

Deploying IBM® Db2 pureScale with TCP/IP private network on AWS

Table of Contents

- Introduction 4**
- Value Propositions with pureScale on AWS.....4**
 - Extreme capacity4
 - Continuous availability4
 - Application transparency5
- Use cases5**
 - Cluster readiness from weeks/months to minutes5
 - Simplified Redeployment5
 - “Fastpass” to new Db2 levels validation with dramatic lower TCO6
 - Lower cost with S3.....6
 - Disaster Recovery with extreme ease7
 - Transferrable skills.....7
- Deployment Overview7**
 - Single Availability Zone (AZ) Deployment Model7
 - Multiple Availability Zones (AZs) Deployment Model (Single region).....9
 - Cross-region Deployment Model9
- Supported Regions and Availability Zones 10**
- Prerequisites and Requirements..... 12**
 - Technical prerequisites for deployment.....12**
 - Host..... 12
 - Operating System Level 12
 - Db2 version..... 12
 - Pacemaker and Spectrum Scale levels 12
 - Cluster interconnect 12
 - Network Interfaces 12
 - Storage..... 12
 - Skills and specialized knowledge 12**
 - Environment configuration requirements 13**
 - AWS account..... 13
 - Resource quotas 13

<i>Architecture Diagrams</i>	14
<i>Security</i>	14
Root Privilege	14
IAM roles for administrating and using the Db2 pureScale deployment	15
S3 buckets access	15
Authentication keys used by the deployment process	16
<i>Costs</i>	16
AWS services	16
Software licenses	16
<i>Sizing</i>	17
Performance	17
<i>Deployment Assets</i>	17
Deployment Steps	17
Set Up Client Connectivity via AWS Transit Gateway	21
<i>Health Check</i>	22
<i>Backup and Recovery</i>	22
<i>Routine Maintenance</i>	22
<i>Reducing and Expanding the Number of Active Db2 pureScale Members</i>	23
Prepare and shut down an EC2 instance	23
Quiesce the member host	23
Stop the instance on the member host	23
Put the host into maintenance mode	24
Halt the host	24
Bring a host back online	24
Start the AWS instance	24
Leave maintenance mode	24
Start the Instance on the host	25
<i>Updating the EC2 instance type</i>	25
<i>Adding a member to an existing Db2 pureScale cluster on AWS</i>	26
Automation documents for adding a member	26
<i>Expanding a file system in an existing Db2 pureScale cluster on AWS</i>	26
Automation documents for filesystem expansion	27
<i>Adding a new filesystem to an existing Db2 pureScale cluster on AWS</i>	27
Create a new multi-attach io2 storage device	27
Attach this multi-attach io2 storage device to all the hosts in the cluster	28

Create a new file system using the new volume	28
Additional tasks	29
<i>Emergency Maintenance</i>	29
<i>Support</i>	30
<i>FAQ</i>.....	30

Introduction

IBM® Db2® pureScale® is an optional Db2 feature that reduces the risk and cost of business growth by providing extreme capacity, continuous availability, and application transparency. These features include:

- Extreme Capacity - Start with a small cluster, grow as you need online.
- Continuous Availability – Design with integrated cluster manager to orchestrate automatic recovery on components failure, allowing transactions to continue on other healthy members in the event of planned or unplanned member outage.
- Application Transparency - Avoid the risk and cost of application changes. As you grow the system, you do not need to make the application cluster-aware in order to get good scalability.
- Workload Consolidation - Map multiple workloads or databases to different subset members within the same Db2 pureScale instance with the capability to shift members from one workload to another in response to workload demands.

The self-managed IBM® Db2 pureScale with TCP/IP interconnect on AWS offering is an extension of the on-premises pureScale with TCP/IP option. They share the same value propositions with the architecture built on familiar and proven design features from Db2 for z/OS database software to deliver exceptional levels of database availability that satisfy the uniqueness of many applications.

Value Propositions with pureScale on AWS

Extreme capacity

The IBM® Db2 pureScale Feature can scale with near-linear efficiency and high predictability. On AWS, several pre-configured cluster sizes are available to suit the desired throughput requirements. Deploying with a pre-configured size is completely automated and fast (in minutes) from the provisioning of AWS EC2 instances along with network and storage configuration, to Db2 pureScale instance setup and configuration. The simplicity and speed of this process makes this the recommended path to deploy new cluster with larger size as opposed to manual reconfiguration on existing deployment using procedures well-documented for on-premises deployments. For savvy pureScale users, the traditional cluster expansion method, which is done by adding new members to the instance, can also be achieved with special instructions provided to handle majority quorum and AWS EC2 instance setup.

Continuous availability

With AWS inherent redundancy in the network component, the pureScale cluster is even more resilient to certain types of failure compared to on-premises deployment with no additional cost (capital and labour) to maintain the high resilient infrastructure.

Application transparency

Whether this is a new deployment on AWS, a move from on-premises pureScale to AWS, or a shift from on-premises or AWS single Db2 instance to pureScale on AWS, applications do not have to be aware of the topology of the database environment, the potential cluster interconnect differences (RDMA Vs non-RDMA), and the AWS region of the deployment. This feature means that applications work just as they did before, yet they can benefit from the enhanced network resiliency, extreme capacity, and continuous availability from the moment that they connect to the pureScale cluster on AWS for the first time.

Use cases

With the pureScale feature on AWS, many existing on-premises practices can be greatly simplified with significant reduction in total cost of ownership, along with numerous new use cases opportunities.

Cluster readiness from weeks/months to minutes

The journey to embark on pureScale typically starts with a joint sizing effort between Db2 technical sales and the client's group of database administrators and application developers. The two groups work together to finalize the hardware and network requirements based on the current database design, throughput expectations at average, and peak periods among other factors. The client's IT department then executes hardware ordering. What follows next can be week(s) or month(s) (in the event of supply chain issues) before all required hardware arrive at the data center where multiple teams (special lab services may be involved) coordinate to assemble the pureScale cluster. Upon completion of the physical setup, system administrators followed by DBAs can proceed to complete the OS and Db2 instance setup. The staged serialized steps above are referred to as the "cluster readiness" step. Database setup and data loading will follow before application validation can commence.

With the pureScale feature on AWS, once the rough sizing of the cluster is determined, the remaining time-consuming cluster readiness step can be shortened from week(s)/month(s) to minutes. The new process is a one-click automated deployment which leverages AWS cloud foundation that handles the EC2 instances, network, multi-attached share storage, Db2 install, and instance setup on all hosts. Once completed, the pureScale instance is ready for data loading and application validations. In fact, multiple concurrent cluster setup and application validation is possible to further reduce the time to production – all with a fraction of the cost in the traditional on-premises scenario.

