcs-aspera-server ~]# ls -lart /db2backups/db2inst1
total 0
drwxr-xr-xrs.  3 root     root     22 Mar 20 15:03
drwxr-xr-xrs.  2 nobody   nobody  256 Mar 20 15:04
[root@hmcs-aspera-server ~]# df -k /db2backups/db2inst1
Filesystem                          1K-blocks     Used Available Use% Mounted on
192.168.102.36:/db2backups/db2inst1 157286400 18711808 138574592  12%/db2backups/db2inst1
148347840   6% /db2backups/db2inst1
[root@hmcs-aspera-server ~]# exit
Logout
Connection to 192.168.102.170 closed.
```

8. Access the transferred contents from the target Power Virtual Server.

```
# hostname
prod-dal12
# ls -lart /db2backups/db2inst1/*/*
/db2backups/db2inst1/ecom_backup_offline/dir2
Total 19539448
drwxr-xr-s  4 root     system        256 Mar 20 12:28
drwxr-xr-s  2 root     system        256 Mar 20 12:29
-rw-r--r--  1 root     system 10001694720 Mar 20 12:43 ECOM.0.db2inst1.DBPART000.20240320122901.002

/db2backups/db2inst1/ecom_backup_offline/dir1
Total 17835432
drwxr-xr-s  4 root     system        256 Mar 20 12:28
drwxr-xr-s  2 root     system        256 Mar 20 12:29
-rw-r--r--  1 root     system 9129037824 Mar 20 12:43 ECOM.0.db2inst1.DBPART000.20240320122901.001
```

From the PowerVS Db2 DBA account, enter the following command to restore the database from the transferred backup files:

```
$ time db2 "RESTORE DB ECOM FROM /db2backups/db2inst1/ecom_backup_offline/dir1,
/db2backups/db2inst1/ecom_backup_offline/dir2 INTO ECOM REPLACE EXISTING COMPRLIB libdb2compr.a
WITHOUT PROMPTING"

real    2m8.10s
user    0m0.00s
sys     0m0.00s
$ date
Wed Mar 20 15:12:14 CDT  2024
$ db2 list db directory

 System Database Directory

 Number of entries in the directory = 1

Database 1 entry:

 Database alias                       = ECOM
 Database name                        = ECOM
 Local database directory             = /home/db2inst1
 Database release level               = 15.00
 Comment                              =
 Directory entry type                 = Indirect
 Catalog database partition number    = 0
 Alternate server hostname            =
 Alternate server port number         =
```

Once the restore has been completed, it is up to the customer to perform functional testing, validate integrity of data, declare target system to be operational and shut down the source database.

Set Up Power Virtual Server Backup

Reference: https://cloud.ibm.com/docs/power-iaas?topic=power-iaas-backup-strategies#baas

One aspect of making the migrated database operational on Power Virtual Server is to set up backups. Customer has options to make use of available backup capabilities sourced from the IBM Cloud Catalog and they can also elect to install and configure their own backup software.

The following slides demonstrate how "Secure Automated Backup with Compass", available in the IBM Cloud Catalog, can be used to establish an effective, cloud-native backup solution for the Db2 database on Power Virtual Server.

For AIX workloads, the Compass offering provides the following capabilities that are relevant for Db2 backups:

1. AIX operating systems

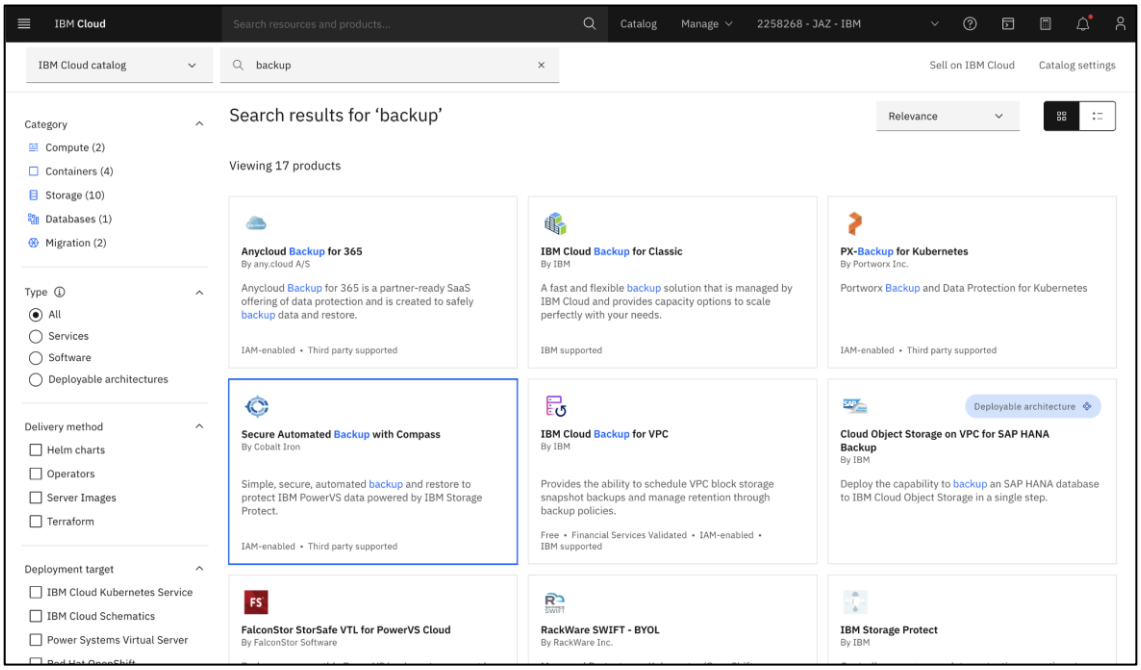
    1. File-level backup and restore
    2. Image-level backup and restore
    3. Policy management down to the directory and file object or type levels
    4. Backup and archive features that includes long-term retention of data

2. DB2 on AIX databases

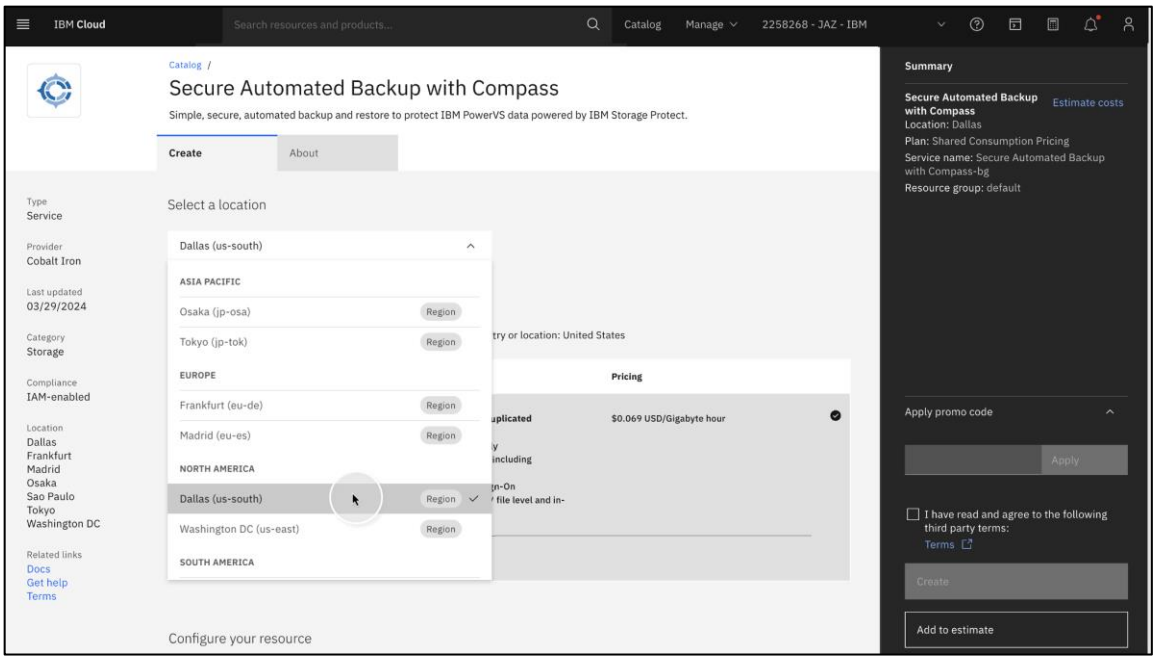
    1. DB2-integrated backup and restore of DB2 databases
    2. DB2-integrated archive logging of DB2 databases

The reference link supplied for this selection overviews this backup offering. Support is provided by Cobalt Iron and you will need to have login credentials to Cobalt Iron to access the Cobalt Iron documentation.

In the following IBM Cloud Catalog entry for Compass, customer can reference more information about Compass and view a representative demo.



When Customer decides to set up this utility, the proper locations for the backup service should be specified. In this case, that location is Dallas.

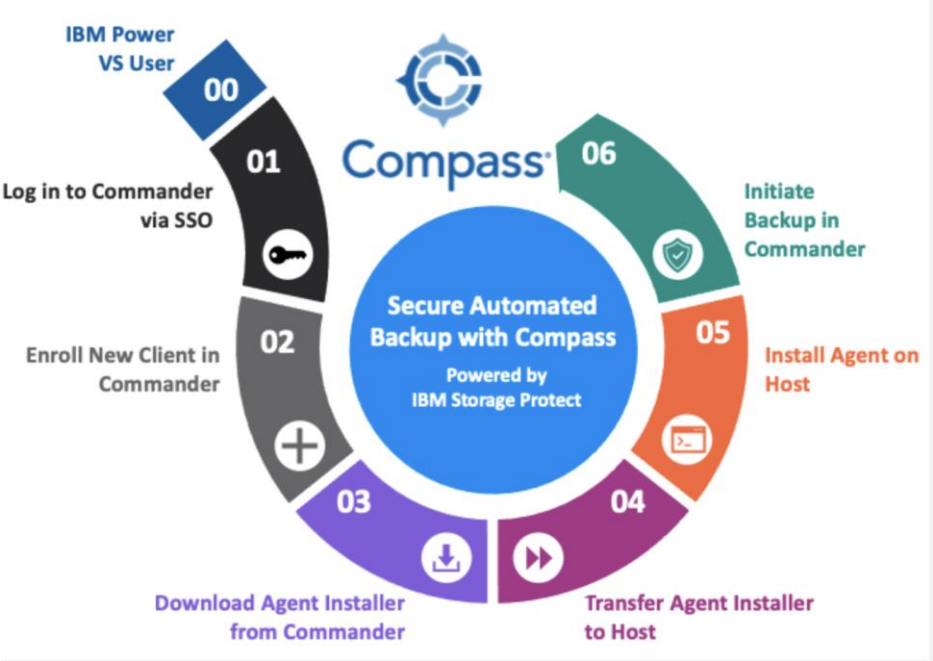


It will take some time to initialize the service. Once established, it will be time to select resources to back up.

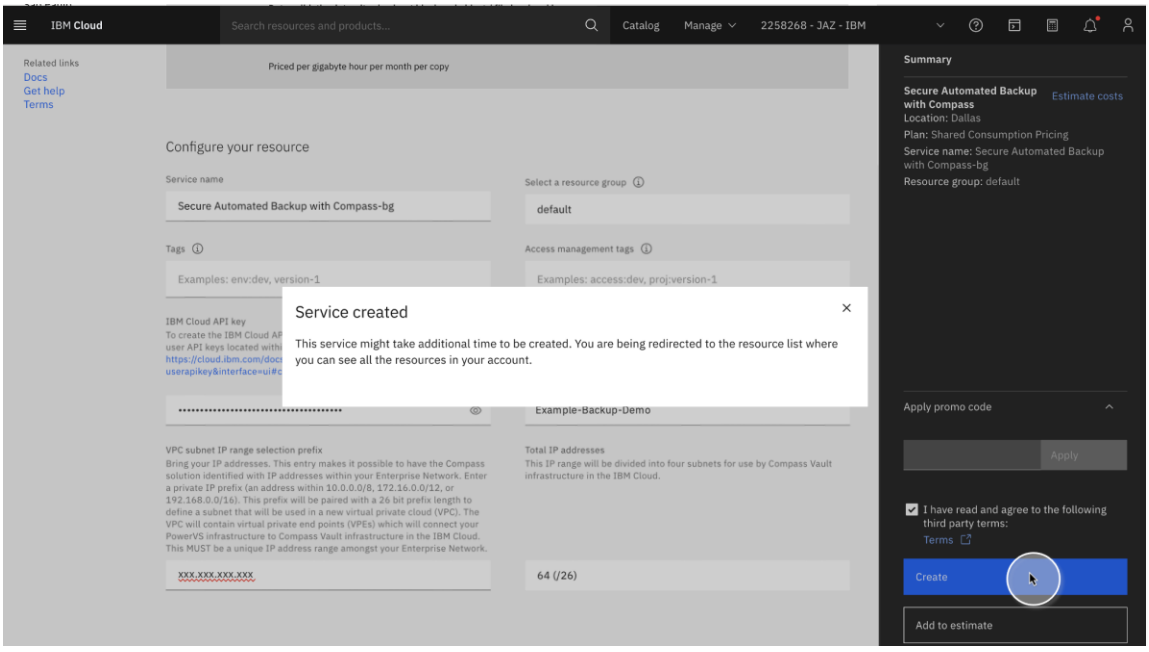
Compass Commander provides the administrative interface. The IBM Cloud Catalog link below will provide you with more information about Compass and allow you to review a representative demo of the available functionality.

https://cloud.ibm.com/catalog/services/secure-automated-backup-with-compass#about

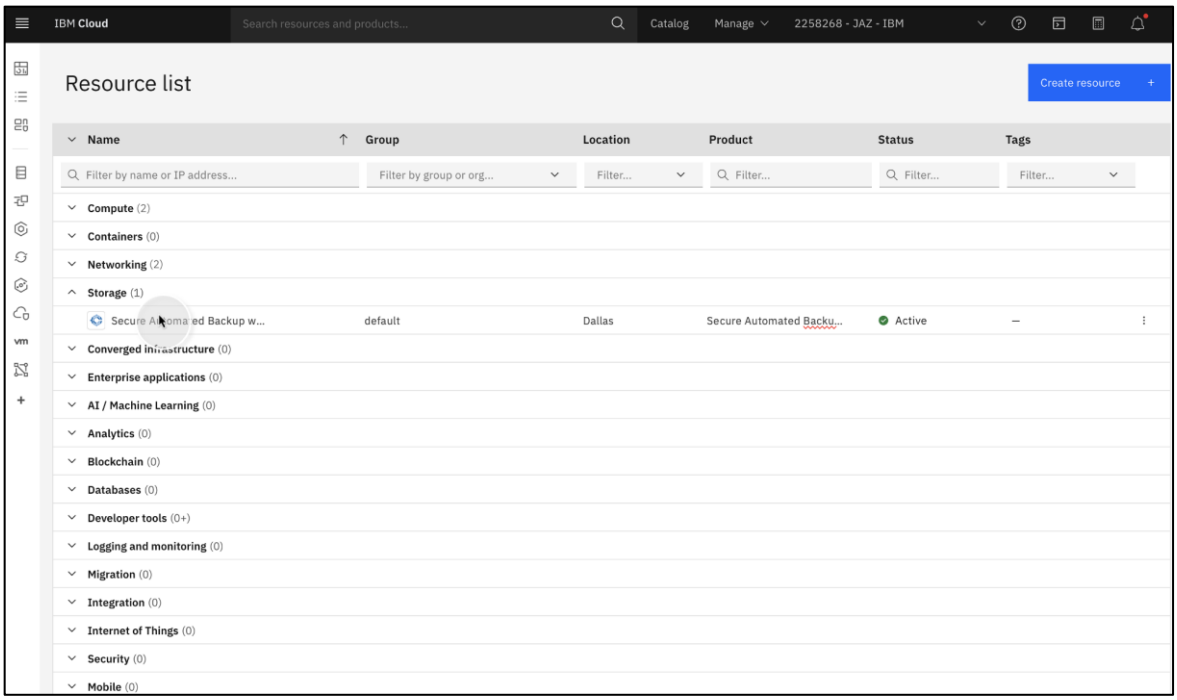
The following diagram summarizes the administrative workflow:



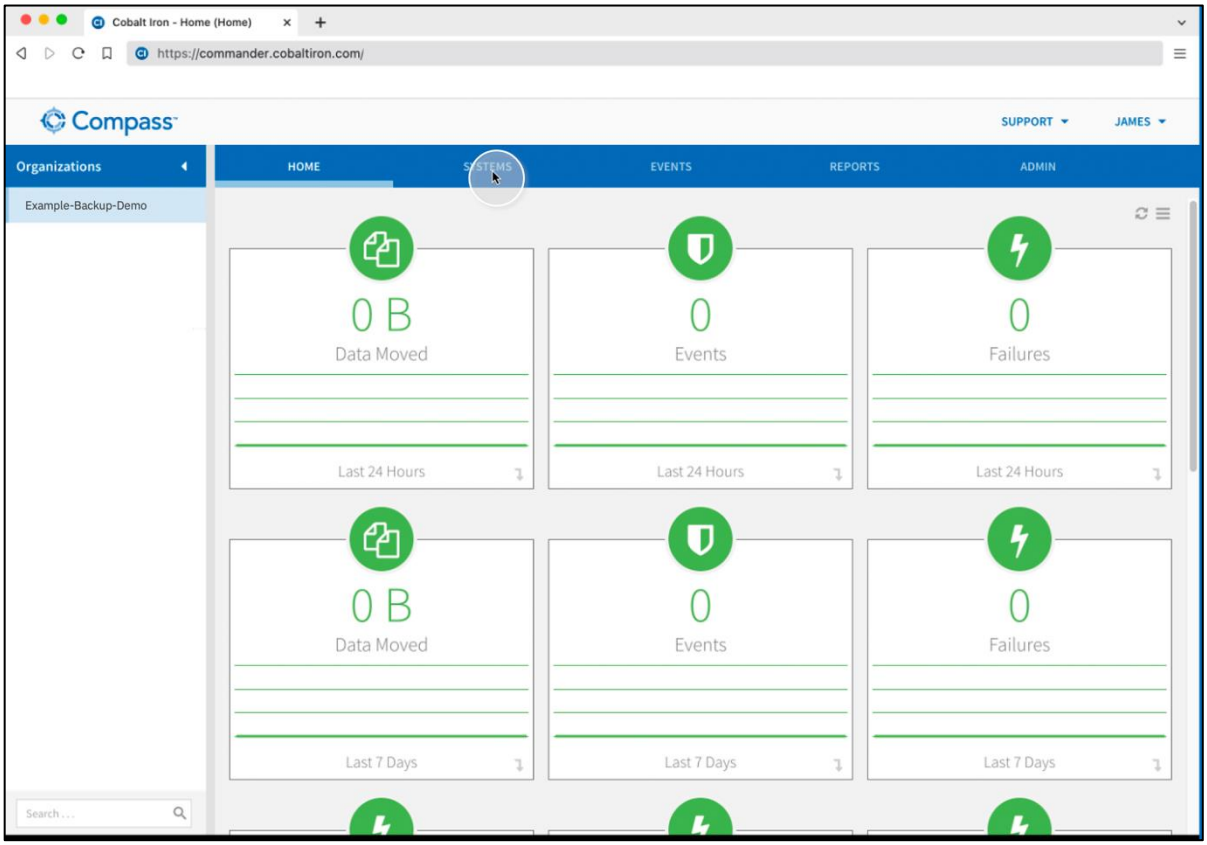
