

Db2 Click to Containerize

Tridex Db2 LUW



George Baklarz
Principal Data & AI Technical Specialist – Americas



Agenda

- pureScale on AWS Update
- Introduction to Db2U and Containerization
- Db2 Click to Containerize Update
- Shift Scenarios
- Resources

IBM Db2 pureScale Primer

Db2 pureScale is an active-active database cluster solution for *transactional workloads*. It reduces the risk and cost of business growth.

IBM® Db2® pureScale® leverages the parallel sysplex architecture and data sharing technology to provide mainframe-class continuous availability (5 9s)¹

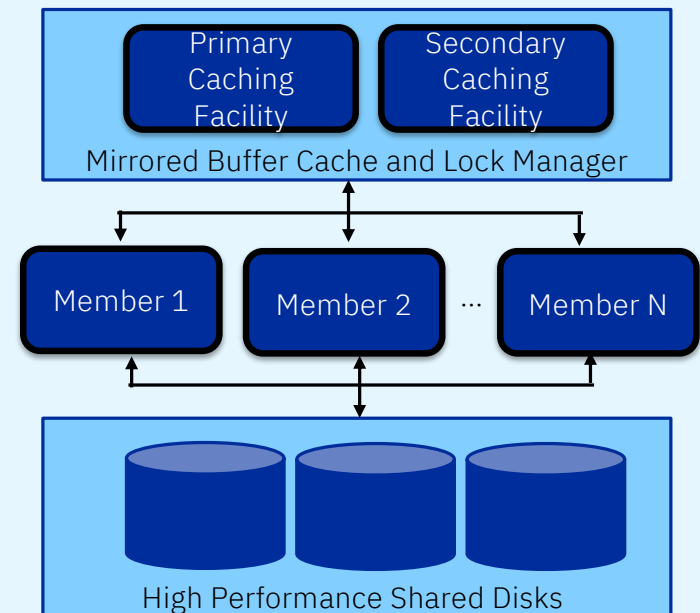
1. **Always on - No unplanned downtime**
2. **Extreme scalability – Unlimited capacity**
3. **Application transparency – No application changes, no data redistribution, and no performance tuning required**

¹[An IBM case study on PUMA with Db2 pureScale](#)



Business Challenge: Flexible scaling, microservices, and continuous availability of Db2 to interact with an increasingly digital world.

Solution: Add Db2 pureScale feature to support flexible scaling. Develop new functions as microservices in Red Hat® OpenShift® containers for faster digital innovation.



Db2 pureScale on AWS Marketplace



The screenshot shows the AWS Marketplace product page for IBM Db2 pureScale. The page includes the AWS Marketplace logo, navigation menu, and product details. The product is titled "Db2 pureScale" and is by "IBM Data and AI". The latest version is 2. The description states that the product eliminates unplanned downtime by providing mission-critical workloads with continuous availability, application transparency, and shared Linux/Unix BYOL. The typical total price is \$2.165/hr. The page also features a "Product Overview" section with a detailed description of the product's benefits and a "Highlights" section with three key features.

Db2 pureScale
By: [IBM Data and AI](#) Latest Version: 2

The IBM® Db2® pureScale® on AWS feature eliminates unplanned downtime by providing your mission critical workloads with continuous availability, application transparency, and shared Linux/Unix BYOL.

Continue to Subscribe
Save to List

Typical Total Price
\$2.165/hr
Total pricing per instance for services hosted on c6i.12xlarge in US East (N. Virginia). [View Details](#)

Overview Pricing Usage Support Reviews

Product Overview

Mission critical workloads require continuous availability at scale. With best-in-class failure detection and recovery, eliminate unplanned downtime while achieving high performance with IBM® Db2® pureScale®. pureScale® leverages the IBM® Db2® parallel sysplex architecture, providing mainframe-class continuous availability that runs anywhere, whenever you need it. pureScale® enables shared cluster scale out, starting with a small cluster, and allowing you to grow your workloads as much or as little as you need. With IBM® Db2® pureScale® there are no application changes to make, data to redistribute, or performance tuning to do. Provide application transparency through automatic load balancing across all active members. Achieve optimal resource utilization at all times, which helps to keep application response times low, while reducing the risk and cost of application changes. Start building more, while managing less.

Highlights

- Cluster of any size ready in minutes.
- Dramatic reduction in total cost of ownership.
- Disaster Recovery with extreme ease.

Version	2 Show other versions
By	IBM Data and AI
Categories	Databases & Analytics Platforms Databases
Operating System	Linux/Unix, SUSE 12 SP5

<https://aws.amazon.com/marketplace/pp/prodview-fyclwzxhlopby>

Db2 pureScale on AWS Documentation

Deploying IBM Db2 pureScale with TCP/IP private network on AWS.pdf 12 / 24 100%

Introduction

- Value Propositions with pureScale on AWS
- Use cases
- Deployment Overview
- Supported Regions
- Prerequisites and Requirements
- Architecture Diagrams
- Security
- Costs
- Sizing
- Deployment Assets
- Health Check
- Backup and Recovery
- Routine Maintenance
- Emergency Maintenance
- Support
- FAQ

Architecture Diagrams

Figure 3 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

12

<https://ibm.biz/purescaleaws>

The Challenges of Containerizing Databases

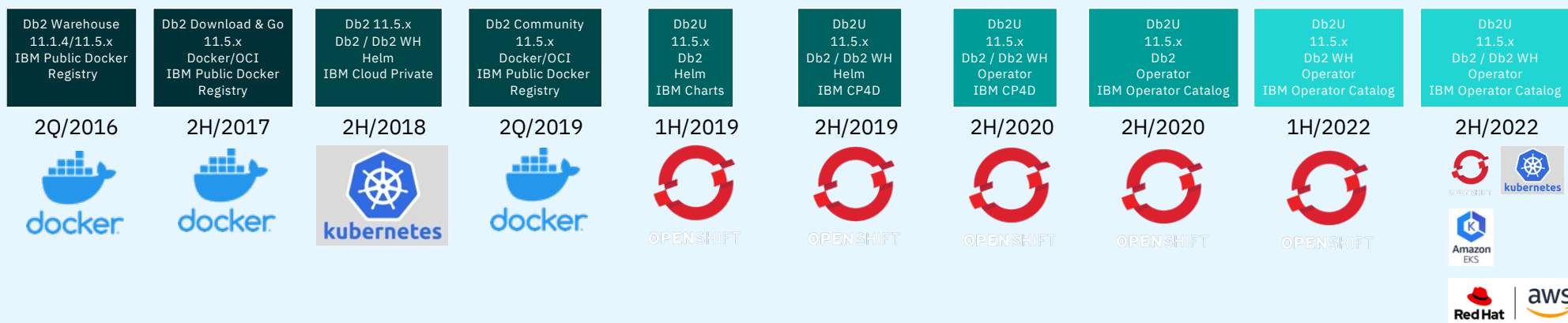
If a containerized application has a built-in database, that database disappears when the container shuts down and is recreated from scratch when it boots back up again.

Many databases were not designed to be cloud native, compatible with containers or orchestrated by Kubernetes.

Because of the benefits of moving to a containerized environment, there's a great deal of interest from enterprises in finding ways to run SQL and no-SQL databases on Kubernetes.

The trick is to provide developers with a data fabric that just works, without forcing developers to struggle with security, auditability or scalability.

