IBM OpenPOWER for DB2 Warehouse Mini Deployment Guide

Version 1.0

09/12/17

TABLE OF CONTENTS

Introduction

This document, along with referenced links, describes a comprehensive set of build instructions, rules, and automation tools for building an IBM® OpenPOWER cluster for DB2 Warehouse Local.

This guide detail steps on how to build a DB2 Warehouse Local Mini config with 3 nodes and locally attached storage (SSDs) (See hardware BOM for detail)

Notes: if you plan to set up a larger node config, Please review the DB2 Warehouse sizing guide, and adjust GPFS FPO config accordingly. This guide does not cover this modifications. Contact IBM for further help.

Review "dashDB Local Software and Hardware prerequisites":
https://www.ibm.com/support/knowledgecenter/en/SS6NHC/com.ibm.swg.im.dashdb.doc/admin/local_prereqs.html#local_prereqs

Review "Spectrum Scale FPO Performance Tuning Guide"
https://www.ibm.com/developerworks/community/wikis/form/anonymous/api/wiki/fa32927c-e904-49cc-a4cc-870bcc8e307c/page/c90999f2-60cd-4f6f-8430-29efb5f01c3b/attachment/7fc9bf19-cabb-425b-b100-067cb3a661ad/media/GPFS%20Shared%20Nothing%20Cluster%20Performance%20Tuning%20Guide_V0.7-3.1.pdf

Step 1: Acquire the hardware

Review DB2 Warehouse - Mini Config Proposal.

https://github.ibm.com/anhdang/DB2-Warehouse-Local-Mini/blob/master/docs/DB2_Warehouse_Mini_Hardware_Proposal.pdf

Go to the following link to obtain the bill of materials with the required parts.

https://github.ibm.com/anhdang/DB2-Warehouse-Local-Mini/blob/master/docs/DB2_Warehouse_Mini_Specs_BOM.pdf

Contact an IBM representative for ordering and purchasing assistance.

https://www-01.ibm.com/marketing/iwm/dre/signup?source=MAIL-power&disableCookie=Yes

Step 2: Acquire the Software

The following are a list of Software Images that are needed for deploy dashDB Local using in this example.

- **dashDB Local image**

  You can obstain dashDB image via 2 methods:

  1. https://hub.docker.com/. To access the image in this location, obtain a Docker Hub ID,

and register for the dashDB Local trial license (https://www.ibm.com/us-en/marketplace/ibm-dashdb-local). After you complete the registration and submit your Docker Hub ID to IBM, you will be granted access to the dashDB Local private repository (ibmdashdb/local) and will be able to download the product image.

(https://www.ibm.com/support/knowledgecenter/en/SS6NHC/com.ibm.swg.im.dashdb.doc/admin/local_containers.html)

2. The Box file-sharing site. To access the image in this location, contact your IBM Support representative. You must also register for the dashDB Local trial license.

- **Docker Engine**

  You must install Docker Engine 1.12.6 or higher. For instructions, see Installing Docker Engine 1.12.6 for IBM dashDB Local.

- **IBM Spectrum Scale FPO** software and license (Spectrum Scale 4.2.1.2 or later)

  In a multi-node deployment, you must set up a POSIX-compliant cluster file system that is mounted on /mnt/clusterfs, with a minimum of 500 GB. Examples of these file systems are IBM Spectrum Scale (formerly known as IBM GPFS), GFS2, and VxFS.

  Confirm if the version of GPFS is supported/tested by with the latest OS Kernel version.