To back up a new Db2 instance on Power Virtual server, a backup administrator would enroll a new client in Commander, Download an Agent Installer from Command, transfer the Agent installer to the Host, then install the agent on the host.



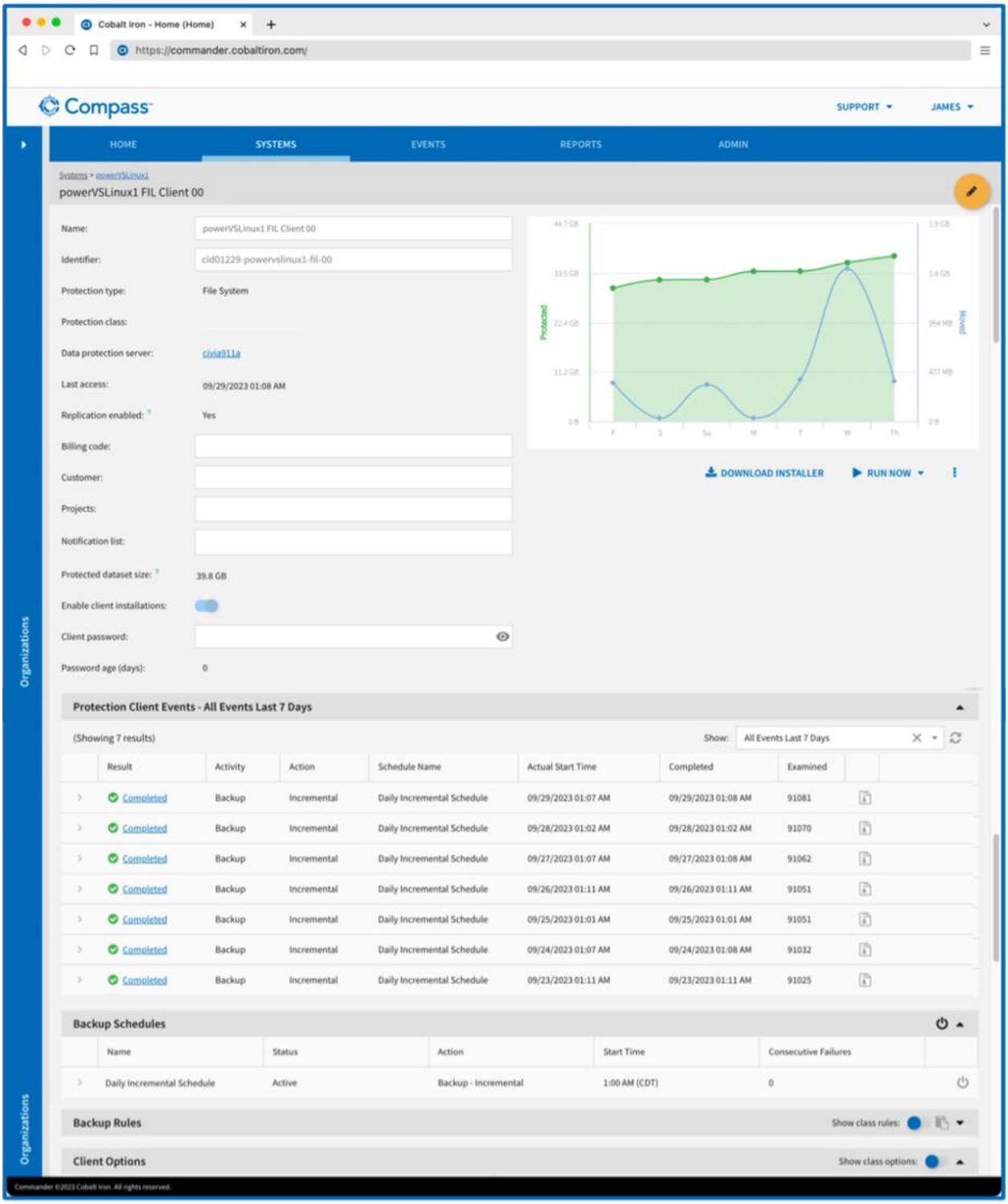
In the following snapshot, we can see that the backup service is active for the Dallas location.



Opening the utility reveals an interface which allows you to download agents and setup AIX file system backups and Db2 database specific backups.



The following screen presents a more detailed view of the interface:

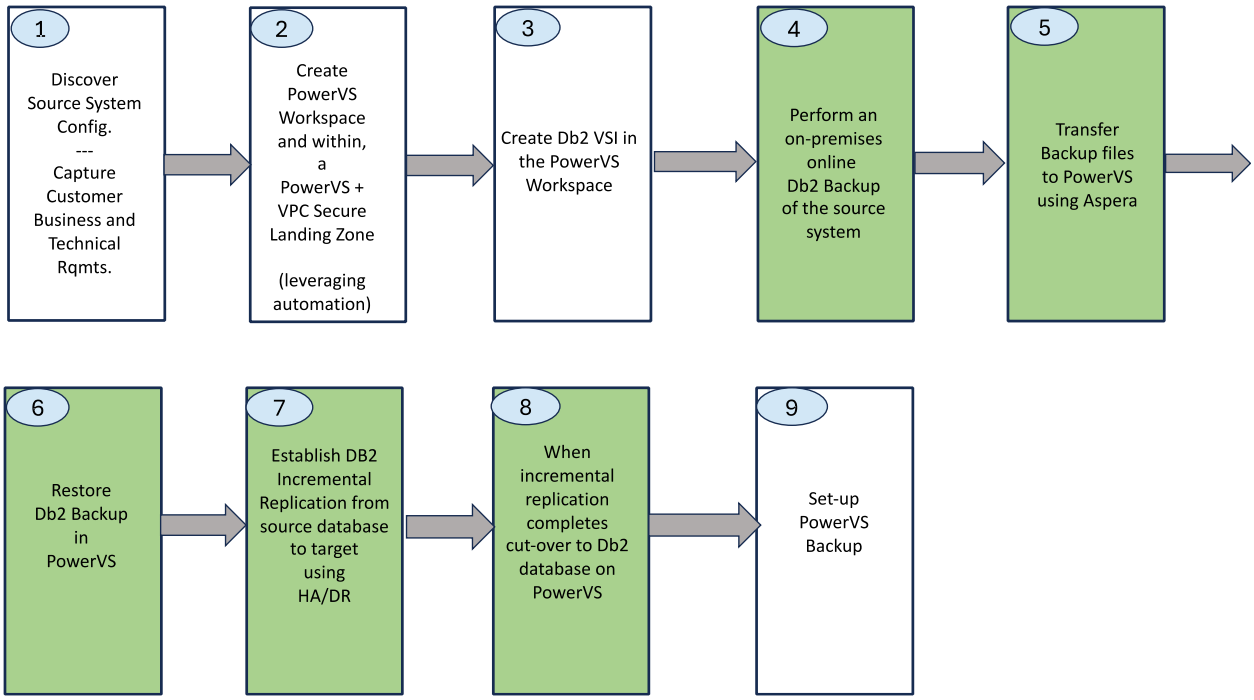


Option 2: Database Backup and Restore, followed by HADR synchronization

This option presents a procedure that uses **db2 backup** to take a point in time backup of a Db2 database resident on Power, transfers the backup file contents to PowerVS local file storage, then applies **db2 restore** to restore the database contents to the PowerVS-resident Db2 database. After the restore is done, incremental replication using Db2 HA/DR is applied to "catch up" the target database with the source.

Step Sequence

The following illustrates the Option 2 activity flow:



For option 2, we will assume discovery and requirements capture has been completed and the reader is familiar with steps 1,2,3 and 9 as discussed in Option 1. We will focus on Steps 4-8 as applied specifically for this option,

References:
https://www.ibm.com/support/pages/step-step-procedures-set-hadr-replicatin-between-db2-databases
https://www.ibm.com/docs/en/db2/11.1?topic=hadr-synchronization-mode
https://ibm.github.io/db2-hadr-wiki/hadrSyncMode.html

The following instructions are derived from the reference sources above, with commands tailored for AIX.

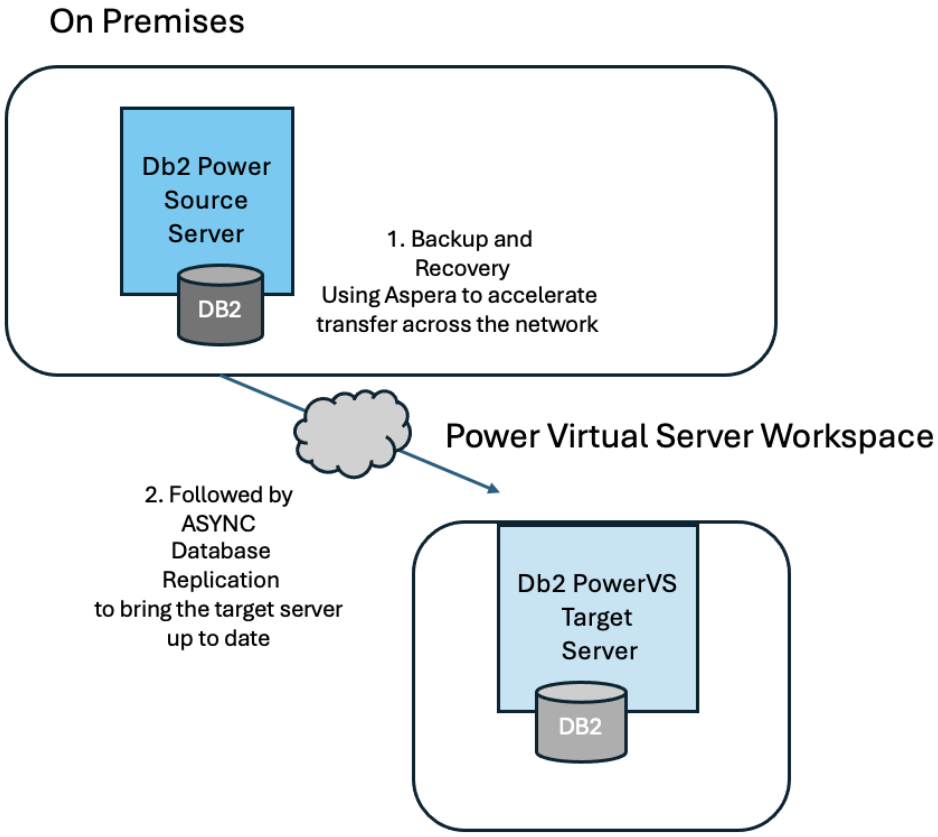
Background:

HADR synchronization mode is controlled by the database configuration parameter hadr_syncmode. See also High availability disaster recovery (HADR) synchronization mode

HADR provides 4 synchronization modes to suit a diverse range of operational environment. Database configuration parameter hadr_syncmode can be set to one of SYNC, NEARSYNC, ASYNC, or SUPERASYNC.

- SYNC - Transactions on primary will commit only after relevant logs have been written to disk on both primary and standby.
- NEARSYNC - Transactions on primary will commit only after relevant logs have been written to disk on primary and received into memory on standby.
- ASYNC - Transactions on primary will commit only after relevant logs have been written to local disk and sent to standby.
- SUPERASYNC - Transactions on primary does not wait for replication of logs to the standby.

Goal for this option is to set up asynchronous replication to capture incremental logs after the backup, between on-premises source and Power Virtual Server target environment. As is done throughout this procedure, the database name will be assumed to be ECOM.



Db2 Backup and Restore

On the source Power database (Primary) server:

1. Create sample db using the **create database** command.