Db2 Containerization Evolution



Db2 Containerization Support

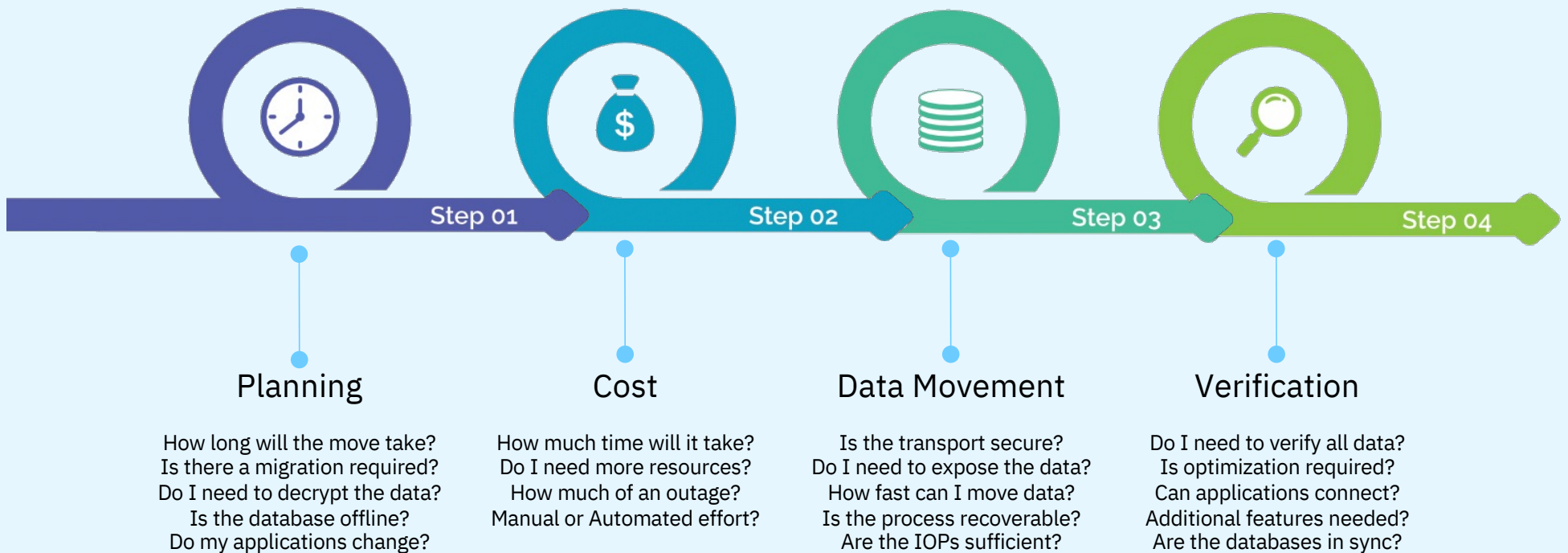
The screenshot shows the Red Hat Ecosystem Catalog search results for IBM Db2 container images. The page title is "Certified container images" and the subtitle is "Container images offer lightweight and self-contained software to enable deployment at scale." The search results are displayed in a grid of cards, each representing a different IBM Db2 container image. The cards are organized into three rows and three columns. The first row contains three cards: "db2-11.5.0", "db2instb-v11.5.0", and "db2auxiliaryauth-v11.5.0". The second row contains three cards: "db2tools-v11.5.0", "db2rest-v11.5.0", and "db2graph-v11.5.0". The third row contains one card: "db2-operator-v11.5.0". Each card includes the IBM logo, the image name, a brief description, and the update date. The left sidebar contains filters for Provider (IBM), Category (Database & Data Management), Product (IBM Db2 on Cloud Pak for Data), Image Type (Container Application), Architecture (amd64), and Release Category (Generally Available). The search results section is currently empty.



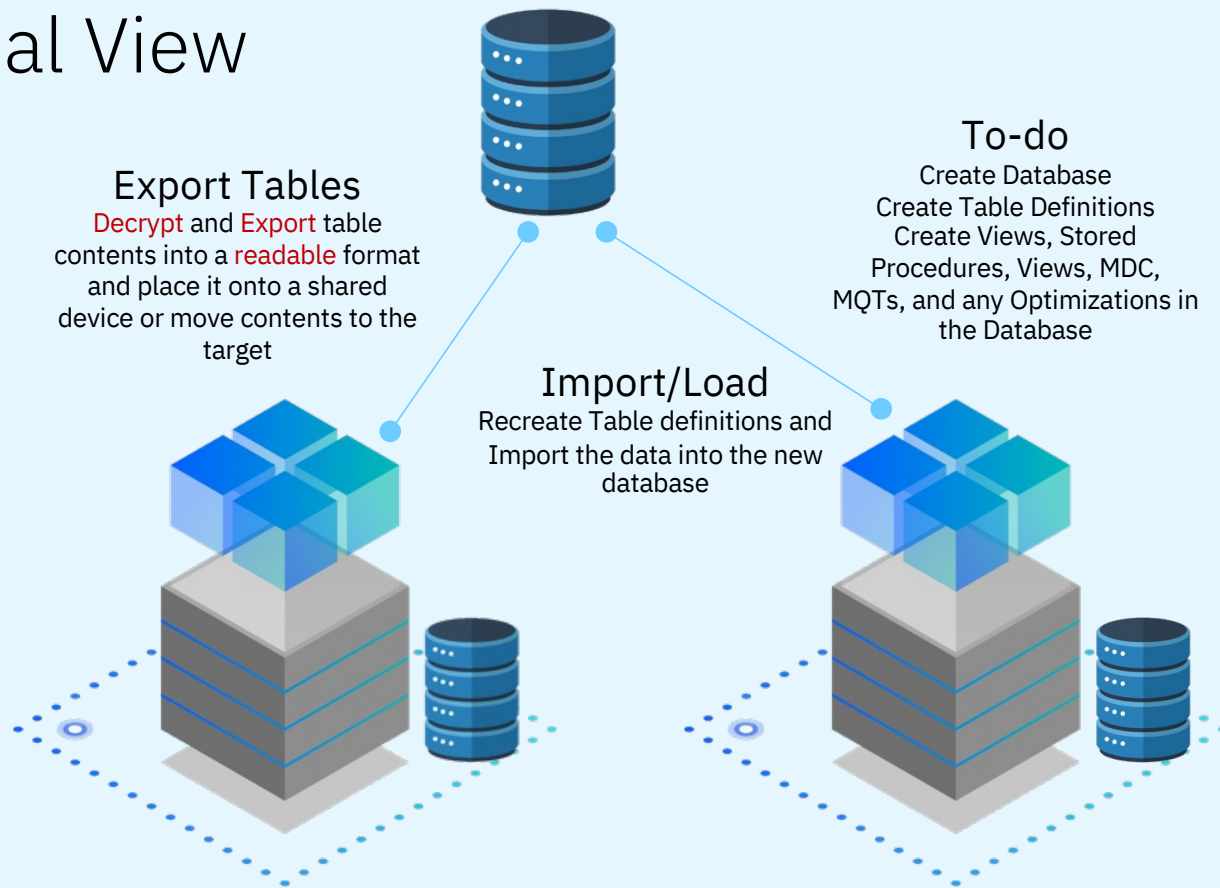
Cloud Pak for Data
Everything you need for enterprise data and AI, on any cloud

The diagram illustrates the Cloud Pak for Data architecture. It shows a stack of layers: Ingest, Analyze, Organize, and Collect. To the right, there are three boxes representing the capabilities of the platform: Business users and applications, Comprehensive Information (Architecture Design, Unified experiences for all your capabilities), and A Hybrid-Cloud world (Provider / Infrastructure / Data Sources). The bottom right corner features logos for AWS, Azure, and On-Prem.

