Business Continuity with IBM Db2 HADR & Linux Pacemaker on Lenovo ThinkSystem

Last update: 17 September 2025

Version 1.0

Technical overview of Db2 HADR on Lenovo ThinkSystem

Explores the use cases of IBM Db2 HA in a mission critical landscape

Explains the best practices deploying IBM Db2 HADR

Describes advantages of using IBM Db2 HADR

Vijayakumar Kulageri
Mahendra Alhat

Table of Contents

1 Introduction

As enterprises strive to achieve the highest possible data availability, service availability & application availability, IBM® Db2® for Linux provides various built-in high availability features. Db2 further provides high availability solutions by using enterprise system resources with broad support for clustering software, such as Linux® Pacemaker.

This document describes the Db2 high availability functions and features, focusing on High Availability and Disaster Recovery (HADR) in the OLTP environment. The document provides a detailed description of HADR, including setup, configuration, administration, monitoring, and preferred practices of deploying HADR on a enterprise grade Lenovo ThinkSystem infrastructure stack.

This document explains how to configure cluster software Pacemaker with Db2 and show how to use pacemaker to automate HADR takeover in an SAP landscape scenario. Db2 also provides unprecedented enterprise-class disaster recovery capability. This edition of the Lenovo Press document covers single system High Availability in detail. The document also explains the best practices to run a database best suited for SAP workloads in attaining the best possible performance results.

This document is intended for database administrators and information management professionals who want to design, implement, and support a highly available application using the Db2 database.

2 Authors

Vijayakumar Kulageri is a Principal Solutions Architect & also Heads the SAP Solutions business for Lenovo Infrastructure Solutions Group Asia Pacific & Japan. Vijay with 20+ years of experience in the SAP ecosystem is mainly specialized on design and deployment of complex business continuity solutions. Currently Vijay wears a dual hat within Lenovo where he leads the Lenovo ISG APJ region for SAP Solutions business and is also an active member of Lenovo's Worldwide Centre of Competence (CoC) for SAP. Vijay is based in Bengaluru, India.

Mahendra Alhat is an Advisory Engineer, a seasoned technology professional with 18+ years of experience in designing, implementing, and optimizing cutting-edge data centre solutions. As an Advisory Engineer and Technology Consultant, he specializes in Hyperconverged Infrastructure (HCI), SAP HANA, virtualization, and cloud-integrated platforms, helping enterprises achieve agility, scalability, and operational efficiency. As part of the Professional Services team, Mahendra collaborates with global clients to deliver end-to-end consultancy—from strategic planning and proof-of-concept (PoC) to large-scale deployments and performance tuning. His deep technical acumen, vendor-agnostic approach, and commitment to best practices make him a trusted advisor for organizations modernizing their IT infrastructure. Mahendra is based in Mumbai, India.

3 Why High Availability & Disaster Recovery

Definition: High Availability

Refers to the ability of a database system to remain operational and accessible with minimal downtime, even in the event of hardware or software failures.

Key Features:

- **Redundancy:** Multiple database instances or nodes.
- **Failover Mechanisms:** Automatic switching to a standby system in case of failure.
- **Load Balancing:** Distributes traffic across multiple servers to prevent overload.
- **Clustering:** Grouping of servers to act as a single system.

Goal:
Ensure continuous availability and minimize downtime.

Definition: Disaster Recovery

It is a strategy to restore database services and data access after a catastrophic event such as natural disasters, cyberattacks, or major system failures.

Key Features:

- **Backup and Restore:** Regular backups stored offsite or in the cloud.
- **Replication:** Data copied to a geographically distant location.
- **Recovery Time Objective (RTO):** Maximum acceptable downtime.
- **Recovery Point Objective (RPO):** Maximum acceptable data loss.

Goal:
Recover data and resume operations after a major disruption.

Difference Between HA and DR

Feature	**High Availability (HA)**	**Disaster Recovery (DR)**
Purpose	Minimize downtime during minor failures	Recover from major disasters
Scope	Local or regional	Often cross-region or cross-country
Downtime Tolerance	Seconds to minutes	Minutes to hours (or more)
Data Loss Tolerance	Near-zero (synchronous replication)	Some loss acceptable (based on RPO requirements)

Feature	High Availability (HA)	Disaster Recovery (DR)
Implementation	Clustering, failover, load balancing	Backups, replication, DR sites
Cost	Higher (due to real-time redundancy)	Variable (depends on backup and DR strategy)

4 High Availability solutions with Db2

4.1 High Availability Disaster Recovery (HADR)

The High Availability Disaster Recovery (HADR) feature provides a high availability solution for both partial and complete site failures. In an HADR environment, log data is shipped continuously from a primary database to one or more standby databases and reapplied to the standby databases. When the primary database fails, applications are redirected to a standby database that automatically takes over the role of the primary database.

A partial site failure can be caused by a hardware, network, or software (Db2 or operating system) failure. Without HADR, a partial site failure requires restarting the database management system server that contains the database. The length of time that it takes to restart the database and the server where it is located is unpredictable. It can take several minutes before the database is brought back to a consistent state and made available.

A complete site failure can occur when a disaster, such as a fire, causes the entire site to be destroyed. However, because HADR uses TCP/IP for communication between the primary and standby databases, they can be situated in different locations. For example, the primary database might be at your head office in one city, and a standby database might be at your sales office in another city. If a disaster occurs at the primary site, data availability is maintained by having the remote standby database take over as the primary database with full Db2 functionality. After a takeover operation occurs, you can bring the original primary database back up and return it to its primary database status; this procedure is known as *failback*. You can initiate a failback if you can make the old primary database consistent with the new primary database. After you reintegrate the old primary database into the HADR setup as a standby database, you can switch the roles of the databases. This operation would enable the original primary database to be the primary database again.

With HADR, you base the level of protection from potential loss of data on your configuration and topology choices. Some of the key choices that you must make are as follows:

What level of synchronization will you use?

Standby databases are synchronized with the primary database through log data that is generated on the primary and sent to the standby databases. The standby databases constantly roll forward through the logs. You can choose from four different synchronization modes. In order of most to least protection, these modes are SYNC, NEARSYNC, ASYNC, and SUPERASYNC.

Will you use a peer window?

The HADR Peer Window is a configurable time period during which the primary and standby databases in an HADR setup are allowed to have a slight divergence in their log positions (measured in log sequence numbers - LSNs) and still be considered "in sync" or in the PEER state.

When the standby is within this window, the primary database treats it as a synchronous peer, even though the replication is technically operating in asynchronous mode. This is the key to understanding its value.

How many standby databases will you deploy?

With HADR, you can use up to three standby databases. With multiple standby databases, you can achieve both your high availability and disaster recovery objectives with a single technology. For more information, see HADR multiple standby databases.

Reads on standby

You can use the reads on standby feature to direct read-only workload to one or more standby databases without affecting the HA or DR responsibility of the standby. This feature can help reduce the workload on the primary without affecting the main responsibility of the standby.

Unless you have reads on standby enabled, applications can access the current primary database only. If you have reads on standby enabled, read-only applications can be redirected to the standby. Applications connecting to the standby database do not affect the availability of the standby if a failover occurs.

Delayed replay

You can use delayed replay to specify that a standby database is to remain at an earlier point in time than the primary, in terms of log replay. If data is lost or corrupted on the primary, you can recover this data on the time delayed standby.

Rolling updates and upgrades

Using an HADR setup, you can make various types of upgrades and Db2 fix pack updates to your databases without an outage. If you are using multiple standby databases, you can perform an upgrade while keeping the protection provided by HADR. For more information, see Applying rolling updates to a Db2 high availability disaster recovery (HADR) environment. Table 1 below contains an overview of which HADR functionality is supported by each type of HADR setup.

Table 1 : HADR Functionality support

Functionality or feature	Principal standby	Auxiliary standby
Synchronization mode	All modes are supported	SUPERASYNC mode only
Reads on standby	Supported	Supported
Delayed replay	Supported	Supported
Log spooling	Supported	Supported
Pacemaker as cluster manager for automated failover to HADR standby	Supported	Not supported
Peer window	Supported	Not supported
Network address translation (NAT)	Supported	Supported
Automatic client reroute (ACR)	Supported	Supported
Client affinities	N/A	N/A

With HADR, a standby database can take over in seconds. Further, you can also configure an automated failover of client (application) connections by incorporating cluster solutions like Linux Pacemaker.

4.2 Db2 high availability (HA) feature

The Db2 high availability (HA) feature enables the integration between Db2 and a cluster managing software.

The idea of clustering is to present to the users a single machine, when in fact the system has multiple nodes to serve client applications. Many clusters act in an active/passive configuration where only one node performs work, with the other nodes that are standing by as the backup if there is a failure. Some cluster solutions are sophisticated enough to allow load balancing between the nodes in an active/active configuration, thus maximizing the performance of the applications, and providing more cost-effective use of the resources.

When you stop a database manager instance in a clustered environment, you must make your cluster manager aware that the instance is stopped. The Db2 High Availability Feature provides a communication route for enabling the database manager to communicate with your cluster manager when the instance configuration changes, such as stopping a database manager instance, which require changes to the cluster.

The Db2 High Availability using the Pacemaker feature is composed of the following elements:

⇨ Pacemaker is an open-source, high availability cluster manager software integrated with Db2® Advanced Edition and Db2 Standard Edition on Linux®. It provides high availability and disaster recovery capabilities for on-premises deployments and non-container cloud environments.

⇨ In Db2 11.5.5 and later, HADR automation is supported with Pacemaker as the cluster manager. This includes supporting automation for multiple HADR databases per instance and supporting automation of multiple HADR instances all within one common Pacemaker/Corosync cluster.

As depicted in Figure 1, an automated failover in an IBM Db2 database configuration is achieved by using Linux pacemaker clustering software. In this section we consider only a two-node high availability configuration, where in a primary node/secondary (HA) node are hosting IBM Db2 database with right set of Db2 database configurations, which will be discussed in the next sections.

After the database is installed & configured on both physical server nodes, the IBM Db2 HADR feature is setup between the two database nodes which initiates a synchronous data replication between the primary & the secondary database. Once the database instances are in full sync mode, Linux pacemaker clustering software is configured on both database nodes, where in an active-passive cluster configuration is created (Primary node – Actively serving the application & Standby node – Passively waiting & just receiving the data replication from primary node). Next step is to add the cluster resources which are to be managed by the pacemaker cluster, namely a) Network resource 2) Db2 database resources 3) Quorum devices.

The above configuration results in a two-node automated IBM Db2 database cluster which offers the highest level of data & service availability to business-critical applications by automatically initiating the DB services failover to the secondary node in case of a primary node issue which can be ranging from operating system issue or database services themselves. The applications connecting to the database servers are always routed through the VIP (Virtual IP) which is a cluster managed resource, which makes sure that the VIP is always directing the network packets to the current active node.

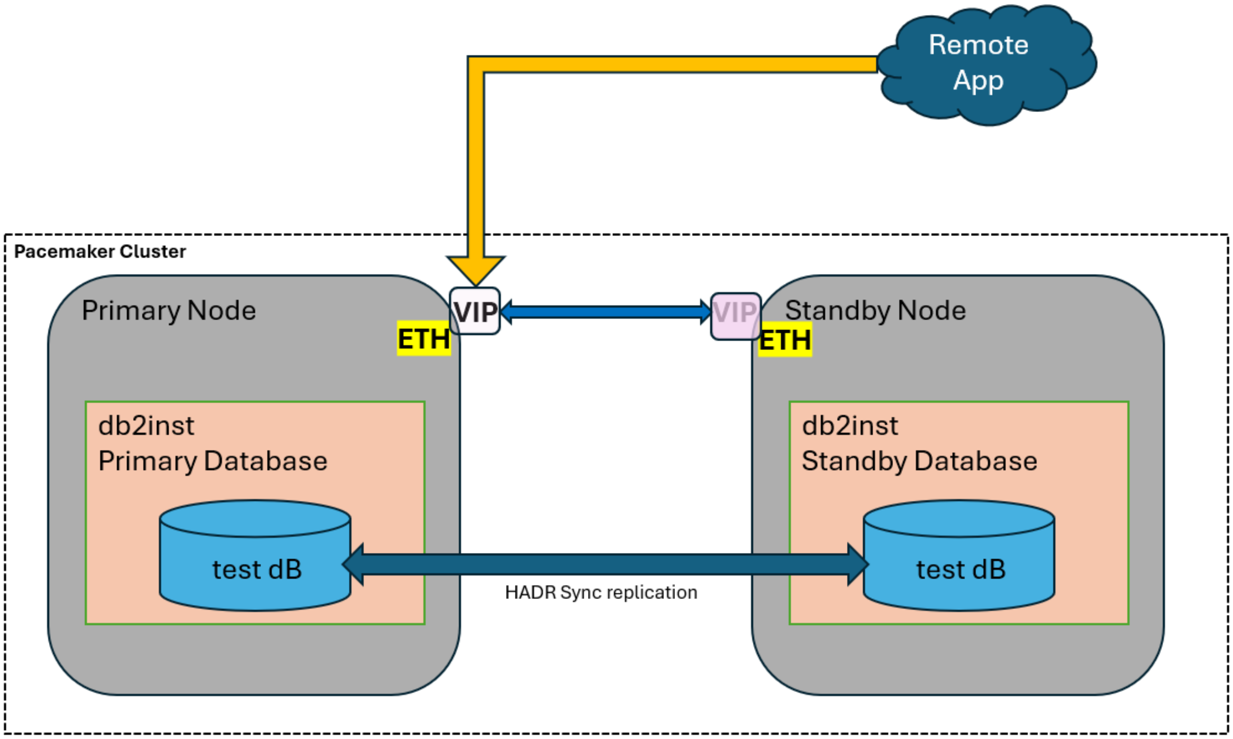


Figure 1 : IBM Db2 HADR with Pacemaker

5 Prerequisites for an integrated solution using pacemaker

Before you can integrate Pacemaker as a cluster management solution, hardware and software prerequisites need to be addressed.