```
$ create database ECOM
```

Above command is a simple example. This command has many options, please see https://www.ibm.com/docs/en/db2/11.5?topic=commands-create-database for examples.

2. Enable it for log archiving.

```
$ db2 update db cfg for ECOM using LOGARCHMETH1 "DISK: /dbuser/archived_logs"
```

Above command will enable database for log archiving. All the transaction logs will get archived to this location, so make sure you have enough disk space to accommodate the archived transaction logs. This will also place the db in backup pending state.

3. Take an offline backup. This is the basic form of the command.

```
$ db2 "backup database ECOM"
```

4. Set up HADR cfg parameters on Primary database. Note that parameters are set to establish asynchronous updates to the standby (target PowerVS) server.

```
$ db2 update db cfg for ECOM using HADR_LOCAL_HOST <IP ADDRESS OF PRIM>
$ db2 update db cfg for ECOM using HADR_LOCAL_SVC <PORT # on PRIM>
$ db2 update db cfg for ECOM using HADR_REMOTE_HOST <IP ADDRESS OF STNDBY>
$ db2 update db cfg for ECOM using HADR_REMOTE_SVC <PORT # on STNDBY>
$ db2 update db cfg for ECOM using HADR_REMOTE_INST <INSTNAME OF STNDBY>
$ db2 update db cfg for ECOM using LOGINDEXBUILD ON
$ db2 update db cfg for ECOM using HADR_ROLE PRIMARY
$ db2 update db cfg for ECOM using HADR_SYNCMODE ASYNC
```

5. Take an online backup to be used for setting HADR. Note that we use the same backup parameters that were used in Option 1 to apply optimization.

```
$ time db2 "backup database ECOM on all dbpartitionnums online to
/db2backups/db2inst1/ecom_backup_online/dir1,/db2backups/db2insti1/ecom_backup_online/dir2
COMPRESS COMPRLIB libdb2compr.a"
Part Result
---- --------------------------------------------------------------------------
0000 DB20000I  The BACKUP DATABASE command completed successfully.

Backup successful.  The time stamp for this backup image is 20240320122901

real   14m39.70s
user   0m0.01s
sys.   0m0.00s
```

<u>On the target PowerVS database (standby) server:</u>

Ensure both the servers are on the same db2level so that a mismatch situation does not occur.

Run "db2level" command on both the servers to check whether they are on the same DB2 Version and Fix Pack.

6. Transfer the backup image (from the primary machine) to the STANDBY MACHINE. Use Aspera, sftp, or whatever secure transfer protocol is available.

7. Perform a db2 restore. The basic form of restore is shown below.

