



Technical Report

Empower Your Data Warehouse with IBM InfoSphere Warehouse and NetApp Storage Systems

Jawahar Lal and Bobby Oommen, NetApp
June 2011 | TR-3943

Design and implement an InfoSphere Warehouse solution on a NetApp storage system

Data warehouses are no longer used just for data mining, analytical processing, and market research. Modern data warehouses are expected to do not only cause and impact analysis of past events but also to provide option-based predictive analytics with actionable recommendations and cost analysis for every option. In addition, front-line business operations rely on data warehouses. The increasing use of predictive analytics for decision making and the reliance of front-line operations on data warehouse analytics mean that these analytics are more intense, sophisticated, and complex. A data warehouse solution empowered by IBM® InfoSphere® Warehouse and NetApp® storage technology meets all of these challenges and also lowers the total cost of ownership and results in higher return on investment.

TABLE OF CONTENTS

1	DOCUMENT PURPOSE	5
2	OVERVIEW	5
2.1	BUSINESS NEED	5
2.2	SOLUTION OVERVIEW	6
2.3	SOLUTION COMPONENTS	7
2.4	KEY BENEFITS OF INFOSPHERE WAREHOUSE	8
2.5	IBM DB2 DATA PARTITION FEATURES	9
2.6	WHY NETAPP STORAGE FOR INFOSPHERE WAREHOUSE?	10
2.7	NETAPP SNAPSHOT TECHNOLOGY	11
2.8	NETAPP STORAGE LAYERS	12
2.9	SNAP CREATOR FRAMEWORK	13
3	ASSUMPTIONS	15
4	ARCHITECTURE	16
4.1	SOFTWARE COMPONENTS	17
4.2	SERVER SPECIFICATIONS	18
4.3	STORAGE SPECIFICATIONS	19
4.4	NETWORK SPECIFICATIONS	19
5	CONFIGURATION OVERVIEW AND CONSIDERATIONS	21
5.1	SERVER CONFIGURATION	21
5.2	DATABASE DESIGN	21
5.3	STORAGE SYSTEM CONFIGURATION	26
5.4	VOLUME CONSIDERATION	32
6	INSTALLATION AND CONFIGURATION	32
6.1	INSTALLATION	32
6.2	DB2 ADMINISTRATOR AND USER ACCOUNTS	33
6.3	INSTALLING DB2	34
6.4	CREATING A DB2 HOME FILE SYSTEM ON A NETAPP STORAGE SYSTEM	34
7	CREATING DB2 INSTANCES AND DATABASES ON A NETAPP STORAGE SYSTEMS	34
7.1	CREATING A DB2 INSTANCE	34
7.2	UPDATING THE DB2NODE.CFG FILE	36
7.3	CREATING A DATABASE ON A NETAPP STORAGE SYSTEM VOLUME	37
7.4	CREATING DATABASE PARTITION GROUPS	37
7.5	CREATING A BUFFER POOL	39
7.6	PLACING THE DATABASE TRANSACTION LOG IN THE NEW DIRECTORY	39
7.7	CREATING TABLE SPACES	40

7.8	CREATING TABLES AND POPULATING DATA.....	41
8	INFOSPHERE WAREHOUSE BACKUP AND RESTORE.....	41
8.1	BASIC CONFIGURATION	42
8.2	DATABASE BACKUP USING SNAP CREATOR FRAMEWORK	43
8.3	DATABASE RECOVERY USING SNAP CREATOR	44
9	APPENDIXES.....	46
9.1	DATABASE MANAGER CONFIGURATION PARAMETERS FOR THE CORE WAREHOUSE	46
9.2	DATABASE CONFIGURATION PARAMETER SETTINGS.....	48
9.3	KERNEL PARAMETER SETTINGS.....	50
9.4	NETWORK CONFIGURATION DETAILS	51
9.5	SNAP CREATOR CONFIGURATION FILE FOR TEST ENVIRONMENT	52

LIST OF TABLES

Table 1)	Server details.....	15
Table 2)	Storage system details.	15
Table 3)	Software components.....	17
Table 4)	Server specifications.	18
Table 5)	Storage specifications.	19
Table 6)	Network specifications.....	19
Table 7)	DB2 registry variable settings.....	21
Table 8)	Database manager configuration parameters.....	22
Table 9)	Database configuration parameters.	22
Table 10)	Table space considerations.....	24
Table 11)	Storage controller specifications.....	26
Table 12)	Administration node specifications.....	27
Table 13)	File system name and purpose on the administration node.....	27
Table 14)	Management node specifications.....	27
Table 15)	File system name and purpose on the management node.....	27
Table 16)	Storage layout, first data node specifications.....	28
Table 17)	Storage layout, second data node specifications.....	29
Table 18)	Storage layout, file system details.....	30
Table 19)	Application server module details.....	31
Table 20)	Application server module, file system details.....	31
Table 21)	Database partitions.....	36
Table 22)	Database configuration parameters.....	39
Table 23)	Table space details.....	40
Table 24)	Snap Creator basic configuration.....	42

Table 25) Kernel parameter settings.	50
Table 26) Network configuration details.	51

LIST OF FIGURES

Figure 1) Logical relationship between aggregate, volume, and physical disks.....	13
Figure 2) Snap Creator architecture and components.	14
Figure 3) Conceptual design of InfoSphere Warehouse on NetApp.	16
Figure 4) Configuration overview: InfoSphere Warehouse architecture.....	17
Figure 5) InfoSphere Warehouse network architecture.....	20
Figure 6) Database partition group layout.	38
Figure 7) Snap Creator backup process flow.	44
Figure 8) Snap Creator database restore process flow.....	45

1 DOCUMENT PURPOSE

NetApp's solution design and deployment guides assist customers in designing and implementing mission-critical business solutions to meet specific business requirements. These guides are developed based on real-time field experiences with NetApp customers, real-world simulations, and NetApp engineering lab validations.

This guide is intended to help customers in the design, implementation, and administration of an InfoSphere Warehouse solution with NetApp storage.

IBM InfoSphere Warehouse offers a comprehensive, scalable, and robust data warehouse solution. Deploying this solution with NetApp storage offers additional cost-saving benefits such as NetApp Snapshot™ based backup and restore, cloning, and Flash Cache. The purpose of this document is to bring together IBM's and NetApp's best practices for planning, configuring, designing, implementing, and administering IBM InfoSphere Warehouse with NetApp storage. It offers guidance to customers who want to integrate the following components:

- InfoSphere Warehouse 9.7 software
- NetApp FAS or IBM N series storage
- IBM Power® (P740) running an AIX® operating system

This document covers key components of the InfoSphere Warehouse and NetApp storage technology in detail.

2 OVERVIEW

2.1 BUSINESS NEED

In last two decades, data warehouse usage has evolved from information about the past to predictive analytics about the present and future that involve complex analysis and diagnosis of the business problem. Initially, data warehouses were used to gather information about what had happened. But understanding what has happened is no longer enough for today's decision makers. They want to develop models based on the gathered information that can produce actionable recommendations and also provide detail about cost and impact. They also want transparent and traceable analytics to back up both the diagnosis of the business problem and its solution.

Businesses are demanding more precisely defined options, making these analytics more intensive and complex than the simple statistics that served 10 years ago. Forecasting the impact of implementing various options is increasingly critical for business decision makers. In addition, decision makers are looking to data warehouses to support front-line operational business processes; for example, enabling customer-facing employees in their day-to-day decision making as they process orders; answer questions; and service requests.

The emphasis on predictive analytics, operational business intelligence, faster reaction to events, and deeper analysis of options results in complex and demanding technical requirements. Some key requirements for the current generation of data warehouses are:

- **Complex analytics.** Sophisticated business strategies demand complex analysis of the warehouse data. Also, the volume of data volume to be analyzed has grown explosively. Most important business decisions today are based on the analytics that make high data availability an important requirement in the data warehouse environment.
- **Frequent updates of data.** Instead of weekly or nightly batch updates, today's data warehouses often require frequent or continuous online updates. Stringent data latency requirements are becoming common; for example, data must be online and available for query within a specified

number of hours, minutes, or seconds of receipt. And high data volume ingest requirements are also becoming common.

- **Mixed workload.** Instead of periodic scheduled reporting or analysis, today's data warehouses often must meet multiple concurrent service levels, including interactive query, large long-running jobs, and a mix of concurrent queries and updates.
- **Many concurrent users.** Instead of a few users feeding a relatively small group of management decision makers, modern data warehouses often support front-line operations and large communities of employees, partners, or customers who are concurrently accessing and updating data.
- **High data availability.** As enterprises rely increasingly on the data warehouse to support critical decision making, marketing research, and operational business processes, high data availability requirements often attain mission-critical status, so that the data must operate on a continuous or near continuous schedule.

2.2 SOLUTION OVERVIEW

IBM offers two types of data warehouse solutions: InfoSphere Warehouse and Smart Analytics System. InfoSphere Warehouse offers an “à la carte” approach; customers can customize the data warehouse components according to their needs. The Smart Analytics System comes as an appliance; the customer must buy the complete solution. For information about Smart Analytics System design and implementation, see the IBM Web site. (www-01.ibm.com/software/data/infosphere/smart-analytics-system/)

InfoSphere Warehouse provides a foundation that meets the entire spectrum of data warehousing needs, from large enterprise data warehouses to high-performance departmental data marts to packaged data warehouse solutions for small and medium-sized businesses. The InfoSphere Warehouse offers greater simplicity, flexibility, scalability, and supportability. Integrating InfoSphere Warehouse methodology and innovative storage technologies can help customers to reduce implementation time and complexity and improve the availability and scalability of the warehouse environment.

Several customers have customized the InfoSphere Warehouse methodology to match their existing server, network, and/or storage assets. The customization approach helps them to use their existing infrastructure and assets and to reduce their overall deployment time.

The data warehouse solution described in this document is one of the integration examples that uses InfoSphere Warehouse Enterprise Edition configuration and NetApp storage. This configuration has been tested in the NetApp lab. The solution components of InfoSphere Warehouse Enterprise Edition are POWER7[®] servers, network switches, and a NetApp storage system. This document describes the steps necessary to design and deploy a scalable InfoSphere Warehouse in NetApp storage environment. It covers the following topics:

- InfoSphere Warehouse architecture
- Detailed implementation specifications, including preparing the servers, storage, and database layout
- Installing DB2[®] 9.7 InfoSphere Warehouse on a POWER7 server in a NetApp environment
- Creating a partition database in a NetApp environment

The InfoSphere Warehouse solution covered in this document was inspired by IBM Smart Analytics System 7700. The solution was tested by IBM and NetApp teams using standard test methodologies under a real-time workload scenario.

2.3 SOLUTION COMPONENTS

The InfoSphere Warehouse is a methodology that breaks a complex concept down into more easily understood units, or modules. The modular approach simplifies capacity planning, sizing, and scaling for the warehouse environment. The InfoSphere Warehouse has many components that can be grouped into modules based on functionality. Depending on business requirements, the components in the following modules can be installed on one or more servers:

- Management module
- Administration module
- Data module
- Application server module
- Failover and HA modules
- Business intelligence module

MANAGEMENT MODULE

This module contains InfoSphere Warehouse components that are primarily used to manage the cluster, hosts, network, storage, and application functionality. Examples of InfoSphere components in this group are System Control Console, IBM Systems Director, DB2 Performance Expert, NetApp storage management products, and cluster deployment and host management consoles. This module can be installed on one or more servers.

ADMINISTRATION MODULE

Depending on size and throughput requirement, an InfoSphere Warehouse deployment can have one or more administration modules. InfoSphere Warehouse deployments have multiple database partitions. The catalog database partition (partition 0), which stores system catalog data and nonpartitioned data belonging to the core warehouse database resides on this module. The catalog partition acts as a coordinator for user connections. The administration module is one of the most important modules for the warehouse, and it should be installed on a powerful machine. For our test environment, the administration module was installed on a Power 740 server with 128GB of memory and two 8-core POWER7 processors. For external storage we used a NetApp FAS3272 active-active cluster storage system with four DS4243 disk shelves.

DATA MODULE

A typical InfoSphere Warehouse deployment consists of one or more data modules. Each data module contains servers, storage, network switches, and multiple partitions of the database. The data modules are used for storing warehouse table data and indexes. Our test environment had two data modules, each with eight database partitions. Each module was created on an IBM Power 740 server with 128GB of memory and two 8-core POWER7 processors. For storage, each module has a dedicated FAS6210 active-active controller pair attached to eight DS4243 disk shelves.

APPLICATION SERVER MODULE

The InfoSphere Warehouse application server module manages all application connections. An optional but strongly recommended module of the data warehouse solution, it consists of the following components:

- Administration Console and Workload Manager
- IBM Data Server Client
- WebSphere® Application Server

This module can be installed on one or two Power 740 servers, each configured with 32GB of memory and two quad-core POWER7 processors. For storage, a NetApp FAS3270 storage system with four DS4243 disk shelves was used. Each disk shelf was populated with twenty-four 300GB SAS disk drives.

FAILOVER AND HA MODULES

The failover module provides high availability for the critical components of an InfoSphere Warehouse environment. Administration and data module components are considered critical for the InfoSphere Warehouse environment, so IBM and NetApp recommend implementing high availability for components in these two modules. In our test configuration, we did not implement the HA module.

BUSINESS INTELLIGENCE MODULE

The primary objective of the business intelligence module is to deliver the packaged reports and analysis for assessing the performance of specific functional domains. This module is implemented using Cognos® 8 Business Intelligence software. It can contain two or more nodes; the maximum number depends on the size of the data warehouse environment. We did not implement this module in the lab.

This module terminology is used throughout the document for logically grouping the InfoSphere components. This list is not comprehensive; for a complete list of components, see the IBM InfoSphere Warehouse Information Center Web page.

(<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp>)

2.4 KEY BENEFITS OF INFOSPHERE WAREHOUSE

Modular architecture. The InfoSphere Warehouse breaks down the complex data warehouse concept and architecture into modules that are easy to understand and manage. As business requirements change and more data sources are needed, the modular architecture can easily scale the Warehouse to meet increased workload demands.

Massively parallel processing (MPP) support. MPP is recognized as the only truly scalable architecture for data warehousing. MPP allows the system to use multiple servers and CPUs to process data in parallel, resulting in performance improvement through a “divide and conquer” approach. MPP improves performance and lowers I/O requirements, resulting in lower hardware and infrastructure support costs.

Database partitioning. InfoSphere Warehouse transparently splits the database across multiple partitions and uses the compute power of multiple servers to satisfy requests for large amounts of information. For more information, see section 2.5, “IBM DB2 Data Partition Features.”

Multidimensional clustering (MDC). MDC offers a flexible method to continuously and automatically cluster table data in multiple dimensions. This improves query performance by reducing I/O requirements.

Materialized query table (MQT). It’s not unusual for a business user to make the same query time and time again. MQTs improve the performance of such queries by caching the results. When the query is submitted again, the database engine can return the data from the MQT.

Mixed workload management. InfoSphere Warehouse is optimized for mixed workloads, or random I/O, which is common for a modern data warehouse. DB2 Workload Manager allows administrators to prioritize queries coming from different users and applications and to control the number of underlying resources dedicated to those processes.

Performance optimization. InfoSphere Warehouse includes IBM Optim™ Performance Manager, which enables IT staff to identify, diagnose, solve, and prevent performance problems in DB2 and associated applications.

DB2 compression. The DB2 Storage Optimization feature transparently compresses data on disk to decrease disk space and storage infrastructure requirements and to improve query performance.

Cubing Services. Cubing Services provides a multidimensional view of data stored in a data warehouse. Cubing Services enables creating, editing, importing, exporting, and deploying cube models over the relational warehouse schema. Cubing Services also offers optimization techniques to dramatically improve the performance of OLAP queries.

Design Studio for data mining. The data mining capability of InfoSphere Warehouse supports standard data mining models (clustering, associations, classification, and prediction) and can be developed by using drag and drop in the Design Studio tool.

Label-based access control (LBAC) for security. The security features in DB2 protect data and comply with regulatory requirements. Users can be granted privileges limited to a particular data resource, or they can be given predefined roles. LBAC allows access to tables at the row level, the column level, or both.

High availability. InfoSphere Replication Server technology is included in all editions of InfoSphere Warehouse. This feature enables queue replication between a single DB2 source and target, for high availability, high throughput, and business continuity.

Database administration. Optim Database Administrator, which is included in all editions of InfoSphere Warehouse, helps save time and reduce errors associated with complex database changes.

Data movement and transformation. The SQL Warehousing (SQW) tool, part of Design Studio, is a graphical tool that generates SQL for warehouse maintenance and administration, replacing hand-coded SQL. SQW complements and works with InfoSphere Information Server and third-party ETL products.