Important Note: *In Db2 11.5.8 and later, Mutual Failover high availability is supported when using Pacemaker as the integrated cluster manager. In Db2 11.5.6 and later, the Pacemaker cluster manager for automated fail-over to HADR standby databases is packaged and installed with Db2.*
In Db2 11.5.5, Pacemaker is included and available for production environments. In Db2 11.5.4, Pacemaker is included as a technology preview only, for development, testing, and proof-of-concept environments.

5.1 Db2 high availability (HA) feature

The integrated Pacemaker high availability (HA) solution is available on the following Linux® distributions:
Intel Linux for Db2 11.5.8 and future fix packs in the same release:

  - Red Hat® Enterprise Linux (RHEL) 8.4 and up
  - SuSE Linux Enterprise Server (SLES) 15 SP3 and up

- For Db2 11.5.7 and future fix packs in the same release:
  - Red Hat Enterprise Linux (RHEL) 8.1 and up
  - SuSE Linux Enterprise Server (SLES) 15 SP1 and up
- For Db2 11.5.6, the level must be one of the following:
  - Red Hat Enterprise Linux (RHEL) 8.1 and 8.2
  - SuSE Linux Enterprise Server (SLES) 15 SP1 and SP2
- For Db2 11.5.4 and 11.5.5, the level must be one of the following:
  - Red Hat Enterprise Linux (RHEL) 8.1
  - SuSE Linux Enterprise Server (SLES) 15 SP1

5.2 Cluster Software

Db2 supports Pacemaker as its integrated cluster manager solution only where the Pacemaker software stack being used is supplied by Db2 directly. This corresponds to a specific Db2 release and is configured entirely using the new Db2cm utility or as instructed by the Db2 Support. For support of the Pacemaker software from Db2, it is required that the configuration provided as well as the Pacemaker software stack remain unchanged.

For version 11.5.5 and version 11.5.4, the Pacemaker version supported by Db2 must be downloaded from this public IBM® website: Db2 Automated HADR with Pacemaker (Need IBM id). There are specific compressed tar files available for each Linux distribution and architecture.

For version 11.5.6 and later releases, the Pacemaker software is included in the Db2 Install image. On-premises deployments do not require any additional downloads and additional packages using Pacemaker, such as the Booth Cluster Ticket Manager, which are not supported.

5.3 QDevice quorum mechanism

This is the recommended quorum mechanism for a production system. It requires a third host to install the corosync-qnetd software on and to act as the arbitrator. The host itself is not required to be part of the cluster and does not require the Db2 server to be installed.

Disk space required on HADR nodes: 10MB (in addition to corosync)

Qnetd server host minimum requirements:

- 2 vCPU
- 8 GB memory
- 10 MB of free disk space + 2 MB per additional cluster configured to use this host as a QDevice.

Other requirements:

- The host used must be accessible via TCP/IP to the other two hosts in the cluster.
- The cluster hosts must be able to communicate with the QDevice host by using the IP address that is specified in their /etc/hosts file.
- All clusters using the QNetd server must have unique cluster names.

5.4 Virtual IP address (VIP)

Virtual IP is often setup per HADR enabled database in Db2 HADR for the purpose of enabling automatic client to reroute when failover occurs. Configuration of VIP is described in the following sections.

5.5 Db2 high availability disaster recovery (HADR) configuration

- Ensure that HADR databases exist on different systems.
- Ensure that all HADR databases are started in their respective primary and standby database roles, and that all HADR primary-standby database pairs are in peer state.

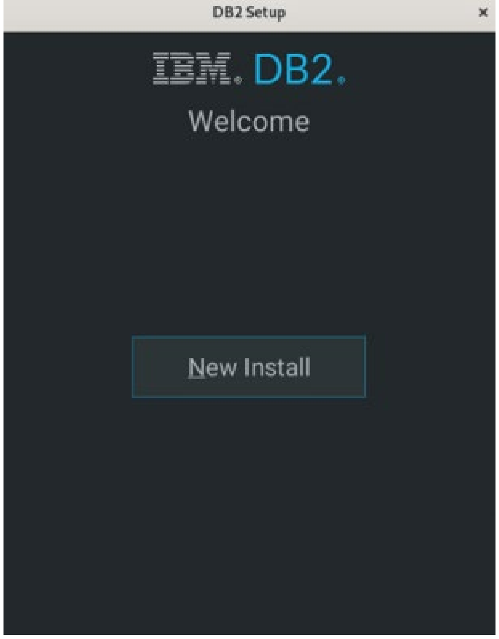
- Ensure that you are using either the SYNC HADR synchronization mode or NEARSYNC HADR synchronization mode.
- Set the **hadr_peer_window** configuration parameter to the recommended value of 120 seconds (60 seconds minimum) for all HADR databases.
- Disable the Db2 fault monitor.

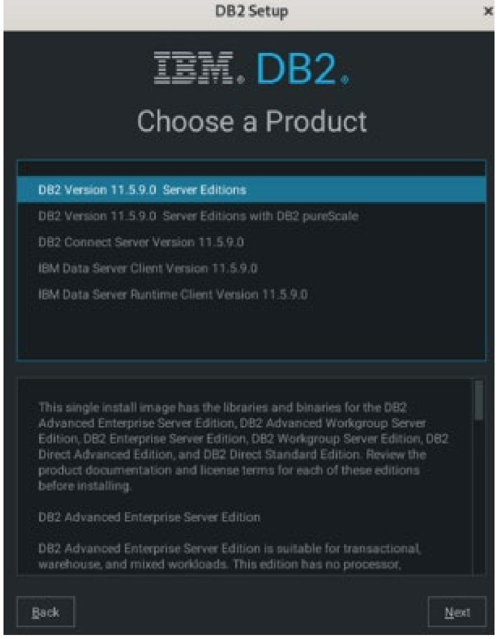
6 Installation & Configuration of Db2 for SAP Applications

***Important Note**:* We have used Db2 version 11.5 server edition for the following documentation. Also, for SAP specific workload scenarios it is highly recommended to use the SAP SWPM based installation of IBM Db2 database, which ensures the installation & configuration of critical parameters which are necessary for SAP applications accessing the IBM Db2 database. *Please refer to the official SAP Note 2751102 for standard parameter settings*

Post downloading the Db2 rpm from here, you can kickstart the installation of Db2 by running the following command :

server01:/ db2dump/server_dec # **./** db2**setup**





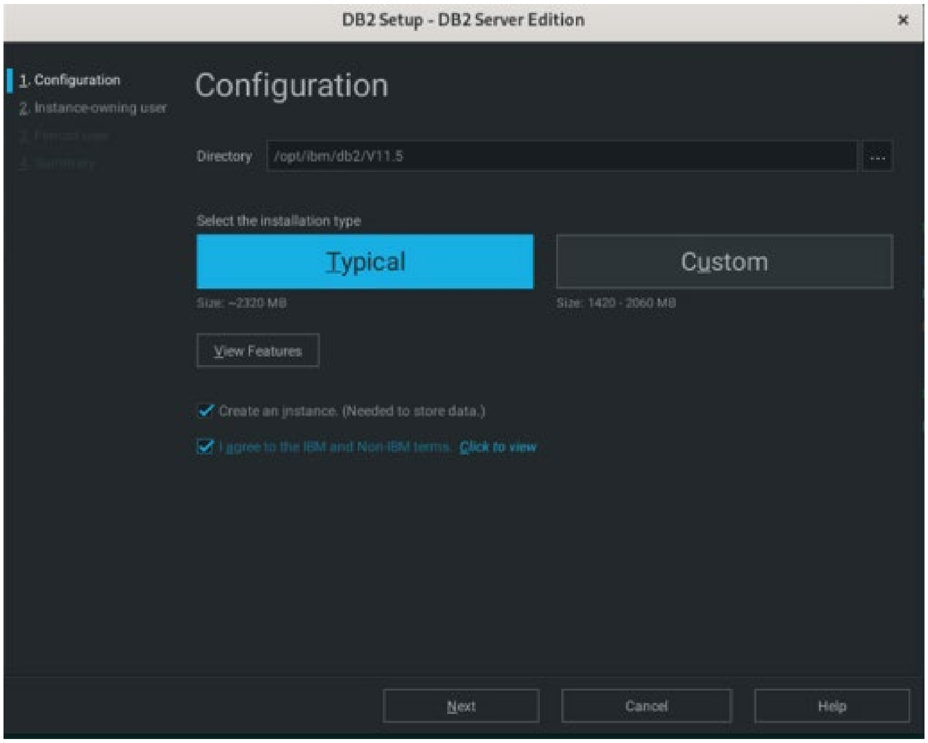
DB2 Version 11.5.9.0 Server Editions
DB2 Version 11.5.9.0 Server Editions with DB2 pureScale
DB2 Connect Server Version 11.5.9.0
IBM Data Server Client Version 11.5.9.0
IBM Data Server Runtime Client Version 11.5.9.0

This single install image has the libraries and binaries for the DB2 Advanced Enterprise Server Edition, DB2 Advanced Workgroup Server Edition, DB2 Enterprise Server Edition, DB2 Workgroup Server Edition, DB2 Direct Advanced Edition, and DB2 Direct Standard Edition. Review the product documentation and license terms for each of these editions before installing.

DB2 Advanced Enterprise Server Edition

DB2 Advanced Enterprise Server Edition is suitable for transactional, warehouse, and mixed workloads. This edition has no processor,

Post selection of the right Db2 version for the installation, you need to select a directory on the node which will be the Db2 installation directory.



Configuration

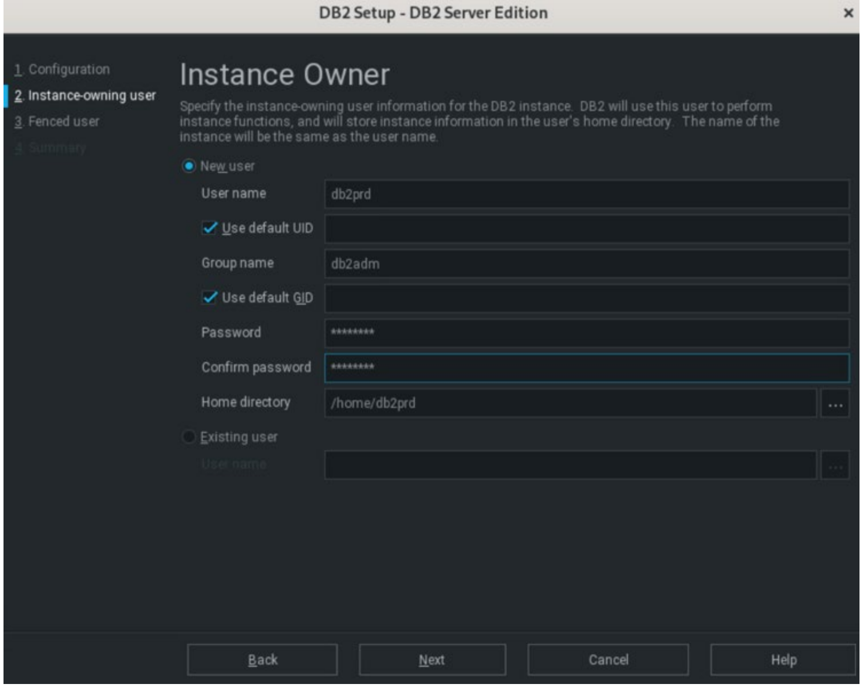
Directory: /opt/ibm/db2/V11.5

Select the installation type

☑ Create an instance. (Needed to store data.)