Simplified Redeployment

The need for a redeployment may stem from incorrect initial cluster sizing, data center move, planned or unplanned outage causing hardware components to be unavailable for long duration, etc. For on-premises scenario, replacing or redeploying out-of-commission cluster requires substantial planning, capital investment in advance, and is usually a labour-intensive process.

With the pureScale feature on AWS, redeployment is as simple as a click of a button and a new larger cluster will be ready in minutes. If the databases were backed up regularly to S3, they can be easily restored to the new cluster. *(Note: certain restrictions exist when storing to a subset of the members in terms of database vs tablespace restore. It also requires a full offline database backup prior to first activation. No such restriction exists when restoring to a superset of members.)* To destroy the out-of-commission cluster on AWS is equivalent to destroying all EC2 instances from the console with a click of few buttons. Unlike on-premises, there is no need to perform the dropping of the instances and de-installation of Db2.

“Fastpass” to new Db2 levels validation with dramatic lower TCO

Today, corporate guidelines often mandate internal certification of new Db2 fix packs and mod packs levels in test environment prior to deployment in production environment. Such certification typically requires months of planning in advance, mostly due to capital and resource constraints. On-premises hardware investment also carries a high cost of maintenance in terms of regular operating system, security fixes and hardware components upgrade.

By and large, the new level validations focus mostly on compatibility with existing applications and are generally cluster-interconnect agnostic, which means that whether the on-premises deployments use RDMA or not is irrelevant to the validations. These use cases are prime candidates to move to pureScale on AWS to avoid the cost of owning and maintaining hardware stack. When the need to validate a new level arises, simply spin up as many new pureScale clusters as needed to perform the validation and destroy them once the validation is completed. The various pre-configured pureScale sizes also provide an opportunity to evaluate if a smaller or larger cluster provides any cost savings or performance benefits for new or existing applications with the new Db2 levels.

Lower cost with S3

If the current on-premises deployment utilizes AWS S3 for backup and restore, load and ingest on a regular basis, moving the pureScale cluster to AWS can lower the cost in terms of in-bound and out-bound network traffic. Network traffic cost within AWS is always lower than from outside. If the current on-premises deployment does not utilize S3 today, the pureScale feature on AWS opens the door for low-cost storage with resiliency without incurring the cost of maintaining a resilient storage infrastructure in the data center.

Disaster Recovery with extreme ease

While Db2 pureScale provides the five nines high availability within a single AZ, disaster recovery covers the last mile of a truly highly available cluster in the event of the local AZ or region failure. With pureScale, the DR strategy is HADR where a second pureScale cluster is set up in a different AZ of the same or different region. For on-premises deployment, the DR site must be carefully chosen, followed by the week(s)/month(s) of hardware procurement, physical setup, and software installation before HADR setup can be done and validated.

With pureScale feature on AWS, setting up the standby pureScale cluster in a DR site amounts to the same one-click deployment. The only difference is choosing a different region available as the DR site. In less than an hour, the instance will be ready for HADR setup. In fact, multiple pureScale clusters can be set up in different regions using the same process concurrently. The HADR pairs can be set up one at a time to determine the optimal location that yields the best performance before shutting down other regions. The same HADR simulation available for on-premises customers can be used on AWS.

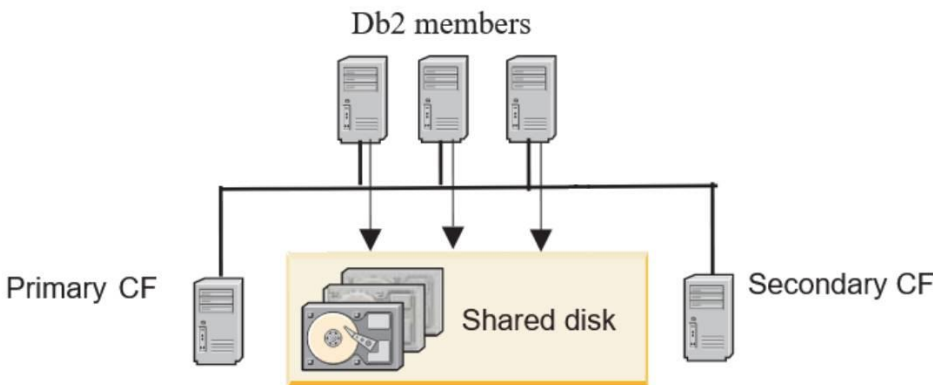
Transferrable skills

For seasoned on-premises Db2 pureScale users, the cluster on AWS has the same look and feel. Procedures for cluster management, online software updates, host maintenance, instance and databases monitoring, performance tuning, problem determination and troubleshooting techniques, etc. are all transferrable. Applications transparency allows seamless move from clusters in traditional data centers to AWS with a simple change of end point IP address in the client configuration.

Deployment Overview

Single Availability Zone (AZ) Deployment Model

The self-managed IBM® Db2 pureScale with TCP/IP interconnect on AWS deployment model provisions a highly available database cluster in a single availability zone (AZ) of a region. The topology consists of an odd number of members (minimum three), dual cluster caching facilities (CFs). Each host is configured with multiple TCP/IP network interfaces and shared storage.



Multiple cluster sizes (small, medium, and large) are offered with key differences in the number of members as well as varying number of cores and RAMs in each host to best fit the demand of the application need. Each host is provisioned with AWS c6i instance enabled with EBS Multi-Attach feature. As such, pureScale must be deployed in regions that support both. The estimated time to provision a five hosts pureScale cluster (3 members and 2 CFs) which includes deploying all EC2 instances, installing the Db2 pureScale software, and setting up the cluster is approximately 30 - 40 minutes.

List of Required Resources

Below is a sample list of AWS resources required in a deployment with three Db2 members and two CFs.

- 1 x Virtual Private Cloud (VPC)
 - IPv4 CIDR 10.0.0.0/16
- 1 x Internet Gateway
- 1 x Route Table
- 2 x Subnets
 - One public subnet with IPv4 CIDR 10.0.0.0/24
 - One private subnet with IPv4 CIDR 10.0.1.0/24
- 1 x Network Access Control List (ACL)
- 1 x Security Group (firewall)
- 5 x EC2 Instances, 7x for XLarge clusters with the following per instance:
 - 1 x EBS Volume
 - 2 x Network Interfaces
 - One part of public subnet
 - One part of private subnet
 - EBS-Optimized
 - Clustered Placement Group
- 4 x EBS Multi-Attach Volume
 - (Optional) 4 x EBS Snapshot

- 1 x S3 Bucket
 - Used for storing Cloud Formation Templates when adding volumes or members

Multiple Availability Zones (AZs) Deployment Model (Single region)

The multiple AZs deployment model for pureScale on AWS is designed to address the disaster recovery needs. To create a disaster recovery site in separate AZs within the same VPC :

- Set DREnabled to True, choose a hostname for the DR site and select a separate AZ for the DR site.
 - See the “deployment steps” below for more details on the parameters
- Follow the same procedures as on-premises to set up HADR in Db2 pureScale environment, see IBM documentation on [Deploying HADR in a Db2 pureScale environment](#).
- Note: setting DREnabled to true will create two pureScale clusters with their own EC2 resources. This comes with additional costs.