Migrating to Containers



Migrating a Database A Traditional View

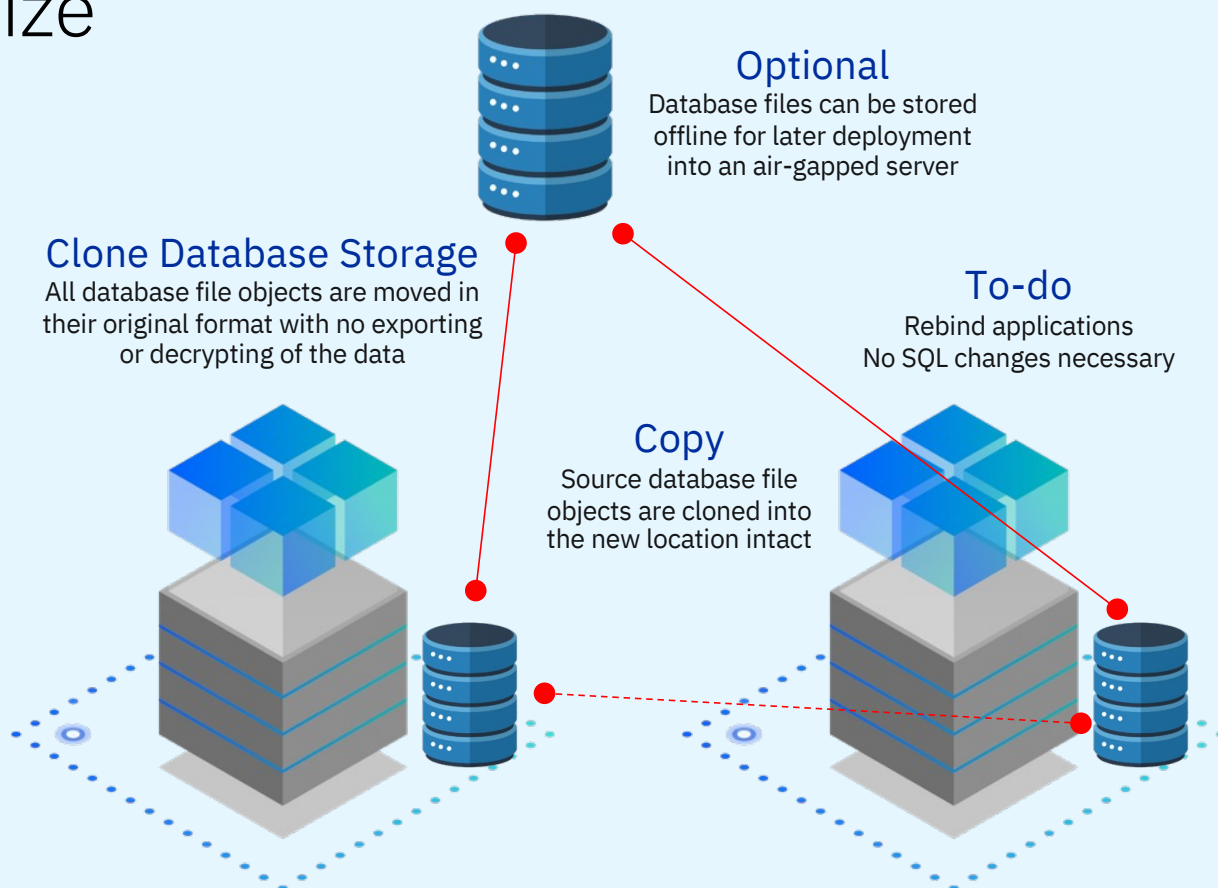


Db2 Click to Containerize

The fastest way to move your data from an On-premise Db2 Linux system to a Cloud Environment

- Move to OpenShift, Kubernetes, or Cloud Pak for Data
- Containerize your data or move to a traditional server instance
- Command line or UI tool for flexibility
- With automatic database upgrades

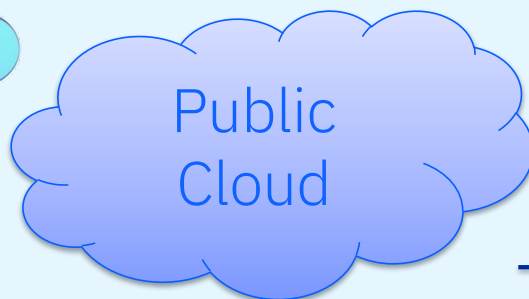
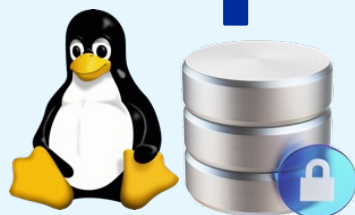
More info: ibm.biz/c2cguide



Choice of Destination and Linux Distributions

Db2 Linux Server x64
or powerLE

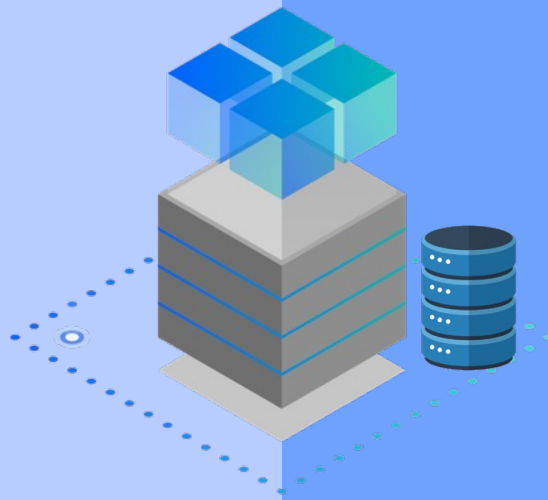
- 10.5 fp7
- 11.1
- 11.5



Which Approach to Use?

Database Migration

- Move between different O/S (AIX, zOS)
- Change configuration and codepages
- Unsupported database versions
- All or subset of data
- SQL Access only
- Topology change (MLNs for DPF)
- Object recreation required
- Data must be exported and decrypted

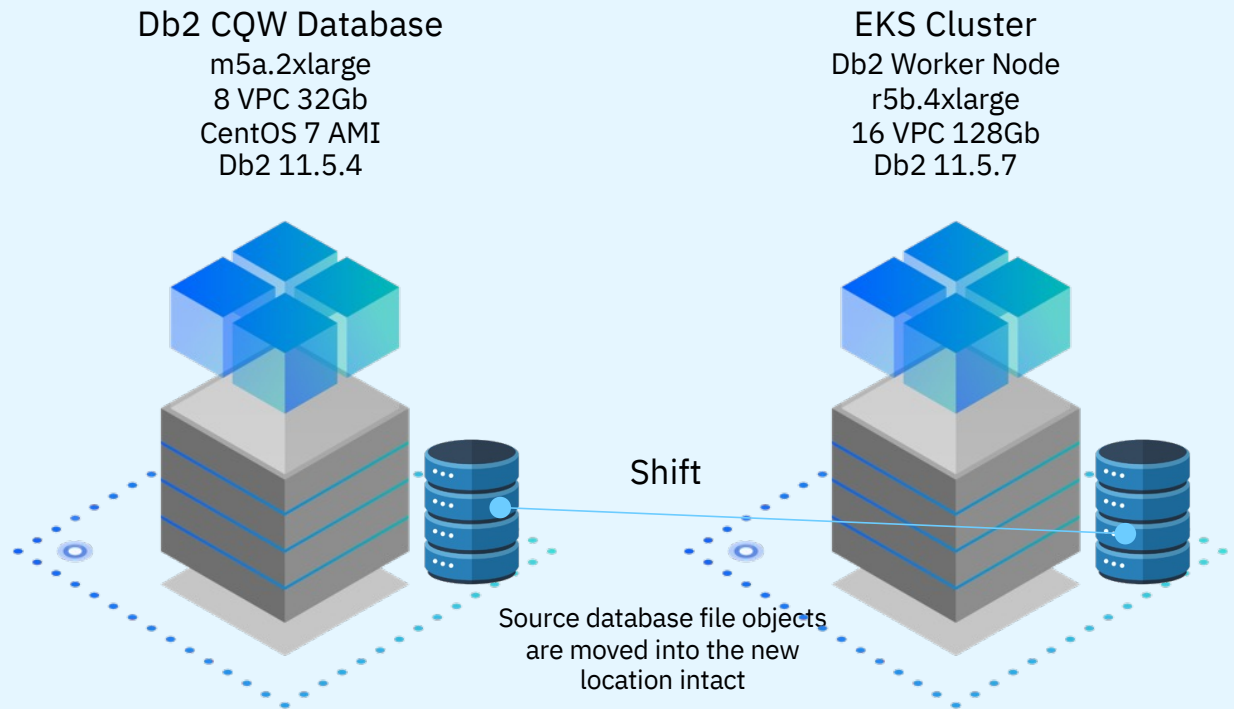


Db2 Click to Containerize

- Move only Linux to Linux environment
- All database features moved intact
- Current releases only 10.5, 11.1, 11.5
- Entire database movement
- Direct file access, no SQL used
- No change in configuration allowed
- No rebuild of objects required
- No exposure of the data