☑ I agree to the IBM and Non-IBM terms. *Click to view*

Next, enter the Db2 instance, instance owner & the Db2 home directory



Instance Owner

Specify the instance-owning user information for the DB2 instance. DB2 will use this user to perform instance functions, and will store instance information in the user's home directory. The name of the instance will be the same as the user name.

● New user

User name: db2prd

☑ Use default UID

Group name: db2adm

☑ Use default GID

Password: ********

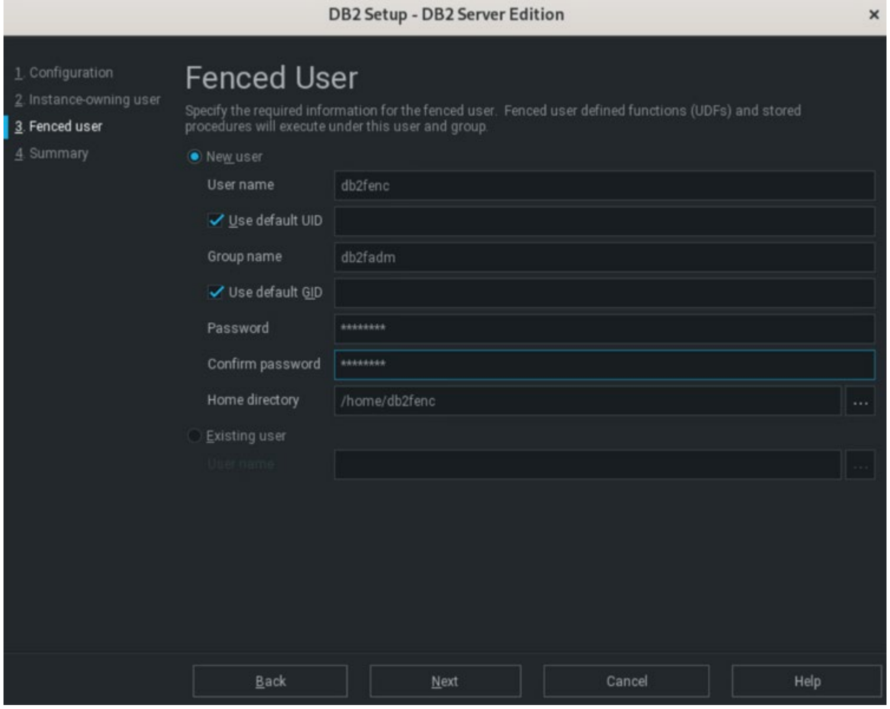
Confirm password: ********

Home directory: /home/db2prd

○ Existing user

User name:

For advanced security purposes, IBM recommends creating a separate fenced user for UDF's which can run outside of the core database address space.



Fenced User

Specify the required information for the fenced user. Fenced user defined functions (UDFs) and stored procedures will execute under this user and group.

● New user

User name: db2fenc

☑ Use default UID

Group name: db2fadm

☑ Use default GID

Password: ********

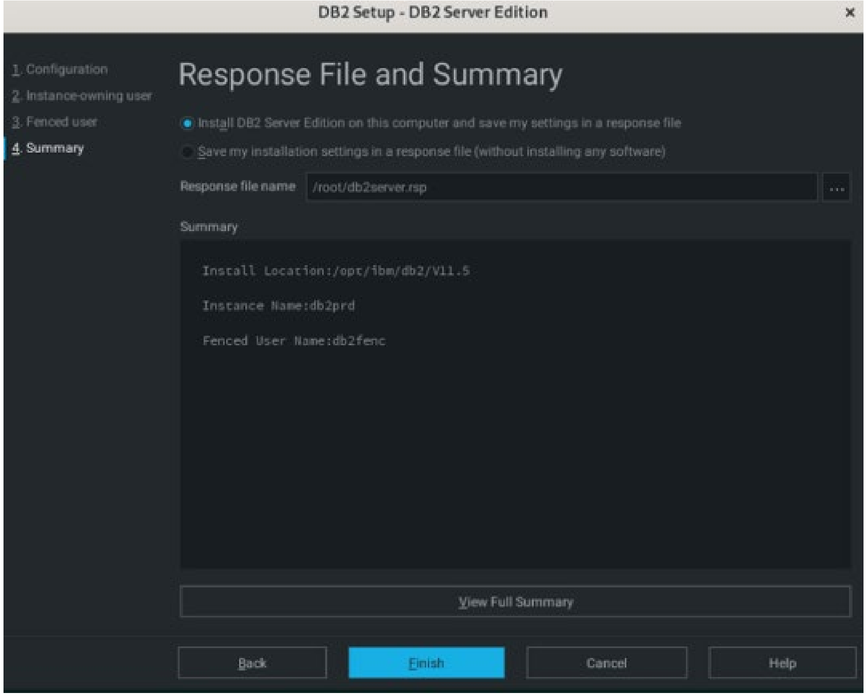
Confirm password: ********

Home directory: /home/db2fenc

○ Existing user

User name

In Db2, a response file is an English-only text file that contains setup and configuration information, allowing for unattended or silent installation and uninstallation of Db2 products and features. It specifies configuration parameters and the products/components to install or uninstall, enabling automation and consistency across multiple installations or systems.



Response File and Summary

● Install DB2 Server Edition on this computer and save my settings in a response file

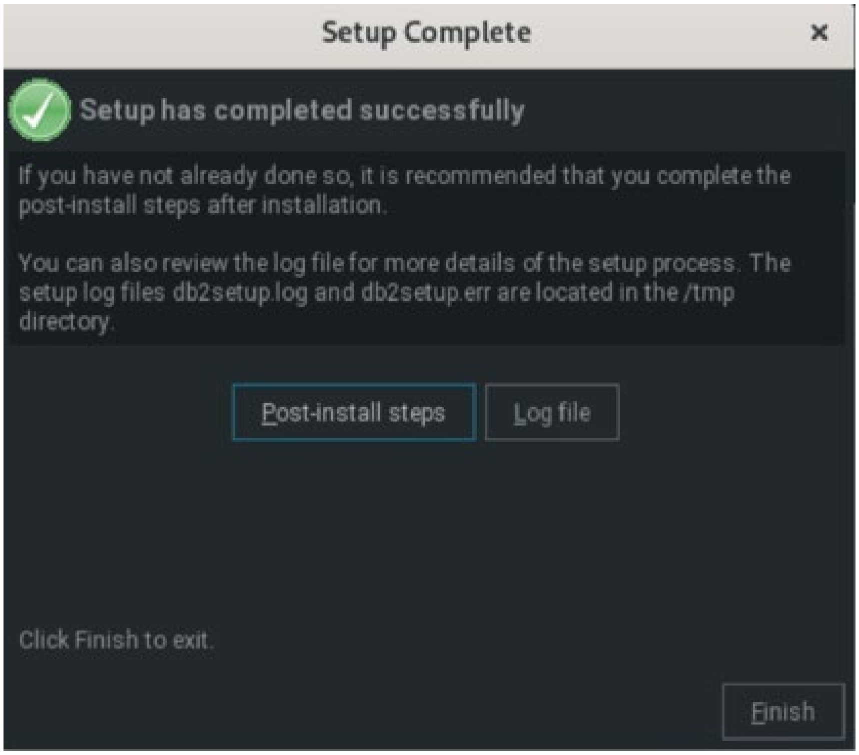
○ Save my installation settings in a response file (without installing any software)

Response file name: /root/db2server.rsp

Summary

```
Install Location:/opt/ibm/db2/V11.5

Instance Name:db2prd

Fenced User Name:db2fenc
```

This should install a Db2 instance successfully. At the completion step we recommend going through the post install steps to further optimize the installation & to understand the best practices of managing a Db2 database.


Setup Complete

Setup has completed successfully

If you have not already done so, it is recommended that you complete the post-install steps after installation.

You can also review the log file for more details of the setup process. The setup log files db2setup.log and db2setup.err are located in the /tmp directory.

Click Finish to exit.

Important Note: In this test, since we are using IBM Db2 HADR it is necessary to follow the above steps on the secondary node to complete a successful Db2 database instance installed with the exact same parameters as used in the primary node.
Please note: the database will be created on the primary node, backed up & then restored on the secondary node.

6.1 SAP Application specific parameter settings on Db2

Please refer to the following mandatory parameter settings requirements in an SAP application scenario on a Db2 database, this would only be needed if the Db2 database is not installed using the SAP SWPM.

If you are running a Db2 database server for an ISV application such as SAP, some best practice guidelines that take into account the specific application might be available. The most straightforward mechanism is the Db2 registry variable DB2_WORKLOAD, which can be set to a value that enables aggregated registry variables to be optimized for specific environments and workloads. Mandatory setting for DB2_WORKLOAD in an SAP application scenario is “DB_WORKLOAD=SAP”

Note: If there are non SAP ISV applications accessing DB2, then the following values are also valid, DB2_WORKLOAD = 1C, CM, COGNOS_CS, FILENET_CM, MAXIMO, MDM, TPM, WAS, WC, and WP .

For many ISV applications, such as SAP Business One, the AUTOCONFIGURE command can be successfully used to define the initial configuration. However, it should not be used in SAP NetWeaver installations, because an initial set of Db2 configuration parameters is applied during SAP installation.

Pay special attention to SAP applications when using partitioned database environments. SAP uses partitioned database environment mainly in its SAP NetWeaver Business Intelligence (Business Warehouse) product. The suggested layout has the Db2 system catalog, the dimension and master tables, plus the SAP base tables on Partition 0. This leads to a different workload on this partition compared to other partitioned database environments. Because the SAP application server runs on this partition, up to eight processors might be assigned to just this partition. As the SAP BW workload becomes more highly parallelized, with many short queries running concurrently, the number of partitions for SAP BI is typically smaller than for other applications. In other words, more than one CPU per data partition is required.

Please also refer to the following SAP notes for detailed explanation on the Db2 parameter settings in an SAP environment.

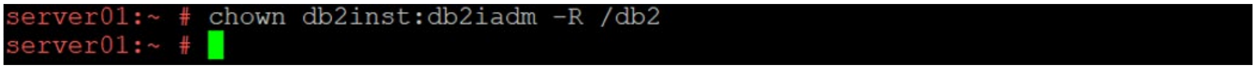
2751102 - DB6: Db2 11.5 Standard Parameter Settings
3520953 - DB6: Using Db2 12.1 with SAP Applications
2751085 - DB6: Using Db2 11.5 with SAP Applications
101809 - DB6: Supported Db2 Versions and Fix Pack Levels

7 Configuration of Db2 HADR

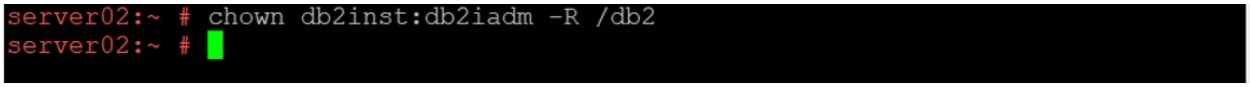
Once the Db2 database instance is successfully installed on both the primary & the secondary nodes, the next step is to configure a HADR replication between the nodes. Please follow the steps below to configure the same.