```
$ db2 "restore database ECOM"
```

The following is the restore command to execute corresponding with the fully parameterized backup command.

```
$ time db2 "RESTORE DB ECOM FROM /db2backups/db2inst1/backup_online/dir1,
/db2backups/db2inst1/backup_online/dir2 INTO ECOM REPLACE EXISTING COMPRLIB libdb2compr.a
WITHOUT PROMPTING"

real    2m8.10s
user    0m0.00s
sys     0m0.00s
$ date
Wed Mar 20 15:12:14 CDT  2024
$ db2 list db directory

 System Database Directory

 Number of entries in the directory = 1

Database 1 entry:

 Database alias                       = ECOM
 Database name                        = ECOM
 Local database directory             = /home/db2inst1
 Database release level               = 15.00
 Comment                              =
 Directory entry type                 = Indirect
 Catalog database partition number    = 0
 Alternate server hostname            =
 Alternate server port number         =
```

Followup Db2 Temporary HADR Configuration to Synchronize Source and Target Databases

8. Set up HADR cfg parameters on standby database.

```
$ db2 update db cfg for ECOM using HADR_LOCAL_HOST <IP ADDRESS ON STANDBY>
$ db2 update db cfg for ECOM using HADR_LOCAL_SVC <PORT # ON STANDBY>
$ db2 update db cfg for ECOM using HADR_REMOTE_HOST <IP ADDRESS ON PRIMARY>
$ db2 update db cfg for ECOM using HADR_REMOTE_SVC <PORT # ON PRIMARY>
$ db2 update db cfg for ECOM using HADR_REMOTE_INST <INSTNAME ON PRIMARY>
$ db2 update db cfg for ECOM using HADR_ROLE STANDBY
$ db2 update db cfg for ECOM using HADR_SYNCMODE ASYNC
```

9. Execute db2 start hadr on database <DBNAME> as standby

```
$ db2 start hadr on database ECOM as standby
```

<u>On the source Power database (Primary) server:</u>

10. Starting up HADR on the primary server

```
$ db2 start hadr on database ECOM as primary
```

Verify that HADR is running using the following command:

```
$ db2pd -db ECOM -hadr
```

You should see output similar to this:

```
Database Member 0 -- Database ECOM -- Active -- Up 0 days 00:23:17 -- Date 06/08/2011 13:57:23

                                 HADR_ROLE = PRIMARY
                               REPLAY_TYPE = PHYSICAL
                            HADR_SYNCMODE = ASYNC
                               STANDBY_ID = 1
                            LOG_STREAM_ID = 0
                               HADR_STATE = PEER
                        PRIMARY_MEMBER_HOST = hostP.ibm.com
                           PRIMARY_INSTANCE = db2inst
                             PRIMARY_MEMBER = 0
                        STANDBY_MEMBER_HOST = hostS1.ibm.com
                           STANDBY_INSTANCE = db2inst
                             STANDBY_MEMBER = 0
                        HADR_CONNECT_STATUS = CONNECTED
                   HADR_CONNECT_STATUS_TIME = 06/08/2011 13:38:10.199479 (1307565490)
                 HEARTBEAT_INTERVAL(seconds) = 25
                        HADR_TIMEOUT(seconds) = 100
               TIME_SINCE_LAST_RECV(seconds) = 3
                    PEER_WAIT_LIMIT(seconds) = 0
                  LOG_HADR_WAIT_CUR(seconds) = 0.000
         LOG_HADR_WAIT_RECENT_AVG(seconds) = 0.006298
       LOG_HADR_WAIT_ACCUMULATED(seconds) = 0.516
                      LOG_HADR_WAIT_COUNT = 82
     SOCK_SEND_BUF_REQUESTED,ACTUAL(bytes) = 0, 50772
     SOCK_RECV_BUF_REQUESTED,ACTUAL(bytes) = 0, 87616
                PRIMARY_LOG_FILE,PAGE,POS = S0000009.LOG, 1, 49262315
                STANDBY_LOG_FILE,PAGE,POS = S0000009.LOG, 1, 49262315
                      HADR_LOG_GAP(bytes) = 0
          STANDBY_REPLAY_LOG_FILE,PAGE,POS = S0000009.LOG, 1, 49262315
            STANDBY_RECV_REPLAY_GAP(bytes) = 0
                          PRIMARY_LOG_TIME = 06/08/2011 13:49:19.000000 (1307566159)
                          STANDBY_LOG_TIME = 06/08/2011 13:49:19.000000 (1307566159)
                   STANDBY_REPLAY_LOG_TIME = 06/08/2011 13:49:19.000000 (1307566159)
              STANDBY_RECV_BUF_SIZE(pages) = 16
                  STANDBY_RECV_BUF_PERCENT = 0
                STANDBY_SPOOL_LIMIT(pages) = 0
                      PEER_WINDOW(seconds) = 0
                  READS_ON_STANDBY_ENABLED = Y
        STANDBY_REPLAY_ONLY_WINDOW_ACTIVE = N
```

Determine if Databases are Consistent

Reference: https://ibm.github.io/db2-hadr-wiki/hadrMonitoring.html#db2pd__hadr

Enter the following command from the source database.