Industry and business process models. InfoSphere Warehouse customers can benefit from IBM's industry models, which combine deep expertise and industry best practices in a form that both business and IT communities can use to accelerate the development and deployment of industry solutions.

2.5 IBM DB2 DATA PARTITION FEATURES

IBM InfoSphere Warehouse uses DB2 as the database component. IBM DB2 9.7 Enterprise Server Edition for Warehouse offers the database partition feature (DPF), a robust performance and scalability feature that customers can use to create a database with multiple partitions. (A database partition is part of a database that consists of its own data, indexes, configuration files, and transaction logs.) Tables can be located in one or more database partitions. Processors associated with each database partition are used to satisfy data requests. Data retrieval and update requests are decomposed automatically into subrequests and are executed in parallel among the applicable database partitions.

The following are some of the foundation features of IBM DB2 9.7 that make it a best fit for OLTP and warehouse environments:

- A shared-nothing parallel database architecture that provides performance, scalability, and high data availability
- Ability to deliver a solid physical database design, I/O performance, efficient cross-partition network utilization, and cost-based query optimization
- Mixed workload management to satisfy multiple service-level objectives on a single system managing a single logical copy of the data

Most DPF deployments are in the areas of data warehousing and business intelligence. DPF is highly recommended for large complex query environments, because its shared-nothing architecture allows outstanding scalability.

For in-depth information about partitioned database configuration best practices, see the InfoSphere Balanced Warehouse documentation.

Best Practice

A typical partitioned database system usually has one processor core per data partition. If the catalog partition is heavily used, IBM and NetApp recommend allocating a processor core for it as well. If the system supports a large number of concurrent active users, two cores per partition might be required.

Note: Starting with DB2 version 9.5, the DB2 database partition feature is available only in all IBM InfoSphere Warehouse product editions.

2.6 WHY NETAPP STORAGE FOR INFOSPHERE WAREHOUSE?

This section describes the key benefits of storing InfoSphere Warehouse on NetApp storage.

Unified storage. With a unified, multiprotocol architecture, NetApp storage solutions take advantage of the benefits of Ethernet and work as a "unification engine," supporting NFS, CIFS, iSCSI, and Fibre Channel over Ethernet (FCoE) in the same system, as well as leveraging traditional Fibre Channel storage investments.

Fast backup. Database backup windows are becoming history. Businesses are demanding 24/7 data availability. NetApp offers excellent value for database customers in DB2 backup and recovery. NetApp storage-efficient Snapshot technology can be used to perform backups with no downtime or performance penalty. Snapshot backups are completed in just seconds, regardless of the size of the database or the level of activity on the NetApp storage system. This innovative approach dramatically reduces backup window requirements and allows DBAs to schedule frequent database backups.

Quick recovery. Using the Data ONTAP® SnapRestore® technology, an entire database can be restored in just seconds from a Snapshot backup. Because no data copying is involved, a large amount of time is saved when a database is returned to the state it was in at the time the Snapshot copy was created. Additionally, the Snapshot creation process is very time efficient, and it allows retaining a large number of Snapshot copies. The SnapRestore process can also reduce the amount of time needed to perform a roll-forward recovery operation against a database.

High availability. Organizations of all sizes now require 24/7 availability. Companies cannot tolerate scheduled downtime or the extended periods of slow system response that are often caused by traditional database backup methods. Snapshot copies, on the other hand, can be created in a matter of seconds without any impact on system response time, affording high availability and uninterrupted system response.

High reliability. The RAID architecture used for NetApp storage systems offers greater reliability than direct-attached storage (DAS). If a RAID member disk fails, it is automatically reconstructed (using parity disk data) without any user intervention. NetApp supports single parity as well as Double Parity RAID (RAID-DP®). RAID-DP is considered approximately 10,000 times more reliable than traditional RAID. For more information, see [RAID-DP: NetApp Implementation of Double-Parity RAID for Data Protection](#).

Storage virtualization. Data ONTAP decouples physical storage from the volumes and allows creating virtual volumes, called NetApp FlexVol® volumes. A FlexVol volume can be resized as the application requires and can be cloned in matter of seconds, without any additional storage requirement.

Performance boost with Flash Cache. NetApp Flash Cache optimizes the performance of random read-intensive workloads and delivers performance that is comparable to that of solid state disks, without the complexity of another storage tier. This intelligent read cache speeds access to data, reducing latency by a factor of 10 or more compared to disk drives. Faster response times can translate into higher throughput for random I/O workloads. There is no need to move data from tier to tier to optimize performance and cost. Active data automatically flows into Flash Cache because every volume and LUN behind the storage controller is subject to caching.

Prioritize workload with FlexShare. You can give caching priority to your most important volumes and LUNs when the load is heaviest by using NetApp FlexShare® software in combination with Flash Cache

cards. You can also tune Flash Cache to match your specific workload with software settings that let you cache only metadata or cache new data when it is written to disk.

Performance improvement with FlexCache. NetApp FlexCache[®] software creates a caching layer in the storage infrastructure that automatically adapts to changing usage patterns, eliminating performance bottlenecks. In addition, FlexCache automatically replicates and serves hot data sets anywhere in the infrastructure by using local caching volumes. Key benefits of FlexCache include eliminating storage bottlenecks automatically without tedious administration, improved read performance in distributed application and testing environments, and simplified tiered storage layers. Read more at http://media.netapp.com/documents/NET_DS_FlexCache_R3.pdf.

NetApp Unified Connect. NetApp offers the industry's first end-to-end network convergence—NetApp Unified Connect concurrent support for FCoE and IP storage traffic (iSCSI, NFS, CIFS) over a shared 10GB Ethernet wire, from server to storage. With NetApp Unified Connect you can build a flexible and efficient shared infrastructure today as your foundation for future-ready IT.

Automation with Snap Creator Framework. Every environment is different and may require significant customization and automation to meet business needs. Creating scripts or using third-party tools to automate database management functions such as backup, restore, clone, and replication can be very costly and also require significant time to implement. NetApp Snap Creator is an automation solution that eliminates the need to create scripts and also offers the flexibility to integrate with any application environment. This solution is fully supported by NetApp and is freely available to customers, who can download it from the software download page on the NetApp Support site.

2.7 NETAPP SNAPSHOT TECHNOLOGY

NetApp Snapshot is a feature of the WAFL[®] (Write Anywhere File Layout) storage virtualization technology that is a feature of Data ONTAP, the microkernel that ships with every NetApp storage system. A Snapshot copy is a locally retained point-in-time “frozen” image of a WAFL volume that provides easy access to old versions of files, directory hierarchies, and/or LUNs (logical unit numbers). The high performance of NetApp Snapshot copies makes them highly scalable. A Snapshot copy takes only a few seconds to create, regardless of the size of the volume or the level of activity on the NetApp storage system. After a Snapshot copy has been created, changes to data objects are reflected in updates to the current version of the objects, as if the Snapshot copy did not exist. Meanwhile, the Snapshot versions of the data remain frozen. The Snapshot does not move data blocks and therefore the time needed to create a Snapshot copy is independent of the size of the database. Snapshot copies incur no performance overhead; users can store up to 255 Snapshot copies per volume on the storage system, all of which can be accessible as read-only, online versions of the data.

NetApp Snapshot can be integrated in the DB2 database backup policy. NetApp Snap Creator is a free tool that enables integrating backup policies and automating backup and restore processes for the whole environment. The tool offers many features and functions in addition to backup and restore. Snapshot copies vastly improve the frequency and reliability of backups, because they incur virtually no performance overhead and can be safely created while a database is up and running. Similarly, restore operations can be automated with Snap Creator. For more information on Snap Creator, see section [2.9](#).

NetApp SnapRestore is a feature of Data ONTAP that provides a way to restore the entire database or parts of the database to the state it was in at the point in time when any available Snapshot copy was taken. Because no data is copied, a large amount of time is saved as the file system is returned to its earlier state. The restore process can be completed in a few minutes, independently of the size of the database.

In addition, when low-impact Snapshot backups have been created frequently throughout the day, fewer transaction logs need to be reapplied as part of the recovery process, resulting in a dramatic reduction in recovery time. In other words, the mean time to recover (the time needed for restore and recovery) is reduced dramatically to several minutes, compared to several hours with conventional backup methods.

The Snapshot backups are stored on the same NetApp storage system as the database. Therefore NetApp recommends using Snapshot backups as a supplement, not a replacement, for backups to a

second location, whether backing up to disk or to tape. Although backups to a second location are still necessary, there is only a slight possibility that they will be needed for a restore and recovery. Most database restore and recovery actions can be handled by database restore using Snapshot copies. Restores from a second location (disk or tape) are necessary only if the primary storage system holding the Snapshot copies is damaged or if there is a need to restore a database from a backup that is no longer available in the form of a Snapshot copy.

For more information on using the Snapshot, SnapRestore and other NetApp products for data protection, see <http://now.netapp.com/NOW/knowledge/docs/bpg/dp/>.

2.8 NETAPP STORAGE LAYERS

Data ONTAP 7 and higher offers a storage virtualization approach that decouples volume from physical storage disk. The physical storage disks are tied to a storage layer named *aggregate*. The volumes that are virtualized are known as *flexible volumes* or NetApp FlexVol volumes. NetApp supports the following logical storage layers:

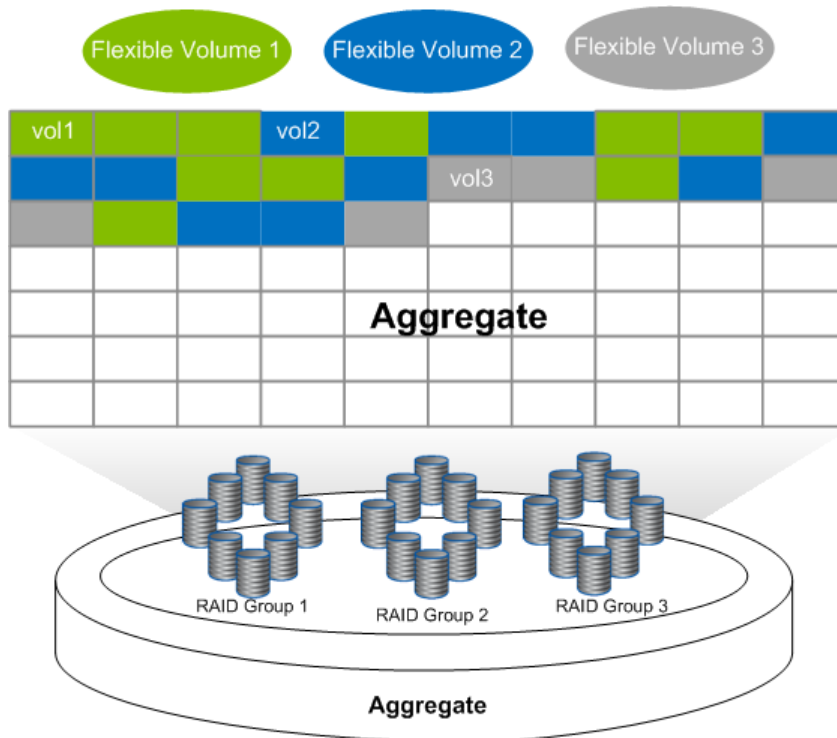
Aggregate. A pool of physical disks that adds a layer of abstraction between the physical disks and the volumes. It possesses its own RAID configuration, plex structure, and set of disks. You can create one or more flexible volumes in an aggregate.

Flexible volume (FlexVol volume). The NetApp FlexVol approach to storage virtualization allows creating one or more FlexVol volumes (virtual volumes) in an aggregate. The volume can grow or shrink according to the needs of the application. All FlexVol volumes share the physical storage resources, such as disks, RAID configuration, and plex structure of the common containing aggregate. FlexVol volumes improve the utilization of storage and save the expense of acquiring more disk space. They also improve I/O performance and reduce bottlenecks by distributing volumes across all available disk drives. NetApp Snapshot technology is available at FlexVol level. Each volume can have up to 255 Snapshot copies.

FlexVol technology offers a feature called FlexClone[®]. You can create a FlexClone volume from the FlexVol volume itself, or you can use an existing Snapshot copy from some point in the past. A FlexClone volume looks exactly like the volume it was created from (its parent FlexVol volume or another FlexClone volume), but it uses no additional physical storage. Figure 1 illustrates the logical relationship between aggregates, flexible volumes, and physical disks.

LUN (logical unit number). When dealing with SAN or iSAN (FCoE or iSCSI), you are making block allocations, and block allocations are managed by using LUNs. A LUN is just a big allocated file sitting on a volume file system. To create a LUN, you must first create a volume and then make it accessible by mapping it to an initiator group. Initiator groups contain one or more FCP WWNs or iSCSI IQNs, which map to a LUN on the NetApp storage system. The mapping process is also known as LUN masking.

Figure 1) Logical relationship between aggregate, volume, and physical disks.

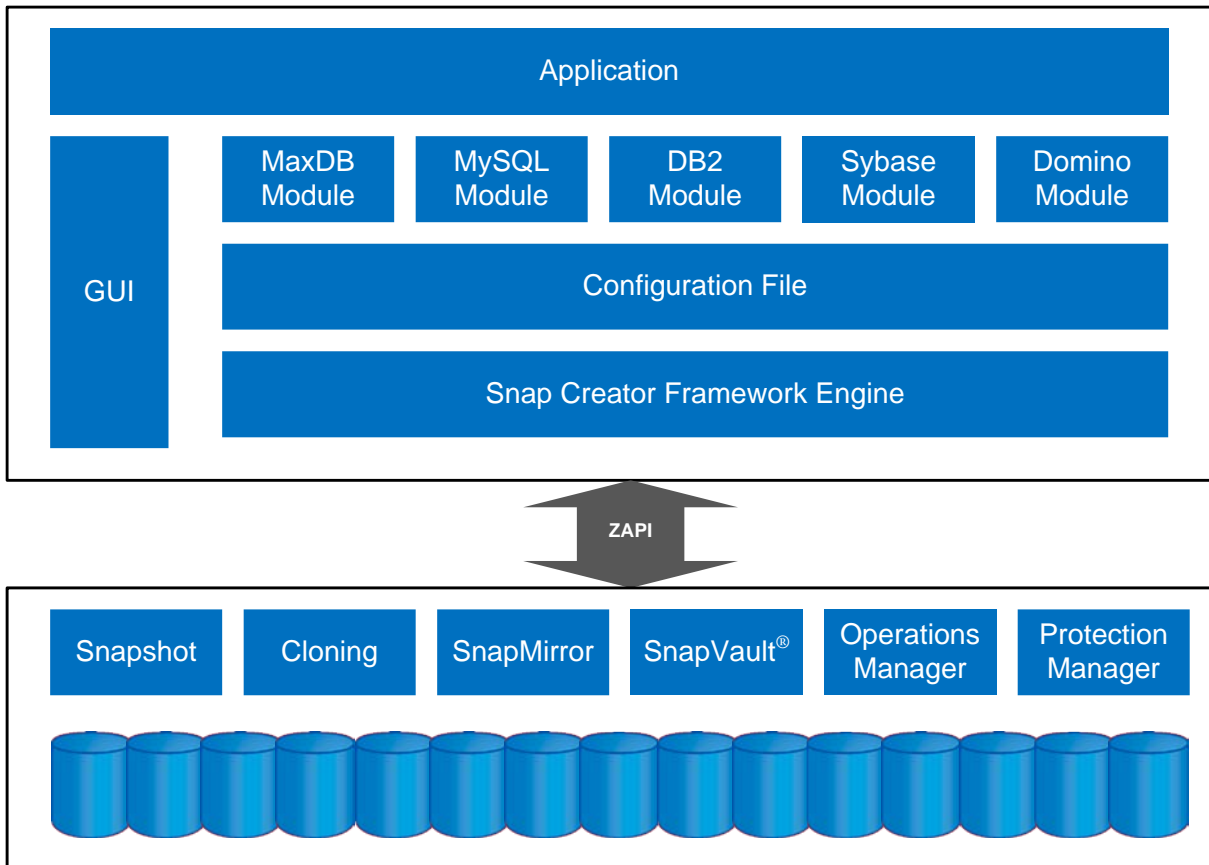


2.9 SNAP CREATOR FRAMEWORK

NetApp Snap Creator is a framework that offers rich features and allows customers to integrate DB2 and other applications and components to leverage Snapshot technology. It is a platform-independent and flexible framework that is fully supported and freely available to all NetApp customers. It can help customers to automate backup, restore, replication, cloning, and data protection processes. It can also be deployed in virtualized environments such as AIX VIOS and VMware®. It eliminates the need to create and manage custom scripts and provides a flexible and reliable solution. Snap Creator comes with default plug-ins, including a plug-in for DB2 and a set of APIs, which can be used to develop custom plug-ins. Plug-ins may be needed for application consistency and recovery of the environment. Snap Creator allows policy-based Snapshot management and role-based access, and it is integrated with a number of other NetApp products. It has a robust error-handling process, and it can send alerts to an e-mail distribution list or communicate with external monitoring systems. As shown in Figure 2, Snap Creator architecture consists of three components:

- **Snap Creator engine.** The engine is the main component of Snap Creator; it can be installed on a central backup server or on the DB2 server where the database is installed. Snap Creator can also be integrated with an existing backup architecture. Snap Creator communicates with the NetApp storage system by using ZAPI calls.
- **Snap Creator configuration file.** The configuration file is the brain behind Snap Creator, and it can be customized as needed. Snap Creator comes with a default configuration file that can be used as a template to create a configuration file for the database. More than one configuration file per database is supported, but only one file can be active at a time.
- **Application Modules (DB2 in this case).** The database modules are used for application consistency for the Snapshot copy. In the case of DB2, the application-consistent Snapshot copies are created by putting the database in an I/O suspend mode during the Snapshot creation process. Once the Snapshot is created, the DB2 modules put the database back into normal mode. For more information, see the Snap Creator Framework documentation on the [NetApp Support](#) site.