7.1 Configuration steps for HADR

Assign proper access privileges to the instance user/group the Db2 directory on both nodes



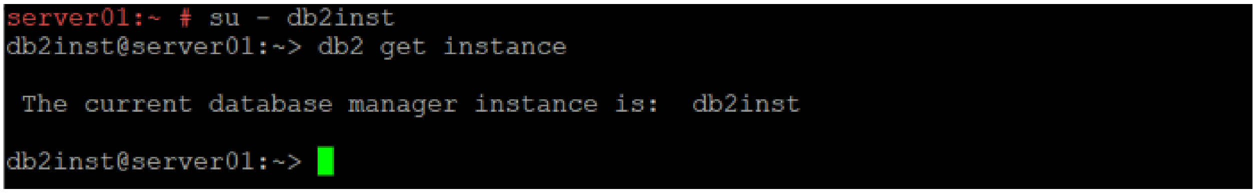
```
server01:~ # chown db2inst:db2iadm -R /db2
server01:~ # 
```



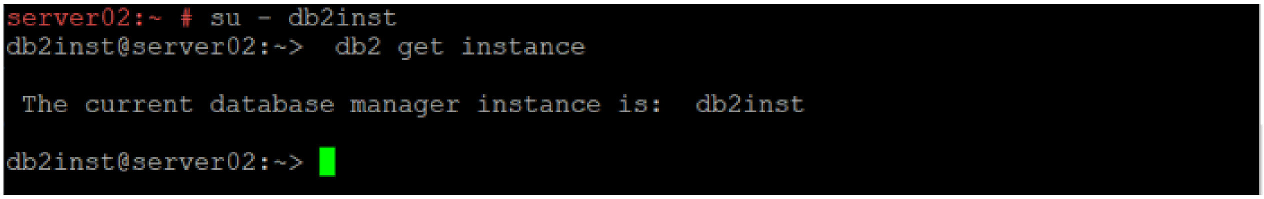
```
server02:~ # chown db2inst:db2iadm -R /db2
server02:~ # 
```

Check the Db2 database instance name created on the primary node & secondary during the installation

“Below commands will be performed with user owning Db2 instance”



```
server01:~ # su - db2inst
db2inst@server01:~> db2 get instance

 The current database manager instance is:  db2inst

db2inst@server01:~> 
```



```
server02:~ # su - db2inst
db2inst@server02:~>  db2 get instance

 The current database manager instance is:  db2inst

db2inst@server02:~> 
```

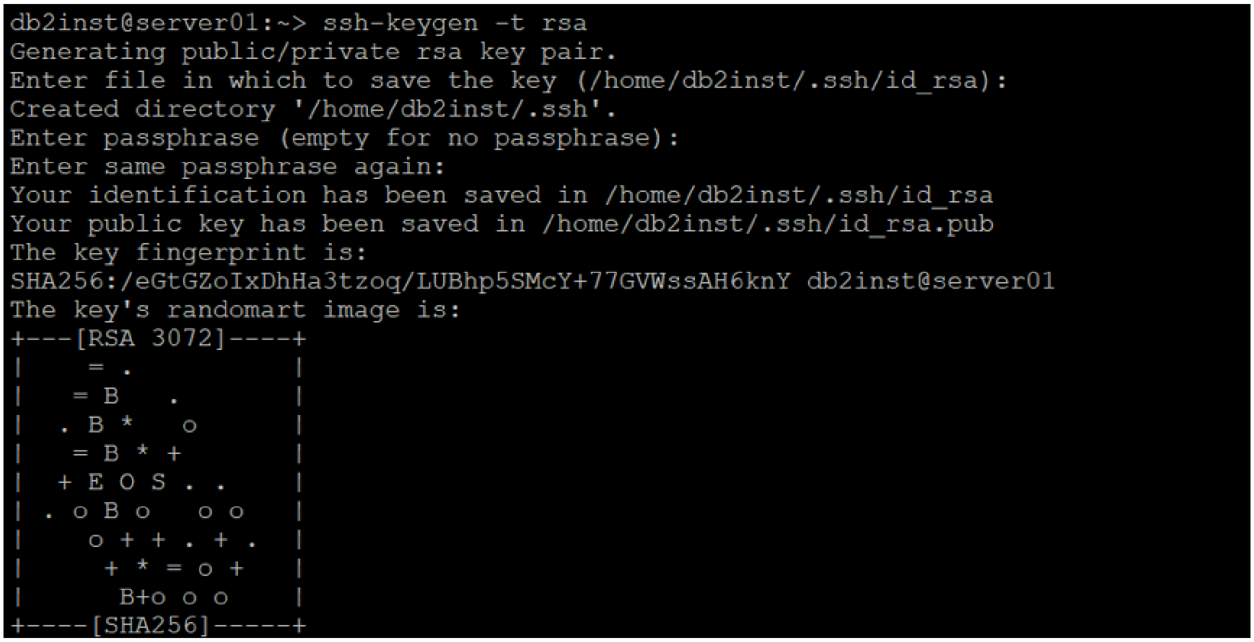
Set the password less ssh access for instance user between the primary & the secondary node

#ssh-keygen -t rsa

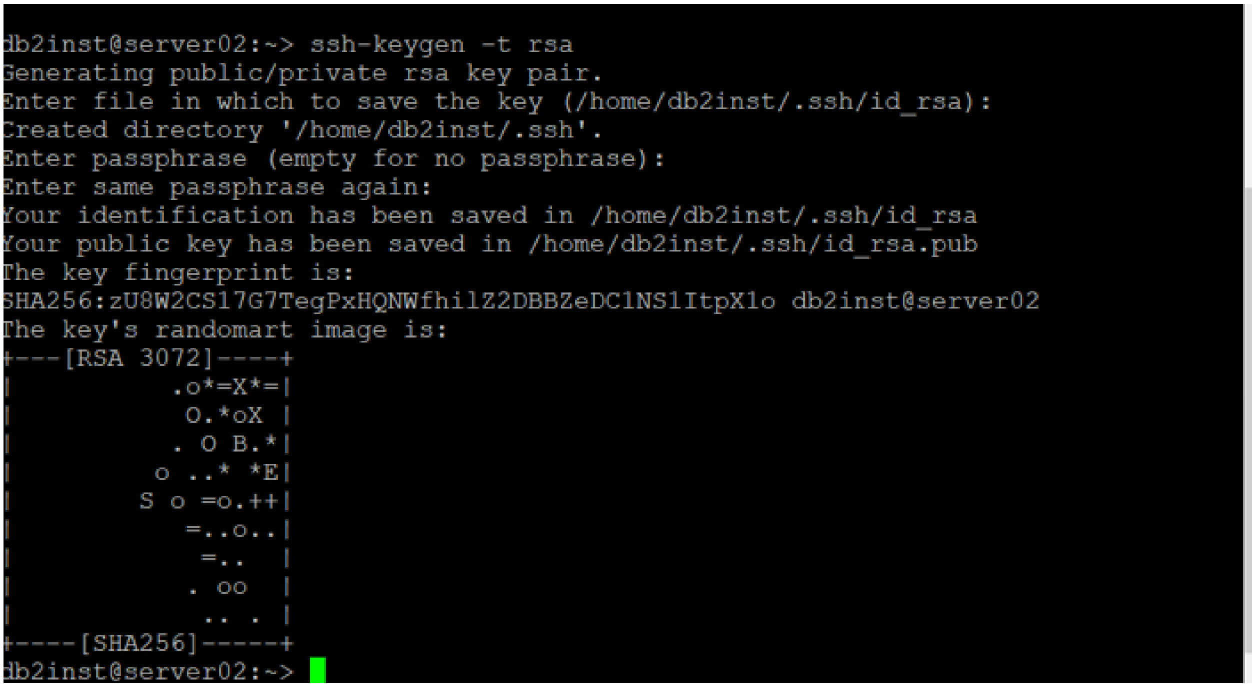
#ssh-copy-id server02

#ssh-copy-id server01

#ssh-copy-id quorumnode



```
db2inst@server01:~> ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/db2inst/.ssh/id_rsa):
Created directory '/home/db2inst/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/db2inst/.ssh/id_rsa
Your public key has been saved in /home/db2inst/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:/eGtGZoIxDhHa3tzoq/LUBhp5SMcY+77GVWssAH6knY db2inst@server01
The key's randomart image is:
+---[RSA 3072]----+
|    = .          |
|   = B   .       |
|  . B *   o      |
|   = B * +       |
|  + E O S . .    |
| . o B o   o o   |
|    o + + . + .  |
|     + * = o +   |
|      B+o o o    |
+----[SHA256]-----+
```



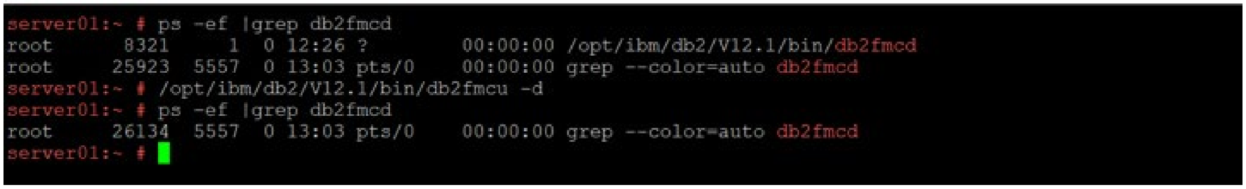
```
db2inst@server02:~> ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/db2inst/.ssh/id_rsa):
Created directory '/home/db2inst/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/db2inst/.ssh/id_rsa
Your public key has been saved in /home/db2inst/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:zU8W2CS17G7TegPxHQNWfhilZ2DBBZeDC1NS1ItpX1o db2inst@server02
The key's randomart image is:
+---[RSA 3072]----+
|          .o*=X*=|
|           O.*oX |
|          . O B.*|
|         o ..* *E|
|        S o =o.++|
|           =..o..|
|            =..  |
|           . oo  |
|            .. . |
+----[SHA256]-----+
db2inst@server02:~> 
```

Disable the db2fmcd monitor on both nodes

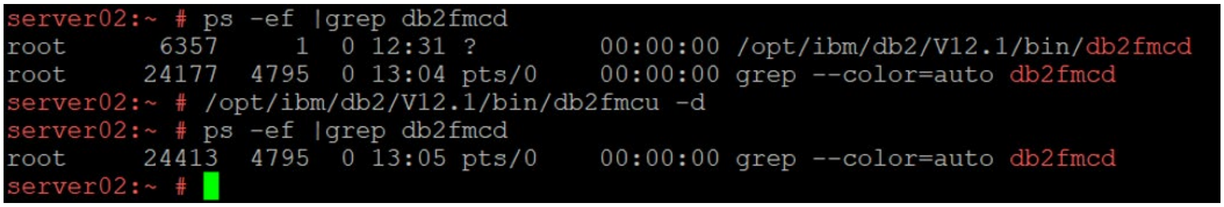
#ps -ef |grep db2fmcd

#db2fmcu -d

#ps -ef |grep db2fmcd



```
server01:~ # ps -ef |grep db2fmcd
root      8321     1  0 12:26 ?        00:00:00 /opt/ibm/db2/V12.1/bin/db2fmcd
root     25923  5557  0 13:03 pts/0    00:00:00 grep --color=auto db2fmcd
server01:~ # /opt/ibm/db2/V12.1/bin/db2fmcu -d
server01:~ # ps -ef |grep db2fmcd
root     26134  5557  0 13:03 pts/0    00:00:00 grep --color=auto db2fmcd
server01:~ #
```

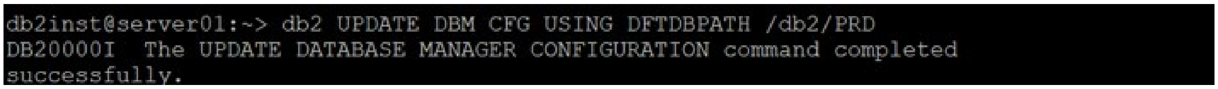


```
server02:~ # ps -ef |grep db2fmcd
root      6357     1  0 12:31 ?        00:00:00 /opt/ibm/db2/V12.1/bin/db2fmcd
root     24177  4795  0 13:04 pts/0    00:00:00 grep --color=auto db2fmcd
server02:~ # /opt/ibm/db2/V12.1/bin/db2fmcu -d
server02:~ # ps -ef |grep db2fmcd
root     24413  4795  0 13:05 pts/0    00:00:00 grep --color=auto db2fmcd
server02:~ #
```

Set the default database path for the database on the primary node

“Below commands will be performed with the user owning the Db2 instance

db2inst@server01:~> db2 UPDATE DBM CFG USING DFTDBPATH /db2/PRD



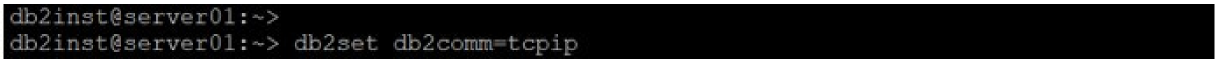
```
db2inst@server01:~> db2 UPDATE DBM CFG USING DFTDBPATH /db2/PRD
DB20000I  The UPDATE DATABASE MANAGER CONFIGURATION command completed
successfully.
```