Cross-region Deployment Model

The cross-region deployment model for pureScale on AWS addresses the high availability, and disaster recovery needs in different regions.

Cross-region deployment can be configured with the following steps using AWS Transit gateways:

- Set up a single site pureScale cluster in any region and any AZ within the region using the one-click automated deployment.
- Choose any other region and any AZ within the region that meets the requirement and set up a second pureScale cluster using the same one-click automated deployment.
- Set up disjoint set of IP addresses between the primary and standby cluster during one-click automated deployment by selecting distinct value for VpcCidr in stack details.
- Once both clusters are up and running, configure Transit gateways, Transit gateway attachments to a VPC, and Transit gateway peering attachments following the AWS documentation on Transit gateways: <https://docs.aws.amazon.com/vpc/latest/tgw/tgw-transit-gateways.html>
- Follow the same procedures as on-premises to set up HADR in Db2 pureScale environment, see IBM documentation on Deploying HADR in a Db2 pureScale environment: <https://www.ibm.com/docs/en/db2/12.1?topic=environments-scenario-deploying-hadr-in-db2-purescale-environment>.
- Clients can connect from another region to the primary cluster. If this is desired, configure Transit gateways, Transit gateway attachments to a VPC, and Transit gateway

peering attachments with the client’s VPC, primary cluster VPC and standby cluster VPC following the AWS documentation on Transit gateways.

Cross-region deployment can be configured with the following steps using VPC peering connections:

- Set up a single site pureScale cluster in any region and any AZ within the region using the one-click automated deployment.
- Choose any other region and any AZ within the region that meets the requirement and set up a second pureScale cluster using the same one-click automated deployment.
- Set up disjoint set of IP addresses between the primary and standby cluster during one-click automated deployment by selecting distinct value for VpcCidr in stack details.
- Once both clusters are up and running, configure VPC peering connections following the AWS documentation on VPC peering: <https://docs.aws.amazon.com/vpc/latest/peering/what-is-vpc-peering.html>
- Follow the same procedures as on-premises to set up HADR in Db2 pureScale environment, see IBM documentation on Deploying HADR in a Db2 pureScale environment: <https://www.ibm.com/docs/en/db2/12.1?topic=environments-scenario-deploying-hadr-in-db2-purescale-environment>.
- Clients can connect from another region to the primary cluster. If this is desired, configure VPC peering connections with the client’s VPC, primary cluster VPC and standby cluster VPC following the AWS documentation on VPC peering.

Note: Geographically Dispersed pureScale Cluster (GDPC) is not available with pureScale AWS offering.

Supported Regions and Availability Zones

Regions	Availability Zones	Supported
US East (N. Virginia)	us-east-1a	Yes
	us-east-1b	Yes
	us-east-1c	Yes
	us-east-1d	Yes
	us-east-1e	No
	us-east-1f	No
US East (Ohio)	us-east-2a	Yes
	us-east-2b	Yes
	us-east-2c	Yes
US West (N. California)	us-west-1a	Yes
	us-west-1b	Yes
US West (Oregon)	us-west-2a	Yes
	us-west-2b	Yes

	us-west-2c	Yes
	us-west-2d	Yes
Canada (Central)	ca-central-1a	Yes
	ca-central-1b	Yes
	ca-central-1d	No
EU (Frankfurt)	eu-central-1a	Yes
	eu-central-1b	Yes
	eu-central-1c	No
EU (Ireland)	eu-west-1a	Yes
	eu-west-1b	Yes
	eu-west-1c	Yes
EU (London)	eu-west-2a	Yes
	eu-west-2b	Yes
	eu-west-2c	Yes
EU (Stockholm)	eu-north-1a	Yes
	eu-north-1b	Yes
	eu-north-1c	Yes
Asia Pacific (Hong Kong)	ap-east-1a	Yes
	ap-east-1b	Yes
	ap-east-1c	Yes
Asia Pacific (Singapore)	ap-southeast-1a	Yes
	ap-southeast-1b	Yes
	ap-southeast-1c	Yes
Asia Pacific (Sydney)	ap-southeast-2a	Yes
	ap-southeast-2b	Yes
	ap-southeast-2c	Yes
Asia Pacific (Seoul)	ap-northeast-2a	Yes
	ap-northeast-2b	Yes
	ap-northeast-2c	Yes
	ap-northeast-2d	Yes
Asia Pacific (Tokyo)	ap-northeast-1a	Yes
	ap-northeast-1c	Yes
	ap-northeast-1d	Yes
Asia Pacific (Mumbai)	ap-south-1a	Yes
	ap-south-1b	Yes
	ap-south-1c	Yes
Middle East (Bahrain)	me-south-1a	Yes
	me-south-1b	Yes
	me-south-1c	Yes

Prerequisites and Requirements

Technical prerequisites for deployment

Host

AWS EC2 c6i instance

Operating System Level

One of:

- Red Hat Enterprise Linux 9.4
- SUSE Linux Enterprise Server level 15 Service Pack 5

Db2 version

12.1.1.1.0

Pacemaker and Spectrum Scale levels

The versions bundled and set up automatically with the above Db2 version.

Cluster interconnect

TCP/IP

Network Interfaces

At least two IPv4 capable TCP/IP interfaces. One used as private communication between members and CFs. The second one used by Spectrum Scale for heart-beat and client connections.

Storage

EBS Multi-attach storage

Skills and specialized knowledge

The deployment requires a moderate level of familiarity with AWS services, including knowledge on deployment and management of solution. For more information and to get started, see [Getting Started Resource Center](#) and [AWS Training and Certification](#) for

information on how to design, deploy, and operate your infrastructure and applications on the AWS Cloud.

A general understanding of Db2 pureScale architecture and instance management skills are required:

- Introduction to a Db2 pureScale environment - <https://www.ibm.com/docs/en/db2/12.1?topic=editions-introduction-db2-purescale-environment>
- Instance management in a Db2 pureScale environment - <https://www.ibm.com/docs/en/db2/12.1?topic=administration-instance-management-in-db2-purescale-environment>

Environment configuration requirements

AWS account

If you don't already have an AWS account, create one at <https://amazon.com> by following the on-screen instructions. Part of the sign-up process involves receiving a phone call and entering a PIN using the phone keypad. Your AWS account is automatically signed up for all AWS services. You are charged only for the services you use. Ensure the AWS account is properly configured in accordance with the sections below.