  [See GPFS readme notes]
  (https://www.ibm.com/support/knowledgecenter/SSFKCN/com.ibm.cluster.gpfs.doc/gpfs_faqs/gpfsclustersfaq.html#linux)

  The following link detail instruction on how to Obstain Spectrum Scale Software:

  https://www.ibm.com/us-en/marketplace/scale-out-file-and-object-storage

  IBM Spectrum Scale Manuals and Support Portals

  https://www.ibm.com/support/knowledgecenter/en/STXKQY/ibmspectrumscale_welcome.html

- **Ubuntu OS** 16.04 or higher for PPCLE (this will be download automatically during cluster genesis deployment) or customer provides iso image.
  (Please check availability of RHEL OS on PPCLE support for DB2 Warehouse)

- **Optional**: RStudio container, Jupyter containers

Notes:

Internet Access to the deployer node is required to download various github/OpenSrc packages nessasary for the cluster genesis process. (see Appendix)

Step 3: Choose the basic configuration parameters

To facilitate faster automated configuration of the overall solution, collect the parameters in *Table 1* before starting. This data is edited into a *config.yml* file, which is used to automatically configure and deploy the entire solution.

Table 1. Configuration parameters

<table>
<tr><th>Parameter</th><th>Description</th><th>Example</th></tr>
<tr><td>Domain name</td><td></td><td>Ibm.com</td></tr>
<tr><td>Upstream DNS servers</td><td>While a domain name system (DNS) server is configured within the cluster, upstream DNS servers must be defined because the names cannot otherwise be resolved.</td><td>*4.4.4.4, 8.8.8.8 as default public upstream DNS servers</td></tr>
<tr><td>Deployment node host name</td><td>The name of the deployment node.</td><td>depnode</td></tr>
<tr><td>Management network IP address</td><td>Management for the cluster takes place on its own internal network.</td><td>192.168.3.3.24</td></tr>
<tr><td>Data network IP address</td><td>Labeled interconnect in the config.yml file in the example below.</td><td>10.0.0.1/24</td></tr>
<tr><td>Management switch IP address</td><td>Labeled ipaddr-mgmt-switch in the config.yml file in the example below.</td><td>192.168.3.5</td></tr>
<tr><td>Data switch IP addresses</td><td>Labeled ipaddr-data-switch in the config.yml file in example below.</td><td>1.2.3.178</td></tr>
<tr><td>Default login data</td><td>Both IDs and passwords.</td><td>BMC network, OS Mgmt network</td></tr>
<tr><td>Data node hostnames and IPs addresses</td><td>Each node in the cluster needs a host name and an IP address for each of the management and data networks.</td><td>
<table>
<tr><th>Name</th><th>Management</th><th>Data IP</th></tr>
<tr><td>bluhelix-1</td><td>192.168.3.102</td><td>10.0.0.2</td></tr>
<tr><td>bluhelix-2</td><td>192.168.3.104</td><td>10.0.0.4</td></tr>
<tr><td>bluhelix-3</td><td>192.168.3.106</td><td>10.0.0.6</td></tr>
</table>
</td></tr>
</table>

Go to the following link to see more options in the *config.yml* file.

https://github.ibm.com/anhdang/dashDBLocal-Mini/blob/master/dashDB.node.config.yml

Step 4: Prepare the deployer node / Laptop

The deployer node is used to obtain the latest software and deployment tools from GitHub and populate the cluster. The deployer node can be established as a temporary

or permanent server. It can be set up as an IBM POWER8® LC or x86 server with the following minimum characteristics:

- Two cores and 32 GB RAM
- Three network-interface connections: 1 GbE Intelligent platform management interface (IPMI), 1 GbE (Mgmt), and 10 GbE (high-speed) .
- Ubunutu 16.04 LTS must be installed before beginning with deployment.

If you do not already have Ubuntu, it is available at the following sources:

- Power8-LC servers: https://www.ubuntu.com/download/server/power8
- x86 servers: https://www.ubuntu.com/download/server

Step 5: Node and Rack Build

Node Config

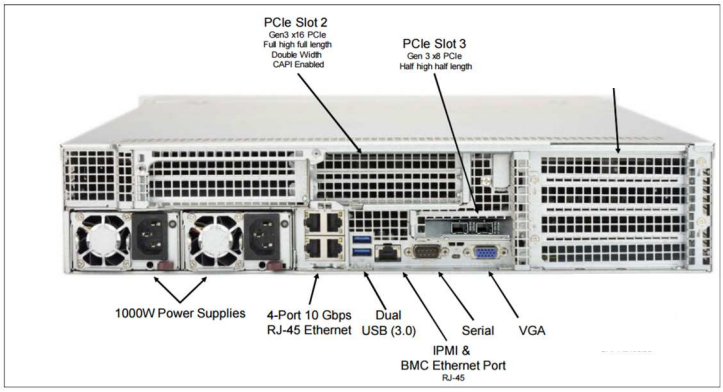
This document prescribes the IBM S822LC (MTM: 8001-22C) as dashDB node.

```
IBM S822LC Server Specification:
https://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/redp5407.html?Open
```

The following figures specify placements of hard drives and network adapter(s) in the server.

