Microsoft

# Deploy IBM Db2 pureScale on Azure

By Larry Mead, Benjamin Guinebertière, Alessandro Vozza, and Jonathon Frost

AzureCAT, CSE EMEA, DMJ

December 2019

# Contents

Authored by Larry Mead, Benjamin Guinebertière, Alessandro Vozza, and Jonathon Frost. Edited by Nanette Ray. Reviewed by Mukesh Kumar, AzureCAT, CSE EMEA, DMJ, and the IBM Db2 pureScale team.

# Introduction

Enterprises have long used traditional relational database management system (RDBMS) platforms to cater to online transaction processing (OLTP) needs. These days, many are migrating their mainframe-based database environments to the Azure cloud as a way to expand capacity, reduce costs, and maintain a steady operational cost structure. And, migration is often the first step in modernizing a legacy platform.

The AzureCAT and DMJ teams recently worked with an enterprise that rehosted its IBM Db2 environment running on z/OS to IBM Db2 pureScale on Azure. The Db2 pureScale database cluster solution provides high availability and scalability on Linux operating systems. We successfully ran Db2 as a standalone, scale-up instance on a single virtual machine (VM) in a large scale-up system on Azure prior to installing Db2 pureScale.

Red Hat Linux 7.4 was used to test the configuration described in this document. This version is available from the Azure Marketplace. Before choosing a Linux distribution, make sure to verify the currently supported versions—see the documentation for IBM Db2 pureScale.

While not identical to the original environment, IBM Db2 pureScale on Linux delivers similar high availability and scalability features as IBM Db2 for z/OS running in a Parallel Sysplex configuration on the mainframe. In our customer's scenario, the cluster is connected via iSCSI to a shared storage cluster. We used the GlusterFS file system, a free, scalable, open source distributed file system specifically optimized for cloud storage. However, IBM no longer supports this solution. To maintain your support from IBM, you need to use a supported iSCSI-compatible file system. Microsoft offers Storage Spaces Direct (S2D) as an option.

This guide describes the steps we took during the migration so you can take advantage of what we learned. Installation scripts are available in the Db2onAzure repository on GitHub. These scripts are based on the architecture we used for a typical medium-sized OLTP workload.

Consider this guide and the scripts as a starting point for your Db2 implementation plan. Your business requirements will differ, but the same basic pattern applies. This architectural pattern may also be used for online analytical processing (OLAP) applications on Azure.

This guide does not cover differences and possible migration tasks for moving an IBM Db2 for z/OS database to IBM Db2 pureScale running on Linux. Nor does it provide equivalent sizing estimations and workload analyses for moving from Db2 z/OS to Db2 pureScale. To help you decide on the best Db2 pureScale architecture for your environment, we highly recommend that you complete a full sizing estimation exercise and establish a hypothesis. Among other factors, on the source system make sure to consider Db2 z/OS Parallel Sysplex with Data Sharing Architecture, Coupling Facility configuration, and DDF usage statistics.

**NOTE:** This guide describes one approach to Db2 migration, but there are others. For example, Db2 pureScale can also run in virtualized on-premises environments. IBM supports Db2 on Microsoft Hyper-V in various configurations. For more information, see Db2 pureScale virtualization architecture in the IBM Knowledge Center.

## Architecture

To support high availability and scalability on Azure, we set up a scale-out, shared data architecture for Db2 pureScale—the following architecture was used for our customer migration.