Set default service name for the database on both nodes

db2inst@server01:~> db2 UPDATE DBM CFG USING SVCENAME sapdb2PRD

Set the database default communication via TCP-IP on the primary node

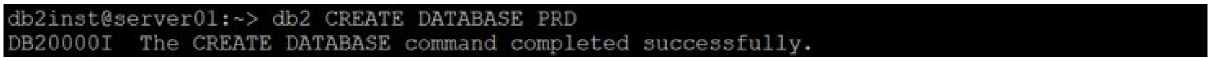
db2inst@server01:~> db2set db2comm=tcpip



```
db2inst@server01:~>
db2inst@server01:~> db2set db2comm=tcpip
```

Create the database on primary node. Note: Do not create the database on secondary node

db2inst@server01:~> db2 CREATE DATABASE PRD



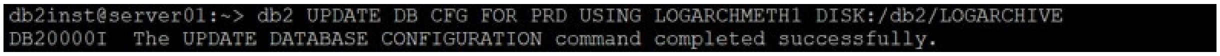
```
db2inst@server01:~> db2 CREATE DATABASE PRD
DB20000I  The CREATE DATABASE command completed successfully.
```

Following settings are mandatory to initiate HADR on a database. Run the following on primary node

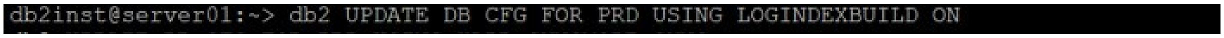
db2inst@server01:~> db2 UPDATE DB CFG FOR PRD USING LOGARCHMETH1 DISK:/db2/FEP/log_archive

db2inst@server01:~> db2 UPDATE DB CFG FOR PRD USING LOGINDEXBUILD ON

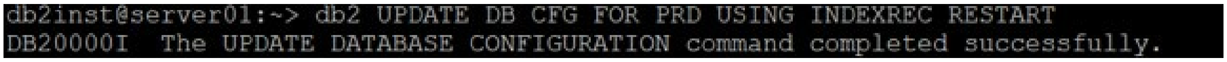
```
db2inst@server01:~>    db2 UPDATE DB CFG FOR PRD USING INDEXREC RESTART
```



```
db2inst@server01:~> db2 UPDATE DB CFG FOR PRD USING LOGARCHMETH1 DISK:/db2/LOGARCHIVE
DB20000I  The UPDATE DATABASE CONFIGURATION command completed successfully.
```



```
db2inst@server01:~> db2 UPDATE DB CFG FOR PRD USING LOGINDEXBUILD ON
```

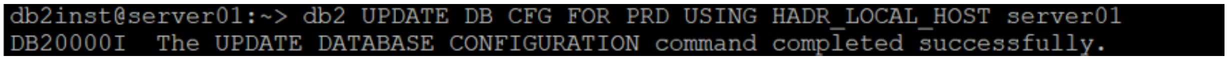


```
db2inst@server01:~> db2 UPDATE DB CFG FOR PRD USING INDEXREC RESTART
DB20000I  The UPDATE DATABASE CONFIGURATION command completed successfully.
```

Important Note: *Make sure of assigning proper directory/file read-write access to the instance owner (in this case "*db*2inst") to the directory where LOGARCHIVE files will be stored.*

Set the local host for HADR configuration as the primary node. In this test, server01 is the hostname of the primary node. Run it on the primary node

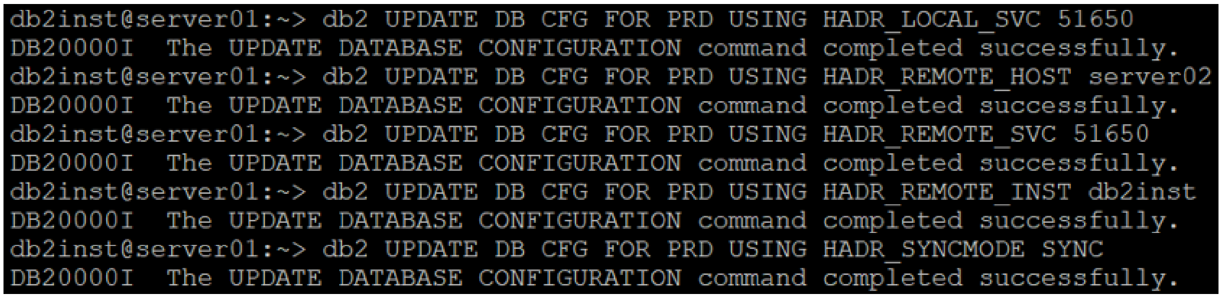
```
db2inst@server01:~>    db2 UPDATE DB CFG FOR PRD USING HADR_LOCAL_HOST server01
```



```
db2inst@server01:~> db2 UPDATE DB CFG FOR PRD USING HADR_LOCAL_HOST server01
DB20000I  The UPDATE DATABASE CONFIGURATION command completed successfully.
```

Set the local service name for HADR (by default it is "51650"), set the remote host entry in the HADR configuration which is nothing but the secondary node (in this test it is "server02"), select the same remote service name & select the remote instance which we recommend being same as the primary node (in this test it is " db2prd"). Set the replication mode as sync (It is essential to select the sync-mode or the replication mode to sync in a typical intra-dc HA setup to achieve the best possible RTO & RPO). Run the following on primary node. As an industry best practice, it is always recommended that the application teams plan, decide & specify the right RPO & RTO targets.

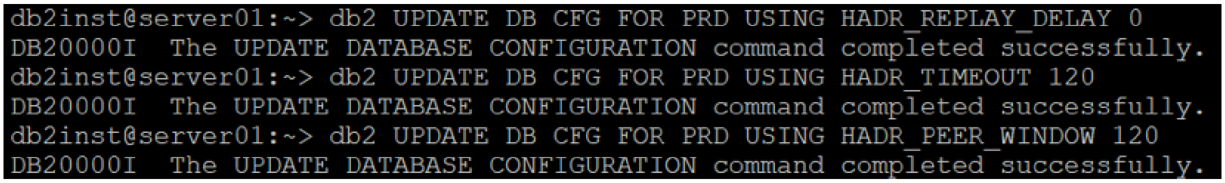
```
db2inst@server01:~>    db2 UPDATE DB CFG FOR PRD USING HADR_LOCAL_SVC 51650
db2inst@server01:~>    db2 UPDATE DB CFG FOR PRD USING HADR_REMOTE_HOST server02
db2inst@server01:~>    db2 UPDATE DB CFG FOR PRD USING HADR_REMOTE_SVC 51650
db2inst@server01:~>    db2 UPDATE DB CFG FOR PRD USING HADR_REMOTE_INST    db2PRD
db2inst@server01:~>    db2 UPDATE DB CFG FOR PRD USING HADR_SYNCMODE SYNC
```



```
db2inst@server01:~> db2 UPDATE DB CFG FOR PRD USING HADR_LOCAL_SVC 51650
DB20000I  The UPDATE DATABASE CONFIGURATION command completed successfully.
db2inst@server01:~> db2 UPDATE DB CFG FOR PRD USING HADR_REMOTE_HOST server02
DB20000I  The UPDATE DATABASE CONFIGURATION command completed successfully.
db2inst@server01:~> db2 UPDATE DB CFG FOR PRD USING HADR_REMOTE_SVC 51650
DB20000I  The UPDATE DATABASE CONFIGURATION command completed successfully.
db2inst@server01:~> db2 UPDATE DB CFG FOR PRD USING HADR_REMOTE_INST db2inst
DB20000I  The UPDATE DATABASE CONFIGURATION command completed successfully.
db2inst@server01:~> db2 UPDATE DB CFG FOR PRD USING HADR_SYNCMODE SYNC
DB20000I  The UPDATE DATABASE CONFIGURATION command completed successfully.
```

Set the HADR_REPLAY, TIMEOUT & PEER WINDOW values as per your typical environment or the values used in our test are proven to work in majority of the end user scenario’s. Run the following on the primary node.

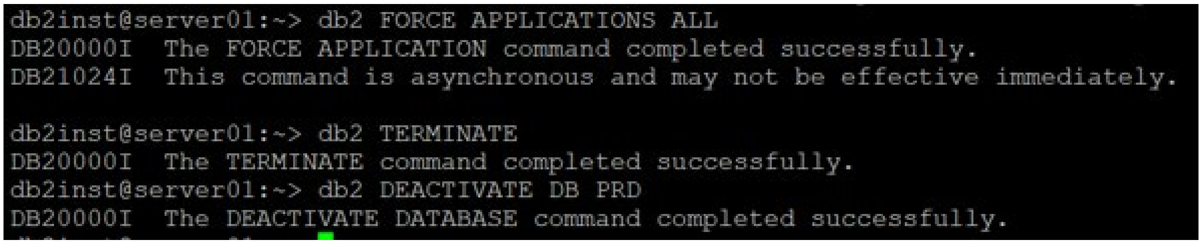
```
db2inst@server01:~>    db2 UPDATE DB CFG FOR PRD USING HADR_REPLAY_DELAY 0
db2inst@server01:~>    db2 UPDATE DB CFG FOR PRD USING HADR_TIMEOUT 120
db2inst@server01:~>    db2 UPDATE DB CFG FOR PRD USING HADR_PEER_WINDOW 120
```



```
db2inst@server01:~> db2 UPDATE DB CFG FOR PRD USING HADR_REPLAY_DELAY 0
DB20000I  The UPDATE DATABASE CONFIGURATION command completed successfully.
db2inst@server01:~> db2 UPDATE DB CFG FOR PRD USING HADR_TIMEOUT 120
DB20000I  The UPDATE DATABASE CONFIGURATION command completed successfully.
db2inst@server01:~> db2 UPDATE DB CFG FOR PRD USING HADR_PEER_WINDOW 120
DB20000I  The UPDATE DATABASE CONFIGURATION command completed successfully.
```

Terminate & deactivate the database on the primary node

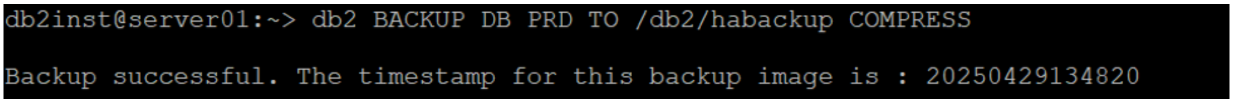
```
db2inst@server01:~>    db2 FORCE APPLICATIONS ALL
db2inst@server01:~>    db2 TERMINATE
db2inst@server01:~>    db2 DEACTIVATE DB PRD
```



```
db2inst@server01:~> db2 FORCE APPLICATIONS ALL
DB20000I  The FORCE APPLICATION command completed successfully.
DB21024I  This command is asynchronous and may not be effective immediately.

db2inst@server01:~> db2 TERMINATE
DB20000I  The TERMINATE command completed successfully.
db2inst@server01:~> db2 DEACTIVATE DB PRD
DB20000I  The DEACTIVATE DATABASE command completed successfully.
```

Backup the database on the primary node.

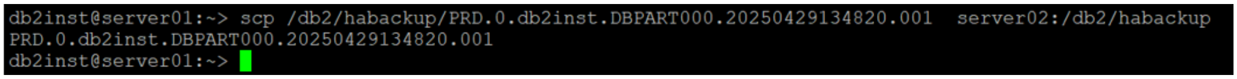
```
db2inst@server01:~>    db2 BACKUP DB PRD TO / db2/habackup COMPRESS
```



```
db2inst@server01:~> db2 BACKUP DB PRD TO /db2/habackup COMPRESS

Backup successful. The timestamp for this backup image is : 20250429134820
```

Copy the backup image from primary node to the secondary node.