Demonstration Scenario

Table	Size
Database Size (RAW)	100Gb
Customers	2000000
Addresses	941566
Customer Demographics	1920800
Date Dimensions	73049
Household Demographics	7200
Income Bands	20
Items	202960
Promotions	1000
Stores	394
Store Sales	262185014
Time Dimensions	36000

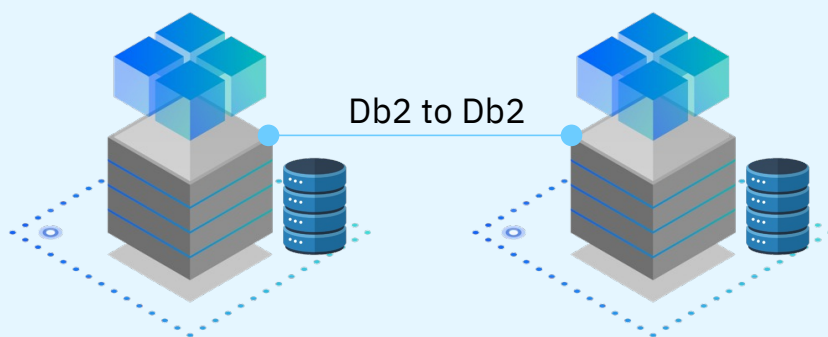


Shift Targets

- The Shift targets have been expanded in the newest release
 - Continues to support OpenShift 3.11, and 4+ clusters
 - CP4D V3.5 and V4+ are supported
 - **New!** Kubernetes Distributions (1.19+) are supported
 - **New!** MicroK8S, MiniKube for small systems
- Operating System Support
 - CentOS 6, 7, 8, RHEL 6, 7, 8, CentOS Stream
 - Ubuntu 18.04, 20.04, 22.04
 - Power LE
 - SUSE 15

Shift Modes

- The Shift mode has been expanded
 - Shift between Db2 instance and another Db2 instance (traditional Db2)
 - Create Database on target if it doesn't exist (Db2 instance only)
 - Set up HADR at target and initialize source and target (Pod or Instance)
 - Create cloned databases without requiring target information (Pod or Instance)
 - Useful in air-gapped environments
 - Clone copy to and from a POD
 - Reverse shift (back-out) from POD back to an instance

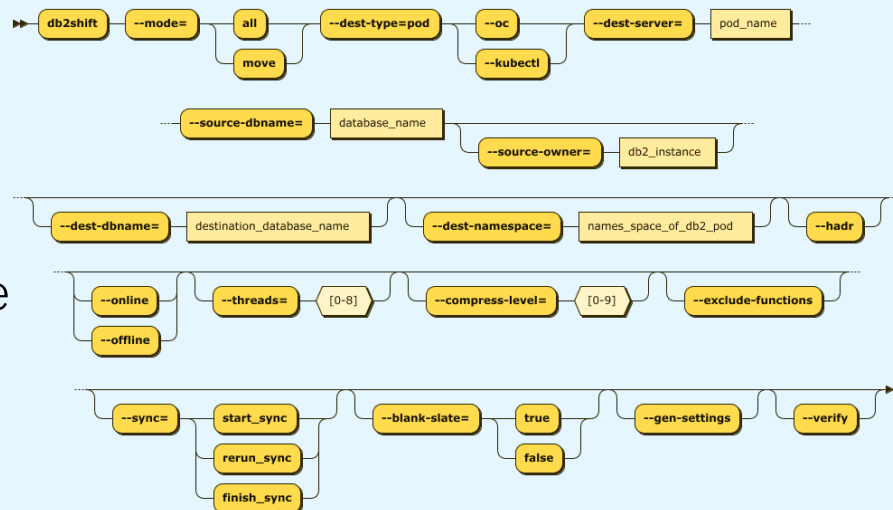


Db2 Shift Scenarios

- The Db2 Shift utility provides support for 9 scenarios
 - Shift Operations
 - Shift a Db2 database to OpenShift, Kubernetes or CP4D
 - Shift a Db2 database to another Db2 instance
 - Clone and Deploy a Database
 - Create a Cloned copy of the Db2 database for later deployment
 - Deploy a clone into an OpenShift, Kubernetes or CP4D container
 - Deploy a clone into another Db2 instance
 - HADR Operations
 - Initialize HADR between Source and Target POD
 - Initialize HADR between Source and Target Instance
 - Initialize DMC and LDAP Authentication for CP4D
 - Copy Cloned Databases to a POD

Containerize Db2 to OpenShift, Kubernetes, or CP4D

- Containerizing an existing Db2 database on an on-premise system to OpenShift, Kubernetes, or CP4D is the most common scenario for using Db2 Shift
- In order to move a Db2 database to a POD you will require the following information:
 - Source Database details
 - Destination location and POD details
 - Type of Containerization environment
 - Shift Options
- The target of the Db2 Shift operation can be OpenShift, a Kubernetes cluster, or Cloud Pak for Data



Containerize Db2 to OpenShift, K8s or CP4D Dialog

- The Source and Destination Details are the only required fields

```
Db2 Shift [/home/c2cdb2/c2c]
                                Shift to Db2U on OpenShift or Kubernetes
Source Details
Database Name : flights_
Instance Owner : db2inst1.....

Destination Details
Target Location :  OpenShift  Kubernetes
Database Name : db2oltp_
POD Project : db2u.....+
POD Name : c-demo-db2u-0.....+

Metadata :  Refresh  None  Settings Only  Verify
Sync Options :  None  Initialize  Refresh  Finalize
Move Options :  Database  Database/LDAP/DMC  Move Database for HADR
Exclude Routines :  Do not move external routines
Database Mode :  Online Move  Offline Move
Thread count : 0
Compression : 4
Overrides : .....+

ESC+Quit ^?+Field Help ^A+Analyze ^X+Review and Execute
```

- The additional fields provide more options for the Shift process

Optional Fields

- Metadata
 - Used for gathering information on the source and target destination
 - No need to use this unless you want to test the connections between the source and the target (verify) or you are shifting a database that needs to be offline (upgraded)
- Sync Options
 - By default, all shifts are performed from beginning to end
 - You can shift incrementally by using the Initialize/Refresh/Finalize Options
- Move Options
 - Use Database for OpenShift and K8s, Database/LDAP/DMC for CP4D, and HADR if you want the target POD created as a standby for the primary database
- Exclude Routines
 - Do not move any external routines found on the source database library path

Optional Fields (Continued)

- Database Mode
 - The database can be online during a shift operation (11.5 only)
 - For databases that require an upgrade (10.5 and 11.1), the database needs to be offline for the shift to work properly (Db2 requirement)
- Thread Count
 - The Db2 shift utility can use up to 8 threads when communicating with the target location
 - The more threads, the more throughput, but at the expense of CPU resources
 - The limit is usually the Ethernet performance, not the threads
- Compression
 - The data stream can be compressed when communicating with the target
 - If the network is >1Gb/sec then there is no benefit from using this setting
 - Anything less than 1Gb/sec may find compression helps at the expense of CPU cycles

Optional Fields (Continued)