Figure 2) Snap Creator architecture and components.



3 ASSUMPTIONS

This document describes the implementation of IBM InfoSphere Warehouse Enterprise Edition in an AIX and NetApp storage system environment. It is assumed that you have basic knowledge of DB2 and the operation of NetApp storage systems. It is also assumed that you possess basic AIX administration skills. All examples in this technical report were tested using DB2 Enterprise Server Edition for Warehouse, version 9.7 with FixPak 2 or later running under AIX 6.1 TL5 and NetApp FAS6210 and FAS3270 storage systems running Data ONTAP 8.0.1.

It is also assumed that you have the documentation for InfoSphere Warehouse 9.7 that is relevant to your operating system version. You can find reference documents at the IBM DB2 information center page (<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp>) or download manuals from the IBM support site (www.ibm.com/support/docview.wss?rs=71&uid=swg27015148).

The InfoSphere Warehouse deployment in our test lab was inspired by an IBM Smart Analytics E7700 configuration that groups InfoSphere Warehouse components into modules: the management module, administration module, data module, application server module, failover and HA module, and business intelligence module.

For the configuration described in this document, it is assumed that the warehouse design that you are implementing has one management module, one administration module, and two data modules. For the components in our test environment, we used the following naming conventions.

- We used four P740 servers for testing the solution. The management module and user module were installed and configured on two servers. Two additional servers were used for the data modules.

Table 1) Server details.

Server Type	Name	Purpose
P740	mgmtnode	Management module
P740	adminnode01	Administration module
P740	datanode01	First node for the data module
P740	datanode02	Second node for the data module

- We used one FAS3270C and two FAS6210C NetApp storage systems. Each system consists of an active –active pair of controllers named netapp01, netapp02, netapp03, netapp04, netapp05, and netapp06.

Table 2) Storage system details.

Storage System	Quantity	Controller Name	Purpose
FAS3270C	1	netapp01, netapp02	Storage for adminnode01 and mgmtnode01
FAS6210C	2	netapp03, netapp04, netapp05, netapp06	Storage for datanode01 and datanode02

- The name of DB2 instance used was `db2inst1`.

4 ARCHITECTURE

At a high level, InfoSphere Warehouse architecture components are logically grouped as management, administration, data, application server, failover and HA, and business intelligence modules. In a typical production environment, each of these modules is installed on a different server to create a complete warehousing solution.

Note: Although you can install all of the component groups on a single server for proof-of-concept purposes, IBM does not recommend installing all of the component groups on a single server for typical production environments

Figures 3 and 4 illustrate the conceptual and physical designs of the InfoSphere Warehouse lab setup.

Figure 3) Conceptual design of InfoSphere Warehouse on NetApp.

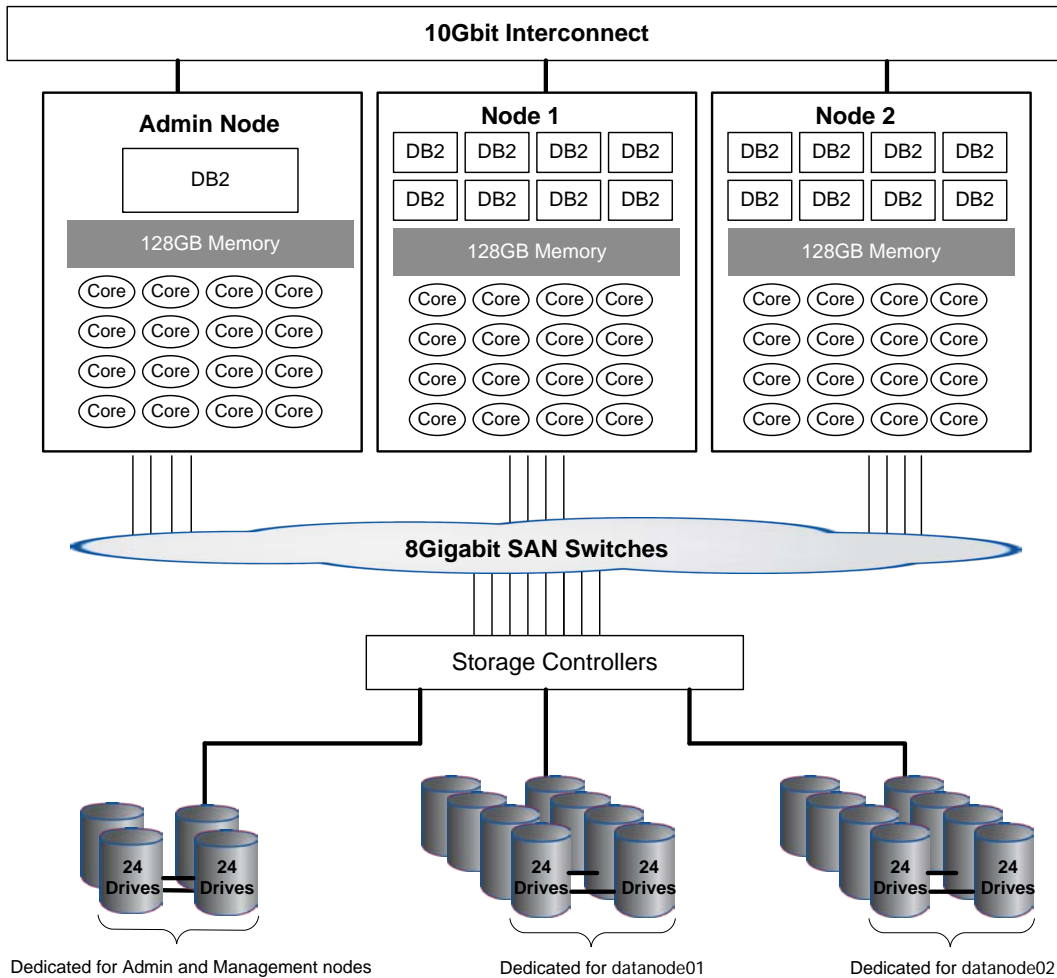
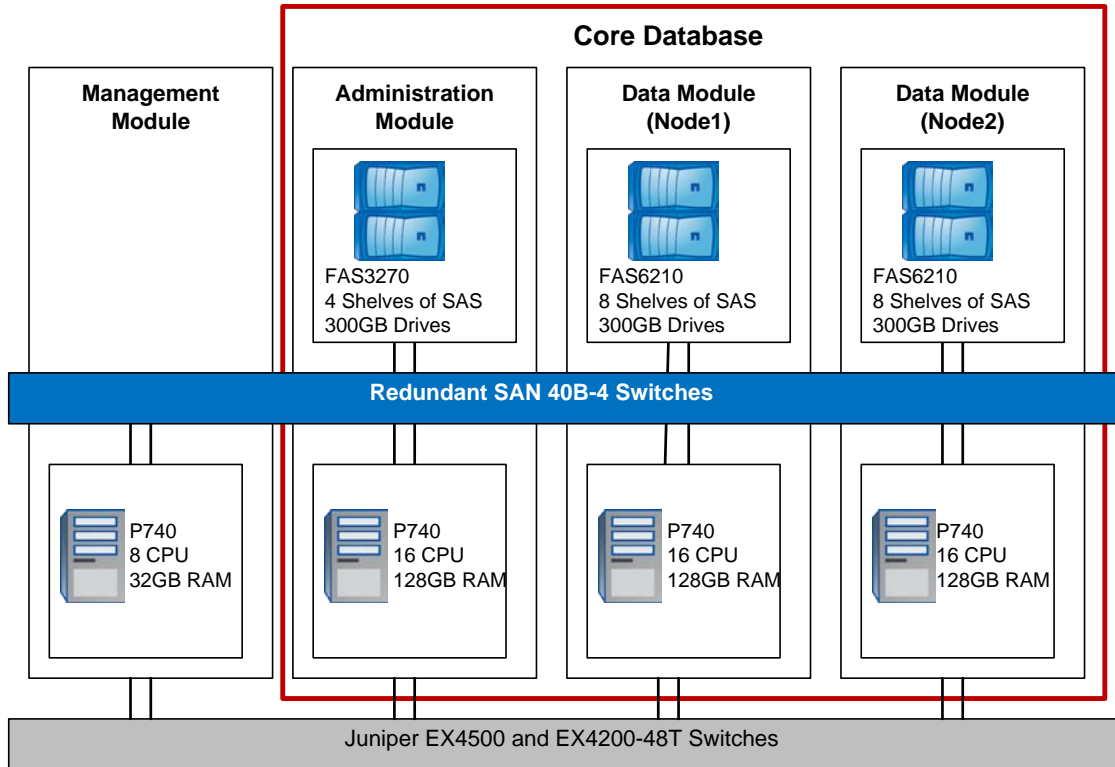


Figure 4) Configuration overview: InfoSphere Warehouse architecture.



4.1 SOFTWARE COMPONENTS

The InfoSphere Warehouse configuration we tested included the following software components from IBM.

Table 3) Software components.

Operating system software stack	<ul style="list-style-type: none"> • IBM AIX 6.1 TL6 SP1 • NFS mount for DB2 home file system. • IBM Reliable Scalable Cluster Technology (RSCT) 3.1.0.1 • IBM Systems Director Platform Control for Power 2.0 • IBM Systems Director Express Edition 6.2.0.1
IBM InfoSphere Warehouse Enterprise Edition 9.7.2	<ul style="list-style-type: none"> • IBM DB2 9.7 Enterprise Server Edition (ESE) Fix Pack 2 • DB2 Performance Expert 3.2.2 • InfoSphere Warehouse Administration Console • InfoSphere Warehouse SQL Warehousing Tool • InfoSphere Warehouse Mining Box, Cubing Services, Intelligent Miner[®], Design Studio • IBM DB2 9.7 Client and AlphaBlox[™]
Java [®] SDK	IBM Java SDK 1.6 SR7 and IBM Java SDK 1.6 SR6 (32-bit)
Tivoli [®]	IBM Tivoli System Automation for Multiplatforms 3.1.0.7
NetApp software	Data ONTAP 8.0.1 7-Mode FC Host Utilities for AIX 5.3

4.2 SERVER SPECIFICATIONS

A typical data warehouse environment is large and complex and requires high compute power to meet business needs. To build an enterprise-class InfoSphere Warehouse solution, you need IBM POWER7 systems running the AIX operating system to meet intense I/O requirements. For our test lab, we used the following server and storage configurations for each node type.

Table 4) Server specifications.

Administration Node
<ul style="list-style-type: none">▪ One IBM Power 740▪ 16 core 3.55GHz POWER7 processors (dual socket 8 core)▪ 128GB memory (32 x 4GB DIMMS)▪ One integrated dual-port 10Gb Host Ethernet Adapter (HEA)▪ One dual-port 10Gb Fibre Ethernet PCIe adapter (One of these ports is used for connectivity to the internal application network and the other is used for connectivity to the corporate network.)▪ Four dual-port 8Gbps Fibre Channel PCIe adapters for connectivity to the SAN switches▪ Two integrated Hardware Management Console (HMC) ports▪ FAS3270C storage 2 DS4243 disk shelves connected to each controller
Data Node
<ul style="list-style-type: none">▪ Two IBM Power 740▪ 16 core 3.55GHz POWER7 processors (dual socket 8 core)▪ 128GB memory (32 x 4GB DIMMS)▪ One integrated dual-port 10Gb Host Ethernet Adapter (HEA)▪ One dual-port 10Gb Fibre Ethernet PCIe adapter (One of these ports is used for connectivity to the internal application network and the other is used for connectivity to the corporate network.)▪ Four dual-port 8Gbps Fibre Channel PCIe adapters for connectivity to the SAN switches▪ Two integrated Hardware Management Console (HMC) ports▪ FAS6210C storage with two active-active controllers▪ 4 DS4243 disk shelves connected to each controller (8 disk shelves for each data node)
Management Node
<ul style="list-style-type: none">▪ One IBM Power 740▪ 8 core 3.72GHz POWER7 processors (dual socket quad-core)▪ 32GB memory (8 x 4GB DIMMS)▪ One integrated dual-port 10Gb Host Ethernet Adapter (HEA).▪ One dual-port 10Gb Fibre Ethernet PCIe adapter (One of these ports is used for connectivity to the internal application network and the other is used for connectivity to the corporate network.)▪ Four dual-port 8Gbps Fibre Channel PCIe adapters for connectivity to the SAN switches▪ Two integrated Hardware Management Console (HMC) ports▪ FAS3270C storage with two DS4243 disk shelves connected to each controller

IBM DB2 9.7 ESE for Warehouse was installed on the administration node and data nodes. WebSphere application server, DB2 client, and other necessary components were installed on the management node. The architecture allows installing the following additional nodes:

- Business intelligence
- Failover and HA

For more information on required components for these nodes, refer to the InfoSphere Warehouse information center: <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp>.

4.3 STORAGE SPECIFICATIONS

The typical workload in a data warehouse is I/O throughput intensive, with operations such as large data loads and index builds, creation of materialized views, and queries over large volumes of data. The underlying storage system for a data warehouse should be designed to meet these heavy requirements. Storage configurations for a data warehouse should be chosen based on the I/O bandwidth that they can provide, and not necessarily on their overall storage capacity. The NetApp storage system product line offers customers great flexibility and features to address these intense workload needs. For our test lab we used the following NetApp storage systems:

Table 5) Storage specifications.

Node Name	Storage Controller	Disk Shelf
Administration Node	FAS3270C	4 DS4243 with 300GB SAS drives
Management Node		
Data Node 1	FAS6210C	8 DS4243 with 300GB SAS drives
Data Node 2	FAS6210C	8 DS4243 with 300GB SAS drives

4.4 NETWORK SPECIFICATIONS

The network infrastructure for InfoSphere Warehouse is supported by 10GB Ethernet and 1GB Ethernet. The 10Gb Ethernet is created by using two Juniper EX4500 switches (internal application switches) that are uplinked together. One VLAN is created on this Ethernet to support server-to-server communications on the internal application network. These switches are uplinked to the virtual chassis described below to provide server-to-external-storage communication for the SAN switches and the NetApp storage systems.