Power Systems 8001-22C (Machine Type-Model)

OS Boot Drives	Slot 1, 2
SSDs Drives	Slot 3,4,5,6,7,8,9,10



PCI Slot 1 (Build-In LOM)	4-Port 1/10G RJ45 Network LOM
PCI Slot 3	10G/40G Dual Port Network Adapter

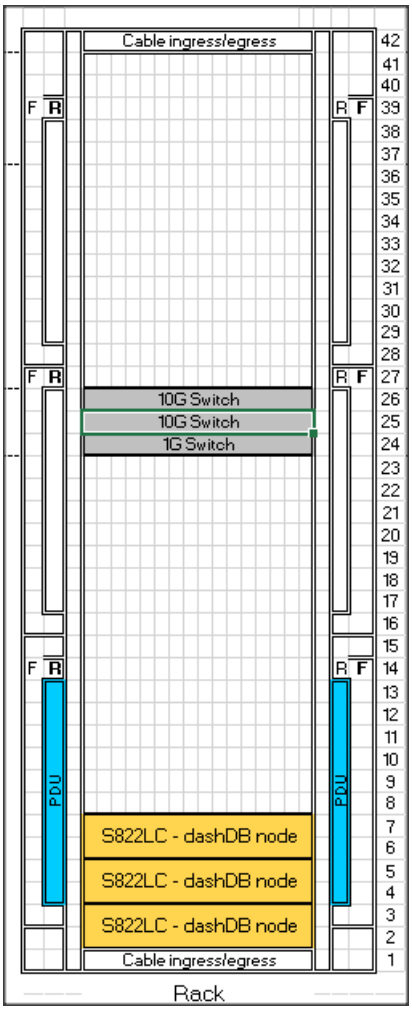
Note: While these servers are capable of sharing ports (multi-function ports), automation requires the port to be set up as a baseboard management controller (BMC) for data only.

Rack Config

User(s) should focus on the the following attributes when placing component in the rack

- Rack modularity (DB2 warehouse allow config to expand between 3-60 nodes)
- Consistency (rack to rack)
- Serviceablility, Cooling and Shipping

Figure below shows an example DB2 Ware house Mini Rack as example.



Networking

DB2 warehouse cluster network diagram.

Notes:

- The Port assignment for each node should match Automate Config File in **Step 3** in the node config sections

- Please ignore Switch #2 if you prefer Non-Redundant High Speed Network.