- Overrides
 - All source database settings are retained at the target location
 - Instance settings do NOT move to the target
 - You may need to adjust the target's instance settings
 - This field provides a way of setting instance values that will be updated during the shift process
 - The Analyze option (^A) will provide a summary of the differences between the source and target and allow you to select which settings to override

```
Db2 Shift [/home/c2cdb2/c2c]
Review the settings below and select which ones should be overridden. Use ^?
to view details of the parameter and links to the documentation.

Parameter          Source              Destination

EXT_PROC_COUNT

The system contains external procedures which are not in the standard
Db2 library. You will need to manually recreate and catalog them.

ESC=close

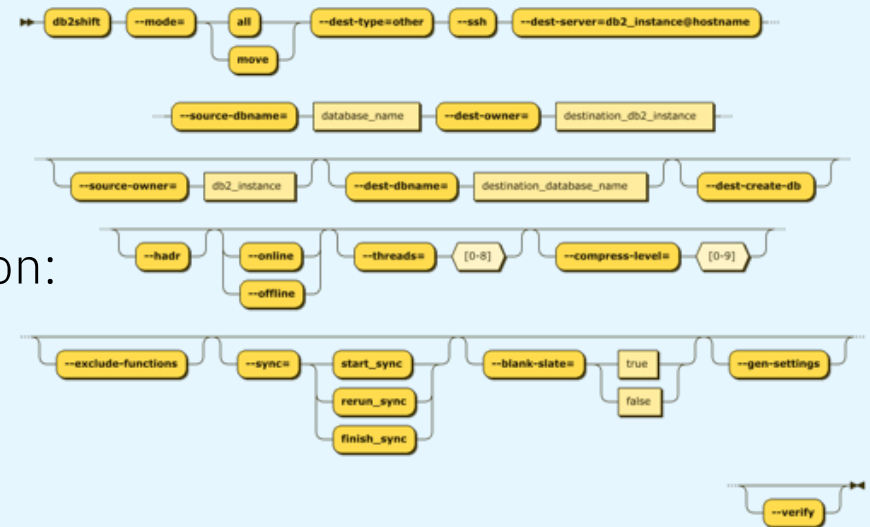
_ cur_eff_code_lvl  V:10 R:5 M:5 F:0 I:0 SB Automated migration
_ encrypted_database YES Encrypted database
o repl_site_id     0
o authentication   SERVER SERVER_ENCRYPT
o instance_memory  3468896 598630

^Q=Quit ^F=Field Help ^X=Generate Overrides
```

Overrides : instance_memory 3468896_-----

Shift Db2 to another Instance

- This format of the Db2 Shift will take an existing Db2 database on an on-premise system, and shift it to another traditional Db2 system to:
 - Another on-premise server
 - A Cloud virtual machine (i.e. EC2)
- This does not containerize Db2!
- The command requires the following information:
 - Source Database details
 - Destination location
 - Shift Options
- The db2shift program assumes that you are currently connected to the instance that has the Db2 database and have ssh connectivity to the target server



Shift Db2 to another Instance

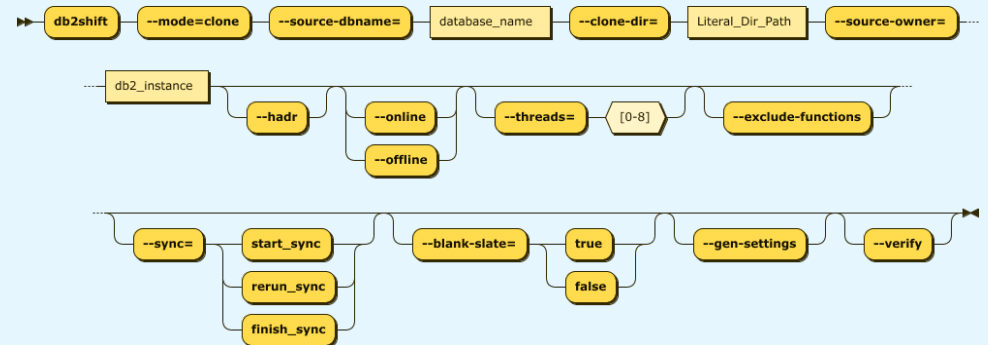
- The Source and Destination Details are the only required fields

```
Db2 Shift [/home/db2inst1/c2c]
Shift Db2 to another Db2 instance
Source Details
Database Name : banking_
Instance Owner : db2inst1_-----
Destination Details
Database Name : banking_ ● Force Database Creation
Instance Owner : db2inst1_-----
Server Address : 10.0.0.2_-----+
Metadata      : ● Refresh ○ None ○ Settings Only ○ Verify
Sync Options  : ● None ○ Initialize ○ Refresh ○ Finalize
HADR Option   : ○ Move Database for HADR
Exclude Routines : ○ Do not move external routines
Database Mode : ● Online Move ○ Offline Move
Thread count  : 4
Compression   : 0
Overrides     : █_-----+
Enter INSTANCE overrides as setting value; setting value; etc...
ESC•Quit ^F•Field Help ^A•Analyze ^X•Review and Execute
```

- The **FORCE DATABASE CREATION** option will create the database at the target
- The additional fields provide more options for the Shift process and are identical to the Db2 Shift to POD command

Clone a Database

- The Db2 Shift clone option is used to take an existing Db2 database that is currently on-premise, and clone it into a directory
 - This cloned database can be transported to another server and deployed at that location
 - The advantage of cloning is that the destination does not need to be connected to the source location and the deployment of the clone can be done at a more convenient time
- The Db2 Shift program requires the following information:
 - Source Database details
 - Clone Options
- The destination details are not required to clone a database



Create a Clone of a Database

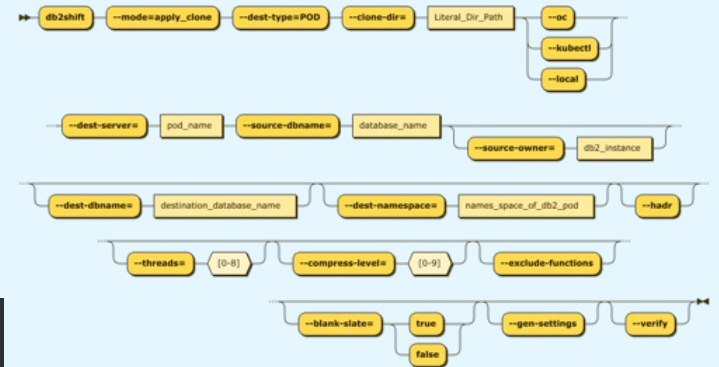
- The Source is required as well as the directory to place the cloned copy

```
Db2 Shift [/home/c2cdb2/c2c]
Clone a Database for Future Deployment
Source Details
Database Name : flights_
Instance Owner : db2inst1
Directory : /tmp/cache
Metadata :  Refresh  None  Settings Only  Verify
Sync Options :  None  Initialize  Refresh  Finalize
Exclude Routines :  Do not move external routines
Database Mode :  Online Move  Offline Move
Thread count : 4
ESC•Quit ^?•Field Help ^D•Directory ^X•Review and Execute
```

- The additional fields provide more options for the Shift process and are identical to the Db2 Shift to POD command
- HADR and Compression options are not used

Deploy a Clone to a POD

- This Db2 Shift option will take a database clone and deploy it into a Db2u pod running on OpenShift, Kubernetes or CP4D
- The requirements for deploying to a POD are identical to a direct shift
- The command requires the following information:
 - Database name and clone location
 - Destination POD details
 - Type of Containerization environment
 - Shift Options



```

Db2 Shift [/home/c2cdb2/c2c]
Deploy a Clone Copy into a Container
Source Details
Database Name : flights
Instance Owner : db2inst1
Directory : /tmp/cache
-----
Destination Details
Target Location :  OpenShift  Kubernetes  Local
Database Name : flights
POD Project : db2u
POD Name : c-demo-db2u-0
-----
Metadata
HADR Option :  Refresh  None  Settings Only  Verify
Exclude Routines :  Do not move external routines
Thread count : 4
Compression : 4
Overrides :
-----
ESC=Quit ^?=Field Help ^O=Select Directory ^A=Analyze ^X=Review and Execute
    
```

Deploy a Clone an Instance

- This Db2 Shift option will take a database clone and deploy it into another Db2 instance running natively (in any environment including Cloud VMs)
- The requirements for deploying to an Instance are identical to a direct shift
- The command requires the following information:
 - Database name and clone location
 - Destination instance details
 - Shift Options