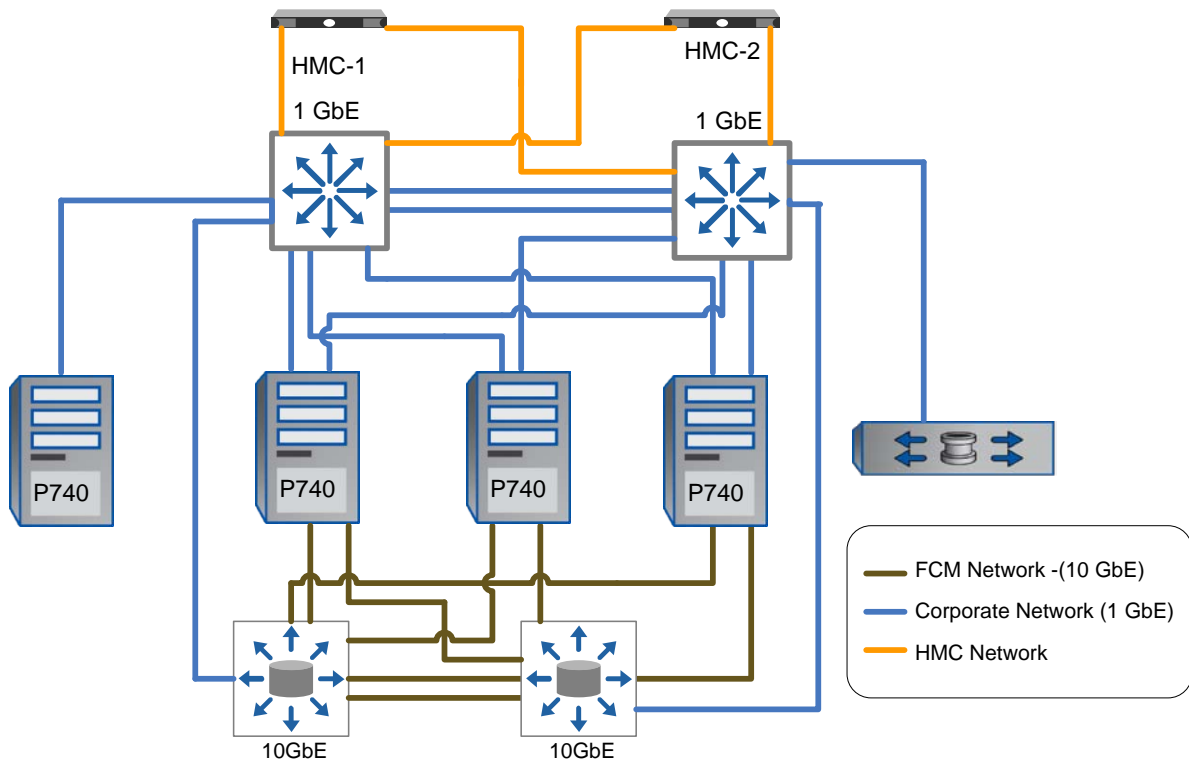
The 1Gb Ethernet fabric uses one or more Juniper EX4200-48T switches (management switches) stacked to form a virtual chassis. A virtual chassis is a feature of the Juniper EX4200-48T switch that allows multiple switches to operate as a unified single high-bandwidth switch with a 128Gbps backplane when they are interconnected. This Ethernet fabric supports external storage management functions for the NetApp storage systems, the SAN switches, and the HMCs. Figure 5 shows the network architecture discussed in this paper and used in our lab. Table 6 describes the networks that we used in the lab.

Table 6) Network specifications.

Network Name	Purpose	Implementation
Interpartition fast communication network (FCM)	<ul style="list-style-type: none"> Interpartition communications between database partitions on different physical servers (DB2 FCM) 	<ul style="list-style-type: none"> 10Gb HEA connected to 10Gb Ethernet switch 10Gb Fiber Ethernet PCIe connect to 10Gb Ethernet switch, link aggregated for ports on servers It is recommended that this network be on private network with dedicated switches

Network Name	Purpose	Implementation
Corporate network	<ul style="list-style-type: none"> This network provides connectivity between each database node and the corporate network LAN-based database backups 	10Gb Fiber Ethernet PCIe on server connect to corporate network fabric
Hardware Management Console network	<ul style="list-style-type: none"> Manages and monitors the IBM Power servers 	Created using integrated flexible service processor ports reserved for this purpose and communication runs through 1Gb Ethernet fabric

Figure 5) InfoSphere Warehouse network architecture.



Appendix 9.4 “[Network Configuration Details](#)” describes the port and IP assignments for the network and other components used for the configuration in our lab environment.

5 CONFIGURATION OVERVIEW AND CONSIDERATIONS

When setting up an InfoSphere Warehouse environment, there are many options that you must consider and plan for to achieve optimal performance, high availability, scalability, and manageability. The database system is the primary application in the system; therefore database configuration, underlying storage configuration, and IBM Power System configuration must be incorporated in the InfoSphere Warehouse planning and design phase.

5.1 SERVER CONFIGURATION

By default, AIX V6.1 provides the optimized tunable values for the best performance of the system and application. AIX Virtual Memory Manager (VMM) classifies memory into computational and noncomputational memory. Computational memory includes application memory (for example, DB2 process text, data, and stack segments) and kernel memory. Noncomputational memory is classified as file system cache, which includes file system data from JFS (Journaled File System), JFS2, NFS (Network File System), GPFS™ (General Parallel File System), or any file system type.

You must create required user accounts, create mount points, and configure HBA settings. For storage manageability, installing FC Host Utility (Host Attach Kit) for AIX is also recommended. The host attach utility can be downloaded from the software download page on the NetApp Support site.

For more information on steps for server configuration and installing host attach, refer to [TR-3772: IBM DB2 on NetApp: Deployment and Best Practice Guide](#).

5.2 DATABASE DESIGN

DATABASE PARAMETERS

IBM DB2 offers a large number of registry, database manager, and database parameters that allow DBAs to tweak the environment based on business requirements. Performance is a key requirement for all database environments. Based on customer experience and lab tests, NetApp and IBM recommend changing the default setting for some of the performance-related parameters. The recommended parameter settings are good for most data warehouse environments. Database performance should be monitored regularly, because the recommended setting may need further tweaking based on workload and application changes. The following tables describe the recommended settings for DB2 registry, database manager, and database configuration parameters.

Table 7) DB2 registry variable settings.

Parameter Name	Default Value	Recommended Value	Purpose
DB2_EXTENDED_OPTIMIZATION	Off	ON	Specifies whether or not the query optimizer uses optimization extensions to improve query performance.
DB2_ANTIJOIN	NO	EXTEND	When EXTEND is specified, the optimizer searches for opportunities to transform both NOT IN and NOT EXISTS subqueries into anti-joins.
DB2COMM		TCPIP	Sets communication protocols for the current DB2 instance to TCPIP.
DB2_PARALLEL_IO		*	Indicates that all table spaces have parallel I/O set to ON.

Parameter Name	Default Value	Recommended Value	Purpose
DB2RSHCMD		/usr/bin/ssh	Specifies that the ssh remote shell program is used for all remote commands instead of the default rsh remote shell program.

Table 8) Database manager configuration parameters.

Parameter Name	Default Value	Recommended Value	Purpose
CPUSPEED	-1	2.7e-07	Specifies the CPU speed, in milliseconds per instruction. Used by the query optimizer to estimate the cost of performing certain operations.
COMM_BANDWIDTH	-1	100	Specifies the value for the communications bandwidth, in MB/sec. Used by the optimizer to estimate the cost of performing certain operations.
NUMDB	8	1	Specifies the number of local databases that can be concurrently active.
DIAGPATH		/db2home/asupwh/d b2dump Where asupwh is the instance name	Provides an alternate path for diagnostic files. The default path is on the NFS-mounted db2home instance directory.
SYSADM_GROUP			Defines the group name with SYSADM authority.
SHEAPTHRES	0	1400000	
SVCENAME			Specifies the TCP/IP port for the environment.
FCM_NUM_BUFFERS	AUTOMATIC	16384 AUTOMATIC	Specifies memory allocation for FCM. FCM is used for efficient communication between database partitions.
HEALTH_MON	ON	OFF	Specifies health monitor status.

Table 9) Database configuration parameters.

Parameter Name	Default Value	Recommended Value	Purpose
LOCKLIST	AUTOMATIC	16384	Defines the amount of storage allocated for the lock list.
MAXLOCKS	AUTOMATIC	10	Defines the percentage of the lock list held by an application that must be filled before the DB manager performs lock escalation.
PCKCACHESZ	AUTOMATIC	-1	Specifies the memory allocation for caching.

Parameter Name	Default Value	Recommended Value	Purpose
SORTHEAP	AUTOMATIC	35000	Defines the maximum number of pages to be used for sorts.
LOGBUFSZ	8	2048	Specifies the amount of the database heap to use for buffering for log records.
UTIL_HEAP_SZ	5000	65536	Specifies the amount of memory that can be used by utilities such as backup, restore, and load.
STMTHEAP	AUTOMATIC	10000	Specifies the size of the statement heap.
LOGFILSIZ	1000	12800	Defines the size of the primary and secondary log files.
LOGPRIMARY	3	50	Defines the number of primary log files.
LOGSECOND	2	0	Defines the number of secondary log files.
NEWLOGPATH			Allows changing the location of log files.
MIRRORLOGPATH			Specifies the location for the mirror log path.
NUM_IOSERVERS	AUTOMATIC	12	Specifies the number of I/O servers for the DB.
NUM_IOCLEANERS	AUTOMATIC	7 adminnode, AUTOMATIC for others	Specifies the number of asynchronous page cleaners for the database.
DFT_PREFETCH_SZ	AUTOMATIC	384	Determines the default prefetch size of table spaces.
CHNGPGS_THRESH	60	30	Specifies the level (%) of changed pages at which the asynchronous page cleaners are started.
WLM_COLLECT_INT	0	20	Specifies a collect and reset interval, in minutes, for workload management.

Any value not listed in these tables should be left at its default value. Use these values as a starting point, rather than using the defaults, and tune your environment explicitly for your particular workload.

Appendix [9.1](#) describes database manager configuration parameter settings, and Appendix [9.2](#) describes the database configuration parameter for our test environment.

TRANSACTION LOGGING

Database transaction logs play a critical role in database recovery. Therefore, logging strategy for a database must be closely aligned with the backup and restore strategy that is chosen. IBM DB2 database uses two strategies for managing log files: *circular logging* and *archival logging*. When circular logging is used, a finite set of log files is constantly reused. Because the size and number of log files used are configurable, storage space requirements can be calculated and/or controlled. However, when circular logging is used, roll-forward recovery and user exit programs are not supported. For our test environment, we used circular logging.

When archival logging is used, log files are retained after they have been populated and can be used to perform roll-forward recovery operations on the database. Because the number of log files used is determined by the volume of database activity, the amount of storage needed for archival logging cannot

be predetermined. For OLTP environments, NetApp and IBM strongly recommend enabling archive logging.

Best Practice

For an archive logging enabled database, you must place transaction logs on a separate volume from the data and system catalog volume. NetApp and IBM also recommend placing archive logs on their own separate volume.

BUFFER POOL DESIGN

The buffer pools are used by database manager to cache data in database memory. For every table space page size used, there must be at least one buffer pool with the same page size. Like table spaces, buffer pools can have a page size of 4KB, 8KB, 16KB, or 32KB. In a warehouse, the database manager effectively manages data, index, and temporary pages in a single buffer pool. Frequently accessed data such as dimension tables and lookup tables is often stored on the administration node, where the buffer pool hit ratio is typically high. If this type of frequently accessed data is also stored on data nodes, a separate buffer pool might improve performance.

Contrary to OLTP, data warehouse reads are sequential in nature. To improve sequential read and table-scan performance, NetApp and IBM recommend creating a block-based buffer pool. A block-based buffer pool consists of a block area and a page area. The block area consists of blocks, where each block contains a specified number of contiguous pages. It is used for sequential prefetching workloads only. The page area is used for all other I/O workloads.

When a block-based buffer pool is configured, the DB2 prefetchers perform read requests of contiguous blocks of pages on disk into contiguous pages in memory. Prefetching using block I/O is more efficient than the page-based vectored I/O used by the DB2 software by default. The prefetchers can also increase the efficiency of buffer pool reuse by reducing buffer pool thrashing when large volumes of data are read.

To handle dirty page lists proactively, the value of the database configuration parameter `CHNGPGS_THRESH` was changed to 30 from its default value of 60. Lowering this parameter value triggers cleaning the dirty page more frequently and improves overall database performance, including some write-intensive database utilities and applications.

TABLE SPACE

IBM DB2 table space is a logical layer that is used to organize and place database object (table, index, and so on) data on physical storage disks. The number of table spaces needed depends on the size and performance requirements of the warehouse. Table 10 describes some important guidelines for database object placement:

Table 10) Table space considerations.

Object Type	Table Space Recommendation
Large tables	Place in a separate table space across the data nodes.
Medium-sized tables	Logically group together and place into table spaces with five or more tables in each table space. These table spaces are placed across the data nodes.
Medium-sized table indexes	Place within the data table space or separate into another shared table space for indexes.
Small tables	Place many tables into one single-partition table space on an administration node.
Small table Indexes	Place into the same single-partition data table space on an administration node.

Object Type	Table Space Recommendation
Replicated tables	These tables are used for joins with tables in multiple partitions and are replicated across the partition. NetApp and IBM recommend placing these tables into a separate table space.
Materialized query tables	Place in separate table spaces that are placed across the data nodes.
Large table index	Place in a separate, dedicated table space.
Range-partitioned tables	Use one table space per range for large tables.

DATABASE PERFORMANCE

IBM DB2 comes with Self-Tuning Memory Manager (STMM). STMM is on by default; it can be turned off. This DB2 feature simplifies the task of memory configuration by automatically setting values for several memory configuration parameters. When enabled, the memory tuner dynamically distributes available memory resources among the buffer pools, locking memory, package cache, and sort memory. The STMM tuning is good for most environments. If your environment needs specific tuning, you can tweak it by using database manager database configuration parameters, along with several registry variables, for optimum performance.

`DB2_PARALLEL_IO` is one of the most important variables for performance. Set this variable when DB2 is used in conjunction with a NetApp storage system. (The system command `db2set` is used to display, set, or remove values for DB2 registry and environment variables. After setting any registry variable, the DB2 database manager must be stopped and restarted before the changes take effect.)

The `DB2_PARALLEL_IO` Registry Variable

When reading data from or writing data to table space containers, DB2 can use parallel I/O if the number of containers in the database is greater than 1. However, there are situations in which it would be beneficial to have parallel I/O enabled for single container table spaces. For example, if the container is created on a NetApp storage system volume or `qtree`, performance may be improved if read and write calls are issued in parallel.

To force DB2 to use parallel I/O for a table space that has only one container, use the `DB2_PARALLEL_IO` registry variable. This variable can be set to asterisk (*), meaning that every table space in every database is to use parallel I/O; or it can be set to a comma-separated list of table space IDs. For example, to turn parallel I/O on for all table spaces, execute the following command:

```
$ db2set DB2_PARALLEL_IO=*
```

The following command can be used to turn parallelism on for selected table spaces:

```
$ db2set DB2_PARALLEL_IO=1,2,4,8
```

The DB2 Configuration Advisor

As mentioned earlier, IBM DB2 provides two sets of configuration parameters, one for database manager and one that can be used to tune a database for optimum performance. A database that uses the default database configuration parameter values may require additional tuning. So if your experience with DB2 is limited, how do you begin to tune your database by modifying the values of these configuration parameters?

The solution is to use the DB2 Configuration Advisor. This tool asks you a series of questions about your database environment and uses your answers to recommend configuration parameter values that will improve overall database performance. The Configuration Advisor is accessed from the DB2 Control Center.

5.3 STORAGE SYSTEM CONFIGURATION

Storage design for InfoSphere Warehouse is based on the principle that every data module has a set of dedicated physical disks for data storage, and that all database objects in the data warehouse are distributed equally across these physical disks. This modular storage design means that all database objects within each data node have equal access to I/O transaction performance available within each data node.

NetApp FAS storage systems are designed to meet scalability, high-availability, and flexibility requirements for demanding workload environments. IBM DB2 is thoroughly tested and verified on NetApp FAS systems in IBM and NetApp labs. NetApp RAID-DP architecture offers a high degree of resilience and also leverages a large spindle count to deliver better performance. NetApp's lab tests have proven that large aggregates deliver better performance compared to multiple smaller aggregates on the same number of physical storage disks.

The storage layout for our test environment was designed keeping resilience, workload balance, and performance in mind. We created a single large aggregate on each storage controller. To balance the workload, we created FlexVol volumes and LUNs for each database partition on both controllers of the storage system cluster. At the server level, we created volume groups using LUNs from each controller; that is, each volume group was spread across both controllers. To further organize the data, we created logical volume groups and file systems on the servers.

Our lab environment was targeted to meet an 8GB/sec throughput requirement for a 30TB size warehouse. We used two NetApp FAS6210 pairs in active-active cluster mode for data nodes. Each storage controller was connected to four DS4243 disk shelves. For the administration and management nodes we used a FAS3270 cluster with four DS4243 disk shelves. Each disk shelf was populated with 24 x 300GB 15K rpm SAS drives. Table 11 summarizes the technical specifications of the storage controllers used in our test lab.

Table 11) Storage controller specifications.

	FAS6210	FAS3270
RAM	48GB	32GB
Onboard 10GbE	8	4
Onboard 8Gb FC	8	8
Onboard GbE	8	4
Storage networking supported	FC, FCoE, iSCSI, NFS, CIFS, FTP, HTTP	FC, FCoE, iSCSI, NFS, CIFS, FTP, HTTP
OS versions	Data ONTAP 8	Data ONTAP 8

For detailed technical specifications for FAS6210 and FAS3270, see the NetApp data storage systems Web page: www.netapp.com/us/products/storage-systems/.