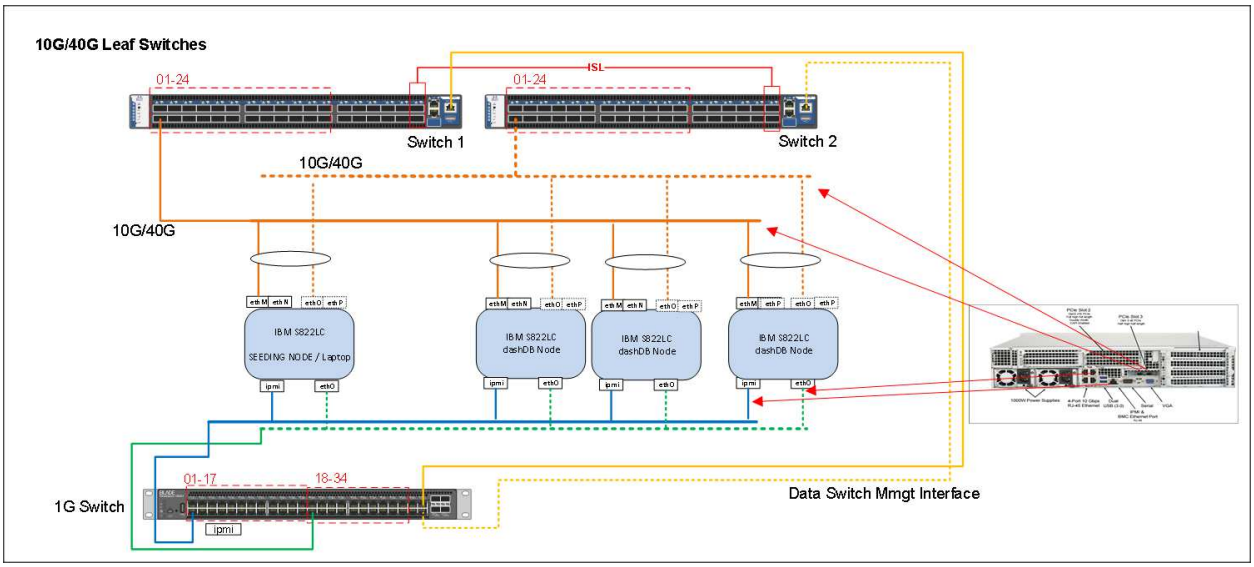


Figure 1. Network Diagram

Example P2P Port Labeling using the diagram above:

Notes:

- These Port number should match what is showed in the "dashDB.node.config.yml" used in **Step 5**
- If you Do not use redundant Highspeed network, you can ignore the connections to the second 10G switch (labeled 10G_TOR2)
- Seeding node network connections are similar to all other nodes,use any available ports on the network switch(es)

		10GbE	10GbE
		10G_TOR_1	10G_TOR_2
Server #	Name <opt>	P2P Data network Cable Label	P2P Data network Cable Label
1	hnode-1	1A/hnode-1/SVR1/slot 3/T1 <> 10G_TOR_1/Port1	1A/hnode-1/SVR1/slot 3/T2 <> H_TOR_2/Port1
2	dnode-1	1A/dnode-1/SVR2/slot 3/T1 <> 10G_TOR_1/Port2	1A/dnode-1/SVR2/slot 3/T2 <> H_TOR_2/Port2
3	dnode-2	1A/dnode-2/SVR3/slot 3/T1 <> 10G_TOR_1/Port3	1A/dnode-2/SVR3/slot 3/T2 <> H_TOR_2/Port3

		1GbE	1G_IPMI
		1G_TOR_1	1G_TOR_1
Server #	Name <opt>	P2P Mgmt RJ4-5 Cable Label	P2P IPMI RJ-45 Cable Label
1	hnode-1	1A/hnode-1/SVR1/LOM/T1 <> 1G_TOR_1/Port1	1A/hnode-1/SVR1/LOM/impi <> 1G_TOR_1/Port19
2	dnode-1	1A/dnode-1/SVR2/LOM/T1 <> 1G_TOR_1/Port2	1A/dnode-1/SVR2/LOM/impi <> 1G_TOR_1/Port20
3	dnode-2	1A/dnode-2/SVR3/LOM/T1 <> 1G_TOR_1/Port3	1A/dnode-2/SVR3/LOM/impi <> 1G_TOR_1/Port21

Step 6: Configure the cluster using the Cluster Genesis tool

This step covers the power on, initialization, configuration, and installation of a cluster solution. This deployment kit provides an automated method to quickly and more predictably go from assembly to a tuned operational state of the cluster's infrastructure. This is referred to as *hardware genesis*.

Genesis occurs once at the beginning of the cluster solution lifecycle. The open-sourced automation scripts are available and can be reused for maintenance and cluster expansion.

The Cluster Genesis tool automatically initializes and configures the hardware by accomplishing the following tasks:

- Reading the *config.yml* files with edited environment-specific changes
- Driving the BMCs to populate the IP addresses to the nodes
- Detecting and populating relevant configuration data to the deployer node
- Deploying the required operating system images to the server nodes
- Configuring the network switches
- Configuring all server management and data nodes (network interfaces, GPU drivers, and so on)

When the Cluster Genesis tool completes its process, control of the cluster is transferred to the operations manager.