Resource quotas

You can request additional quota increase and view your current quota limits here: <https://us-east-2.console.aws.amazon.com/servicequotas/home?region=us-east-2>

Resource	Number of Resources
VPCs	1
Route Tables	1
AWS Identity and Access Management (IAM) security groups	1
EC2 Instances - c6i.8/12/16xlarge	5 (7 for large clusters)
EBS Volumes – io2	4 – Local storage (alternatively can use gp2/3) 4 – Shared Storage (Logs, Archlogs, Data, Filesystem)
IOPS for Provisioned IOPS SSD (io2) volumes	20K, 28K and 40K for (x)small, medium, and large cluster sizes respectively.
EBS Snapshots	4+

Architecture Diagrams

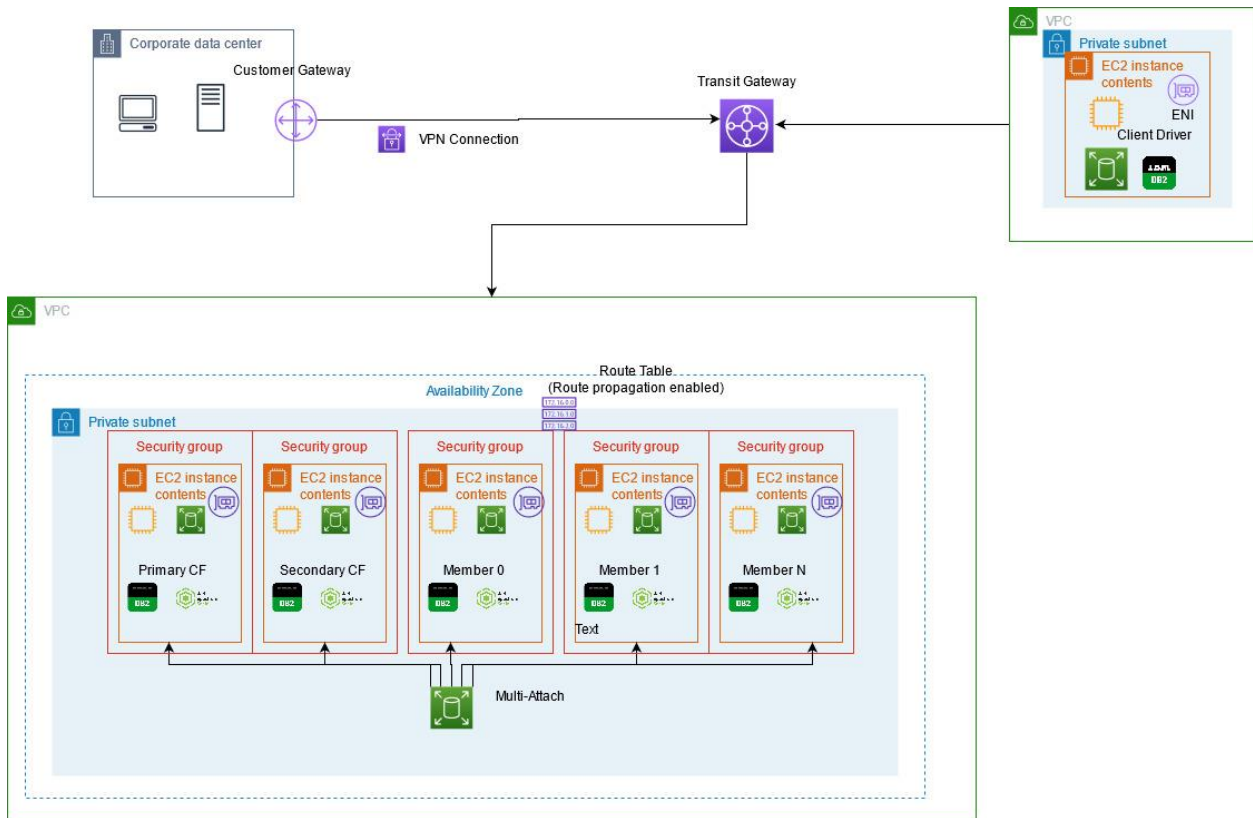


Figure 1 IBM Db2 pureScale architecture on AWS

Details in the above figure:

- A VPC configured with public and private subnets according to AWS best practices
- EC2 Instances as members or caching facilities, each with the following:
 - EBS Volumes for local storage
 - Network Interface for public and private subnets
- EBS Multi-Attach storage which is attached to all EC2 instances
- AWS Transit Gateway for communication with clients hosted in separate VPC.
 - VPN Connection for on premise clients.

Security

Root Privilege

The deployment of the Db2 pureScale cluster on AWS using the Cloud Formation template does not require the usage of an AWS root account. It is recommended to use an administrative AWS

account, which is authorized to create and show the state of EC2 resources. The use of an AWS root account should not be used to deploy or operate a pureScale cluster.

IAM roles for administrating and using the Db2 pureScale deployment

IAM roles with the minimum required privileges should be used for tasks associated with the instances used in the Db2 pureScale deployment. The deployment of Db2 pureScale requires permissions to create EC2 instances, security groups, configure subnets and routing, create VPCs, create Transit Gateways and provision multi-attach EBS storage, attaching these to the EC2 instances. An administrative IAM role should be used which provides these capabilities.

Two IAM roles are created during deployment. These roles allow the Automations for expanding a file system and adding a member to create the required resources and configure the hosts. The AutomationServiceRole can be assumed by the Automations and gives it the required permission to create EC2 resources and add them to the Cloud Formation Stack. The SSMIAMRole is used to configure the SSM agent on the hosts and allows for the Automations to run commands to configure the hosts. See “Adding a Member” and “Expanding a File System” below for more details on how these IAM roles are used.

Other IAM roles should be created as needed, following the principle of least privilege. These might be limited to viewing the EC2 instance status and being able to start/stop these resources. Such a role would be suitable for maintenance activities for the Db2 pureScale cluster.

S3 buckets access

Db2 pureScale supports access to S3 buckets for backup, restore, load, and ingest utilities. There is no association between the local Db2 instance ID and the S3 accounts. Any Db2 instance can access any S3 buckets provided the appropriate S3 buckets use ID and passwords are cataloged in the database. Refer to the following links for detail.

- Remote Storage requirements:
<https://www.ibm.com/docs/en/db2/12.1?topic=reference-remote-storage-requirements>
- S3 user ID and password management:
<https://www.ibm.com/docs/en/db2/12.1?topic=commands-catalog-storage-access>

A S3 bucket with the name “db2-ps-automation-<Stack ID>” is created during deployment. This is used to store updated templates when automatically adding members or expanding file systems. The updated templates can be deleted after the Automations are run, but can also be kept to track the history of changes to the stack. This bucket will not be deleted when deleting a stack if it is not empty. Manually delete the objects in the bucket to fully delete the stack.