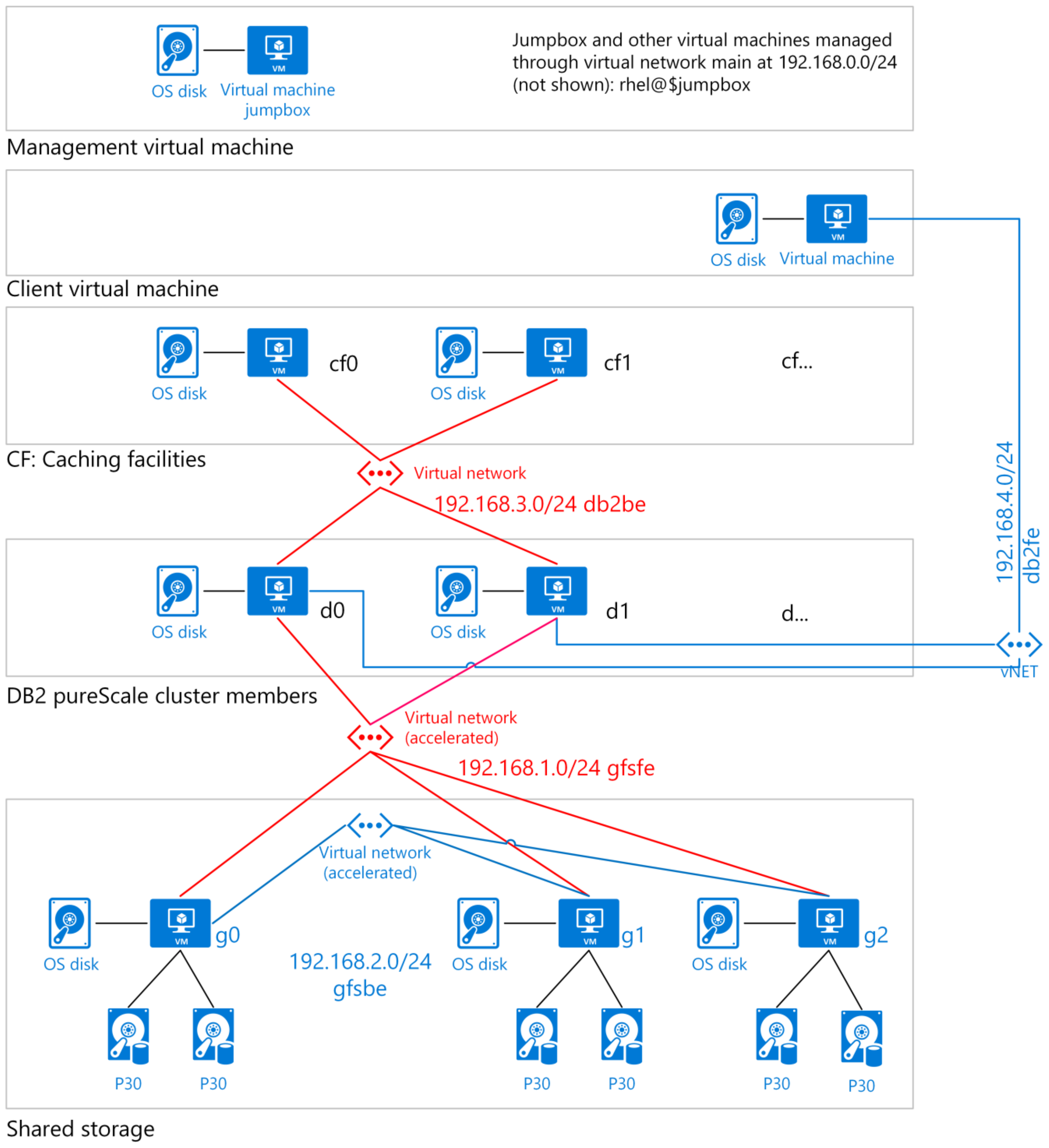

Figure 1. Db2 pureScale on Azure VMs showing storage and networking

Figure 1 depicts the logical layers needed for a Db2 pureScale cluster. These include a client VM, management VM, caching VMs, database engine VMs and shared storage VMs. In addition to the database engine nodes, the diagram also includes two nodes used for what are known as the cluster caching facilities (CFs). A minimum of two nodes are used for the database engine itself; a Db2 server that belongs to a pureScale cluster is called a member. The cluster is connected via iSCSI to a three-node shared storage cluster to provide scale-out storage and high availability. Db2 pureScale is installed on Azure virtual machines running Linux.

Consider our approach as a template that you can modify as needed to suit the size and scale needed by your organization. Our architectural approach is based on the following:

- Two or more database members are combined with at least two CF nodes, which manage a global buffer pool (GBP) for shared memory and global lock manager (GLM) services to control shared access and lock contention from multiple active members. One CF node acts as the primary and the other as the secondary, failover CF node. To avoid creating an environment that has a single point of failure, a minimum of four nodes are required for a Db2 pureScale cluster.
- High-performance shared storage (shown in P30 size in Figure 1), which is used by each of the nodes in the shared storage cluster.
- High-performance networking for the data members and shared storage.