All Genesis Cluster tool procedures are built into automation described in *Perform the deployment of* **CLUSTER GENESIS** on page 14. Go to the following link for more information about this process in the Genesis deployment README file.

https://github.ibm.com/open-power-solution-genesis/accelerated-db/blob/initial_dev/README.md

Go to the following link for more information about the procedure overview and deployment automation procedures for DB2 Warehouse Mini Config.

https://github.ibm.com/anhdang/dashDBLocal-Mini/blob/master/README.md

Obtain the default configuration file

The Genesis automation uses a configuration file to specify the target cluster configuration. The deployment tooling uses this YAML text file to specify the IP address locations of the managed switches and the system nodes attached to the switches as well as other useful details for deployment process:

- Review/Update the following Sections: "*reference-architecture*", "*networks*", "*node-templates*".

https://github.ibm.com/anhdang/dashDBLocal-Mini/blob/master/dashDB.node.config.yml

or use the alternate config yml file if the genesis tool does not have admin access to the network switches:

https://github.ibm.com/anhdang/dashDBLocal-Mini/blob/master/dashDB.node.switch-passive.config.yml

Customize the configuration file for the environment

The *config.yml* file contains a lot of configuration information. To enable a cluster tailored to specific environment, edit the .yml file with the configuration parameters that were collected in *Step 3*: CHOOSE THE BASIC CONFIGURATION PARAMETERS, replacing the Red text with your data. The following excerpt focuses on the lines to edit.

```
reference-architecture:
# dashDB Node roles
    db2_nodes_layout: "scripts/db2_nodes.cfg"   ← DB2 Node configured role
# Select db2Local image location: "db2_local_image" or  "db2_online_image"
    db2_image_selection: db2_local_image    ← DB2 docker image location, choose one
    db2_local_files: "packages/db2local/*.tar"
    db2_online_dockerhub: {name: "dockerID" , password: "docker_password"}

# Cluster File System / Disk layout  ← update section for Spectrum disk layout if different than recommended BOM.
    gpfs_install: "packages/gpfs/*.deb"   image    ← GPFS install image
    gpfs_nodes_layout: "scripts/gpfs-fpo-node.cfg"   ← GPFS Node layout
    gpfs_disks_layout: "scripts/gpfs-fpo-pool_ConfigA.cfg"  ← GPFS FPO drive layout
```

```
      gpfs_clusterfs_mnt: "/mnt/clusterfs"  ← GPFS mount point

# DockerVG , recommended for High Performance Docker deployment
# dockerpool can reside on saparate LVM (not yet support in this deployment script)
      dockerpool_disk: "/dev/sdb"   ← Docker Thinpool volume



~ ~ ~ ~ ~ ~ ~ licensing comment and YAML ~ ~ ~ ~ ~
ipaddr-mgmt-network: 192.168.3.0/24
ipaddr-mgmt-switch:
   rack1: 192.168.3.5              ← Type your management switch IP address here.
ipaddr-data-switch:
   rack1: 1.2.3.178                ← Type your data switch IP address here.
 ~ ~ ~ ~ ~ ~ ~ YAML and comments ~ ~ ~ ~ ~
networks:
   external:
      description: Organization site or external network
      addr: 1.2.3.4/24             ← Type your subnet address here.
      broadcast: 1.2.3.255           ← Type your broadcast IP here.
      gateway: 1.2.3.1             ← Type your gateway IP here.
      dns-nameservers: 1.2.3.4     ← Type your nameserver IP here.
      dns-search: aus.stglabs.ibm.com
      method: static
      eth-port: eth10
   interconnect:
      description: Private 10G Data Network to Interconnect Cluster
      addr: 10.0.0.0/24
      broadcast: 10.0.0.255
      method: static
      eth-port: eth11
 ~ ~ ~ ~ ~ ~ ~ bunch of YAML and comments ~ ~ ~ ~ ~
node-templates:
   controller1:
      hostname: bluhelix             ← Type your hostname here.
      userid-ipmi: ADMIN           ← Type your userid here.
      password-ipmi: admin         ← Type your password here.
      cobbler-profile: ubuntu-16.04.1-server-ppc64el
 ~ ~ ~ ~ ~ ~ ~ bunch of YAML and comments ~ ~ ~ ~ ~
```