Authentication keys used by the deployment process

The user is not required to create a key as part of the deployment process. An existing key is required by the Cloud Formation template and is applied to the Db2 pureScale cluster hosts. See the “Deployment” section on pre-requisites for information on creating an authentication key. After creating the key, the new key can be chosen from the drop-down option in the Security section of the deployment template (under KeyName).

Costs

AWS services

Users are responsible for the cost of the AWS services and any third-party licenses not included as part of Db2 license.

The AWS CloudFormation template includes configuration parameters that can be customized. Some of the settings, such as the instance type, affect the cost of deployment. For cost estimates, see the pricing pages for each chosen AWS service. Prices are subjected to change.

After the deployment is completed, create [AWS Cost and Usage Reports](#) to deliver billing metrics to an Amazon Simple Storage Service (Amazon S3) bucket in your account. These reports provide cost estimates based on usage throughout each month and aggregate the data at the end of the month. For more information, see [What are AWS Cost and Usage Reports?](#)

To reduce the cost of keeping all the AWS EC2 instances online, consider putting Db2 pureScale members in maintenance mode and shutting down those hosts. See the section “Reducing and Expanding the number of active Db2 pureScale Members” for details on this procedure.

Software licenses

The IBM® Db2 pureScale Feature on AWS employs Bring Your Own License (BYOL) model. New IBM Db2 license can be purchased from [Passport Advantage](#) (<https://www.ibm.com/software/passportadvantage/index.html>) or an [IBM representative](#). For assistance with Passport Advantage, see [Passport Advantage Online for customers](#) (https://www.ibm.com/software/passportadvantage/pao_customer.html).

Follow the steps under the Procedure in Applying Db2 licenses (<https://www.ibm.com/docs/en/db2/12.1?topic=licenses-applying-db2>) to apply the license manually after the installation.

Sizing

The IBM® Db2 pureScale Feature on AWS provides four pre-canned configurations – x-small, small, medium, and large. Below is the detail on each configuration with the expected throughput.

	X-Small	Small	Medium	Large
EC2 instance type	c6i.4xlarge	c6i.8xlarge	c6i.12xlarge	c6i.16xlarge
Number of Members	3	3	3	5
Number of CFs	2	2	2	2
Number of CPUs	16	32	48	64
RAM	32	64	96	128
IOPS	5000	5000	7000	10000
Network Bandwidth	Up to 12.5GbE	12.5GbE	18.75GbE	25GbE

Performance

The performance of these instance types has been tested with an OLTP workload, achieving the following SQL statement rates. These are intended as a rough guide to the expected performance – each workload has unique characteristics and should be tested to ensure that the chosen instance type is sufficient to meet the workload requirements.

Deployment Type	Instance type	90% read / 10% write SQL statements / sec	70% read / 30% write SQL statements / sec
X-Small	c6i.4xlarge	110,000	70,000
Small	c6i.8xlarge	190,000	125,000
Medium	c6i.12xlarge	440,000	300,000
Large	c6i.16xlarge	770,000	350,000

Deployment Assets

The following subsections cover both the initial deployment of the Db2 pureScale cluster as well as resources for setting up the AWS Transit Gateway for client connectivity.

Deployment Steps

The deployment of Db2 pureScale on AWS non-RDMA takes approximately 30 - 35 minutes. Follow the steps below to get started:

Prerequisites:

- An AWS account
- AWS Key Pair – If one has not been created, you can create one by following the guide here: [Create a key pair using Amazon EC2](#)

Instructions:

1. Start at the [IBM Db2 pureScale on RHEL](#) or [IBM Db2 pureScale on SLES](#) page . Click “Subscribe” (or “Continue to Subscribe” if you have already visited this page before) to see the license terms.
2. Click “Continue to Configuration” to choose a fulfillment option. Select “CloudFormation template” to deploy the solution and choose the region where this pureScale cluster will be deployed.

IBM Db2 pureScale

[< Product Detail](#) [Subscribe](#) [Configure](#)

Configure this software

Choose a fulfillment option and software version to launch this software.

Fulfillment option
Db2 pureScale on AWS

Software version
1 (Jun 29, 2022)

Whats in This Version
Db2 pureScale
running on c6i.12xlarge
[Learn more](#)

Region
US East (N. Virginia)

Use of Local Zones or WaveLength infrastructure deployment may alter your final pricing.

Product Code: d248upng7ehfhg50hcwfk0xtu
[Release notes \(updated June 29, 2022\)](#)

3. Click “Continue to Launch” to review the launch configuration details.
4. Click “Launch” to open the CloudFormation “Create stack” page, starting with “Specify template”. The correct template is automatically selected. Click “Next” to proceed to configure the stack.
5. On the Specify Details page, change the stack name if needed. Review the parameters for the template. Provide values for the parameters that require input. For all other parameters, review the default settings and customize them as necessary. When you finish reviewing and customizing the parameters, choose Next.

Parameter label (name)	Default	Description
Stack Name	-	<i>Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).</i>
DeploymentType:	Medium	<i>Choose the size of deployment - Extra Small (3 members, 16 CPU, 32GB RAM, up to 12.5GbE), Small (3 Members, 32 CPU, 64GB RAM, 12.5GbE), Medium (3 Members, 48 CPU, 96GB RAM, 18.75GbE), Large (5 Members64 CPU, 128GB RAM, 25GbE) or</i>
InstanceLocalStorageSize	100	<i>Set the size of the local storage for each instance in the cluster (in GB)</i>
Hostname	ps-aws	<i>This hostname will be used to name the instances, postfixed by the number of the instance. i.e. hostname-1, hostname-2, etc.</i>
RootPassword	-	<i>Set the password for the root user on each machine.</i>