ADMINISTRATION AND MANAGEMENT NODE

In our test lab, the administration node and management node shared the storage and were connected to FAS3270 storage controllers. In a production deployment, you might want to deploy the administration node on its own storage controller. A large 64-bit aggregate was created on each storage controller. Table 12 describes FlexVol volumes, volume groups, and file system settings for the test lab on the administration node.

Table 12) Administration node specifications.

Storage System	Aggregate Name	Volume Name	LUN	Volume Group	Logical Volume	File System
netapp01	db2aggr	admvol	/vol/admvol/db2home	vgadm	lvdb2home	/db2home/
netapp02	db2aggr	admvol	/vol/admvol/db2home		Lvhome	/home
					Lvstage	/stage
netapp01	db2aggr	datavol	/vol/datavol/dbdata	vgdb	lvadmdata	/db2data/asupwh/
netapp02	db2aggr	datavol	/vol/datavol/dbdata		lvadmlog	/db2log/asupwh/NODE0000

IBM DB2 9.7 ESE was installed and a DB2 instance named `asupwh` was created. The catalog partition was configured on the administration node. To organize data, we created file systems for `db2home`, `data`, and logs. Table 13 describes the purpose of file systems on the administration node.

Table 13) File system name and purpose on the administration node.

File System Name	Purpose	NFS Shared Across All Nodes
/db2home	Used for the DB2 instance home directory.	Yes
/home	Used as a home directory for all users.	Yes
/stage	Used for staging tables, flat files, and other purposes.	Yes
/db2data/asupwh/	Used for storing all permanent table space containers for catalog partition or node 0. Also contains the catalog files for the database and the diagnostic data directory for node 0.	No
/db2log/asupwh/	Holds the primary logs for the catalog partition or node 0.	No

The management node hosts InfoSphere Warehouse applications such as the InfoSphere Warehouse Administration Console, the SQL Warehousing Tool, MiningBox, and Cubing Services. We created storage objects such as volumes and LUNs for the management node on each controller.

Table 14) Management node specifications.

Storage System	Aggregate Name	Volume Name	LUN	Volume Group	Logical Volume	File System
netapp01	db2aggr	mgmtvol	/vol/mgmtvol/lun01	vgmgmt	lvmgmtapp	/mgmtapp
netapp02	db2aggr	mgmtvol	/vol/mgmtvol/lun01		lvbkups	/bkups

Table 15 describes the purpose of file systems created for management node.

Table 15) File system name and purpose on the management node.

File System Name	Purpose	GPFS Shared Across All Nodes
/mgmtapp	Used to install management software applications and tools.	No
/bkups	Used as a directory for system-level backups.	No

DATA NODE

We used two data modules for our test lab. Each data module was installed on a Power 740 server with 16 cores and 128GB RAM. For storage, each data node was connected to a FAS6210 active-active cluster pair and each storage controller was attached to four DS4243 disk shelves. Each disk shelf had 24 300GB 15K rpm disks. To take advantage of large spindle count, we created a large aggregate on each controller with 88 disks and RAID group size 22 disks, and used RAID-DP. Tables 16 and 17 describe the storage layout for the data nodes.

STORAGE LAYOUT FOR FIRST DATA NODE

Table 16) Storage layout, first data node specifications.

Storage System	Aggregate Name	Volume Name	LUN	Volume Group	Logical Volume	File System
netapp03	db2aggr	datavol01	/vol/datavol01/dbdata	Vgdb01		
					lvdbdata01	/dbdata01
netapp04	db2aggr	datavol01	/vol/datavol01/dbdata		lvdblog01	/dblog01
netapp03	db2aggr	datavol02	/vol/datavol02/dbdata	Vgdb02		
					lvdbdata02	/dbdata02
netapp04	db2aggr	datavol02	/vol/datavol02/dbdata		lvdblog02	/dblog02
netapp03	db2aggr	datavol03	/vol/datavol03/dbdata	Vgdb03		
					lvdbdata03	/dbdata03
netapp04	db2aggr	datavol03	/vol/datavol03/dbdata		lvdblog03	/dblog03
netapp03	db2aggr	datavol04	/vol/datavol04/dbdata	Vgdb04		
					lvdbdata04	/dbdata04
netapp04	db2aggr	datavol04	/vol/datavol04/dbdata		lvdblog04	/dblog04
netapp03	db2aggr	datavol05	/vol/datavol05/dbdata	Vgdb05		
					lvdbdata05	/dbdata05
netapp04	db2aggr	datavol05	/vol/datavol05/dbdata		lvdblog05	/dblog05
netapp03	db2aggr	datavol06	/vol/datavol06/dbdata	Vgdb06		
					lvdbdata06	/dbdata06
netapp04	db2aggr	datavol06	/vol/datavol06/dbdata		lvdblog06	/dblog06

Storage System	Aggregate Name	Volume Name	LUN	Volume Group	Logical Volume	File System
netapp03	db2aggr	datavol07	/vol/datavol07/dbdata	Vgdb07		
					lvdbdata07	/dbdata07
netapp04	db2aggr	datavol07	/vol/datavol07/dbdata		lvdblog07	/dblog07
netapp03	db2aggr	datavol08	/vol/datavol08/dbdata	Vgdb08		
					lvdbdata08	/dbdata08
netapp04	db2aggr	datavol08	/vol/datavol08/dbdata		lvdblog08	/dblog08

STORAGE LAYOUT FOR SECOND DATA NODE

Table 17) Storage layout, second data node specifications.

Storage System	Aggregate Name	Volume Name	LUN	Volume Group	Logical Volume	File System
netapp05	db2aggr	datavol09	/vol/datavol09/dbdata	Vgdb09		
					lvdbdata09	/dbdata09
netapp06	db2aggr	datavol09	/vol/datavol09/dbdata		lvdblog09	/dblog09
netapp05	db2aggr	datavol10	/vol/datavol10/dbdata	Vgdb10		
					Lvdbdata10	/dbdata10
netapp06	db2aggr	datavol10	/vol/datavol10/dbdata		Lvdblog10	/dblog10
netapp05	db2aggr	datavol11	/vol/datavol11/dbdata	Vgdb11		
					Lvdbdata11	/dbdata11
netapp06	db2aggr	datavol11	/vol/datavol11/dbdata		Lvdblog11	/dblog11
netapp05	db2aggr	datavol12	/vol/datavol12/dbdata	Vgdb12		
					Lvdbdata12	/dbdata12
netapp06	db2aggr	datavol12	/vol/datavol12/dbdata		Lvdblog12	/dblog12
netapp05	db2aggr	datavol13	/vol/datavol13/dbdata	Vgdb13		
					Lvdbdata13	/dbdata13
netapp06	db2aggr	datavol13	/vol/datavol13/dbdata		Lvdblog13	/dblog13

Storage System	Aggregate Name	Volume Name	LUN	Volume Group	Logical Volume	File System
netapp05	db2aggr	datavol14	/vol/datavol14/dbdata	Vgdb14	Lvdbdata14	/dbdata14
netapp06	db2aggr	datavol14	/vol/datavol14/dbdata		Lvdblog14	/dblog14
netapp05	db2aggr	datavol15	/vol/datavol15/dbdata	Vgdb15	Lvdbdata15	/dbdata15
netapp06	db2aggr	datavol15	/vol/datavol15/dbdata		Lvdblog15	/dblog15
netapp05	db2aggr	datavol16	/vol/datavol16/dbdata	Vgdb16	Lvdbdata16	/dbdata16
netapp06	db2aggr	datavol16	/vol/datavol16/dbdata		Lvdblog16	/dblog16

Each data node has eight partitions created on it, and each data partition has four file systems created for it. Table 18 describes the purpose of each file system.

Table 18) Storage layout, file system details.

File System Name	Purpose	NFS Shared Across All Nodes
/db2home	Used for the DB2 instance home directory for the database partition.	Yes
/dbdataNN	Used for storing all permanent table space containers for data module partitions. Also contains the catalog files for the database and the diagnostic data directory for partition and is used for temporary table spaces.	No
/dblogNN	Holds the primary logs for the database partition NN, where NN is the database partition number.	No
/stage	Used as workspace for the database partition NN.	Yes

APPLICATION SERVER MODULE

The WebSphere application server was installed and configured on the application server node.

Table 19) Application server module details.

Storage System	Aggregate Name	Volume Name	LUN	Volume Group	Logical Volume	File System
netapp01	db2aggr	appvol	/vol/appvol/lun01	vgapp	lvappsrvr01	/IBM/dwe/appserver01
netapp02	db2aggr	appvol	/vol/appvol/lun01		lvishome	/ishome
					lvappsrvr02	/IBM/dwe/appserver02

Table 20 describes the purpose of file systems created for the management node.

Table 20) Application server module, file system details.

File System Name	Purpose	Shared Across All Nodes	Size (GB)
/IBM/dwe/appserver01	Used for installation and working directory for the InfoSphere Warehouse software components.	No	150
/IBM/dwe/appserver02	Used for the binary files for the Cubing Services and other components.	No	150
/ishome	Used as the home directory for the InfoSphere Warehouse administrative user.	No	100

The storage layout described in Tables 12, 14, 16 and 17 was designed keeping NetApp Snapshot based backup and recovery in mind. This layout allows customers to back up their data warehouse database by using NetApp Snapshot technology, and it also leverages Snapshot for replication and cloning.

Note: The storage configuration described in this section assumes that the database is in circular logging mode. Contrary to most warehouse environments, if your business requires using archive logging, consider storing data and logs on separate LUNs on separate volumes. This will simplify Snapshot based backup and recovery of your database.

SECURITY AND ACCESS

Security and access settings are designed according the security policies of each organization. We used the simplest approach and created DB2 user accounts on the NetApp storage system. You can add users by using NetApp System Manager or the CLI on the NetApp storage system. You can download System Manager from the NetApp Support site. For more information on how to use this tool, refer to the appropriate software documentation page, which is also available on the documentation page on the NetApp Support site.

In addition to adding users, you must apply licenses and configure rsh/ssh access. For more information, see [TR-3772: IBM DB2 on NetApp: Deployment and Best Practice Guide](#).

5.4 VOLUME CONSIDERATION

The basic logical data container in DB2 for table data is the table space, which defines the physical relationship of the database objects and the storage on which they reside. A table space consists of one or more containers. A container can be a directory, a file, or a raw device. If a single table space spans more than one container, the DB2 database manager uses parallelism to balance the data load across all containers.

NetApp FlexVol volumes make it easy to manage the placement of database objects on storage system volumes. By placing table space containers in a FlexVol volume, a single Snapshot copy of the storage system volume can be used to back up multiple DB2 objects. For more information on creating and managing FlexVol volumes, refer to the `vo1` command in the Data ONTAP Commands manual documentation, which is available on the NetApp [Support](#) site.

When placing a DB2 database on NetApp storage, there are some important design considerations:

- The root volume should be its own volume. The root volume usually contains data that doesn't change much over time, so it doesn't need to be backed up as often. If a data volume fails, having a separate, still-functioning root volume saves valuable time in the recovery process.
- NetApp and IBM strongly recommend storing all database object files on a separate volume on the NetApp storage system that is not the root volume.
- If NetApp SnapMirror[®] software is used in the backup-and-recovery scenario, it is important to note that SnapMirror is available only to non-root volumes.
- The transaction logs should be kept on a separate storage system volume from the database objects, not on the root volume. This practice decreases the loss of data after a database failure, due to the availability of more current logs. If the transaction logs and the database reside on the same volume, the recovered volume would contain the active logs at the time the Snapshot copy was taken. No log data after the Snapshot copy would be available, causing a loss of data because of the lack of logs available for roll-forward operations.

6 INSTALLATION AND CONFIGURATION

Before you begin installing InfoSphere Warehouse, make sure that the servers you have selected to install each of the components on have the required hardware and software and that you have the correct user authorities to install the product. The installation requirements for data server components, application server components, and client server components are described on the IBM Web site: http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.dwe.install.doc/install_intro.html.

6.1 INSTALLATION

BEFORE YOU START INSTALLATION

You can download InfoSphere Warehouse software images from the [Passport Advantage user Web page](#) to the server, or you can get software disks for the product. In either case, you must have the activation CD, which contains the license profile for the edition that you purchased. Before starting the installation program, you must copy the contents of the activation CD to a directory on the server on which you are installing the product. This directory is called the `profile` directory.

The installation programs install InfoSphere Warehouse in a default directory. In addition, some files that can be used by other compatible products are installed in a separate default directory called the shared resources directory.

Installation Directory

The installation programs install the components of InfoSphere Warehouse in the following default directory:

- With root authority: `/opt/IBM/ISWarehouse`
- Without root authority: `user_home/ISWarehouse`

Shared Resources Directory

The shared resources directory is used by the client component group. If the server on which you install the client components does not already have a shared resources directory, the installation program creates that directory in the following default location:

- With root authority: `/opt/IBM/ISWShared`
- Without root authority: `user_home/ISWShared`

Shared Development Environments (Shell Sharing)

In InfoSphere Warehouse, you can share a common development environment with other compatible Eclipse-based IBM products. This is called shell share, and it allows installation of multiple Eclipse-based products into the same package group so that the combined functions of those products appear in the same Eclipse instance, using the same workspace.

Note: The shared resources directory has nothing to do with shell sharing. The shared resources directory is created to store files that are common across certain IBM products regardless of whether you choose to use shell share. The main purpose of the shared resources directory is to save space by installing only one copy of files that are used by multiple products.

Warehouse Repository Database

InfoSphere Warehouse requires a repository database for storing warehousing metadata. You can create the database before starting the installation, or the installation program can create it. The default name of this database is `SQWCTRL`. The metadata database must be created in a nonpartitioned database environment. First you create a nonpartitioned instance and then you create the database in that instance.

RUNNING THE INSTALLATION

InfoSphere Warehouse offers the following installation methods.

- GUI-based launch pad
- GUI-based installation wizards
- Silent installation on a command line

If you want to install only a specific component product, you can run the installation program for that particular component. To produce this document and test the solution, we used the launch pad method to install InfoSphere Warehouse components.

6.2 DB2 ADMINISTRATOR AND USER ACCOUNTS

If you already have DB2 installed on your UNIX[®] server, it is likely that the appropriate DB2 user accounts already exist. If that is the case, the home directories for these accounts should be moved to a storage system volume so that all remaining servers can access them if one server in the configuration goes down. If you are installing DB2 for the first time on the AIX server, you can create a DB2 administrator and one or more user accounts as part of the DB2 installation process. For instructions on setting up the appropriate accounts during the installation process, refer to the IBM DB2 documentation.

6.3 INSTALLING DB2

IBM DB2 offers multiple installation methods, including setup wizard, response file, CLI (`db2_install`) and payload file deployment. The DB2 Setup wizard is an easy-to-use interface for installing DB2 products and for performing initial setup and configuration tasks. This method also creates DB2 instances and response files that can be used to duplicate the installation on other machines. To produce this document, we used `db2_install` command line installation method option. Although the `db2_install` command installs all components for the DB2 product that you specify, it does not perform user and group creation, instance creation, or configuration. The instance and database creation and configuration are done after the DB2 software is installed.

By default, the installation location is `/opt/IBM/db2/V9.7` for AIX, HP-UX, and Oracle Solaris platforms. The default location for Linux installation is `/opt/ibm/db2/V9.7`. You can change the default location by specifying a fully qualified path.

6.4 CREATING A DB2 HOME FILE SYSTEM ON A NETAPP STORAGE SYSTEM

Part of the DB2 ESE for warehouse installation process involves creating a DB2 home file system (or directory) on the administration server and making that location an NFS mount point on all the other nodes. The disadvantage of this configuration is that if the primary server goes down, a secondary server cannot take over because it cannot access the instance data. A better approach is to create the DB2 home file system on a NetApp storage system volume and make that location an NFS mount point to all nodes involved in the DPF configuration, including the primary server. With this approach, the only change in the installation process is to mount the storage system volume on which the `db2home` file system is stored to each server and specify that location as the location to be used as part of the installation process.

7 CREATING DB2 INSTANCES AND DATABASES ON A NETAPP STORAGE SYSTEMS

The InfoSphere Warehouse integrated installer with the response files option does not create the instance; you should follow the instructions in section 7.1 to create the instance. However, if you used the integrated installer interactively, the installer should have created the instance based on the inputs you provided for the instance owner, the fenced owner, and the administration instance owner. If the instance is not created, then you can create and configure it manually by using the steps described in section 7.1.

7.1 CREATING A DB2 INSTANCE

Creating a DB2 instance in a NetApp storage system environment is a very straightforward process that uses standard DB2 commands. This section describes the steps to create a new DB2 instance in a NetApp storage environment.