```
$ db2pd -db ECOM -hadr
```

Transfer Database Service to Standby Server on PowerVS

The following steps assume that the databases for the on-premises HADR configuration and the remote auxiliary standby database are consistent,

Here are the steps to switch service to the standby instance,

- Let's call the current on-premises PRIMARY "server1" and
- the current on-premises STANDBY "server2

1. ON PRIMARY (server1): db2 connect to ECOM
2. ON PRIMARY (server1): power down the Primary --> db2stop
3. ON STANDBY (server2): db2 takeover hadr on database ECOM
4. The STANDBY instance on server2 is now the primary
5. ON server2: db2pd -db ECOM -hadr (the ROLE should state: PRIMARY)

When ready, stop on-premise database operations permanently.

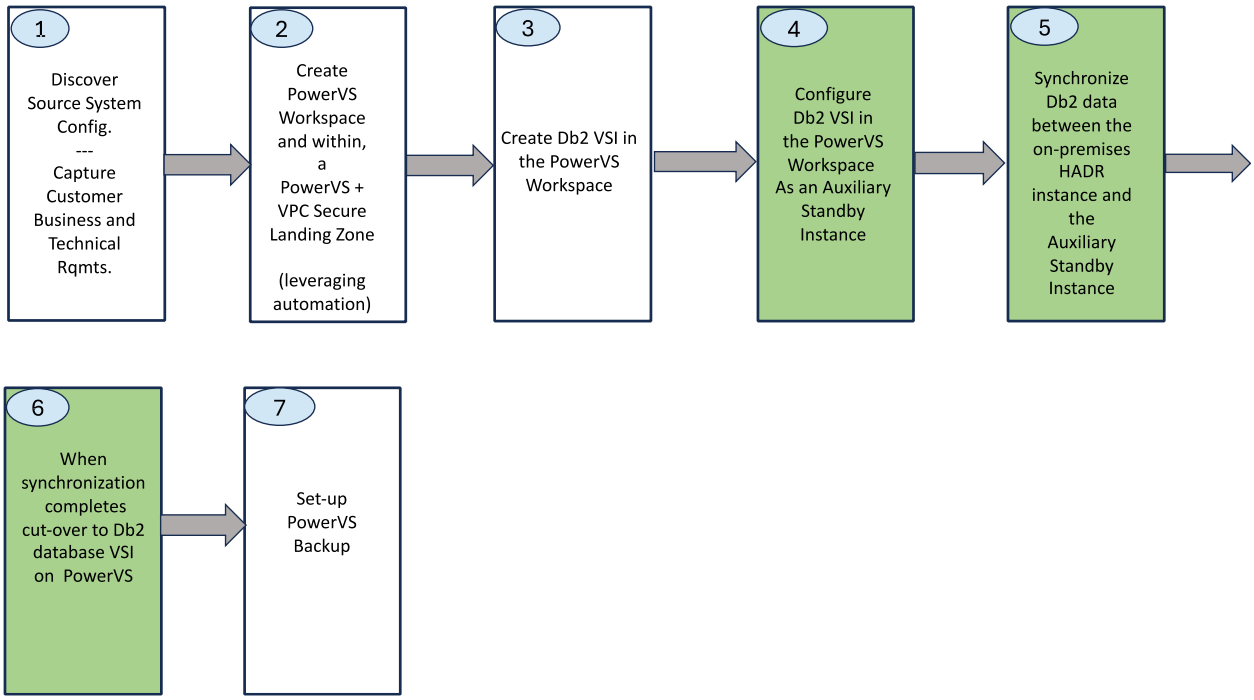
Note:
1. Hostname of the HADR pair cannot be the same on both the servers.
2. The instance name and the underlying userid on UNIX systems can be different. Make sure to update the correct name of the instance for the db cfg parameter HADR_REMOTE_INST to the correct value.

Option 3: Migrate An Existing Power Db2 HADR database to a Single Database Instance On Power Virtual Server.

This section presents a standard approach to add resiliency to a Db2 database that has been migrated to Power Virtual Server, by configuring a new Power Virtual Server auxiliary standby Db2 instance to an existing on-premises Db2 pair. Once consistency has been established, service is transferred to the Power Virtual Server resident instance.

Step Sequence

The following illustrates the Option 3 activity flow:



For option3 we will assume discovery and requirements capture has been completed and the reader is familiar with steps 1,2,3 and 7 as discussed in Option 1. We will focus on Steps 4-6 as applied specifically for this option.

References:

Use this Redbook as a starting point

- https://www.redbooks.ibm.com/redbooks/pdfs/sg247363.pdf , “High Availability and Disaster Recovery Options for DB2 for Linux, UNIX, and Windows”

then consult these resources as required to get additional details.

- https://www.ibm.com/support/pages/steps-adding-new-auxiliary-standby-existing-db2-hadr-pair
- https://www.ibm.com/docs/en/db2/11.5?topic=recover-disaster-recovery
- https://www.ibm.com/docs/en/db2/11.5?topic=server-high-availability-disaster-recovery-hadr
- https://www.ibm.com/docs/en/db2/11.1?topic=hadr-synchronization-mode
- https://ibm.github.io/db2-hadr-wiki/hadrSyncMode.html
- https://www.ibm.com/docs/en/db2/11.5?topic=availability-high-db2-server

Background

Many Db2 configurations make use of an auxiliary standby instance installed at a remote site to provide support for disaster recovery, if services at the primary site be ome unavailable for an extended period of time. The auxiliary standby instance is kept up to data using database replication across the network. This option will use this configuration as a migration mechanism, eventually transferring service to the Power Virtual Server resident instance.

Replication Across a Network

We previously went over using database replication to migrate and synchronize an on-premises database to Power Virtual Server. The same technologies, procedures and considerations apply in setting up an auxiliary standby instance.

An important decision is choosing the type of replication. As you recall, HADR provides (4) synchronization modes to suit a diverse range of operational environment. Database configuration parameter hadr_syncmode can be set to one of SYNC, NEARSYNC, ASYNC, or SUPERASYNC.

Across long distances, SYNC replication is not performant, and therefore is not suitable to maintain a auxiliary standby instance.

Here is a brief description of the three candidate modes of replication:

NEARSYNC
: This mode provides somewhat less protection against transaction loss, in exchange for a shorter transaction response time than that of SYNC mode.

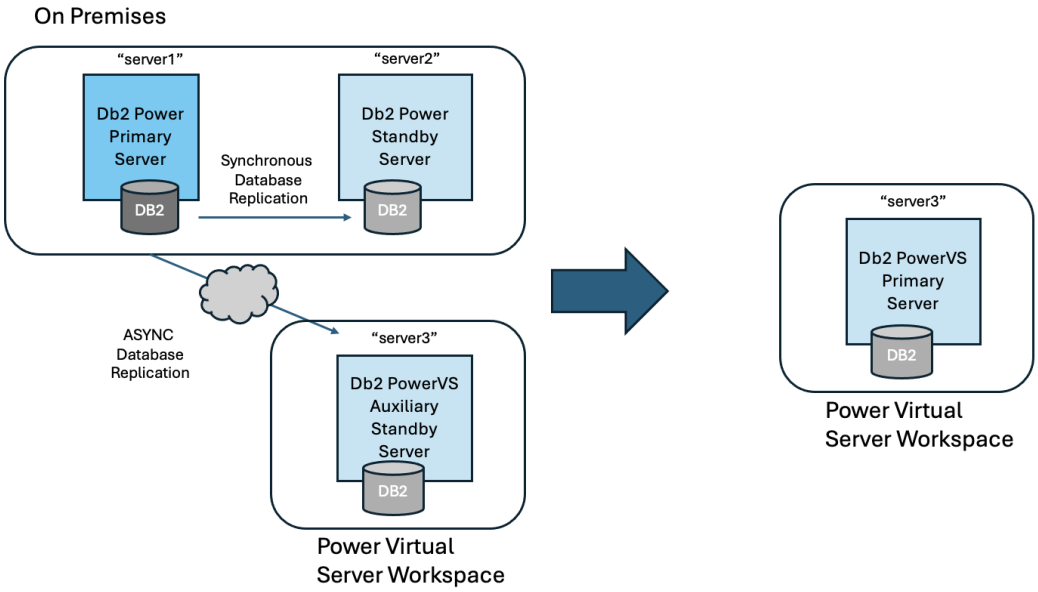
ASYNC
: Compared with the SYNC and NEARSYNC modes, the ASYNC mode results in shorter transaction response times but might cause greater transaction losses if the primary database fails.

SUPERASYNC

This mode has the shortest transaction response time but also has the highest probability of transaction losses if the primary system fails. This mode is useful when you do not want transactions to be blocked or experience longer response times due to network interruptions or congestion.

In this mode, the HADR pair can never be in peer state or disconnected peer state. The log writes are considered successful only when the log records were written to the log files on the primary database. Because the primary system does not wait for acknowledgement from the standby system, transactions might be considered committed when they are still on their way to the standby.

ASYNC mode is the most appropriate choice for database replication across long distances when a single auxiliary standby instance is involved.



The following steps, closely sourced from https://www.ibm.com/support/pages/steps-adding-new-auxiliary-standby-existing-db2-hadr-pair, demonstrate how to connect a Power Virtual server based Db2 instance as an auxiliary standby server.

Key Assumption: (2) Db2 servers has been installed on a Power Virtual server, sized similarly to the primary Db2 database, with identical software version and patch levels.

1. For each database, collect a Db2 online backup from the primary server. Once again, ECOM is the database name used.

```
$ db2 backup db ECOM online to /tmp/backups
```

Record the timestamp of the backup taken.

2. For each database, restore the backups onto the new auxiliary standby Db2 server on Power Virtual Server.

```
$ db2 restore db ECOM from /tmp/backups taken at <timestamp of backup>
```

3. Determine the host name, port number, and instance name that will be in the HADR setup.

Example:

Role	Host name	Port number	Instance name
Primary	server1	60061	db2inst1
Principal standby	server2	60062	db2inst2
Auxiliary standby	server3	60063	db2inst3

4. Stop HADR for each database on the existing Primary and Standby servers.

```
$ db2 STOP HADR ON DATABASE ECOM
```

5. On each server, update the HADR_TARGET_LIST setting for each database as follows:

server1:

```
$ db2 update db cfg for ECOM using HADR_TARGET_LIST server2:60062 | server3:60063 immediate
```

server2:

```
$ db2 update db cfg for ECOM using HADR_TARGET_LIST server1:60061 | server3:60063 immediate
```

server3:

```
$ db2 update db cfg for ECOM using HADR_TARGET_LIST server1:60061 | server2:60062 immediate
```

6. On the new Auxiliary Standby (server3) configure the following settings for each database (these should already be set on the other two servers, since HADR is currently up and running there):

```
$ db2 update db cfg for ECOM using HADR_SYNCMODE SYNC
$ db2 update db cfg for ECOM using HADR_LOCAL_HOST server3
$ db2 update db cfg for ECOM using HADR_LOCAL_SVC 60063
$ db2 update db cfg for ECOM using HADR_REMOTE_HOST server1
$ db2 update db cfg for ECOM HADR_REMOTE_SVC 60061
$ db2 update db cfg for ECOM HADR_REMOTE_INST db2inst1
```

7. Start HADR on each server, for each database.

The correct order for starting HADR is to start HADR on the principal standby first, then the Aux standby, and then start it on the primary.

On Principal Standby (server2):

```
$ db2 start hadr on database ECOM as standby
```

On Auxiliary Standby (server3):

```
$ db2 start hadr on database ECOM as standby
```

On Primary (server1):

```
$ db2 start hadr on database ECOM as primary
```

Verify that the HADR connections are up and running:

```
$ db2pd -db ECOM -hadr
```

Now we must wait until the auxiliary standby instance is fully synchronized. Depending upon the size of the database and the speed of the connection from the on-premises database to the remote auxiliary standby instance, the synchronization could take hours or days.

Determine if Databases are Consistent

Reference: https://ibm.github.io/db2-hadr-wiki/hadrMonitoring.html#db2pd__hadr

Enter the following command from the source database.