AvailabilityZone	us-east-2	<i>Availability Zone of your Deployment</i>
VpcCIDR	0	<i>Value between 0 - 255 used to create a 16-bit CIDR block of the cluster VPC. For example, a value of 10 would result VPC subnet of 10.10.0.0/16.</i>
SecurityGroupCIDR		<i>Please set CIDR to x.x.x.x/32 to allow one specific IP address access, 0.0.0.0/0 to allow all IP addresses access, or another CIDR range.</i>
SharedInstanceStorageFileSystem	db2fs	Set the name of the Db2 instance shared file system.
SharedInstanceStorageMountPoint	/sqllibfs	Set the directory of the shared instance mount point.
SharedInstanceStorageSize	100	Set the size of the shared storage disk for the cluster (in GB).
DatabaseStorageFileSystem	datafs	Set the name of the shared database file system.
DatabaseStorageMountPoint	/datafs	Set the name of the mount point for database shared storage.
DatabaseStorageSize	100	Set the size of the database shared storage for the cluster (in GB).
LogStorageFileSystem	logfs	Set the name of the filesystem for log shared storage.
LogStorageMountPoint	/logfs	Set the name of the mount point for log shared storage.
LogStorageSize	100	Set the size of the shared storage disk for the cluster (in GB).
ArchiveLogStorageFileSystem	archlogfs	Set the name of the filesystem for archive log shared storage.
ArchiveLogStorageMountPoint	/archlogfs	Set the name of the mount point for archive log shared storage.
ArchiveLogStorageSize	100	Set the size of the shared storage disk for the cluster (in GB).
KeyName	-	<i>Name of an existing EC2 KeyPair to enable SSH access to the EC2 instances.</i>
DREnabled	False	<i>Choose whether or not to deploy a disaster recovery site</i>
DRHostname	ps-aws-dr	<i>Name of the hostname for the disaster recovery site</i>

DRAvailabilityZone	-	<p><i>Availability Zone of your disaster recovery Deployment.</i></p> <p><i>Note: this must be selected even if DREnabled is false or the deployment will error out.</i></p>
--------------------	---	--

6. On the “Configure stack options” page, you can specify tags (key-value pairs) for resources in your stack and set advanced options. When you’re done, choose Next.
7. On the “Review” page, review and confirm the template settings.
8. Choose Create to deploy the stack.
9. Monitor the status of the stack. When the status is CREATE_COMPLETE, the AWS Resources have been completed for the Db2 pureScale installation.
10. Approximately 30 minutes after the stack has been completed, verify the cluster has successfully installed Db2 pureScale by connecting to the *first* (hostname-1) host in the cluster. Open the file /var/log/cloud-init-output.log, and go to the bottom. Search for the following line, “*Db2 pureScale deployment has completed successfully! rc: 0.*”.
11. Switch to the instance user and run “db2instance -list” and validate that all members have been added, then run db2start.
12. Update the Db2 instance user password for each host.

Note: After deployment each of the hosts in the pureScale cluster will have a public IP, which can be used to connect via SSH after deployment, assuming the security group CIDR is permissive enough allow the connect through. These IPs will not persist if the instance hosts hibernated/shutdown, though they will persist across reboot. For some organizations, it may be desirable to remove these IP addresses via the EC2 console to meet security requirements.

If a permanent public IP is desired, provision [AWS Elastic IP](#)’s for each of the hosts.

Set Up Client Connectivity via AWS Transit Gateway

Once the pureScale cluster has been deployed, you can deploy additional EC2 Instances to be set up as clients. Clients within the same VPC as the pureScale cluster do not need an AWS transit gateway to communicate with the cluster hosts. However, if your Db2 clients exist within a separate VPC, or with in an on-premise data center, then an AWS transit gateway is the recommended way to enable connectivity between the clients and the Db2 pureScale cluster.

Refer to the following AWS documentation on AWS Transit Gateways to complete the following steps.

<https://docs.aws.amazon.com/vpc/latest/tgw/tgw-getting-started.html>

1. Create an AWS Transit Gateway if one does not already exist as described under the header “Step 1: Create the transit gateway” in the above documentation.
2. Attach the newly deployed Db2 pureScale VPC, as well as any VPC that might need to connect to the pureScale cluster to the transit gateway, as described by “Step 2: Attach your VPCs to your transit gateway” in the above documentation.
3. Setup routes between the VPCs as described by “Step 3: Add routes between the transit gateway and your VPCs”
4. Modify the security group of the pureScale cluster to allow communication with the Db2 clients.

Health Check

Db2 pureScale provides extensive monitoring capabilities through many table functions, administration views, and commands to query the state and alert information of underlying components at host, member, CF, network adapters levels. Overviews and details can be found in the following links:

- Monitoring in a Db2 pureScale environment (<https://www.ibm.com/docs/en/db2/12.1?topic=monitoring-in-db2-purescale-environment>).
- Interfaces for retrieving status information for Db2 pureScale instances (<https://www.ibm.com/docs/en/db2/12.1?topic=instance-interfaces-retrieving-status-information>).

Backup and Recovery

Refer to “Backup and restore operations in a Db2 pureScale environment” topic in IBM Documentation (<https://www.ibm.com/docs/en/db2/12.1?topic=backup-restore-operations-in-db2-purescale-environment>) for details.

Routine Maintenance

Db2 pureScale supports continuous availability through planned maintenance. Refer to “Maintenance in a Db2 pureScale environment” topic in IBM Documentation (<https://www.ibm.com/docs/en/db2/12.1?topic=environment-maintenance-in-db2-purescale>) for details.

In addition to planned maintenance, procedures to perform addition of new members to the cluster to accommodate expected workload increase is available in [Adding new members or an additional cluster caching facility](#).

Reducing and Expanding the Number of Active Db2 pureScale Members

Managing costs effectively can help reduce the Total Cost of Ownership. Once a Db2 pureScale cluster is deployed, some of the Db2 members may be stopped and shut down, reducing the hourly costs of keeping that EC2 instance active. It is strongly recommended that the Db2 pureScale cluster has an odd number of active hosts, so the procedure below is most suited to reducing a seven-host (five Db2 members and two CFs) cluster to five active hosts (three Db2 members and two CFs). Shutting down and restarting hosts is a much simpler procedure than adding new Db2 members to the originally provisioned cluster.

The following guidance is based on the procedure at [Performing maintenance on a Db2 pureScale host](#).

Prepare and shut down an EC2 instance

Quiesce the member host

Perform a member drain operation using the QUIESCE parameter of the db2stop command, as follows:

```
db2stop member <member-id> quiesce 30
```

This stops new connections being made to this member, and will move connections for completed transactions to other members in the cluster. This process is given 30 seconds to complete before the remaining connections are forced off, if any remain active. Using a larger timeout will likely reduce the number of statements forced.

Stop the instance on the member host

Now stop the instance on the member host

```
db2stop instance on <host-name>
```

Put the host into maintenance mode

As the Db2 cluster services administrator, perform the following steps:

- Put the cluster manager into maintenance mode on the host, by issuing the following command:

```
DB2INSTANCE=<instanceName> DB2DIR/bin/db2cm -cm -enter -  
maintenance
```

where DB2DIR represents the installation location of your Db2 copy.

- Put the cluster file system service on the host into maintenance mode, by issuing the following command:

```
DB2DIR/bin/db2cm -cfs -enter -maintenance
```

Halt the host

The member host can be halted at the command line.