1. Log in to the administration node that acts as the NFS server for the DB2 instance home directory and create an instance by using the `db2icrt` command:

```
/opt/IBM/db2/V9.5/instance/db2icrt -u fenced_id instance_name
```

On the AIX platform, the instance name must be the same as the name of the instance owning user. For example, to create an instance named `asupwh`, you would issue the following command:

```
/opt/IBM/db2/V9.7/instance/db2icrt -u asupfenc asupwh
```

Note: Make sure that the NFS home directory has been set up to allow root to write to the home directory, at least for instance creation.

2. After creating an instance, update the following registry parameters.

Parameter Name	Recommended Value	Command
DB2_EXTENDED_OPTIMIZATION	ON	db2set DB2_EXTENDED_OPTIMIZATION=ON
DB2_ANTIJOIN	EXTEND	db2set DB2_ANTIJOIN= EXTEND
DB2COMM	TCPIP	db2set DB2COMM=tcPIP
DB2_PARALLEL_IO	*	db2set DB2_PARALLEL_IO=*
DB2RSHCMD	/usr/bin/ssh	db2set DB2RSHCMD=/usr/bin/ssh

- Next you need to update the database manager configuration parameter by using the `update dbm config` command. For example, to update the `svcename` parameter, using the value listed in the `/etc/services` file, you would execute the following command.

```
db2 update dbm config using svcename xasupwh
```

Where the instance name is `asupwh` and the service name is `xasupwh`.

- The fault monitor daemon that is controlled by DB2 Fault Monitor Controller Utility (FMCU) is not required for Info Sphere warehouse, and it is recommended to turn it off. You can stop the fault monitor and prevent the FMCU from launching on any node by executing the following command:

```
db2fm -i asupwh -D
```

```
db2fmcu -d
```

The `db2fmcu` command must be run as root because it accesses the system's `inittab` file. Run the `db2fmcu` command on all servers.

Note: If you apply a DB2 Fix Pack, the `inittab` file will again be configured to include the fault monitor coordinator. To prevent the fault monitor coordinator from being launched after you have applied a Fix Pack, you must reissue the `db2fmcu -d` command described in step 4.

- By default, the diagnostic files are created in the `~/sql1lib/db2dump` directory, which is on the NFS-mounted `db2home` instance directory. For InfoSphere Warehouse, NetApp and IBM strongly recommend placing the diagnostic files in storage that is not NFS-mounted. Create a directory for the diagnostic files and a symbolic link for the directory on each server.

Node Name	Command to Create Directory	Symbolic Link for Directory
Adminnode	<code>mkdir -p /db2home/asupwh/NODE0000/db2dump</code>	<code>ln -s /db2home/asupwh/NODE0000/db2dump /db2home/asupwh/db2dump</code>
datanode1	<code>mkdir -p /db2home/asupwh/NODE0001/db2dump</code>	<code>ln -s /db2home/asupwh/NODE0001/db2dump /db2home/asupwh/db2dump</code>
datanode2	<code>mkdir -p /db2home/asupwh/NODE0009/db2dump</code>	<code>ln -s /db2home/asupwh/NODE0009/db2dump /db2home/asupwh/db2dump</code>

To place the diagnostic file to a separate directory, you need to update the `diagpath` database manager parameter for the instance by using the following command:

```
db2 update dbm config using diagpath /db2home/bcuaix/db2dump
```

- IBM recommends disabling AIX medium page (64K). To disable support for medium pages, add the following lines to the instance owner's `sql1lib/userprofile` file:

```
export LDR_CNTRL=DATAPSIZE=4K;
```

```
export DB2ENVLIST=LDR_CNTRL;
```

- Set the remaining database manager configuration parameters for the instance as described in Appendix [9.1](#).

7.2 UPDATING THE DB2NODE.CFG FILE

The database partitions and servers mapping is managed in the `db2nodes.cfg` file, which is found in the instance home directory (`~/sqlllib/db2nodes.cfg`). Table 21 describes the design for database partition numbering of the InfoSphere Warehouse instance named `asupwh`.

Table 21) Database partitions.

Partition number	Description	Located on
Partition 0	Functions as catalog partition, coordinator, and single-partition data	Administration module
Partition 1 – Partition 8	Database partitions with partitioned data	Data module 1
Partition 9 – Partition 16	Database partitions with partitioned data	Data module 2

The catalog partition is given the number 0 because there is only one catalog for the database. The database partitions that have the greatest workload and contain the vast majority of data are the data nodes. Because most database administration and performance tuning involves the data nodes, the numbering of these partitions starts at 1 and continues sequentially.

The database partition and server mapping specified in the `db2nodes.cfg` file has the following format:

```
<nodenum> <hostname> <logical port> <netname>
```

Where:

- **nodenum.** A unique number, between 0 and 999, that identifies a database partition in a partitioned database system. For additional database partition servers, the `nodenum` values must be in ascending order.
- **hostname.** The TCP/IP host name of the database partition server for use by database partitions for non-FCM communication. This network is also used for FCM communication if no `netname` is specified.
- **logical port.** Specifies the logical port number for the database partition. This field is used to specify each database partition (node) within a single server. These logical ports correspond to the ports that are specified in the `/etc/services` file for interpartition communications.
- **netname (optional).** Specifies the host name or the IP address of the high-speed interconnect for FCM communication.

Note: Do not modify the `db2nodes.cfg` file while the instance is running.

For our test lab InfoSphere Warehouse environment, the configuration file was similar to the following:

```
0 adminnode01 0
1 datanode01 0
2 datanode01 1
3 datanode01 2
4 datanode01 3
5 datanode01 4
6 datanode01 5
7 datanode01 6
8 datanode01 7
9 datanode02 0
10 datanode02 1
11 datanode02 2
12 datanode02 3
```

```
13 datanode02 4
14 datanode02 5
15 datanode02 6
16 datanode02 7
```

7.3 CREATING A DATABASE ON A NETAPP STORAGE SYSTEM VOLUME

After the instance is created, you can create the database for the InfoSphere Warehouse. The database manager creates the database catalog on the database partition where the `CREATE DATABASE` statement is executed. For this reason, you should make sure that the `CREATE DATABASE` command is executed on the database partition that will serve as the catalog partition. This catalog partition is located on the administration node and has partition number 0 in the `db2nodes.cfg` file.

You can create a DB2 database by using the `CREATE DATABASE` command and specifying the database path and database location.

For example, to create a database named `asupdb` using `/db2data` for the database and `/db2home` for `db2path`, you would execute a `CREATE DATABASE` command similar to the following:

```
CREATE DATABASE asupdb ON /db2data DBPATH ON /db2home USING
CODESET UTF-8
TERRITORY US
COLLATE USING SYSTEM_819_US
PAGESIZE 16384
DFT_EXTENT_SZ 16
AUTOCONFIGURE APPLY NONE;
```

When the core warehouse database is created using the above statement:

- Automatic storage management is enabled for the database and the default page size for buffer pools and table spaces is 16KB.
- File systems in `/db2data/asupwh/NODENNNN` are used for all automatic storage table space containers. Sufficient zeroes are added to the partition number to produce a four-digit suffix represented by `NNNN`.
- The `/db2home/asupwh/NODENNNN` file system is used for the database directory.
- The default extent size is 16 pages, which is equivalent to 256KB when the page size is 16KB.

After the database is created, the catalog partition is restricted to the administration node (which is where the `CREATE DATABASE` command was issued). The default database partition groups span all nodes.

7.4 CREATING DATABASE PARTITION GROUPS

When a database is created, the following database partition groups are created automatically:

- **IBMCATGROUP.** This group consists of one database partition, database partition 0, and contains the catalog tables.
- **IBMTEMPGROUP.** This group contains all of the database partitions. It is the database partition group in which the default `TEMPSPACE1` system-managed temporary table space is created. The page size for `TEMPSPACE1` is determined by the database page size, which is 16KB for InfoSphere Warehouse.
- **IBMDEFAULTGROUP.** This group contains all of the database partitions. It becomes the default partition group for table spaces that do not have explicit partition group assignments.

For an InfoSphere Warehouse, we created two additional database partition groups:

- **PDPG (partitioned data partition group)**. This group contains only the data partition on the data nodes. It does not include the administration or catalog partition. The tables in this group are normally medium-sized to quite large. For our test environment we had 2 data nodes with 8 partitions on each (16 data partitions). Use the following command to create the PDGP:


```
CREATE DATABASE PARTITION GROUP PDPG ON DBPARTITIONNUMS (1-16)
```

- **SDPG (single-partition data partition group)**. This group contains one database partition: database partition 0. The tables in this group are usually small. These tables are typically dimension tables or lookup tables, and they can be static or dynamic.

To create the SDPG for our test lab, we used the following command:

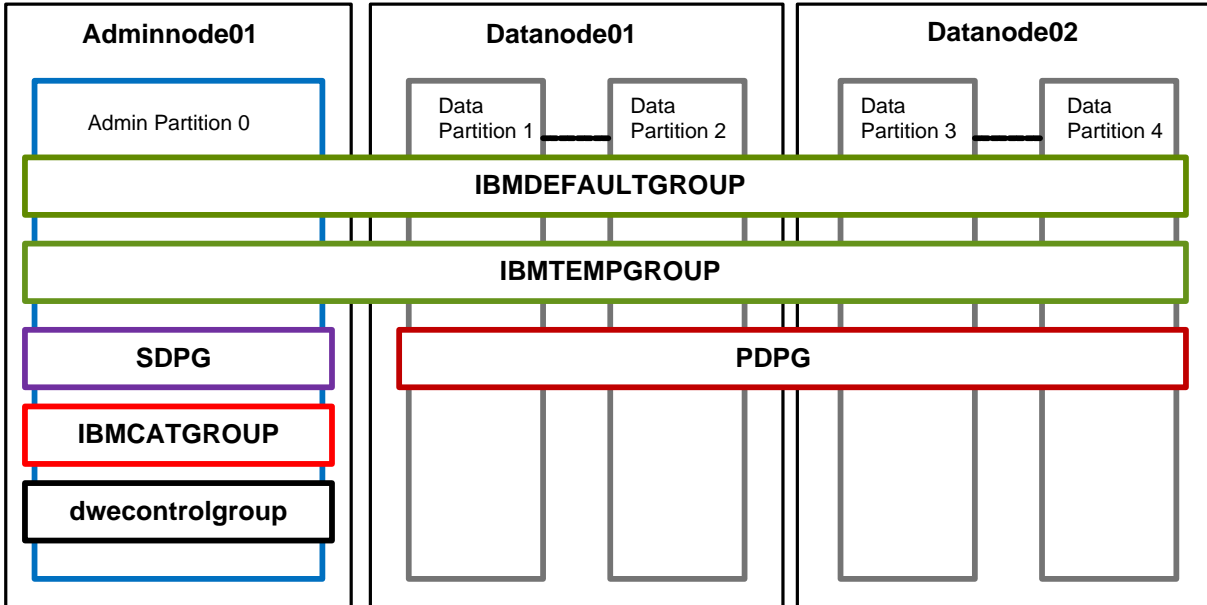
```
CREATE DATABASE PARTITION GROUP SDPG ON DBPARTITIONNUMS (0);
```

After creating the data partition groups, the logical view of our InfoSphere Warehouse groups resembled Figure 6.

- For some InfoSphere Warehouse components, we created one additional partition group named `dwecontrolgroup` on partition 0:

```
CREATE DATABASE PARTITION GROUP dwectrlgroup ON DBPARTITIONNUMS (0);
```

Figure 6) Database partition group layout.



Note: The SMS container for the default TEMPSPACE1 is in the database path. It is important to create an alternative 16KB DMS temporary table space with containers in the same file systems as the data, and then drop the default temporary table space. Dropping the default table space and creating the new table space in the prescribed file system makes sure that the database manager uses the disk space planned for temporary table spaces.

7.5 CREATING A BUFFER POOL

For the InfoSphere Warehouse configuration, the target for the total buffer pool size is approximately 4.5GB per database partition. The first 16MB are allocated to the default buffer pool, which is used for the database catalog and for monitoring tables. The remaining buffer pool memory is allocated to a single buffer pool for both the permanent and temporary table spaces.

Contrary to OLTP, data warehouse reads are sequential. To improve sequential read and table scan performance, IBM recommends creating a block-based buffer pool, which consists of both a block area and a page area. The block area consists of blocks, where each block contains a specified number of contiguous pages. It is used for sequential prefetching workloads only. The page area is used for all other I/O workloads.

When a block-based buffer pool is configured, the DB2 prefetchers perform read requests of contiguous blocks of pages on disk into contiguous pages in memory. Prefetching using block I/O is more efficient than the page-based vectored I/O used by the DB2 software by default. The prefetchers can also increase the efficiency of buffer pool reuse by reducing buffer pool thrashing when large volumes of data are read. Use a **CREATE BUFFERPOOL** statement to create buffer pools. The following statement creates a 4.5GB buffer pool with a 1.5GB block area:

```
CREATE BUFFERPOOL bp_16k ALL DBPARTITIONNUMS
SIZE 300000 PAGESIZE 16K
NUMBLOCKPAGES 100000
BLOCKSIZE 16;
```

Best Practice

For an InfoSphere Warehouse environment, IBM recommends minimizing the number of buffer pools. Based on performance tests, IBM also recommends reserving one-third of the buffer pool for the block area.

7.6 PLACING THE DATABASE TRANSACTION LOG IN THE NEW DIRECTORY

By default, DB2 database logs are initially created in a directory called `SQLLOGDIR`, a subdirectory of the database directory. You can change the location where active logs and future archive logs are placed by changing the value for this configuration parameter to point to either a different directory or to a device.

```
db2 "UPDATE DB CFG FOR asupdb USING NEWLOGPATH /db2logs"
```

The new setting does not become the value of `logpath` until both of the following occur:

- The database is in a consistent state, as indicated by the `database_consistent` parameter
- All applications are disconnected from the database

If the database is configured for roll-forward recovery, archive logs that are currently stored in the database log path directory are not moved to the new location.

In addition to updating the configuration file, you should also update the following logging-related configuration parameters on all database partitions.

Table 22) Database configuration parameters.

DB Configuration Parameter	Recommended Value	Size (MB)
Logprimary	50	2600
Logsecond	0	0

DB Configuration Parameter	Recommended Value	Size (MB)
Logfilsiz	12800	50
Logbufsz	1024	4

For our test lab environment we used circular logging. If archive logging is used, you must set `logarchmeth1` and `failarchpath` to the appropriate values. The same is true for mirror logging. If mirror logging is enabled, you must set the `mirrorlogpath` configuration parameter.

In our test environment, we created 40GB of logging space for each data partition.

Note: If your InfoSphere Warehouse uses archive logging, you should create separate volumes for data and transaction logs on storage to enable database recovery using Snapshot backup images.

7.7 CREATING TABLE SPACES

Before beginning to create table spaces for an InfoSphere Warehouse environment, you must develop a table space design layout. IBM recommends that you use automatic storage management for all user table spaces. As described in this section, you use the `CREATE TABLESPACE` command to create a table space.

```
CREATE TABLESPACE <tbs_name> IN DATABASE PARTITION GROUP <pg_name>
BUFFERPOOL <bp_name>
OVERHEAD 4.0
TRANSFERRATE 0.04;
```

Where:

- `tbs_name` is the table space name
- `pg_name` is the partition group name for the table space
- `bp_name` is the buffer pool name for the table space

For example, to create a table space named `ts_big` in a partition group `pdpg` that uses buffer pool `bp_16k`, you would execute the following command:

```
CREATE TABLESPACE ts_big IN DATABASE PARTITION GROUP pdpg
BUFFERPOOL bp_16k
OVERHEAD 4.0
TRANSFERRATE 0.04;
```

For our test lab deployment we created the following table spaces.

Table 23) Table space details.

Table Space Name	Type	Managed	Partition Group	Buffer Pool	Page Size	Extent Size (pages)
ts_big	Large	database	Pdpg	Bp_16k	16K	16
ts_big_idx	Large	database	Pdpg	Bp_16k	16K	16
db2tmp	Temporary	database	ibmtempgroup	Bp_16k	16K	16
ts_small	Regular	database	sdpg	Bp_16k	16K	16
ts_monitoring	Regular	Database	IBMDEFAULTGROUP	IBMDEFAULTBP	16K	16
temp16k	Temporary	database	ibmtempgroup	bp_16k	16K	16

Table Space Name	Type	Managed	Partition Group	Buffer Pool	Page Size	Extent Size (pages)
usertemp16k	Temporary	database	pdpq	bp_16k	16K	16

When an InfoSphere Warehouse database is created, a system temporary table space named `TEMPSPACE1` and a user temporary table space `USERSPACE1` are created. We created an additional user temporary table space named `usertemp16k` by using the following command:

```
CREATE USER TEMPORARY TABLESPACE usertemp16k IN DATABASE PARTITION GROUP pdpq
BUFFERPOOL bp_16k
OVERHEAD 4.0 TRANSFERRATE 0.04;
```

Depending on your analysis of the table space design that best meets the needs of your data warehouse, the table spaces that you create might look very different from those created in this example.