```
$ db2pd -db ECOM -hadr.
```

Transfer Database Service to Auxiliary Standby server on PowerVS

The following steps assume that the databases for the on-premises HADR configuration and the remote auxiliary standby database are consistent.

Here are the steps to switch service to the auxialiary standby instance,

- Let's call the current on-premises PRIMARY "server1",
- the current on-premises STANDBY "server2" and
- the remote AUXILIARY STANDBY "server3".

1. ON PRIMARY (server1): db2 connect to ECOM
2. ON PRIMARY (server1): power down the primary --> db2stop force
3. ON STANDBY (server2): power down the standby -- > db2stop force
4. ON AUXILIARY STANDBY (server3): db2 takeover hadr on database ECOM by force
5. server3 is now the primary
6. ON server3: db2pd -db ECOM -hadr (the ROLE should state: PRIMARY)

When ready, stop on-premise database operations permanently.

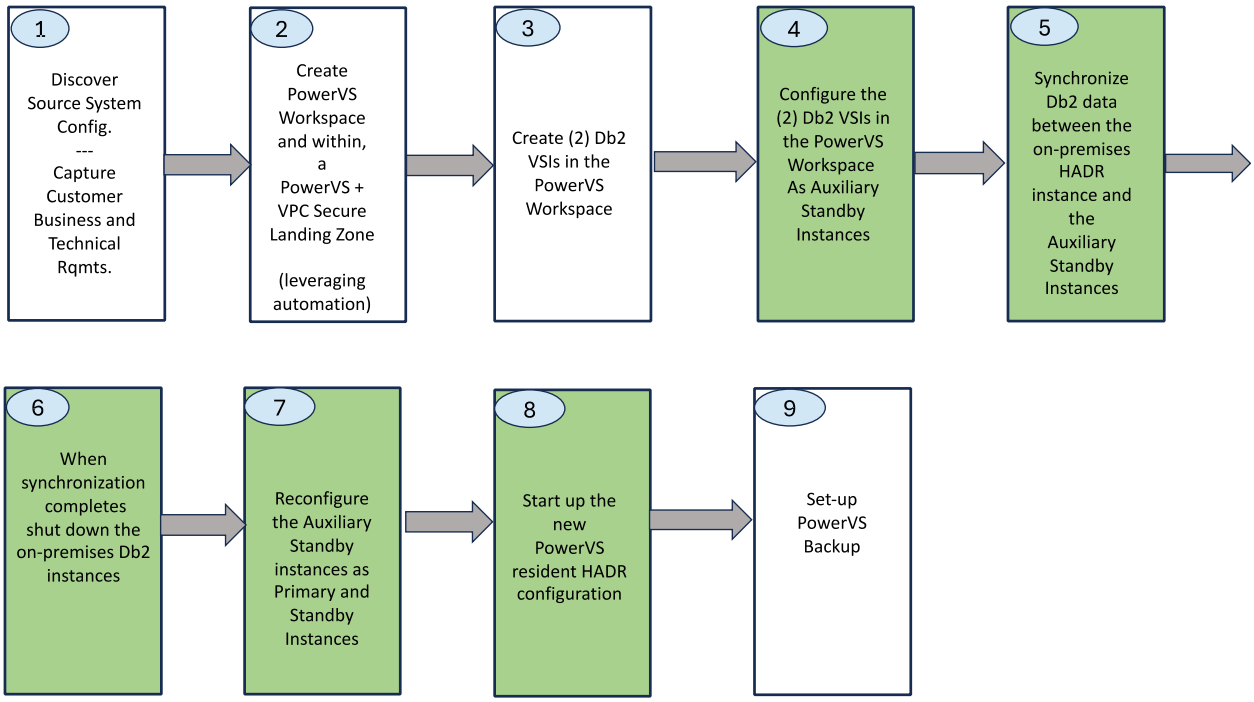
Redirect all applications to the db2inst3 instance.

Option 4: Migrate an Existing On-Premises Power HADR configuration to Power Virtual Server

This section leverages Db2's ability to support two (or more) auxiliary standby services via replication. For an existing on-premises HADR configuration, two auxiliary standby servers are established in Power Virtual Server. One all of the databases are synchronized, the on-premises servers are shut down, and the auxiliary standby servers are reconfigured as a primary/standby HADR configuration pair.

Step Sequence

The following illustrates the Option 4 activity flow:



For option 4 we will assume discovery and requirements capture has been completed and the reader is familiar with steps 1,2,3 and 9 as discussed in Option 1. We will focus on Steps 4-8 as applied specifically for this option.

References:

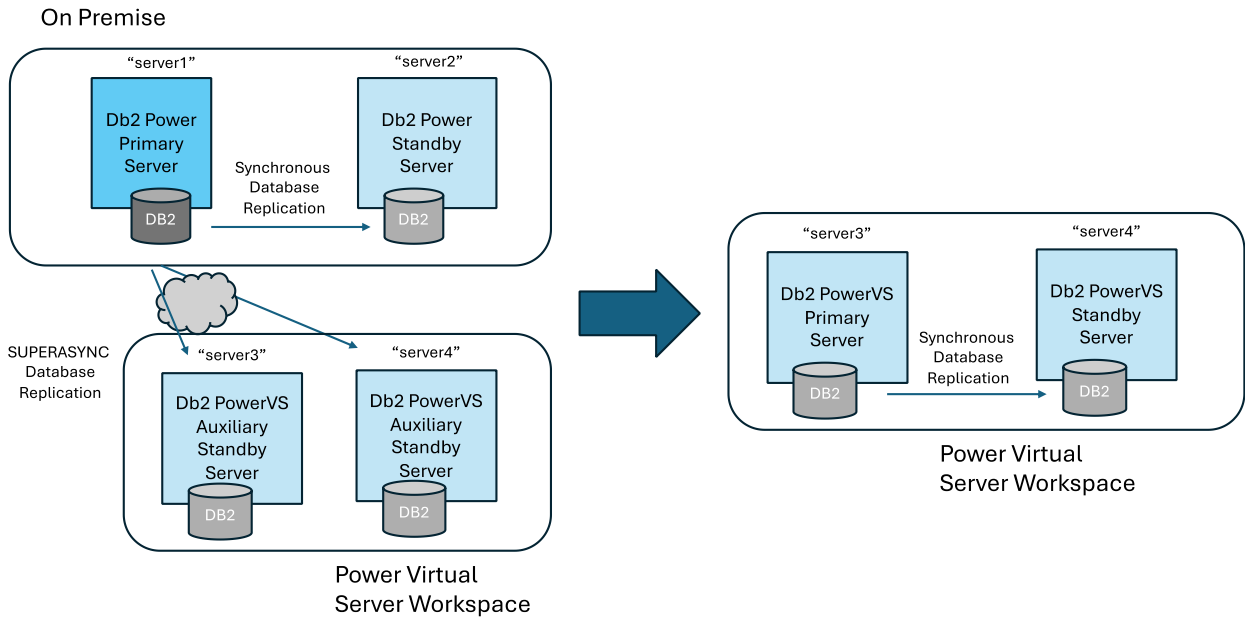
Use this Redbook as a starting point

- https://www.redbooks.ibm.com/redbooks/pdfs/sg247363.pdf , "High Availability and Disaster Recovery Options for DB2 for Linux, UNIX, and Windows"

then consult these resources as required to get additional details.

- https://www.ibm.com/support/pages/steps-adding-new-auxiliary-standby-existing-db2-hadr-pair
- https://www.ibm.com/docs/en/db2/11.5?topic=recover-disaster-recovery
- https://www.ibm.com/docs/en/db2/11.5?topic=server-high-availability-disaster-recovery-hadr
- https://www.ibm.com/docs/en/db2/11.1?topic=hadr-synchronization-mode
- https://ibm.github.io/db2-hadr-wiki/hadrSyncMode.html
- https://www.ibm.com/docs/en/db2/11.5?topic=availability-high-db2-server

This procedure applies SUPERASYNC replication between the primary server and the two Auxiliary Standby instances. SUPERASYNC is approapriate when you have two or more auxiliary standby instances deployed.



The following steps, closely sourced from https://www.ibm.com/support/pages/steps-adding-new-auxiliary-standby-existing-db2-hadr-pair, demonstrate how to connect a Power Virtual server based Db2 instance as an auxiliary standby server, for disaster recovery purposes.

Key Assumption: Two Db2 Power Virtual Servers have been installed, co-resident in a Power Virtual Server Workspace, sized similarly to the primary Db2 database, with identical software version and patch levels, and the Power Virtual Servers having matching or more modern firmware levels than the source database,

1. For each database, collect a Db2 online backup from the primary server. Once again, ECOM is the database name used.