As the system administrator, issue

```
shutdown now
```

or by using the AWS console to stop the EC2 instance.

Bring a host back online

Start the AWS instance

Using the AWS console, start the EC2 instance for the member, following the instructions at [Stop and start your instance](#).

Leave maintenance mode

As the Db2 cluster services administrator, perform the following steps:

- Exit cluster manager maintenance mode on the host by issuing the following command:

```
DB2INSTANCE=<instanceName> DB2DIR/bin/db2cm -cm -exit -  
maintenance
```


where DB2DIR represents the installation location of your Db2 copy.

- Exit cluster file system maintenance mode on the host by issuing the following command:

```
DB2DIR/bin/db2cm -cfs -exit -maintenance
```

where DB2DIR represents the installation location of your Db2 copy.

Start the Instance on the host

Restart the instance on the host by issuing the following command:

```
db2start instance on <host-name>
```

where host-name represents the name of the host of the given member.

Restart the member by issuing the following command:

```
db2start member <member-id>
```

where <member-id> represents the member that you have just taken out of maintenance mode.

Updating the EC2 instance type

EC2 instance types can be changed to scale up or scale down with demand and can be updated while the EC2 instance is offline. To shutdown the EC2 instance, follow the steps under [Prepare and shut down an EC2 instance](#). Once the EC2 instance is offline select the instance on the EC2 dashboard and click “Actions” > “Instance settings” > “Change instance type”. Next, select one of the supported EC2 instance types for IBM® Db2® pureScale® on AWS (c6i.4xlarge, c6i.8xlarge, c6i.12xlarge , c6i.16xlarge). After selecting a new instance type, follow the instructions under [Bring a host back online](#).

Repeat this one at a time until every EC2 instance is updated. While an EC2 instance is being updated, the Db2 instance can stay online on the remaining online hosts.

Once the host is back online, inspect the instance state using “db2instance -list” . All hosts should be online and there should be no alerts.

Adding a member to an existing Db2 pureScale cluster on AWS

Adding a member to IBM® Db2® pureScale® on AWS can be done automatically using the AWS Automation documents in the table below. After navigating to the document for the desired region, click “execute automation” in the top right of the AWS dashboard and fill in the fields under “input parameters”. “Cloudwatch alarm” and “Tags” can be skipped. After filling in the required parameters, click execute in the bottom right and the member will be added in ~20 minutes.

This automation will create a new EC2 instance, attach the volumes, set up the necessary networking and run `db2iupdt -add -member`. The EC2 instance type of the new member will be the same as the existing instances in the cluster. The automation will start `db2iupdt` and exit. Check the `db2iupdt` log on the first host for the status of the adding a member.

Note: Wait for the initial set up to complete before attempting to run the Automations. Do not run Automations concurrently.

The AWS SSM agent must be running on all hosts. The SSM agent will be started by default and the status can be checked with the following command.

```
systemctl status amazon-ssm-agent
```

Automation documents for adding a member

us-east-1
us-east-2
us-west-1
ca-central-1
eu-central-1
eu-west-1
eu-west-2
eu-north-1
ap-east-1
ap-southeast-1
ap-southeast-2
ap-northeast-2
ap-northeast-1
ap-south-1
me-south-1

Expanding a file system in an existing Db2 pureScale cluster on AWS

Expanding a file system on IBM® Db2® pureScale® on AWS can be done automatically using the AWS Automation documents in the table below. After navigating to the document for the

desired region, click “execute automation” in the top right of the AWS dashboard and fill in the fields under “input parameters”. “Cloudwatch alarm” and “Tags” can be skipped. After filling in the required parameters, click execute in the bottom right and the disk will be added in ~5 minutes.

This automation will create a new volume, attach it to the EC2 instances and run `db2cluster -cfs -add -filesystem <filesystem> -disk <disk>` automatically. The IOPS of the new volume will depend on the deployment type. See the table in the Sizing section for more details.

Note: Wait for the initial set up to complete before attempting to run the Automations. Do not run Automations concurrently.

The AWS SSM agent must be running on the first host. The SSM agent will be started by default and the status can be checked with the following command.

```
systemctl status amazon-ssm-agent
```

Automation documents for filesystem expansion

us-east-1
us-east-2
us-west-1
ca-central-1
eu-central-1
eu-west-1
eu-west-2
eu-north-1
ap-east-1
ap-southeast-1
ap-southeast-2
ap-northeast-2
ap-northeast-1
ap-south-1
me-south-1

Adding a new filesystem to an existing Db2 pureScale cluster on AWS

Create a new multi-attach io2 storage device

The following instructions are based on the full documentation available at [Attach a volume to multiple instances with Amazon EBS Multi-Attach](#).

- Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>
- In the navigation pane, choose Volumes.
- Choose Create volume.
- For Volume type, choose Provisioned IOPS SSD (io1) or Provisioned IOPS SSD (io2).
- For Size and IOPS, choose the required volume size and the number of IOPS to provision.
- For Availability Zone, choose the same Availability Zone that the instances are in.
- For Amazon EBS Multi-Attach, choose Enable Multi-Attach.
- Set the encryption status for the volume.
- (Optional) To assign custom tags to the volume, in the Tags section, choose Add tag, and then enter a tag key and value pair. If you have used tags on your pureScale cluster, you should consider adding similar tags to this new volume.
- Click 'Create Volume'

[Attach this multi-attach io2 storage device to all the hosts in the cluster.](#)

You attach a Multi-Attach enabled volume to an instance in the same way that you attach any other EBS volume. For more information, see [Attach an Amazon EBS volume to an instance](#). The Multi-Attach volume must be attached to all of the hosts in the pureScale cluster.

Make a note of the disk device path that the operating system will see this disk under for the given instance. This is easiest to determine by noting the time that the device is associated with the instance. Storage volumes will appear as NVMe devices, typically seen as `/dev/nvme<number>n1`.

As each instance is attached to the volume, it will appear under "Attached instances" as you add further instances.

[Create a new file system using the new volume](#)

On one of the hosts in the cluster, identify the new storage device. For example, this may be `/dev/nvme5n1` if this is the first additional storage volume added to the standard configuration.

More information on this area can be found in the [Creating a shared file system](#) documentation.

Run `db2cm` as the System Administrator to create a Spectrum Scale file system

```
db2cm-create -filesystem mydb2database -disk /dev/nvme5n1 \  
-mount /db2fs/mydb2database
```

This should complete with the message

File system 'mydb2database' has been successfully created.