Note: If you create a table space that does not use automatic storage, you must specify container paths within these file systems.

Note: If the database partition group is not specified, the default database partition group (`IBMDEFAULTGROUP`) is used for `REGULAR`, `LARGE`, and `USER TEMPORARY` table spaces. For `SYSTEM TEMPORARY` table spaces, the default database partition group `IBMTEMPGROUP` is used.

7.8 CREATING TABLES AND POPULATING DATA

After completing the steps described in sections [7.1](#) through [7.7](#), your database is ready for you to create tables and populate with data. You can transform your logical database design to a physical database and create scripts to create tables, or you can reverse engineer an existing database to create table creation scripts. After the tables are created, you can import existing data into the environment and start using your warehouse.

8 INFOSPHERE WAREHOUSE BACKUP AND RESTORE

Data warehouse environments tend to be large, and backing up massive data poses a big challenge for DBAs. In addition to data volume that needs to be backed up, other factors such as system performance, shrinking backup windows, explosive data growth, and decreasing mean time to recover make the use of traditional backup and recovery methods outdated and unfit. On other hand, NetApp Snapshot technology has been proven very effective to perform database backup and restore operations. NetApp Snapshot and SnapRestore are Data ONTAP features that can be used to meet data warehouse backup and recovery challenges. NetApp Snap Creator eliminates the need for scripting and automates the backup and restore process.

As described in section [2.9](#), NetApp Snap Creator Framework has three components: an engine, a configuration file, and application plug-ins. Snap Creator comes with a default configuration file. To configure the DB2 database to do backup and recovery using NetApp Snap Creator, you must create a configuration file. The default configuration file can be customized and used for any environment. The following section describes some required configuration changes you must make to the configuration file settings. For a more information, see the Snap Creator Framework documentation on the NetApp Support site.

8.1 BASIC CONFIGURATION

Snap Creator uses a configuration file to perform its operations. The purpose of each parameter is well documented in the documentation section of the configuration file. Table 24 describes some of the parameters that you must modify.

Table 24) Snap Creator basic configuration.

Parameter Name	Purpose	Required	Test Env. Setting
SNAME	Used to assign a meaningful name to a Snapshot copy. It prefixes the supplied value to the Snapshot name. NetApp and IBM recommend that you set this parameter to be the same as the database name.	Yes	ASUPDB
SNAP_TIMESTAMP_ONLY	Appends the timestamp (YYYYMMDDHHMMSS) to the Snapshot name.	Yes	Y
VOLUMES	Specifies the names of the values used for database storage. Multiple volumes are allowed. The format used for this parameter is controller1:volume1, volume2; contoller2:volume1.	Yes	netapp01:admvol,datavol;netapp02:datavol;netapp03:datavol; netapp04:datavol; netapp05:datavol; netapp06:datavol;
NTAP_USERS	Specifies user credentials to perform Snapshot operation on the NetApp storage.	Yes	netapp01:db2inst1/db2inst1;netapp02:db2inst1/db2inst1; netapp03:db2inst1/db2inst1;netapp04:db2inst1/db2inst1;netapp05:db2inst1/db2inst1; netapp06:db2inst1/db2inst1.
TRANSPORT	Specifies the use of HTTP or HTTPS protocol for API communications.	Yes	HTTP
PORT	Specifies the port used to communicate with the NetApp storage. Normally the value is 80 or 443	Yes	80
NTAP_SNAPSHOT_RETENTIONS	Specifies the number of NetApp Snapshot copies to retain for a given policy.	Yes	daily:7,weekly:4,monthly:1.
NTAP_SNAPSHOT_RETENTION_AGE	Specifies a retention age (in days) for Snapshot copies. When configured, Snapshot copies are deleted only if they are older than the number defined value. NetApp recommends setting this field especially if you have an RPO requirement in a production environment.	Yes	7
DATABASES	Specifies the database name and Instance name.	Yes	ASUPDB:db2inst1
DB2_CMD	Specifies the CLI command.	Yes	db2
APP_NAME	Specifies the call to the default plug-in.	No	

Snap Creator uses application plug-ins for application consistency. The default plug-ins for DB2 are good for a nonpartitioned database backup and recovery. The partition database backup and recovery is handled by using a custom plug-in developed specifically for InfoSphere Warehouse. This custom plug-in

can be downloaded free of charge from the NetApp Communities Web site (<http://communities.netapp.com/docs/DOC-11493>).

To use the custom plug-in, you must make the following changes to the configuration file:

1. Download the `db2_IWH_plug-in.zip` file and copy it to the server where Snap Creator Framework is installed.

2. To perform application-consistent backup after write suspend on the database, update the following Snap Creator configuration file parameter:

```
APP_QUIESCE_CMD01=<Plug-in path>/infoSphereNetAppPlugin <DBNAME> <MODE>
```

The update for our test lab configuration file looked similar to the following:

```
APP_QUIESCE_CMD01=/scServer/infoSphereNetAppPlugin asupdb SUSPEND
```

3. Update the following configuration parameter to allow write resume using the custom plug-in:

```
APP_UNQUIESCE_CMD01=<Path to Plug-in>/infoSphereNetAppPlugin <DBNAME> <MODE>
```

The update for our test lab configuration file looked similar to the following:

```
APP_UNQUIESCE_CMD01=/scServer/infoSphereNetAppPlugin asupdb RESUME
```

Note: For InfoSphere Warehouse or a partition database, `APP_NAME` must be left blank.

The configuration file used for our test lab environment is shown in Appendix [9.5](#).

8.2 DATABASE BACKUP USING SNAP CREATOR FRAMEWORK

After creating a configuration file for the database, you can create the Snapshot backup on the NetApp storage system by using Snap Creator. Snap Creator is invoked by using the `snapcreator` command. You specify configuration (profile) and action options as “snap.”

To back up a database by using Snap Creator, you would execute the following command:

```
./snapcreator --profile <profile name> --action snap --policy <bkup policy> --verbose
```

Where:

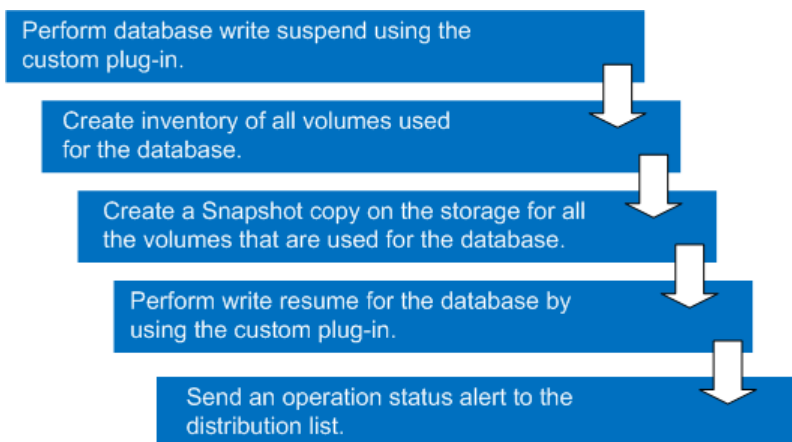
- `profile` is your profile or configuration file name. NetApp and IBM recommend keeping the profile name the same as the database name.
- `action` is the operation you are performing. Valid values are `snap`, `snaplist`, `restore_alt`, `clone_vol`, and so on.
- `bkup_policy` is the backup policy type. Possible policy types are `hourly`, `daily`, `weekly`, and `monthly`.
- `verbose` (optional) shows the Snap Creator operation on the console. Snap Creator saves the output to the logs regardless of whether or not you specify this option.

To create Snapshot based database backup for a database named `ASUPDB` by using a backup policy named `daily`, you would execute the following `snapcreator` command:

```
./snapcreator --profile ASUPDB --action snap --policy daily --verbose
```

Snap Creator initiates and completes background processes after you execute the `snapcreator` command. These background processes are shown in Figure 7.

Figure 7) Snap Creator backup process flow.



When the Snapshot backup process is complete and the Snapshot backup has been created, you can list or view the backup images by using Snap Creator. To list the Snapshot based backups for a database named `ASUPDB`, you would execute the following command:

```
./snapcreator --profile ASUPDB --action snaplist --policy daily
```

8.3 DATABASE RECOVERY USING SNAP CREATOR

Any production database may be subjected to data corruption, either by a malicious process or by user error. To go back to a point in time before the data corruption, restore the database by using the latest available good database backup image. If the database is using NetApp storage, the restore process can be automated with Snap Creator. Snap Creator uses an interactive process that requires user input to restore the database from a Snapshot based backup image.

To restore a database from a Snapshot backup copy, execute the following command:

```
./snapcreator --<profile name> --action restore --policy <bkup policy>
```

Where:

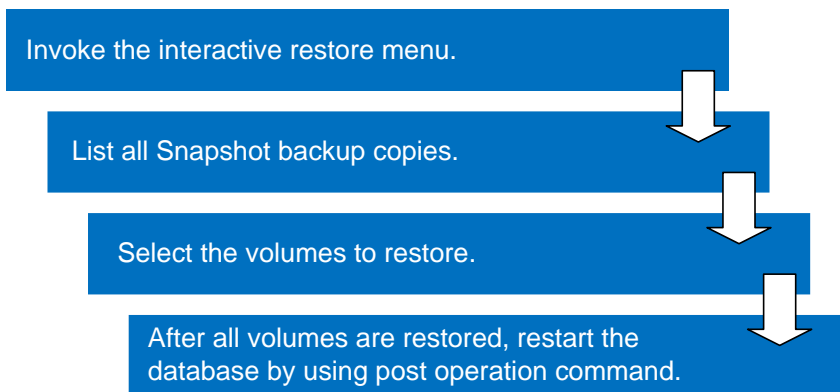
- `Profile` is the profile or configuration file name.
- `action` is the operation you are performing (in this case, the action is `restore`).
- `bkup_policy` is the backup policy type. Possible policy types are hourly, daily, weekly, and monthly.

For example, to restore a database named `ASUDB`, you would execute the following command:

```
./snapcreator --profile ASUPDB --action restore_alt --policy daily
```

After a `snapcreator` command is executed, Snap Creator initiates and completes background processes. Figure 8 shows the background process flow.

Figure 8) Snap Creator database restore process flow.



After the restore is complete, you can restart the database by using the Snap Creator postprocess operation or manually by running the DB2 commands. To restart the database manually, use the following commands to initialize the database:

```
db2inidb [DBName] as mirror  
db2 restart db [DBName] write resume
```

Where:

DBName identifies the name of the database to be initialized.

For example, to initialize a database named ASUPDB for all the nodes, you would execute the following command on the administration node database server:

```
export DB2NODE=0 && db2inidb asupdb as mirror && db2 restart db asupdb write resume &&  
db2 terminate  
  
export DB2NODE=1 && db2inidb asupdb as mirror && db2 restart db asupdb write resume &&  
db2 terminate  
  
export DB2NODE=2 && db2inidb asupdb as mirror && db2 restart db asupdb write resume &&  
db2 terminate  
  
export DB2NODE=3 && db2inidb asupdb as mirror && db2 restart db asupdb write resume &&  
db2 terminate  
  
export DB2NODE=4 && db2inidb asupdb as mirror && db2 restart db asupdb write resume &&  
db2 terminate  
  
export DB2NODE=5 && db2inidb asupdb as mirror && db2 restart db asupdb write resume &&  
db2 terminate  
  
export DB2NODE=6 && db2inidb asupdb as mirror && db2 restart db asupdb write resume &&  
db2 terminate  
  
export DB2NODE=7 && db2inidb asupdb as mirror && db2 restart db asupdb write resume &&  
db2 terminate  
  
export DB2NODE=8 && db2inidb asupdb as mirror && db2 restart db asupdb write resume &&  
db2 terminate  
  
export DB2NODE=9 && db2inidb asupdb as mirror && db2 restart db asupdb write resume &&  
db2 terminate  
  
export DB2NODE=10 && db2inidb asupdb as mirror && db2 restart db asupdb write resume  
&& db2 terminate
```

```

export DB2NODE=11 && db2inidb asupdb as mirror && db2 restart db asupdb write resume
&& db2 terminate

export DB2NODE=12 && db2inidb asupdb as mirror && db2 restart db asupdb write resume
&& db2 terminate

export DB2NODE=13 && db2inidb asupdb as mirror && db2 restart db asupdb write resume
&& db2 terminate

export DB2NODE=14 && db2inidb asupdb as mirror && db2 restart db asupdb write resume
&& db2 terminate

export DB2NODE=15 && db2inidb asupdb as mirror && db2 restart db asupdb write resume
&& db2 terminate

export DB2NODE=16 && db2inidb asupdb as mirror && db2 restart db asupdb write resume
&& db2 terminate

export DB2NODE=17 && db2inidb asupdb as mirror && db2 restart db asupdb write resume
&& db2 terminate

```

You can create scripts for these commands by using a scripting language such as shell script or Perl script.

9 APPENDICES

9.1 DATABASE MANAGER CONFIGURATION PARAMETERS FOR THE CORE WAREHOUSE

The following script shows the database manager configuration parameter settings for our lab environment:

```

Database manager configuration release level           = 0x0d00
CPU speed (millisec/instruction) (CPUSPEED)         = 2.700000e-07
Communications bandwidth (MB/sec) (COMM_BANDWIDTH)  = 1.000000e+02
Max number of concurrently active databases (NUMDB)  = 1
Federated Database System Support (FEDERATED)       = NO
Transaction processor monitor name (TP_MON_NAME)    =
Default charge-back account (DFT_ACCOUNT_STR)       =
Java Development Kit installation path (JDK_PATH)    =
/db2home/asupwh/sqlllib/java/jdk64
Diagnostic error capture level (DIAGLEVEL)          = 3
Notify Level (NOTIFYLEVEL)                          = 3
Diagnostic data directory path (DIAGPATH)           =
/db2data/asupwh/db2dump
Size of rotating db2diag & notify logs (MB) (DIAGSIZE) = 0
Buffer pool (DFT_MON_BUFPOOL)                       = OFF
Lock (DFT_MON_LOCK)                                 = OFF
Sort (DFT_MON_SORT)                                 = OFF
Statement (DFT_MON_STMT)                            = OFF
Table (DFT_MON_TABLE)                               = OFF
Timestamp (DFT_MON_TIMESTAMP)                       = ON
Unit of work (DFT_MON_UOW)                          = OFF
Monitor health of instance and databases (HEALTH_MON) = OFF
SYSADM group name (SYSADM_GROUP)                   = ASUP
SYSCTRL group name (SYSCTRL_GROUP)                  =
SYSMAINT group name (SYSMAINT_GROUP)                =
SYSMON group name (SYSMON_GROUP)                   =
Client Userid-Password Plugin (CLNT_PW_PLUGIN)      =