## Compute considerations

This architecture runs the application, storage, and data tiers on Azure VMs. The setup scripts used create the following:

- A Db2 pureScale cluster. The type of compute resources you need on Azure depend on your setup. In general, there are two approaches you can use:
    - Use a multi-node, high-performance computing (HPC)-style network where multiple small to medium-sized instances access shared storage. For this HPC type of configuration, Azure memory-optimized E-series or storage-optimized L-series virtual machines provide the compute power needed.
    - Use fewer large virtual machine instances for the data engines. For large instances, the largest memory optimized M-series VMs are ideal for heavy in-memory workloads, but a dedicated instance may be required depending on the size of the Logical Partition (LPAR) that is used to run Db2.
- The Db2 CF uses memory-optimized VMs such as E-series or L-series.
- A shared storage cluster that uses Standard_DS4_v2 VMs running Linux.
- A management jumpbox is a Standard_DS2_v2 VM running Linux. An alternative is Azure Bastion, a service you can deploy that provides a secure RDP/SSH experience for all the VMs in your virtual network.
- The client is a Standard_DS3_v2 VM running Windows (used for testing).
- *Optional*. A witness server. This is needed only with certain earlier versions of Db2 pureScale. This example uses a Standard_DS3_v2 virtual machine running Linux.

In either case, a minimum of two Db2 instances are required in a Db2 pureScale cluster. A Cache instance and Lock Manager instance are also required.

## Storage considerations

Like Oracle RAC, Db2 pureScale is a high-performance block I/O, scale-out database. We recommend using the largest Azure Premium Storage available, that suits your needs. For example, smaller storage options may be suitable for development and test environments, while production environments often require larger storage capacity. We chose P30 because of its ratio of IOPS to size and price. Regardless of size, use Premium Storage for best performance.

Db2 pureScale uses a shared everything architecture, where all data is accessible from all cluster nodes. Premium storage must be shared across multiple instances—whether on-demand or on dedicated instances.

A large Db2 pureScale cluster can require 200 terabytes (TB) or higher of Premium shared storage, with IOPS of 100,000. Db2 pureScale supports an iSCSI block interface that can be used on Azure. The iSCSI interface requires a shared storage cluster. We suggest using Shared Storage Direct (S2D), or a tool from another vendor. This type of solution creates a virtual SAN (vSAN) device in Azure. Db2 pureScale uses the vSAN to install the clustered file system used to share data among multiple VMs.[1]

## Networking considerations

IBM recommends InfiniBand networking for all members in a Db2 pureScale cluster; Db2 pureScale also uses RDMA (where available) for the CFs used.

During setup, an Azure resource group is created to contain all the virtual machines. In general, resources are grouped based on their lifetime and who will manage them. The virtual machines in this architecture require accelerated networking, an Azure feature that provides consistent, ultra-low network latency via single root I/O virtualization (SR-IOV) to a virtual machine.

Every Azure virtual machine is deployed into a virtual network that is segmented into multiple subnets: main, Gluster FS front end (gfsfe), Gluster FS back end (bfsbe), Db2 pureScale (db2be), and Db2 purescale front end (db2fe). The installation script also creates the primary NICs on the virtual machines in the main subnet.

Network security groups (NSGs) are used to restrict network traffic within the virtual network and isolate the subnets.

On Azure, Db2 pureScale needs to use TCP/IP as the network connection for storage.

[1] The performance benchmarks for the various vSAN implementations have yet to be established as this writing.

# Solution deployment

To deploy this architecture, download and run the deploy.sh script found in the Db2onAzure repository on GitHub.

The repository also includes scripts you can use to set up a Grafana dashboard that can be used to query Prometheus, the open-source monitoring and alerting system included with Db2.

ⓘ **NOTE:** The deploy.sh script on the client creates private SSH keys and passes them to the deployment template over HTTPS. For greater security, we recommend using Azure Key Vault to store secrets, keys, and passwords.

## How the deployment script works

The deploy.sh script creates and configures the Azure resources that are used in this architecture. The script prompts you for the Azure subscription and VMs for the target environment, and then performs the following operations:

- Sets up the resource group, virtual network, and subnets on Azure for the installation.
- Sets up the NSGs and SSH for the environment.
- Sets up multiple NICs on both the GlusterFS and the Db2 pureScale virtual machines.
- Creates the GlusterFS storage virtual machines. If you use Storage Spaces Direct or another storage solution, see Storage Spaces Direct overview.
- Creates the jumpbox virtual machine.
- Creates the Db2 pureScale virtual machines.
- Creates the witness virtual machine that Db2 pureScale pings. Skip this part of the deployment if your version of Db2 pureScale does not require a witness.
- Creates a Windows virtual machine to use for testing but does not install anything on it.

Next, the deployment scripts set up iSCSI vSAN for shared storage on Azure. In this example, iSCSI connects to GlusterFS. This solution also gives you the option to install the iSCSI targets as a single Windows node. (iSCSI provides a shared block storage interface over TCP/IP that allows the Db2 pureScale setup procedure to use a device interface to connect to shared storage.) For GlusterFS basics, see the Architecture: Types of volumes topic in Getting started with GlusterFS.