The new mount point can be inspected using regular system tools, for example:

```
[root@demo-5 ~]# mount | grep /db2fs/mydb2database
db2fs on /sqllibfs type gpfs (rw,relatime)
mydb2database on /db2fs/mydb2database type gpfs (rw,relatime)
[root@demo-5 ~]# ls -l /db2fs/mydb2database
total 0
[root@demo-5 ~]# ls -ld /db2fs/mydb2database
drwxr-xr-x 2 root root 262144 Oct 15 20:49 /db2fs/mydb2database
```

The new mount point should be present on all the hosts in the cluster. The above command was run host demo-5. We should see the same information on any other host – e.g. on host demo-2:

```
[root@demo-5 ~]# ssh demo-2 mount | grep mydb2
mydb2database on /db2fs/mydb2database type gpfs (rw,relatime)
```

Create any directories needed in this file system and ensure that the directory ownership and permissions are appropriate for the database instance account.

Additional tasks

Refer to the IBM Documentation for further information on using this new storage volume. For example:

- [Adding a storage path to an existing database](#)
- [Changing the existing log path to a new directory](#)

Emergency Maintenance

Db2 pureScale provides multiple utilities to examine and repair the cluster, refer to the following links for detail:

- Status monitoring of a Db2 pureScale instance
(<https://www.ibm.com/docs/en/db2/12.1?topic=environment-status-monitoring-db2-purescale-instance>).
- Troubleshooting the Db2 pureScale Feature
(<https://www.ibm.com/docs/en/db2/12.1?topic=techniques-troubleshooting-db2-purescale-feature>)

Support

Db2 pureScale on AWS is supported through regular IBM Support channel (<https://www.ibm.com/docs/en/db2/12.1?topic=support-contacting>)

FAQ

Q: What are the motivations to release pureScale on AWS?

A: Three reasons:

1. Our response to an increasing number of requests to deploy pureScale on Cloud to target new workloads. For example. a phone app that provides catalog for customers to browse and purchase items.
2. Align with IBM cloud strategy for all offerings - pureScale included, in hybrid mode.
3. This opens up many new use cases for existing on-premises and new cloud only customers. It is important to point that for on-premises customers, this DOES NOT mean they will move to cloud overnight, if at all. It means they can now leverage the speed of deployment on cloud to accomplish their routine that used to takes months with huge infrastructure costs to day(s) or week(s) with significant lower cost of ownership.
 - See the section “Use Cases” in this document for more detail.

Q: How is this new offering different from on-premises?

A: First of all, the current offering is pureScale with commodity Ethernet (TCP/IP) as cluster interconnect. RDMA or equivalent support will be in a future release. So, it is important to set the performance expectation to be similar to that on-premises. In terms of cluster topology, it has the same look and feel as on-premises. The same engine functionality is available on AWS - there is no delta between cloud vs on-premises.

Perhaps, the biggest difference is the up & running experience. For on-premises, the elapsed time from initial preparation to cluster deployed can take multiple months. This offering, on the other hand, can achieve this within 40 minutes for a medium size cluster. This opens a LOT of new use cases with this solution.

Q: What Linux distributions and Db2 versions is this offering supported?

A: This offering is available with the latest mod pack release in the 12.1 stream. The release will be refreshed on a regular cadence with the latest cumulative special build on Redhat 9.4 or SLES 15 SP6 Linux distributions. However, it is important to emphasize that the EC2 instance, the OS installation, Db2 installation and pureScale cluster setup are all automated with a click of a button.

Q: What type of workload is suitable with this offering?

A: With our internal throughput tests using varying workload of R/W ratio on a 3-member (Medium) vs 5-member (Large) clusters,

- write-heavy workload (R/W - 70/30) show modest gains on the large cluster compared to medium - average 350,000 SQL/sec compared to 300,000 SQL/sec on medium
- read-mostly workloads show substantial capacity - averaging over 3/4 of a million SQL statements per second (~770,000 with large cluster compared to ~440,000 with medium cluster)

Refer to the throughput guidelines for each of the deployment sizes in the “Performance” section of this document.

Q: Can this offering be deployed in any region?

A: Yes, any AZ in any region can be chosen for deployment where multi-attach EBS io2 volumes are available. For more information, see the [Supported Regions](#) section.

Q: Is the pureScale solution on AWS using shared storage or NFS or Spectrum Scale?

A: pureScale on AWS has the same prerequisite as on-premises. It supports only one type of cluster file system - Spectrum Scale - to be deployed within the same host as the member and CF hosts (not outside). The deployment takes care of all these, customers do not need to install or configure Spectrum Scale independently. The offering on AWS does use the newly available multi-attach EBS storage to provide shared storage among all hosts.

Q: Is there any differences in managing or monitoring a pureScale instance on AWS?

A: No, the same commands or utilities on-premises are applicable to cloud, for example:

- db2instance

- db2cm
- db2pd
- db2mon
- All MON_GET_* table functions relevant to pureScale are all applicable.

Q: How does future updates of Db2 or OS work with this new offering?

A: Keep in mind that this is a self-managed offering. This means once the initial deployment is completed, the entire stack is owned and managed by you. You can update the OS and Db2 just as on-premises at any time with the following caveats:

- OS level: The new OS level must be one that supported by the Db2 pureScale installed.
- Db2 level: As of the first release, only 11.5.6.0 is supported due to TSA. This limitation will be lifted in 11.5.8.0 which will be picked up automatically from the AWS marketplace for new deployment. There is no need for customers to choose a version of Db2. The latest one from the same release stream will always to used, starting from the next release.

In short, once the next mod pack is out, there are two options for customers who have deployed with the current offering:

1. Use online Db2 update - as in on-premises. In this case, customers will need to download the new Db2 release themselves into their AWS instance and follow the same online update procedures for each host.
2. Deploy new pureScale cluster and start over. New deployment will always use the latest Db2 version within the same major release.

Q: Is the recovery performance the same as on-premises?

A: With support for I/O fencing in Db2 V12.1.1.0, all failures have comparable time-to-recover performance.

Q: When will RDMA-equivalent support be available?

A: This is planned for a future Db2 release.

Q: Can we deploy Pacemaker in such a way as to automate the failover from the primary host to the standby host?

A: Starting in V12.1, the default cluster manager used for pureScale on AWS deployments will be Pacemaker.

Q: Can AWS EBS Snapshot be used in the DR site (in a different region)?

A: Yes. You will need to suspend I/O processes just like an on-premises configuration, by using SET WRITE SUSPEND.

Q: Can altering the configuration of the EC2 instance after deployment, such as lowering IOPS at night for cost reduction, void support from IBM?

A: No, the cluster is under customers' control once deployed just as on-premises. But customers should consider the impact of change, validate it on a test environment before implementing it in production.

Q: Is a do-it-yourself pureScale instance on AWS supported? For example, an instance that is not deployed through Marketplace.

A: No, not unless the cluster was previously approved by Db2 with an official written consent.

Q: Is there any integration with other AWS services?

A: No.