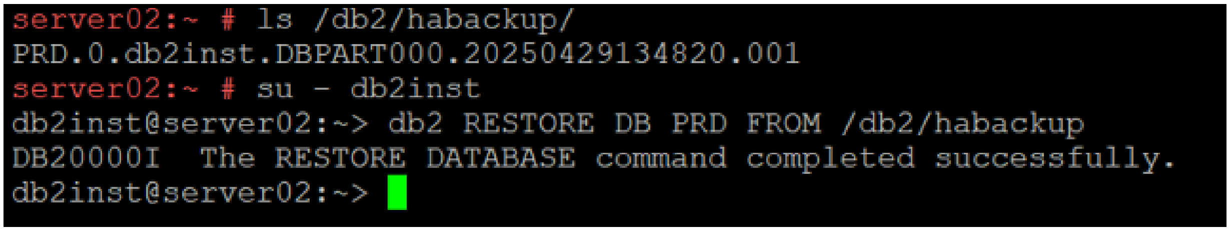
#scp / db2/habackup/PRD.x.xxx.xxx   server02:/ db2/habackup



```
db2inst@server01:~> scp /db2/habackup/PRD.0.db2inst.DBPART000.20250429134820.001  server02:/db2/habackup
PRD.0.db2inst.DBPART000.20250429134820.001
db2inst@server01:~>
```

Restore the copied database backup image in the secondary node.

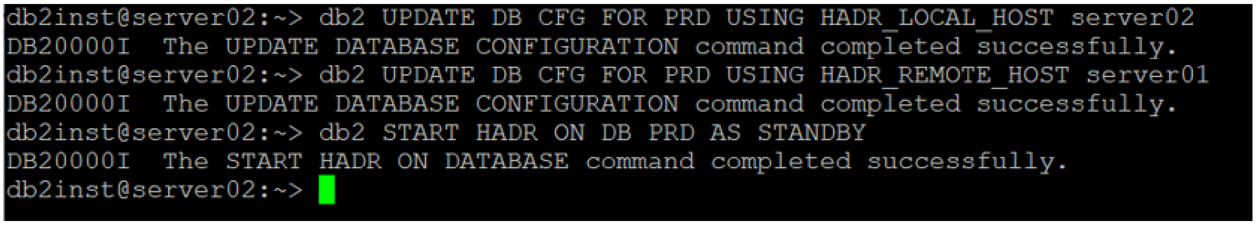
Check file on second server and switch to   Db2 owner user shell.



```
server02:~ # ls /db2/habackup/
PRD.0.db2inst.DBPART000.20250429134820.001
server02:~ # su - db2inst
db2inst@server02:~> db2 RESTORE DB PRD FROM /db2/habackup
DB20000I  The RESTORE DATABASE command completed successfully.
db2inst@server02:~>
```

Set the local host & remote host parameters of HADR on the secondary node & start the HADR on the secondary node as **STANDBY**

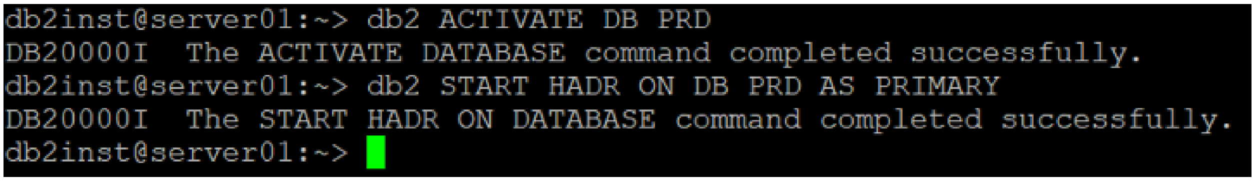
```
db2inst@server02:~>   db2 UPDATE DB CFG FOR PRD USING HADR_LOCAL_HOST server02

db2inst@server02:~>   db2 UPDATE DB CFG FOR PRD USING HADR_REMOTE_HOST server01

db2inst@server02:~>   db2 START HADR ON DB PRD AS STANDBY
```



```
db2inst@server02:~> db2 UPDATE DB CFG FOR PRD USING HADR_LOCAL_HOST server02
DB20000I  The UPDATE DATABASE CONFIGURATION command completed successfully.
db2inst@server02:~> db2 UPDATE DB CFG FOR PRD USING HADR_REMOTE_HOST server01
DB20000I  The UPDATE DATABASE CONFIGURATION command completed successfully.
db2inst@server02:~> db2 START HADR ON DB PRD AS STANDBY
DB20000I  The START HADR ON DATABASE command completed successfully.
db2inst@server02:~>
```

Activate the database & the start HADR on the primary node as **PRIMARY**

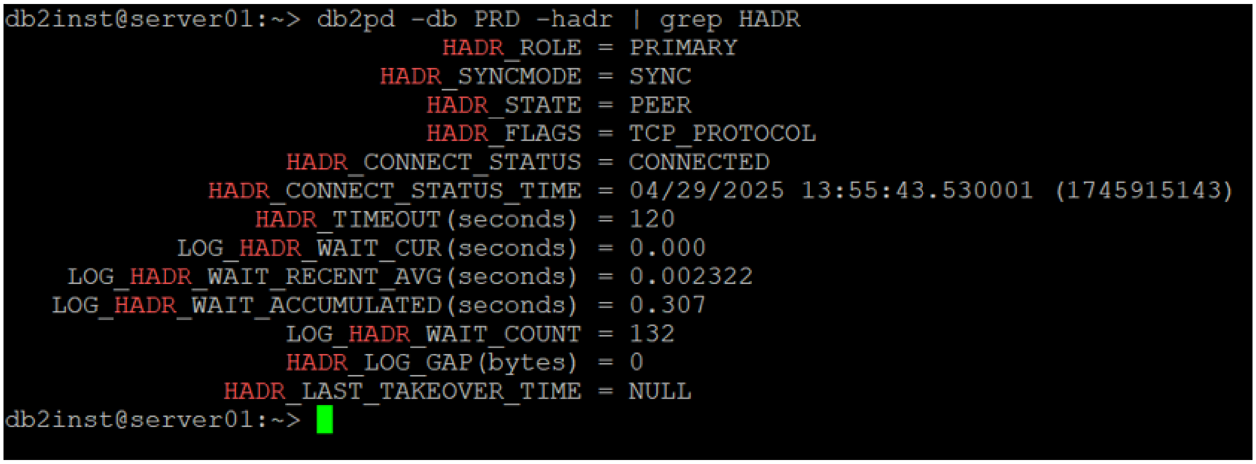
```
db2inst@server01:~>   db2 ACTIVATE DB PRD

db2inst@server01:~>   db2 START HADR ON DB PRD AS PRIMARY
```



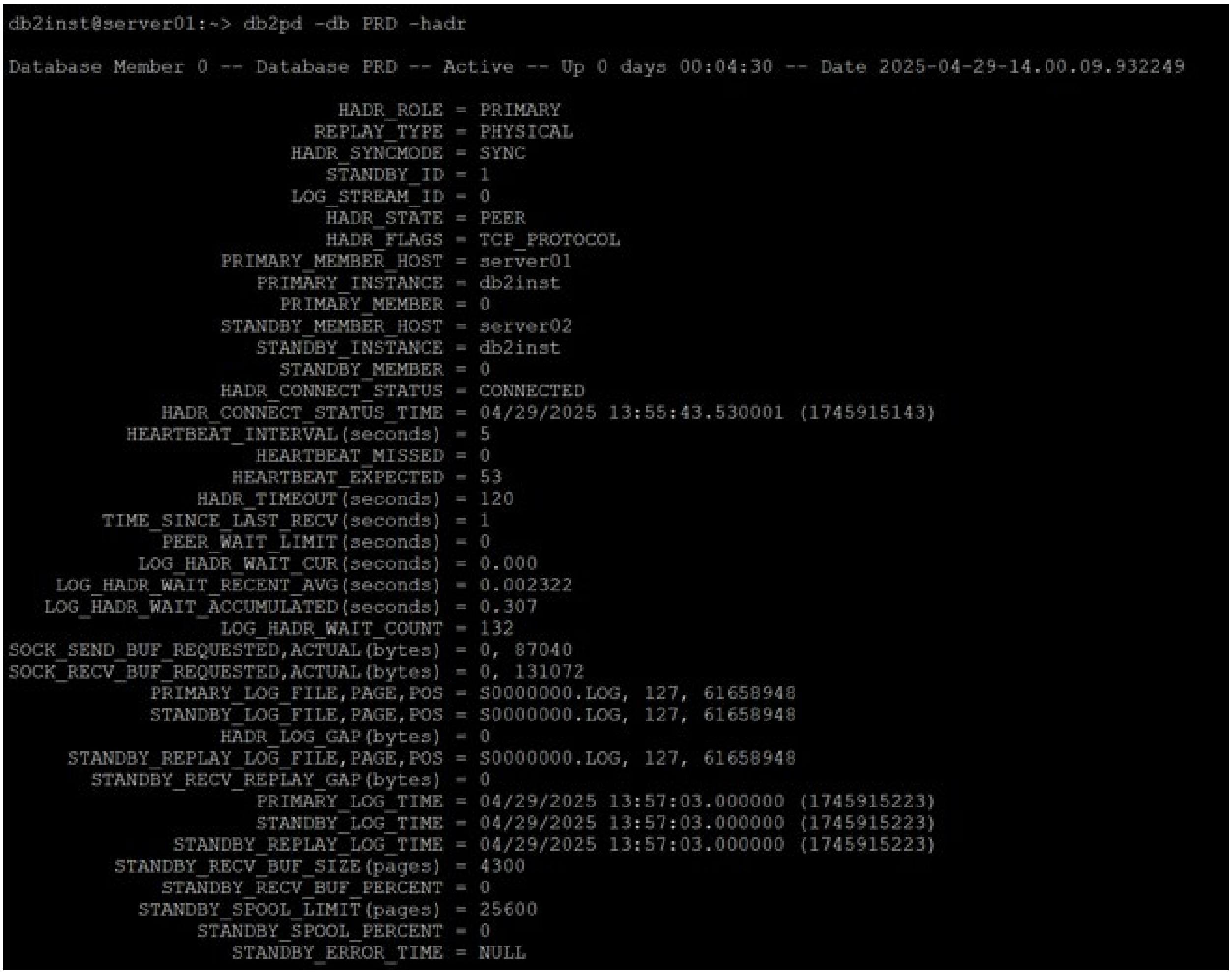
```
db2inst@server01:~> db2 ACTIVATE DB PRD
DB20000I  The ACTIVATE DATABASE command completed successfully.
db2inst@server01:~> db2 START HADR ON DB PRD AS PRIMARY
DB20000I  The START HADR ON DATABASE command completed successfully.
db2inst@server01:~>
```

Check the status of HADR on the primary node

```
db2inst@server01:~>    db2pd -db PRD -hadr | grep HADR
```



```
db2inst@server01:~> db2pd -db PRD -hadr | grep HADR
                            HADR_ROLE = PRIMARY
                        HADR_SYNCMODE = SYNC
                           HADR_STATE = PEER
                           HADR_FLAGS = TCP_PROTOCOL
                  HADR_CONNECT_STATUS = CONNECTED
             HADR_CONNECT_STATUS_TIME = 04/29/2025 13:55:43.530001 (1745915143)
               HADR_TIMEOUT(seconds) = 120
          LOG_HADR_WAIT_CUR(seconds) = 0.000
   LOG_HADR_WAIT_RECENT_AVG(seconds) = 0.002322
  LOG_HADR_WAIT_ACCUMULATED(seconds) = 0.307
                  LOG_HADR_WAIT_COUNT = 132
                 HADR_LOG_GAP(bytes) = 0
              HADR_LAST_TAKEOVER_TIME = NULL
db2inst@server01:~>
```



```
db2inst@server01:~> db2pd -db PRD -hadr

Database Member 0 -- Database PRD -- Active -- Up 0 days 00:04:30 -- Date 2025-04-29-14.00.09.932249

                            HADR_ROLE = PRIMARY
                          REPLAY_TYPE = PHYSICAL
                        HADR_SYNCMODE = SYNC
                           STANDBY_ID = 1
                        LOG_STREAM_ID = 0
                           HADR_STATE = PEER
                           HADR_FLAGS = TCP_PROTOCOL
                  PRIMARY_MEMBER_HOST = server01
                     PRIMARY_INSTANCE = db2inst
                       PRIMARY_MEMBER = 0
                  STANDBY_MEMBER_HOST = server02
                     STANDBY_INSTANCE = db2inst
                       STANDBY_MEMBER = 0
                  HADR_CONNECT_STATUS = CONNECTED
             HADR_CONNECT_STATUS_TIME = 04/29/2025 13:55:43.530001 (1745915143)
          HEARTBEAT_INTERVAL(seconds) = 5
                     HEARTBEAT_MISSED = 0
                   HEARTBEAT_EXPECTED = 53
                HADR_TIMEOUT(seconds) = 120
        TIME_SINCE_LAST_RECV(seconds) = 1
             PEER_WAIT_LIMIT(seconds) = 0
           LOG_HADR_WAIT_CUR(seconds) = 0.000
    LOG_HADR_WAIT_RECENT_AVG(seconds) = 0.002322
   LOG_HADR_WAIT_ACCUMULATED(seconds) = 0.307
                  LOG_HADR_WAIT_COUNT = 132
SOCK_SEND_BUF_REQUESTED,ACTUAL(bytes) = 0, 87040
SOCK_RECV_BUF_REQUESTED,ACTUAL(bytes) = 0, 131072
            PRIMARY_LOG_FILE,PAGE,POS = S0000000.LOG, 127, 61658948
            STANDBY_LOG_FILE,PAGE,POS = S0000000.LOG, 127, 61658948
                  HADR_LOG_GAP(bytes) = 0
     STANDBY_REPLAY_LOG_FILE,PAGE,POS = S0000000.LOG, 127, 61658948
       STANDBY_RECV_REPLAY_GAP(bytes) = 0
                     PRIMARY_LOG_TIME = 04/29/2025 13:57:03.000000 (1745915223)
                     STANDBY_LOG_TIME = 04/29/2025 13:57:03.000000 (1745915223)
              STANDBY_REPLAY_LOG_TIME = 04/29/2025 13:57:03.000000 (1745915223)
         STANDBY_RECV_BUF_SIZE(pages) = 4300
             STANDBY_RECV_BUF_PERCENT = 0
           STANDBY_SPOOL_LIMIT(pages) = 25600
                STANDBY_SPOOL_PERCENT = 0
                   STANDBY_ERROR_TIME = NULL
```

8 Linux Pacemaker cluster configuration

Once the HADR configuration is successfully implemented; in order to achieve the automated failover of Db2 database services, we need to configure the pacemaker cluster software. In this section we will go through the step-by-step process of configuring the pacemaker cluster with Db2 database instances.