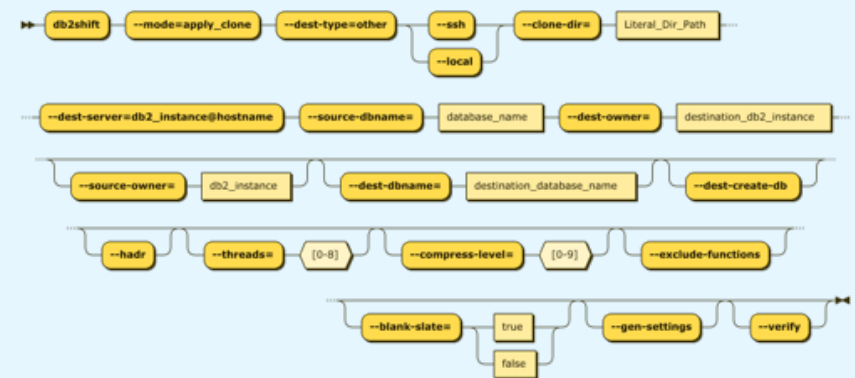
```
Db2 Shift [/home/c2cdb2/c2c]
Deploy a Clone Copy into a Db2 Instance

Source Details
Database Name : flights_
Instance Owner : db2inst1
Directory : /tmp/cache

Destination Details
Target Location : ● Remote ○ Local
Database Name : flights_ ○ Force Database Creation
Instance Owner : db2inst1
Server Address : some.server.com

Metadata : ● Refresh ○ None ○ Settings Only ○ Verify
HADR Option : ● Move Database for HADR
Exclude Routines : ● Do not move external routines
Thread count : 4
Compression : 4
Overrides :

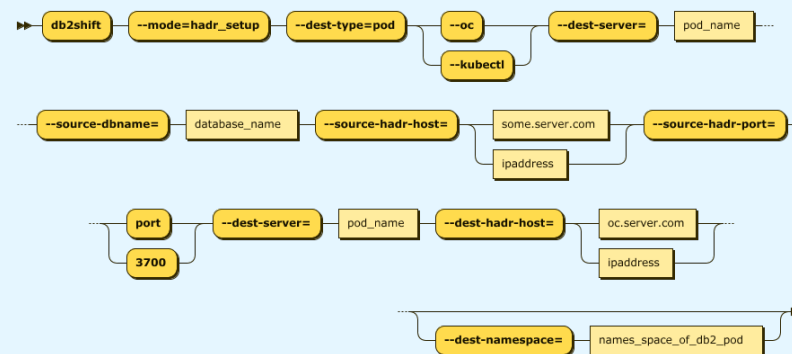
ESC+Quit ^?+Field Help ^D+Select Directory ^A+Analyze ^X+Review and Execute
```



Initialize HADR to a POD

- This Db2 Shift option will take a source and destination (POD) database and start the HADR service between them
 - The Db2u pod must have been created with the HADR option during the shift step
 - Note that a destination HADR port is not required since it is automatically generated for you by the Db2 Shift program
- The command requires the following information:
 - The source database name and server
 - The destination POD and server details

```
Db2 Shift [/home/c2cdb2/c2c/dist]
Initialize HADR Source and Destination Databases (POD)
Source Details
Database Name : |.....
Source HADR IP : .....
Source HADR Port : 3700
Destination Details
Target Location : ● OpenShift ○ Kubernetes
POD Project : .....
POD Name : .....
Target HADR IP : .....
```



Initialize HADR to an Instance

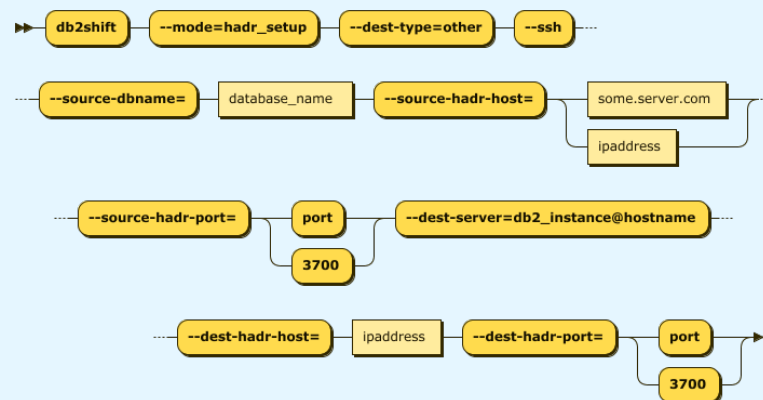
- This menu is similar to the previous one where the HADR service is setup between the source Db2 database and another Db2 instance
 - The Db2 database on the target system must have been created with the HADR option during the Shift operation
- The command requires the following information:
 - The source database name and server
 - The destination server details

```
Db2 Shift [/home/c2cdb2/c2c]
Initialize HADR Source and Destination Databases

Source Details
Database Name : lights
Source HADR IP : some.server.com
Source HADR Port : 3700

Destination Details
Instance Owner : db2inst1
Target HADR IP : other.server.com
Target HADR Port : 3700

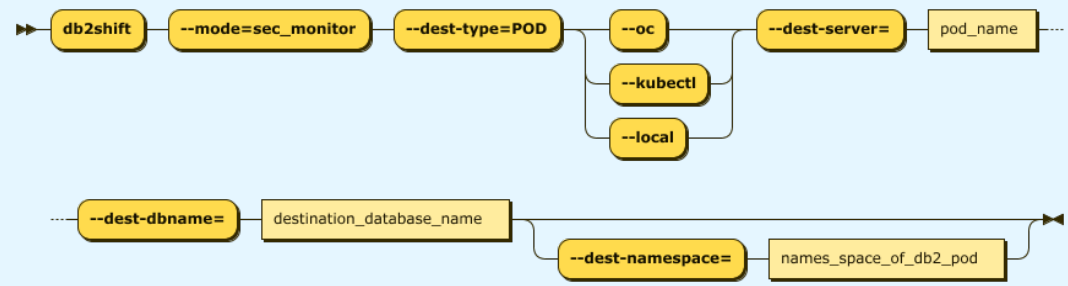
ESC+Quit ^?+Field Help ^X+Review and Execute
```



Initialize LDAP and DMC (CP4D Only)

- Db2 Shift will automatically add the appropriate userids to the LDAP service and reset the Data Management Console (DMC) when shifting to CP4D
 - If you choose to shift a database and set it up as an **HADR** secondary, the LDAP and DMC setup cannot be performed until **after** you have converted the secondary POD into the primary server
- You must supply the name of the database, POD name and project as well as the name of the instance owner to update the LDAP and DMC settings

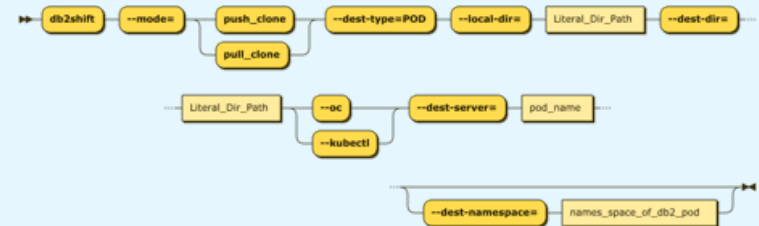
```
Db2 Shift [/home/c2cdb2/c2c]
Update LDAP and DMC Settings
Source Details
Instance Owner : db2inst1
POD Details
Target Location : ● OpenShift ○ Kubernetes
Database Name   : flights
POD Project     : db2u
POD Name        : c-demo-db2u-0
```



Copy a Clone to a POD

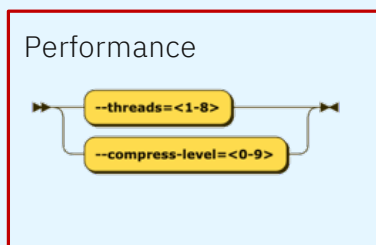
- This Db2 Shift command provides a feature that allows a user to copy an existing database clone copy to a POD, or to retrieve a database clone from a POD
- Once a database clone has been generated, the copy can be moved to any location and then deployed locally
 - This option provides a convenient way of copying the database using Db2 Shift without having to use OpenShift or Kubernetes commands
- The panel requires the following information:
 - Type of copy (From Source to Target or Target to Source)
 - Source cloned database directory
 - Target cloned database directory
 - The destination POD and server details

```
Db2 Shift [/home/c2cdB2/c2C]
Copy or Retrieve Database Clone
Copy Type
Copy Type :  Copy Clone to Target Pod  Copy Target Clone to Local
Source Details
Directory : /tmp/cache.....
POD Details
Directory : /tmp/cache.....
Target Location :  OpenShift  Kubernetes
POD Project : .....
POD Name : .....
```