Editable portions of the Config.yml file

The inventory file

The inventory file is a YAML text file that contains the entire inventory of the cluster, captured during the genesis process. It can be used to feed subsequent automation (management, deployment, and so on). Do not edit this file manually.

Go to the following link for the generic master copy of the latest inventory file.

https://github.com/open-power-ref-design/cluster-genesis/blob/master/master_inventory.yml

The file contains the configuration specifics of each network switch and server node. The *Switches* data structure indicates the types of switches (management, spine, or leaf), their IP addresses, and associated log in credentials. The following sample inventory data structure contains the management and leaf switches attributes.

```
switches:
  mgmt:
  - hostname: mgmtswitch1
    ipv4-addr: 192.168.3.5
    rack-id: rack1
    userid: admin
    password: mspassword
  leaf:
  - hostname: leafswitch1
    ipv4-addr: 192.168.3.6
    rack-id: rack1
    userid: joeleaf
    password: joeleafpassword
```

The *Server Nodes* data structure specifies the type of node controller, its network properties, and its system architecture (ppc64 or x86). The following snippet shows the data structure.

```
  Controller1:
  - hostname: bluhelix-1
    userid-ipmi: ADMIN
    password-ipmi: admin
    port-ipmi: 15
    port-pxe: 16
    port-eth10: 21
    port-eth11: 22
    mac-ipmi: 70:e2:84:14:0a:10
```

```
ipv4-ipmi: 192.168.3.107
rack-id: rack1
template: controller2
architecture: ppc64
chassis-part-number: 8001-22C
chassis-serial-number: 1004C9A
mac-pxe: 70:e2:84:14:0a:12
ipv4-pxe: 192.168.3.108
external-addr: 9.3.3.5
interconnect-addr: 10.0.0.4
```

When Cluster Genesis completes, the *inventory.yml* file is stored on the deployment node in the path */var/oprc.*

Perform the deployment of Cluster Genesis

To deploy the Cluster Genesis tool, run the installation script:

```
$ ./install.sh
```

The installation script checks out the Cluster Genesis from its own GitHub repository. It applies patches and downloads the various dependent packages required for the installation.

After the *install.sh* is run cleanly, start the automated deployment:

```
$ ./deploy.sh myconfig.yml
```

- Deploy.sh will execute the following tasks:
  - "Cluster Genesis" performs: OS install and Config basic Network connectivity to all servers.
  - "DB2 Playbook" ./playbooks/db2_playbook.yml performs DB2 Warehouse application deployment includes the following tasks:

- "./playbooks/tasks/1_create_drive_partition.yml"
- "./playbooks/tasks/2_install_gpfs.yml"
- "./playbooks/tasks/3_config_gpfs.yml"
- "./playbooks/tasks/4_install_docker.yml"
- "./playbooks/tasks/5_install_dashDB.yml"

References

The following links and documents provide more information related to this document:

- IBM Power System® S822LC (8001-22C) reference material located in the IBM Knowledge Center
- dashDB Local Cluster Sizer reference in IBM Knowledge Center

https://www.ibm.com/support/knowledgecenter/en/SS6NHC/com.ibm.swg.im.dashdb.doc/admin/scalingout.html

- IBM open-power-ref-design-toolkit
  `https://github.com/open-power-ref-design-toolkit/`

Printed in the United States of America May 2017

information contained in this document was obtained in specific environments, and is presented as an illustration. The results obtained in other operating environments may vary.

IBM Systems
294 Route 100, Building SOM4
Somers, NY 10589-3216

The IBM home page can be found at **ibm.com®**.

Version 1.1
25 May 2017