Note: In this test, we are using the integrated pacemaker package which is by default packaged with the Db2 database instance & gets installed automatically during the installation of a Db2 database. If you are installing the pacemaker package on a Linux environment (SUSE or RHEL), we recommend checking for previously installed or OS packaged pacemaker software & uninstall it accordingly.

8.1 Installing the Pacemaker cluster software stack

Ensure that you have the Pacemaker cluster software package that is intended for use with Db2 by downloading the package from the IBM Marketing Registration Services site. Before proceeding to the next section, verify that all prerequisites and necessary criteria have been met. For more information on these prerequisites, please refer to the following checklist

Pre-setup checklist

- Instance user ID and group ID are set up.
- The /etc/hosts file includes both hosts, following the format listed in Host file setup.
- Both hosts have TCP/IP connectivity between their Ethernet network interfaces.
- Both the root and instance user IDs can use ssh between the two hosts, using both long and short host names.
- The Pacemaker cluster software has been downloaded to both hosts.
- Ensure that no non- Db2 provided Pacemaker components are installed.
- Ensure system repositories are enabled for pacemaker package dependencies.

Procedure

1. As root on the primary node, extract the tar file in the /tmp folder.
   - cd /tmp
   - tar -zxf Db2_v11.5.4.0_Pacemaker_20200418_<OS Version>_x86_64.tar.gz
   - The above will create the directory Db2_v11.5.4.0_Pacemaker_20200418_<OS Version>_x86_64

2. Verify that the following packages are installed. The output may vary slightly for different architectures and Linux distributions. All packages should include the **db2pcmk** text in the output.

```
[root]# rpm -q corosync
corosync-3.0.3-1.db2pcmk.el8.x86_64
[root]# rpm -q pacemaker
```

```
pacemaker-2.0.2-1. db2pcmk.el8.x86_64
[root]# rpm -q crmsh
crmsh-4.1.0-0. db2pcmk.el8.noarch
```

3. Copy the **db2cm** utility from the cluster software directory into the instance sqllib/bin directory

4. Copy the resource agent scripts (db**2hadr**, **db2inst**, **db2ethmon**) from /tmp/Db2agents into /usr/lib/ocf/resource.d/heartbeat/ on both hosts

5. Repeat steps 1 to 3 on the secondary node.

8.2 The Db2 cluster manager (db2cm) utility

You can use the Db2 cluster manager (db2cm) utility to configure and administer your highly available databases in a Pacemaker-managed Db2 Linux cluster. The Pacemaker cluster software stack must be installed on all hosts in the cluster. The Db2 instances and HADR database should be configured and online before performing the following procedure outlined.

Whether you use the Db2 installer or the db2installPCMK installation script to install Db2, you must meet the basic prerequisites for installing Db2.

Run the db2installPCMK installation script. The db2installPCMK script is in the Db2 install media at the following location:

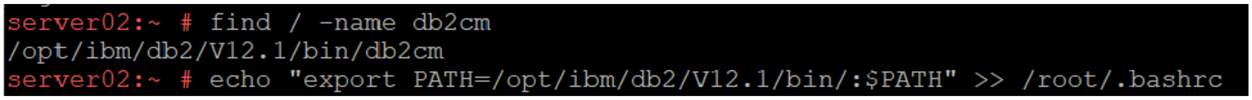
```
# db2/<platform>/pcmk <<where platform refers to the appropriate hardware platform>>
```

To install or update Pacemaker, use the following command:

```
# db2installPCMK -i
```

find db2cm path and add it to bashrc

```
#find / -name   db2cm
#/ db2/ db2PRD/ db2_software/bin/ db2cm
#echo "export PATH=/ db2/ db2PRD/ db2_software/bin:$PATH" >> /root/.bashrc
```



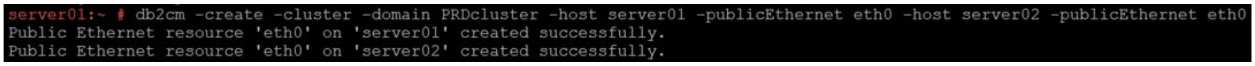
```
server02:~ # find / -name db2cm
/opt/ibm/db2/V12.1/bin/db2cm
server02:~ # echo "export PATH=/opt/ibm/db2/V12.1/bin/:$PATH" >> /root/.bashrc
```

Procedure

The following steps are only required to run once on any one of the hosts by root. There is no need to run them on both hosts. Choose one of the hosts to perform all actions on the same host.

1. Create the Pacemaker cluster and the public network resources by running the following command. This is only required to be run once.

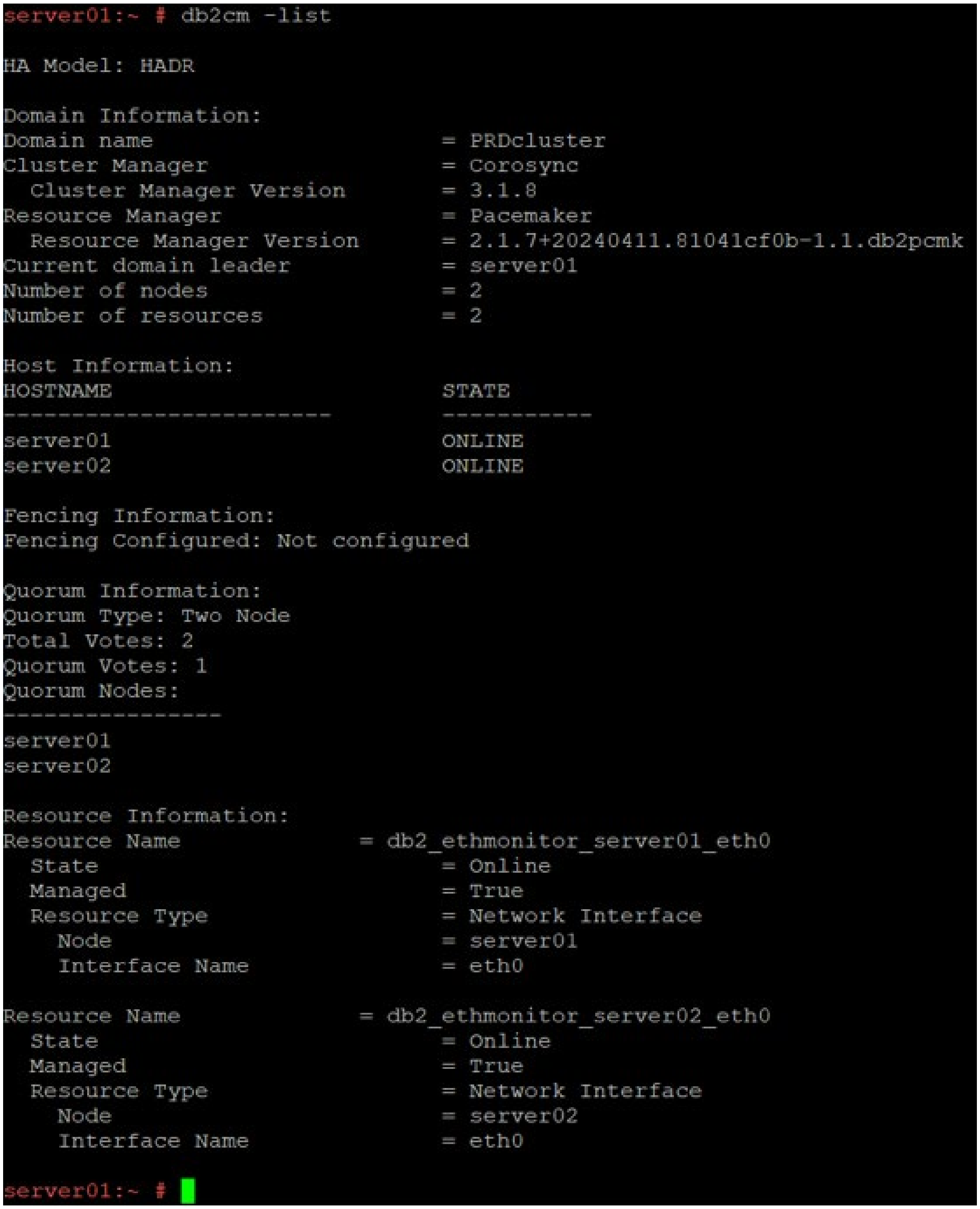
2. # db2cm -create -cluster -domain PRDcluster -host server01 -publicEthernet bond0 -host server02 -publicEthernet bond0

   ```
   server01:~ # db2cm -create -cluster -domain PRDcluster -host server01 -publicEthernet eth0 -host server02 -publicEthernet eth0
   Public Ethernet resource 'eth0' on 'server01' created successfully.
   Public Ethernet resource 'eth0' on 'server02' created successfully.
   ```

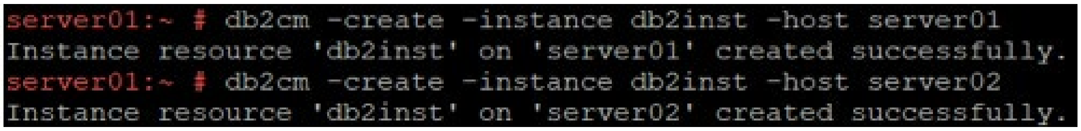
3. Check the result of base cluster formation.

db2cm -list



```
server01:~ # db2cm -list

HA Model: HADR

Domain Information:
Domain name                        = PRDcluster
Cluster Manager                    = Corosync
  Cluster Manager Version          = 3.1.8
Resource Manager                   = Pacemaker
  Resource Manager Version         = 2.1.7+20240411.81041cf0b-1.1.db2pcmk
Current domain leader              = server01
Number of nodes                    = 2
Number of resources                = 2

Host Information:
HOSTNAME                           STATE
------------------------           -----------
server01                           ONLINE
server02                           ONLINE

Fencing Information:
Fencing Configured: Not configured

Quorum Information:
Quorum Type: Two Node
Total Votes: 2
Quorum Votes: 1
Quorum Nodes:
----------------
server01
server02

Resource Information:
Resource Name                = db2_ethmonitor_server01_eth0
  State                            = Online
  Managed                          = True
  Resource Type                    = Network Interface
    Node                           = server01
    Interface Name                 = eth0

Resource Name                = db2_ethmonitor_server02_eth0
  State                            = Online
  Managed                          = True
  Resource Type                    = Network Interface
    Node                           = server02
    Interface Name                 = eth0

server01:~ #
```