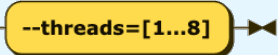


Db2 Shift Performance

- There are two settings which will impact the performance of a Shift operation:
 - Threads - the amount of parallelism that Db2 Shift will use
 - Compression - How much compression will be applied to the data stream during shifts
- The impact of these two parameters can be substantial
- Deciding what setting to use is based on network speed, CPU capacity and overall load on the source system



Threading

- Syntax: 

```
Thread count      : 4
```

- The copy phase of the Db2 Shift program can use multiple threads to transmit data to a destination
- This setting allows you to increase the parallelism up to 8 threads
- As you increase the number of threads, the amount of data being transmitted increases, at the expense of greater CPU usage and network congestion
- The default value is 4 which strikes a balance between overhead and network performance

Compression

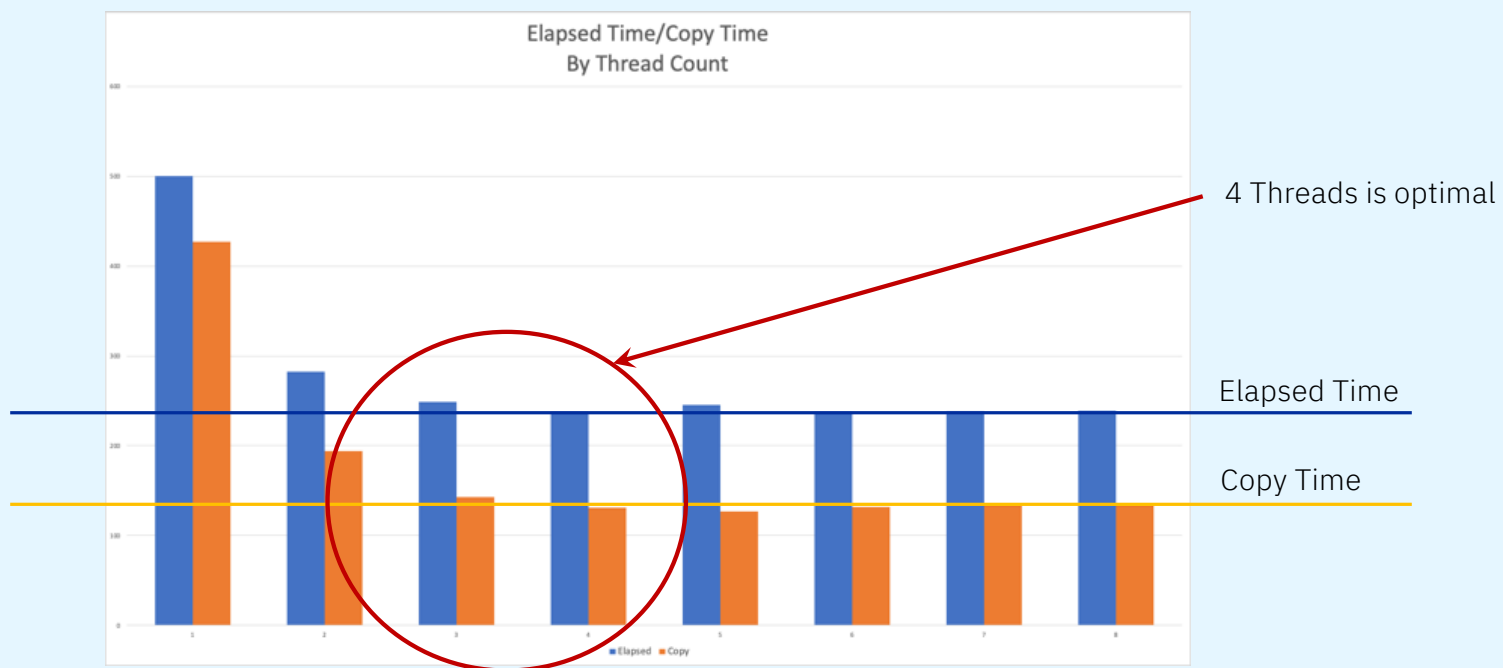
- Syntax: ►► `--compression=[1...8]` ◄◄

```
Compression      : 4
```

- RSYNC compresses the data during the transfer process to allow for faster movement of data
- The amount of compression can be adjusted from 0 to 9 with 0 turning off compression and values between 1 and 9 increasing the amount of compression applied to the data
 - A value of 4 has been found to be a good compromise between compression overhead and data size on slow networks (< 1Gb/s)
 - For high-speed networks, a value of 0 is recommended unless there is a requirement to reduce network traffic
- Higher compression values will result in more CPU usage and **may not significantly reduce** the size of the data stream

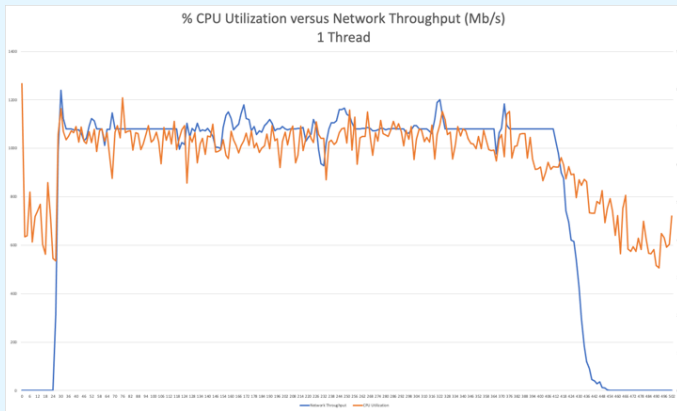
Elapsed Time

- 289M rows of transactional database with database size of 50Gb
- Best elapsed time 237 secs, with 131 sec for the data transfer
- Why 4 threads?