```

```

Client Kerberos Plugin          (CLNT_KRB_PLUGIN) =
Group Plugin                    (GROUP_PLUGIN) =
GSS Plugin for Local Authorization (LOCAL_GSSPLUGIN) =
Server Plugin Mode              (SRV_PLUGIN_MODE) = UNFENCED
Server List of GSS Plugins      (SRVCON_GSSPLUGIN_LIST) =
Server Userid-Password Plugin   (SRVCON_PW_PLUGIN) =
Server Connection Authentication (SRVCON_AUTH) = NOT_SPECIFIED
Cluster manager                 (CLUSTER_MGR) = TSA
Database manager authentication (AUTHENTICATION) = SERVER
Alternate authentication        (ALTERNATE_AUTH_ENC) = NOT_SPECIFIED
Cataloging allowed without authority (CATALOG_NOAUTH) = NO
Trust all clients               (TRUST_ALLCLNTS) = YES
Trusted client authentication   (TRUST_CLNTAUTH) = CLIENT
Bypass federated authentication (FED_NOAUTH) = NO
Default database path           (DFTDBPATH) = /db2home/asupwh
Database monitor heap size (4KB) (MON_HEAP_SZ) = AUTOMATIC(90)
Java Virtual Machine heap size (4KB) (JAVA_HEAP_SZ) = 2048
Audit buffer size (4KB)        (AUDIT_BUF_SZ) = 0
Size of instance shared memory (4KB) (INSTANCE_MEMORY) = AUTOMATIC(29934788)
Agent stack size                (AGENT_STACK_SZ) = 1024
Sort heap threshold (4KB)       (SHEAPTHRES) = 1400000
Directory cache support         (DIR_CACHE) = YES
Application support layer heap size (4KB) (ASLHEAPSZ) = 15
Max requester I/O block size (bytes) (RQRIOBLK) = 32767
Workload impact by throttled utilities(UTIL_IMPACT_LIM) = 10
Priority of agents               (AGENTPRI) = SYSTEM
Agent pool size                 (NUM_POOLAGENTS) = AUTOMATIC(100)
Initial number of agents in pool (NUM_INITAGENTS) = 0
Max number of coordinating agents (MAX_COORDAGENTS) = AUTOMATIC(200)
Max number of client connections (MAX_CONNECTIONS) =
                                AUTOMATIC(MAX_COORDAGENTS)
Keep fenced process             (KEEPFENCED) = YES
Number of pooled fenced processes (FENCED_POOL) =
                                ATOMATIC(MAX_COORDAGENTS)
Initial number of fenced processes (NUM_INITFENCED) = 0
Index re-creation time and redo index build (INDEXREC) = RESTART
Transaction manager database name (TM_DATABASE) = 1ST_CONN
Transaction resync interval (sec) (RESYNC_INTERVAL) = 180
SPM name                        (SPM_NAME) =
SPM log size                     (SPM_LOG_FILE_SZ) = 256
SPM resync agent limit           (SPM_MAX_RESYNC) = 20
SPM log path                     (SPM_LOG_PATH) =
TCP/IP Service name              (SVCENAME) = xasupwh
Discovery mode                   (DISCOVER) = SEARCH
Discover server instance         (DISCOVER_INST) = ENABLE
SSL server keydb file            (SSL_SVR_KEYDB) =
SSL server stash file            (SSL_SVR_STASH) =
SSL server certificate label     (SSL_SVR_LABEL) =
SSL service name                 (SSL_SVCENAME) =
SSL cipher specs                 (SSL_CIPHERSPECS) =
SSL versions                     (SSL_VERSIONS) =
SSL client keydb file            (SSL_CLNT_KEYDB) =
SSL client stash file            (SSL_CLNT_STASH) =
Maximum query degree of parallelism (MAX_QUERYDEGREE) = ANY
Enable intra-partition parallelism (INTRA_PARALLEL) = NO
Maximum Asynchronous TQs per query (FEDERATED_ASYNC) = 0
No. of int. communication buffers(4KB) (FCM_NUM_BUFFERS) = AUTOMATIC(16384)

```

```

No. of int. communication channels (FCM_NUM_CHANNELS) = AUTOMATIC(2048)
Node connection elapse time (sec) (CONN_ELAPSE) = 10
Max number of node connection retries(MAX_CONNRETRIES) = 5
Max time difference between nodes(min) (MAX_TIME_DIFF) = 60
db2start/db2stop timeout (min) (START_STOP_TIME) = 10

```

9.2 DATABASE CONFIGURATION PARAMETER SETTINGS

The following script shows the database configuration settings:

```

Database configuration release level = 0x0d00
Database release level = 0x0d00
Database territory = US
Database code page = 1208
Database code set = UTF-8
Database country/region code = 1
Database collating sequence = SYSTEM_819
Alternate collating sequence (ALT_COLLATE) =
Number compatibility = OFF
Varchar2 compatibility = OFF
Date compatibility = OFF
Database page size = 16384
Dynamic SQL Query management (DYN_QUERY_MGMT) = DISABLE
Statement concentrator (STMT_CONC) = OFF
Discovery support for this database (DISCOVER_DB) = ENABLE
Restrict access = NO
Default query optimization class (DFT_QUERYOPT) = 5
Degree of parallelism (DFT_DEGREE) = 1
Continue upon arithmetic exceptions (DFT_SQLMATHWARN) = NO
Default refresh age (DFT_REFRESH_AGE) = 0
Default maintained table types for opt(DFT_MTTB_TYPES) = SYSTEM
Number of frequent values retained (NUM_FREQVALUES) = 10
Number of quantiles retained (NUM_QUANTILES) = 20
Decimal floating point rounding mode(DECFLT_ROUNDING) = ROUND_HALF_EVEN
Backup pending = NO
All committed transactions have been written to disk = YES
Rollforward pending = NO
Restore pending = NO
Multi-page file allocation enabled = YES
Log retain for recovery status = NO
User exit for logging status = NO
Self tuning memory (SELF_TUNING_MEM) = OFF
Size of database shared memory (4KB)(DATABASE_MEMORY) = AUTOMATIC
(1591552)
Database memory threshold (DB_MEM_THRESH) = 10
Max storage for lock list (4KB) (LOCKLIST) = 16384
Percent. of lock lists per application (MAXLOCKS) = 10
Package cache size (4KB) (PCKCACHESZ) = (MAXAPPLS*8)
Sort heap thres for shared sorts (4KB)(SHEAPTHRES_SHR) = 5000
Sort list heap (4KB) (SORTHEAP) = 35000
Database heap (4KB) (DBHEAP) = AUTOMATIC(1200)
Catalog cache size (4KB) (CATALOGCACHE_SZ) = (MAXAPPLS*5)
Log buffer size (4KB) (LOGBUFSZ) = 2048
Utilities heap size (4KB) (UTIL_HEAP_SZ) = 65536
Buffer pool size (pages) (BUFFPAGE) = 1000
SQL statement heap (4KB) (STMTHEAP) = 10000
Default application heap (4KB) (APPLHEAPSZ) = AUTOMATIC(128)

```



```

Application Memory Size (4KB)          (APPL_MEMORY) = AUTOMATIC(40016)
Statistics heap size (4KB)            (STAT_HEAP_SZ) = AUTOMATIC(4384)
Interval for checking deadlock (ms)    (DLCHKTIME) = 10000
Lock timeout (sec)                    (LOCKTIMEOUT) = -1
Changed pages threshold                (CHNGPGS_THRESH) = 30
Number of asynchronous page cleaners    (NUM_IOCLEANERS) = AUTOMATIC(7)
Number of I/O servers                  (NUM_IOSERVERS) = 12
Index sort flag                        (INDEXSORT) = YES
Sequential detect flag                 (SEQDETECT) = YES
Default prefetch size (pages)          (DFT_PREFETCH_SZ) = 384
Track modified pages                   (TRACKMOD) = NO
Default number of containers           = 1
Default tablespace extentsize (pages) (DFT_EXTENT_SZ) = 16
Max number of active applications      (MAXAPPLS) = AUTOMATIC(202)
Average number of active applications  (AVG_APPLS) = AUTOMATIC(1)
Max DB files open per application      (MAXFILOP) = 61440
Log file size (4KB)                   (LOGFILSIZ) = 12800
Number of primary log files            (LOGPRIMARY) = 50
Number of secondary log files          (LOGSECOND) = 0
Changed path to log files              (NEWLOGPATH) =
Path to log files                      =
                                          /db2logs/asupwh/NODE0001/
Overflow log path                      (OVERFLOWLOGPATH)=
Mirror log path                        (MIRRORLOGPATH) =
First active log file                  =
Block log on disk full                 (BLK_LOG_DSK_FUL) = NO
Block non logged operations             (BLOCKNONLOGGED) = NO
Percent max primary log space by transaction (MAX_LOG)= 0
Num. of active log files for 1 active UOW(NUM_LOG_SPAN)= 0
Group commit count                    (MINCOMMIT) = 1
Percent log file reclaimed before soft ckckpt(SOFTMAX)= 100
Log retain for recovery enabled         (LOGRETAIN) = OFF
User exit for logging enabled           (USEREXIT) = OFF
HADR database role                    = STANDARD
HADR local host name                   (HADR_LOCAL_HOST) =
HADR local service name                (HADR_LOCAL_SVC) =
HADR remote host name                  (HADR_REMOTE_HOST) =
HADR remote service name               (HADR_REMOTE_SVC) =
HADR instance name of remote server(HADR_REMOTE_INST) =
HADR timeout value                     (HADR_TIMEOUT) = 120
HADR log write synchronization mode     (HADR_SYNCMODE) = NEARSYNC
HADR peer window duration (seconds)(HADR_PEER_WINDOW) = 0
First log archive method               (LOGARCHMETH1) = OFF
Options for logarchmeth1               (LOGARCHOPT1) =
Second log archive method               (LOGARCHMETH2) = OFF
Options for logarchmeth2               (LOGARCHOPT2) =
Failover log archive path              (FAILARCHPATH) =
Number of log archive retries on error  (NUMARCHRETRY) = 5
Log archive retry Delay (secs)         (ARCHRETRYDELAY) = 20
Vendor options                         (VENDOROPT) =
Auto restart enabled                   (AUTORESTART) = ON
Index re-creation time and redo index build(INDEXREC) = SYSTEM
Log pages during index build           (LOGINDEXBUILD) = OFF
Default number of loadrec sessions      (DFT_LOADREC_SES) = 1
Number of database backups to retain    (NUM_DB_BACKUPS) = 12
Recovery history retention (days)      (REC_HIS_RETENTN) = 366
Auto deletion of recovery objects       (AUTO_DEL_REC_OBJ) = OFF

```

```

TSM management class          (TSM_MGMTCLASS) =
TSM node name                 (TSM_NODENAME) =
TSM owner                     (TSM_OWNER) =
TSM password                  (TSM_PASSWORD) =
Automatic maintenance         (AUTO_MAINT) = ON
Automatic database backup     (AUTO_DB_BACKUP) = OFF
Automatic table maintenance   (AUTO_TBL_MAINT) = ON
Automatic runstats            (AUTO_RUNSTATS) = ON
Automatic statement statistics (AUTO_STMT_STATS) = ON
Automatic statistics profiling (AUTO_STATS_PROF) = OFF
Automatic profile updates     (AUTO_PROF_UPD) = OFF
Automatic reorganization      (AUTO_REORG) = OFF
Auto-Revalidation             (AUTO_REVAL) = DEFERRED
Currently Committed           (CUR_COMMIT) = ON
CHAR output with DECIMAL input (DEC_TO_CHAR_FMT) = NEW
Enable XML Character operations (ENABLE_XMLCHAR) = YES
WLM Collection Interval (minutes) (WLM_COLLECT_INT) = 20
Monitor Collect Settings
Request metrics               (MON_REQ_METRICS) = BASE
Activity metrics              (MON_ACT_METRICS) = BASE
Object metrics                (MON_OBJ_METRICS) = BASE
Unit of work events           (MON_UOW_DATA) = NONE
Lock timeout events           (MON_LOCKTIMEOUT) = NONE
Deadlock events               (MON_DEADLOCK) = WITHOUT_HIST
Lock wait events              (MON_LOCKWAIT) = NONE
Lock wait event threshold     (MON_LW_THRESH) = 5000000
Number of package list entries (MON_PKGLIST_SZ) = 32
Lock event notification level (MON_LCK_MSG_LVL) = 1
SMTP Server                   (SMTP_SERVER) =
SQL conditional compilation flags (SQL_CCFLAGS) =
Section actuals setting       (SECTION_ACTUALS) = NONE

```

9.3 KERNEL PARAMETER SETTINGS

Table 25 describes the kernel parameter settings used on all nodes in the environment.

Table 25) Kernel parameter settings.

Category	Recommend Parameter Settings
Virtual memory manager (VMM)	Default AIX 6.1 virtual memory manager settings
I/O parameters	j2_minPageReadAhead=32 j2_maxPageReadAhead=512
AIO servers	default AIX 6.1 setting (aio_maxservers = 30 per CPU)
Network parameters	sb_max=1310720 rfc1323=1 ipqmaxlen=250 udp_sendspace=65536 udp_recvspace=655360 tcp_sendspace=221184 tcp_recvspace=221184 jumbo frames enabled and the MTU value set to 9000
Maximum number of AIX processes per user	This system limit is set to 4096

Category	Recommend Parameter Settings
User limits	The following user limit values were set to Unlimited: core size (core) data size (data) file size (fsize) number of open file descriptors (nofiles) stack size (stack), with a hard limit of 4 GB
hdisk device settings	max_transfer=0x100000 queue_depth=128 reserve_policy=no_reserve
Fibre Channel device settings	lg_term_dma=0x1000000 max_xfer_size=0x100000 num_cmd_elems=1024

9.4 NETWORK CONFIGURATION DETAILS

Table 26) Network configuration details.

Network Component	Description	Name	IP Address	Subnet
POWER7 server	Management node	MgmtNode01	10.61.166.141	10.61.166.1
POWER7 server	Administration node	Adminnode01	10.61.166.142	10.61.166.1
POWER7 server	Data node 1	Datanode01	10.61.166.143	10.61.166.1
POWER7 server	Data node 2	Datanode02	10.61.166.144	10.61.166.1
1Gb management switch	Juniper switch for 1 GbE corporate network connection	jnpr4200sw1		10.61.166.1
10Gb Fibre Ethernet	Internal application network switch	jnpr4500sw1		10.61.166.1
10Gb Fibre Ethernet	Internal application network switch	jnpr4500sw20		10.61.166.1
HMC1	Hardware management consoles	Hmc1	10.61.166.148	10.61.166.1
HMC2	Hardware management consoles	Hmc2	10.61.166.149	10.61.166.1
Storage system	FAS3272C Storage system for administration and management nodes	Netapp01	10.61.166.150	10.61.166.1
		Netapp02	10.61.166.151	10.61.166.1
Storage system	FAS6210 storage system for data node 1	Netapp03	10.61.166.152	10.61.166.1
		Netapp04	10.61.166.153	10.61.166.1
Storage system	FAS6210 storage system for data node 2	Netapp05	10.61.166.154	10.61.166.1
		Netapp06	10.61.166.155	10.61.166.1

Network Component	Description	Name	IP Address	Subnet
SAN switches				
SAN switches				

9.5 SNAP CREATOR CONFIGURATION FILE FOR TEST ENVIRONMENT

```
#####
###           Required Options           ###
#####
SNAME=ASUPDB
SNAP_TIMESTAMP_ONLY=Y
VOLUMES=netapp01:admvol,datavol;netapp02:datavol;netapp03:datavol;netapp04:datavol;net
app05:datavol;netapp06:datavol
NTAP_SNAPSHOT_RETENTIONS=daily:7
NTAP_USERS=netapp01:db2inst1/db2inst1;netapp02:db2inst1/db2inst1;netapp03:db2inst1/db2
inst1;netapp04:db2inst1/db2inst1;netapp05:db2inst1/db2inst1;netapp06:db2inst1/db2inst1
NTAP_PWD_PROTECTION=N
TRANSPORT=HTTP
PORT=80
LOG_NUM=10
#####
###           Snapshot Options           ###
#####
NTAP_SNAPSHOT_RETENTION_AGE=7
SNAPDRIVE=N
SNAPDRIVE_DISCOVERY=N
NTAP_SNAPSHOT_DISABLE=N
NTAP_SNAPSHOT_NODELETE=
NTAP_SNAPSHOT_CREATE_CMD01=
NTAP_CONSISTENCY_GROUP_SNAPSHOT=N
NTAP_CONSISTENCY_GROUP_TIMEOUT=medium
NTAP_SNAPSHOT_DELETE_BY_AGE_ONLY=N
#####
###           APP Commands           ###
#####
APP_QUIESCE_CMD01==/scServer/infoSphereNetAppPlugin ASUPDB SUSPEND
APP_UNQUIESCE_CMD01==/scServer/infoSphereNetAppPlugin ASUPDB RESUME
#####
###           Pre Commands           ###
#####
PRE_APP_QUIESCE_CMD01=
PRE_NTAP_CMD01=
PRE_NTAP_CLONE_DELETE_CMD01=
PRE_APP_UNQUIESCE_CMD01=
PRE_RESTORE_CMD01=
PRE_EXIT_CMD01=/scServer/infoSphereNetAppPlugin ASUPDB RESUME
PRE_CLONE_CREATE_CMD01=
#####
### Event Configuration ###
#####
NTAP_ASUP_ERROR_ENABLE=N
SUCCESS_MSG=INFO: SnapCreator Backup for %SNAME ACTION: %ACTION POLICY: %SNAP_TYPE
finished successfully
#####
APP_NAME=
#####
###           DB2 Settings (unix only)           ###
#####
```

```
DB2_DATABASES=asupdb:asupwh  
DB2_CMD=db2
```

NetApp provides no representations or warranties regarding the accuracy, reliability or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information in this document is distributed AS IS, and the use of this information or the implementation of any recommendations or techniques herein is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. This document and the information contained herein may be used solely in connection with the NetApp products discussed in this document.

[Go further, faster®](#)