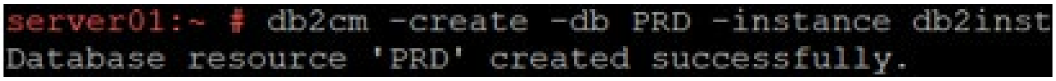
4. Create the instance resource model by running the following commands
5. # db2cm -create -instance db2PRD -host server01
6. # db2cm -create -instance db2PRD -host server02



```
server01:~ # db2cm -create -instance db2inst -host server01
Instance resource 'db2inst' on 'server01' created successfully.
server01:~ # db2cm -create -instance db2inst -host server02
Instance resource 'db2inst' on 'server02' created successfully.
```

7. Create the HADR database resource by running the following commands

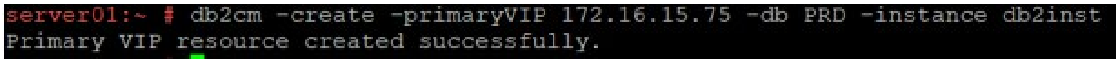
8. # db2cm -create -db PRD -instance db2PRD



```
server01:~ # db2cm -create -db PRD -instance db2inst
Database resource 'PRD' created successfully.
```

9. Create the VIP resource

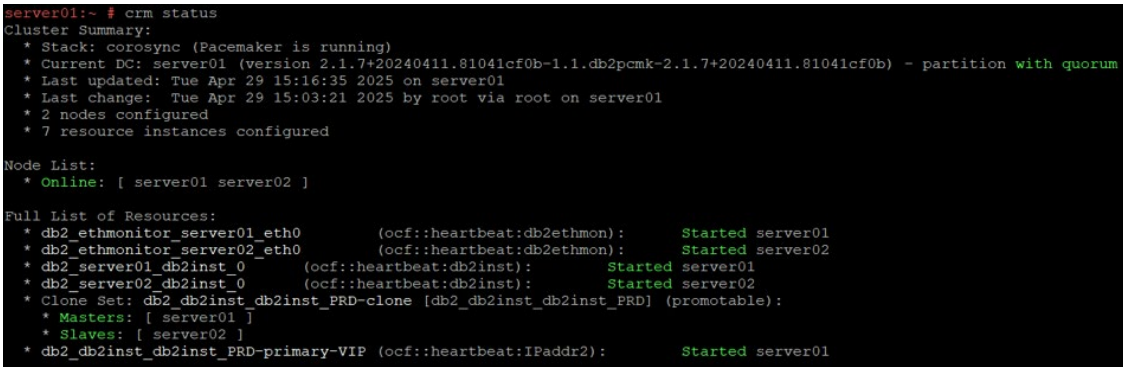
   # db2cm -create -primaryVIP 172.21.33.60 -db PRD -instance db2PRD



```
server01:~ # db2cm -create -primaryVIP 172.16.15.75 -db PRD -instance db2inst
Primary VIP resource created successfully.
```

10. Check the cluster resource post resource addition using crm_status.

    #crm status

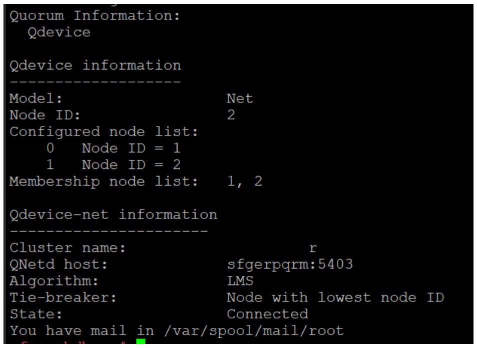


```
server01:~ # crm status
Cluster Summary:
  * Stack: corosync (Pacemaker is running)
  * Current DC: server01 (version 2.1.7+20240411.81041cf0b-1.1.db2pcmk-2.1.7+20240411.81041cf0b) - partition with quorum
  * Last updated: Tue Apr 29 15:16:35 2025 on server01
  * Last change:  Tue Apr 29 15:03:21 2025 by root via root on server01
  * 2 nodes configured
  * 7 resource instances configured

Node List:
  * Online: [ server01 server02 ]

Full List of Resources:
  * db2_ethmonitor_server01_eth0        (ocf::heartbeat:db2ethmon):      Started server01
  * db2_ethmonitor_server02_eth0        (ocf::heartbeat:db2ethmon):      Started server02
  * db2_server01_db2inst_0      (ocf::heartbeat:db2inst):        Started server01
  * db2_server02_db2inst_0      (ocf::heartbeat:db2inst):        Started server02
  * Clone Set: db2_db2inst_db2inst_PRD-clone [db2_db2inst_db2inst_PRD] (promotable):
    * Masters: [ server01 ]
    * Slaves: [ server02 ]
  * db2_db2inst_db2inst_PRD-primary-VIP (ocf::heartbeat:IPaddr2):        Started server01
```

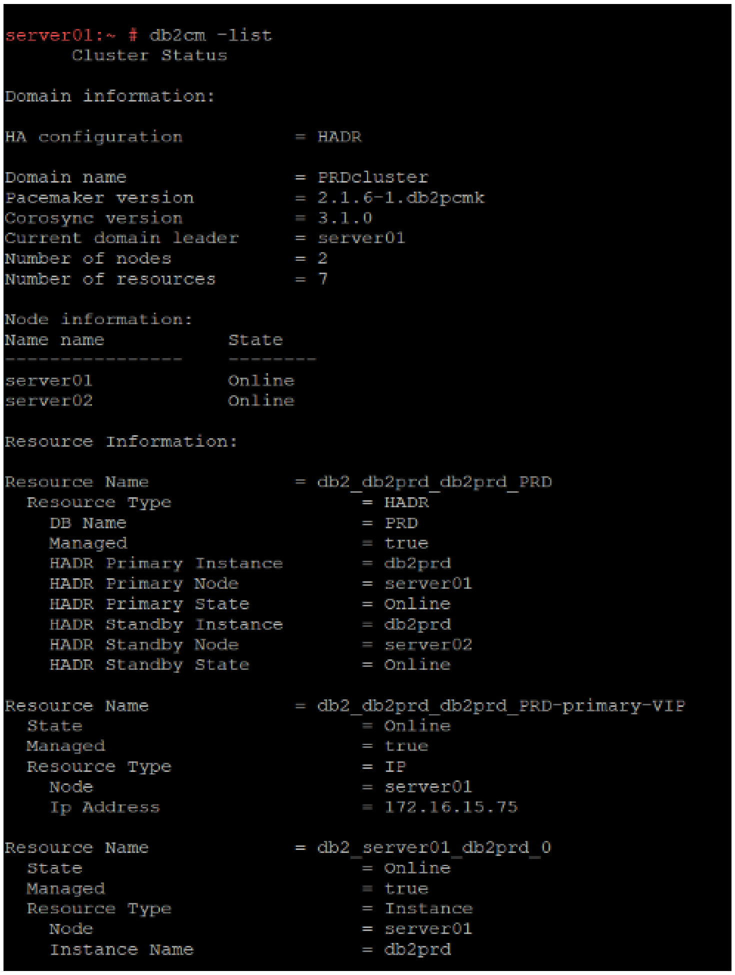
For the quorum node follow 6.2 to install db2cm utility. After the installation you can add the quorum node into the cluster with the command below (update both DB server “/etc/hosts” with quorumnode ip).

# db2cm -create -qdevice quorumnode

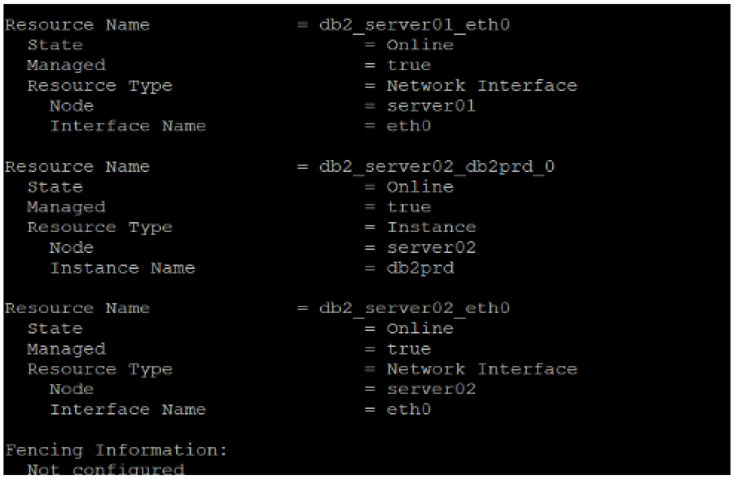
After the quorum node addition, you can use the check db2cm -list command to verify the quorum node.



```
Quorum Information:
  Qdevice

Qdevice information
-------------------
Model:                  Net
Node ID:                2
Configured node list:
    0   Node ID = 1
    1   Node ID = 2
Membership node list:   1, 2

Qdevice-net information
----------------------
Cluster name:                   r
QNetd host:             sfgerpqrm:5403
Algorithm:              LMS
Tie-breaker:            Node with lowest node ID
State:                  Connected
You have mail in /var/spool/mail/root
```



```
server01:~ # db2cm -list
      Cluster Status

Domain information:

HA configuration          = HADR

Domain name               = PRDcluster
Pacemaker version         = 2.1.6-1.db2pcmk
Corosync version          = 3.1.0
Current domain leader     = server01
Number of nodes           = 2
Number of resources       = 7

Node information:
Name name           State
----------------    --------
server01            Online
server02            Online

Resource Information:

Resource Name             = db2_db2prd_db2prd_PRD
  Resource Type                 = HADR
    DB Name                     = PRD
    Managed                     = true
    HADR Primary Instance       = db2prd
    HADR Primary Node           = server01
    HADR Primary State          = Online
    HADR Standby Instance       = db2prd
    HADR Standby Node           = server02
    HADR Standby State          = Online

Resource Name             = db2_db2prd_db2prd_PRD-primary-VIP
  State                         = Online
  Managed                       = true
  Resource Type                 = IP
    Node                        = server01
    Ip Address                  = 172.16.15.75

Resource Name             = db2_server01_db2prd_0
  State                         = Online
  Managed                       = true
  Resource Type                 = Instance
    Node                        = server01
    Instance Name               = db2prd
```



```
Resource Name             = db2_server01_eth0
  State                         = Online
  Managed                       = true
  Resource Type                 = Network Interface
    Node                        = server01
    Interface Name              = eth0

Resource Name             = db2_server02_db2prd_0
  State                         = Online
  Managed                       = true
  Resource Type                 = Instance
    Node                        = server02
    Instance Name               = db2prd

Resource Name             = db2_server02_eth0
  State                         = Online
  Managed                       = true
  Resource Type                 = Network Interface
    Node                        = server02
    Interface Name              = eth0

Fencing Information:
  Not configured
```

9 Resources

Lenovo

Lenovo SR650 V3

Lenovo SAP Solutions

IBM

Db2 best practices

Db2 HADR

SAP

SAP on Db2 best practices

Trademarks and special notices

© Copyright Lenovo 2025.

Lenovo and the Lenovo logo are trademarks or registered trademarks of Lenovo in the United States, other countries, or both. A current list of Lenovo trademarks is available on the Web at https://www.lenovo.com/us/en/legal/copytrade/.

The following terms are trademarks of Lenovo in the United States, other countries, or both:

Lenovo®
ThinkSystem®

The following terms are trademarks of other companies:

Intel® is a trademark of Intel Corporation or its subsidiaries.

Linux® is the trademark of Linus Torvalds in the U.S. and other countries.

IBM® and DB2® are trademarks of IBM in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

References in this document to Lenovo products or services do not imply that Lenovo intends to make them available in every country.

Information is provided "AS IS" without warranty of any kind.

All customer examples described are presented as illustrations of how those customers have used Lenovo products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Information concerning non-Lenovo products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by Lenovo. Sources for non-Lenovo list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. Lenovo has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-Lenovo products. Questions on the capability of non-Lenovo products should be addressed to the supplier of those products.

All statements regarding Lenovo future direction and intent are subject to change or withdrawal without notice and represent goals and objectives only. Contact your local Lenovo office or Lenovo authorized reseller for the full text of the specific Statement of Direction.

Some information addresses anticipated future capabilities. Such information is not intended as a definitive statement of a commitment to specific levels of performance, function or delivery schedules with respect to any future products. Such commitments are only made in Lenovo product announcements. The information is presented here to communicate Lenovo’s current investment and development activities as a good faith effort to help with our customers' future planning.

Performance is based on measurements and projections using standard Lenovo benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Photographs shown are of engineering prototypes. Changes may be incorporated in production models.

Any references in this information to non-Lenovo websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this Lenovo product and use of those websites is at your own risk.