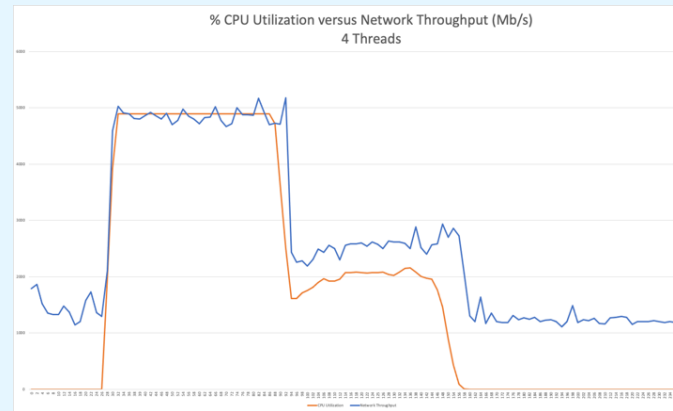


Combined CPU and Network

- CPU Usage is directly correlated to Network Throughput



1 Thread

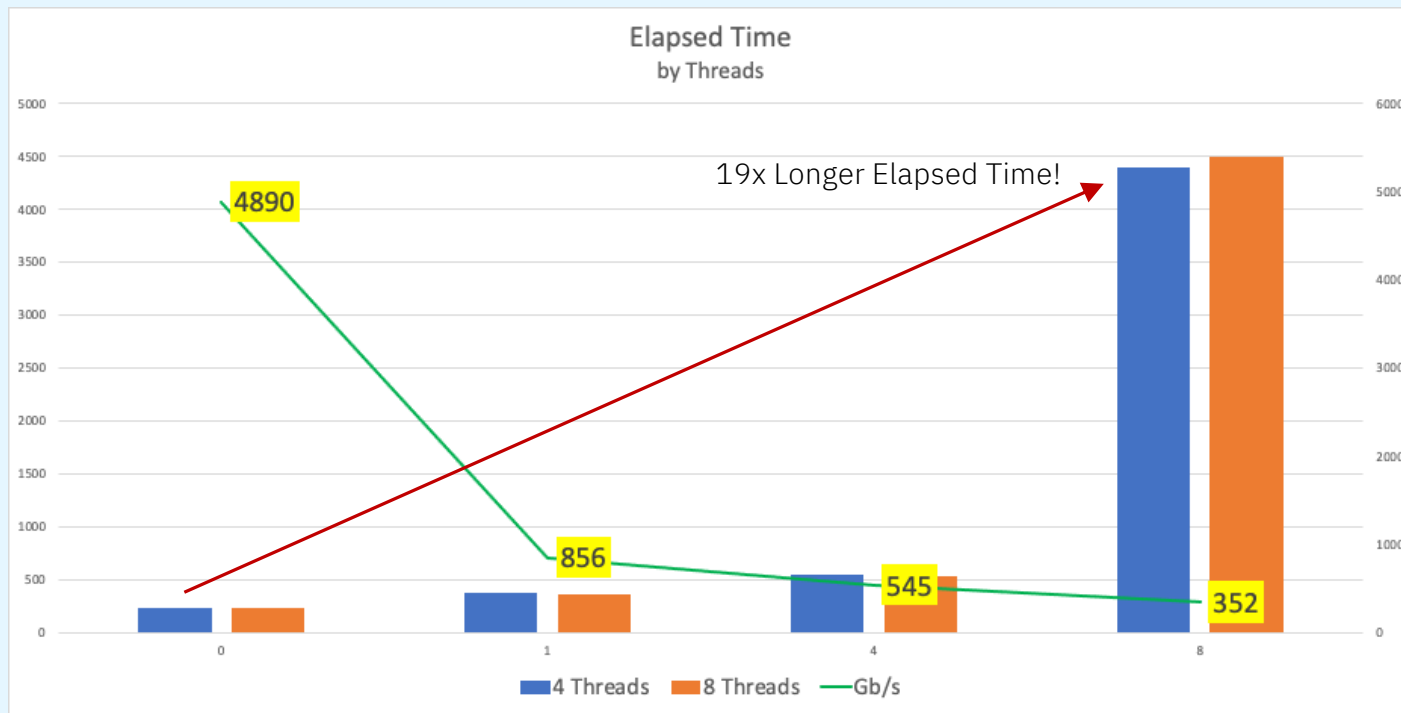


4 Threads

- Maximum network throughput
 - 1 thread was 1240 Mb/sec
 - 4 threads was 4890 Mb/sec
- Throughput was capped by processor speed and network limit of 5Gb/s

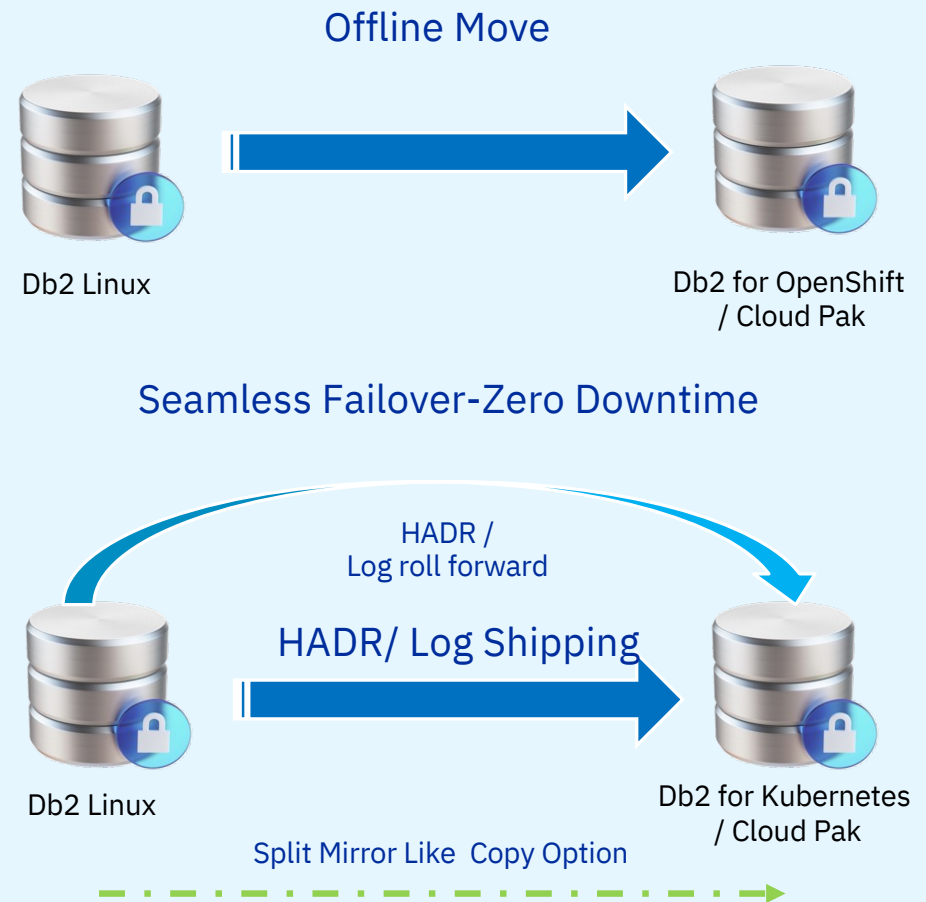
Compression

- The Compression setting has a negative impact on elapsed time
 - It will reduce the amount of network traffic but at the expense of elapsed time and CPU usage

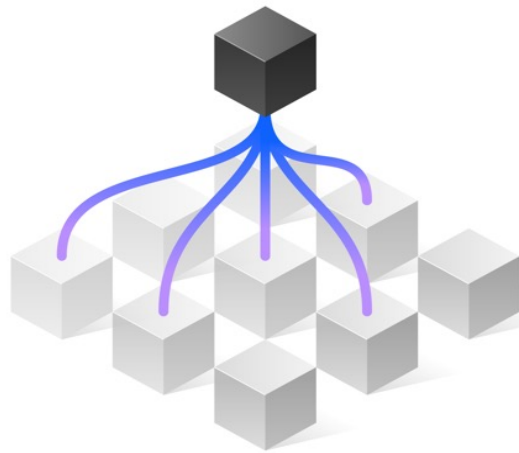


Summary

- Moves Databases to and from:
 - Normal Db2 Instance
 - Containerized Db2 Instance
- Support Multi-tenant Db2
- Supports offline Clones
- Inflight Version Upgrade
- Db2 Warehouse / OLTP
- Volumes = GBs to Terabytes
- Cloud or Local Deployments
 - IBM Cloud Pak For Data Deployments
 - Kubernetes
 - OpenShift
- Tunable Transportation



Resources



Db2 Click to Containerize

Visit ibm.biz/c2cdownload for more details on the Technical Preview of Db2 Click to Containerize

Documentation is available on ibm.biz/c2cebook (PDF) or ibm.biz/c2cguide (HTML)

Self-paced lab is available for business partners and IBMers at ibm.biz/c2cv2lab. Lab details are described in the following PDF ibm.biz/c2ck8slab

Single page fact sheet can be found at ibm.biz/c2cfactsheet

Thank you

© Copyright IBM Corporation 2022

IBM Corporation
New Orchard Road, Armonk, NY 10504

Produced in the United States of America
March 2021

IBM, the IBM logo, ibm.com, IBM Cloud, IBM Cloud Pak, Db2, InfoSphere, DataStage, Cognos and IBM Watson are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Red Hat®, and OpenShift® are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions. It is the user’s responsibility to evaluate and verify the operation of any other products or programs with IBM products and

programs. THE INFORMATION IN THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT. IBM products are warranted according to the terms and conditions of the agreements under which they are provided.

The client is responsible for ensuring compliance with laws and regulations applicable to it. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the client is in compliance with any law or regulation.

Statement of Good Security Practices: IT system security involves protecting systems and information through prevention, detection and response to improper access from within and outside your enterprise. Improper access can result in information being altered, destroyed, misappropriated or misused or can result in damage to or misuse of your systems, including for use in attacks on others. No IT system or product should be considered completely secure, and no single product, service or security measure can be completely effective in preventing improper use or access. IBM systems, products and services are designed to be part of a lawful, comprehensive security approach, which will necessarily involve additional operational procedures, and may require other systems, products or services to be most effective. IBM DOES NOT WARRANT THAT ANY SYSTEMS, PRODUCTS OR SERVICES ARE IMMUNE FROM, OR WILL MAKE YOUR ENTERPRISE IMMUNE FROM, THE MALICIOUS OR ILLEGAL CONDUCT OF ANY PARTY