The deployment scripts execute these general steps:

1. Sets up a shared storage cluster on Azure. This involves at least two Linux nodes.
2. Sets up an iSCSI Direct interface on target Linux servers for the shared storage cluster.
3. Sets up the iSCSI Initiator on the Linux virtual machines that will access the shared storage cluster using iSCSI Target. For setup details, see the How To Configure An iSCSI Target And Initiator In Linux in the RootUsers documentation.
4. Installs the storage layer for the iSCSI interface.

After creating the iSCSI device, the final step is to install Db2 pureScale. As part of the Db2 pureScale setup, IBM Spectrum Scale (formerly known as GPFS) is compiled and installed on the shared storage cluster. This clustered file system enables Db2 pureScale to share data among the multiple virtual machines that run the Db2 pureScale engine. For more information, see the IBM Spectrum Scale documentation on the IBM website.

## Db2 pureScale response file

The GitHub repository includes Db2server.rsp, a response (.rsp) file that enables you to generate an automated script for the Db2 pureScale installation. The following table lists the Db2 pureScale options that the response file uses for setup. You can customize the response file, as needed, for your installation environment. A sample response file (Db2server.rsp) is included—if you use this file, you must edit it before it can work in your environment.

Db2 pureScale response file options

| Screen name | Field | Value |
|---|---|---|
| Welcome | | New Install |
| Choose a Product | | Db2 Version 11.1.3.3. Server Editions with Db2 pureScale |
| Configuration | Directory | /data1/opt/ibm/db2/V11.1 |
| | Select the installation type | Typical |
| | I agree to the IBM terms | Checked |
| Instance Owner | Existing User For Instance, User name | Db2sdin1 |
| Fenced User | Existing User, User name | Db2sdfe1 |
| Cluster File System | Shared disk partition device path | /dev/dm-2 |
| | Mount point | /Db2sd_1804a |
| | Shared disk for data | /dev/dm-1 |
| | Mount point (Data) | /Db2fs/datafs1 |
| | Shared disk for log | /dev/dm-0 |
| | Mount point (Log) | /Db2fs/logfs1 |
| | Db2 Cluster Services Tiebreaker. Device path | /dev/dm-3 |

Db2 pureScale response file options (*continued*)

| Screen name | Field | Value |
|---|---|---|
| Host List | d1 [eth1], d2 [eth1], cf1 [eth1], cf2 [eth1] | |
| | Preferred primary CF | cf1 |
| | Preferred secondary CF | cf2 |
| Response File and Summary | first option | Install Db2 Server Edition with the IBM Db2 pureScale feature and save my settings in a response file |
| | Response file name | /root/DB2server.rsp |

Note the following:

- The values for /dev-dm0, /dev-dm1, /dev-dm2, and /dev-dm3 can change after a reboot on the virtual machine where the setup takes place (d0 in the automated script). To find the right values, you can issue the following command before completing the response file on the server where the setup will be run:

```
[root@d0 rhel]# ls -als /dev/mapper
total 0
0 drwxr-xr-x  2 root root     140 May 30 11:07 .
0 drwxr-xr-x 19 root root    4060 May 30 11:31 ..
0 crw-------  1 root root 10, 236 May 30 11:04 control
0 lrwxrwxrwx  1 root root       7 May 30 11:07 db2data1 -> ../dm-1
0 lrwxrwxrwx  1 root root       7 May 30 11:07 db2log1 -> ../dm-0
0 lrwxrwxrwx  1 root root       7 May 30 11:26 db2shared -> ../dm-2
0 lrwxrwxrwx  1 root root       7 May 30 11:08 db2tieb -> ../dm-3
```

- The setup scripts use aliases for the iSCSI disks so that the actual names can be found easily.
- When the setup script is run on d0, the /dev/dm-* values may be different on d1, cf0 and cf1. The Db2 pureScale setup doesn't care.

## Troubleshooting and known issues

The GitHub repo includes a Knowledge Base (db2-on-azure/doc/KB.md) listing potential issues you may encounter and resolutions you can try. For example, known issues can occur when:

- Trying to reach the gateway IP address
- Compiling GPL
- The security handshake between hosts fails
- The Db2 installer detects an existing file system
- Manually installing GPFS
- Installing Db2 pureScale when GPFS is already created
- Removing Db2 pureScale and GPFS
- Installing a witness VM that is not required by your version of Db2 pureScale.

## Learn more

Creating required users for a Db2 pureScale Feature installation

Db2icrt - Create instance command

Db2 pureScale Clusters Data Solution

IBM Data Studio

Platform Modernization Alliance: IBM Db2 on Azure

Azure Virtual Data Center Lift and Shift Guide