```
$ db2 backup db ECOM online to /tmp/backups
```

Record the timestamp of the backup taken.

2. For each database, restore the backups onto the new auxiliary standby Db2 Power Virtual Server.

```
$ db2 restore db ECOM from /tmp/backups taken at <timestamp of backup>
```

3. Determine the host name, port number, and instance name that will be in the HADR setup.

Example:

Role	Host name	Port number	Instance name
Primary	server1	60061	db2inst1
Principal standby	server2	60062	db2inst2
Auxiliary standby	server3	60063	db2inst3
Auxiliary standby	server4.	60064	db2inst4

4. Stop HADR for each database on the existing Primary and Standby servers.

```
$ db2 STOP HADR ON DATABASE ECOM
```

5. On each server, update the HADR_TARGET_LIST setting for each database as follows:

server1:

```
$ db2 update db cfg for ECOM using HADR_TARGET_LIST server2:60062 | server3:60063 |
server4:60064 immediate
```

server2:

```
$ db2 update db cfg for ECOM using HADR_TARGET_LIST server1:60061 | server3:60063 |
server4:60064 immediate
```

server3:

```
$ db2 update db cfg for ECOM using HADR_TARGET_LIST server4:60064 | server1:60061 |
server2:60062 | immediate
```

server4:

```
$ db2 update db cfg for ECOM using HADR_TARGET_LIST server3:60063 |server1:60061 |
server2:60062 |immediate
```

6. On Auxiliary Standby instance (server3) configure the following settings for each database (these should already be set on the other two servers, since HADR is currently up and running there):

```
$ db2 update db cfg for ECOM using HADR_SYNCMODE SYNC
$ db2 update db cfg for ECOM using HADR_LOCAL_HOST server3
$ db2 update db cfg for ECOM using HADR_LOCAL_SVC 60063
$ db2 update db cfg for ECOM using HADR_REMOTE_HOST server1
$ db2 update db cfg for ECOM using HADR_REMOTE_SVC 60061
$ db2 update db cfg for ECOM using HADR_REMOTE_INST db2inst1
```

7. On Auxiliary Standby instance (server4) configure the following settings for each database:

```
$ db2 update db cfg for ECOM using HADR_SYNCMODE SYNC
$ db2 update db cfg for ECOM using HADR_LOCAL_HOST server4
$ db2 update db cfg for ECOM using HADR_LOCAL_SVC 60064
$ db2 update db cfg for ECOM using HADR_REMOTE_HOST server1
$ db2 update db cfg for ECOM using HADR_REMOTE_SVC 60061
$ db2 update db cfg for ECOM using HADR_REMOTE_INST db2inst1
```

7. Start HADR on each server, for each database.

The correct order for starting HADR is to start HADR on the Principal Standby first, then the Auxiliary Standbys, and then start it on the Primary.

On Principal Standby (server2):

```
$ db2 start hadr on database ECOM as standby
```

On Auxiliary Standby (server3):

```
$ db2 start hadr on database ECOM as standby
```

On Auxiliary Standby (server4):

```
$ db2 start hadr on database ECOM as standby
```

On Primary (server1):

```
$ db2 start hadr on database ECOM as primary
```

Verify that the HADR connections are up and running:

```
$ db2pd -db ECOM -hadr
```

Now we must wait until the auxiliary standby instances are fully synchronized. Depending upon the size of the database and the speed of the connection from the on-premises database to the remote autxiliary standby instance, the synchronization could take hours or days.

Determine if databases are consistent

Reference: https://ibm.github.io/db2-hadr-wiki/hadrMonitoring.html#db2pd__hadr

Enter the following command from the source database.

```
$ db2pd -db ECOM -hadr.
```

Transfer database service to the Auxiliary Standby servers on PowerVS

The following steps assume that the databases for the on-premises HADR configuration and the remote auxiliary standby database are consistent.

Here are the steps to switch service to the auxialiary standby instance,

- Let's call the current on-premises PRIMARY "server1",
- the current on-premises server2 STANDBY "server2"
- the remote server3 AUXILIARY STANDBY "server3", and
- the remote server4 AUXILIARY STANDBY "server4".

1. ON PRIMARY (server1): db2 connect to ECOM
2. ON PRIMARY (server1): power down the primary --> db2stop force
3. ON STANDBY (server2): power down the standby -- > db2stop force
4. ON AUXILIARY STANDBY server4 configure the following:

On AUXILIARY STANDBY instance (server4) configure the following settings for each database:

```
$ db2 update db cfg for ECOM using HADR_SYNCMODE SYNC
$ db2 update db cfg for ECOM using HADR_LOCAL_HOST server4
$ db2 update db cfg for ECOM using HADR_LOCAL_SVC 60064
$ db2 update db cfg for ECOM using HADR_REMOTE_HOST server3
$ db2 update db cfg for ECOM using HADR_REMOTE_SVC 60063
$ db2 update db cfg for ECOM using HADR_REMOTE_INST db2inst3
```

5. ON AUXILIARY STANDBY (server3): db2 takeover hadr on database ECOM by force
6. server3 is now the primary
7: ON server3: db2pd -db <dbname> -hadr (the ROLE should state: PRIMARY)

When ready, stop on-premise database operations permanently.

Redirect all applications to the dbinst3 instance.

Appendix

Seagate Lyve Mobile Solution for Physical Data Transfer

Overview

Reference: https://www.seagate.com/products/cloud/lyve-mass-storage-platform/

Lyve™ Mobile from Seagate® is a high-capacity edge storage solution that enables businesses to aggregate, store, move, and activate their data. Scalable, modular, and vendor agnostic, this integrated solution eliminates network dependencies so you can transfer mass data sets in a fast, secure, and efficient manner. With on-demand consumption delivered as a service, you order and pay only for the devices you need, when you need them.

As applied to RMAN backup and restore, Lyve Mobile can be used as a local store for source system backup. The device can be detached and physically shipped to Seagate. When supplied with Cloud Object Storage credentials, Seagate will, upon receipt, transfer the contents of the device across the network to Cloud Object Storage that a customer allocates, in this case, as staging space for the backup files to be restored to a target PowerVS instance.

Product and Service Brochure

https://www.seagate.com/content/dam/seagate/migrated-assets/www-content/product-content/data-transport/_shared/files/SC704_1-2102US-Lyve-Mobile-Family-Flyer-revised.pdf

Physical Devices and Associated Capacities

https://www.seagate.com/products/data-transport/

Aspera

Overview

References:

Product Overview content:

- https://www.ibm.com/products/aspera
- https://www.ibm.com/docs/en/aspera-on-cloud?topic=cloud-what-is-aspera

Accelerated Network Transfer for Migration (On-premises to Power Virtual Server) Do-it-Yourself solution guide

- https://cloud.ibm.com/media/docs/downloads/power-iaas/accelerated_migration.pdf

PowerVS Aspera server Terraform Automation

- https://github.com/IBM/power-aspera-server

IBM Aspera is a suite of data transfer software for moving amounts of data quickly and securely, regardless of network conditions.

Aspera on Cloud is IBM's on-demand SaaS offering for global content transfer and exchange. The service enables organizations to move large files and data sets – securely and reliably – across on-premises and multi-cloud environments at unrivaled speed.

Using Aspera on Cloud, organizations can store and readily access files and folders in multiple cloud-based and on-premises storage systems. Sharing among users is as easy as browsing or dragging-and-dropping – regardless of where the files are located – freeing collaboration from traditional boundaries among colleagues in both local and remote locations.

Aspera on Cloud uses IBM Aspera's FASP protocol, which overcomes the limitations of other file-transfer technologies. By moving large data sets at maximum speed, reliably and securely – regardless of network conditions, physical distance between sites, and file size, type, or number – Aspera technology enables a new world of collaboration, sharing, and content delivery.