How to Deploy an IBM OpenPOWER Cluster for Accelerated Databases

Version 1.1

Introduction

This document, along with referenced links, describes a comprehensive set of instructions, rules, and automation tools for building an IBM® OpenPOWER-based cluster that is tuned for accelerated databases. Kinetica's GPU database is an example of an accelerated database.

High-level deployment steps

Note: Each step is described in more detail in the sections that follow.

1. Acquire the hardware.
2. Choose the configuration parameter for the solution.
3. Prepare the deployer node.
4. Rack and cable the hardware.
5. Configure the cluster using the Cluster Genesis tool.
6. Complete the post Cluster Genesis configurations.
7. Operations manager.
8. Install the applicable accelerated database software. (Not automated in this deployment kit.)

Step 1: Acquire the hardware

Go to the following link to view the *Accelerated Database Design Proposal*, which shows the required hardware.

https://github.com/open-power-ref-design/accelerated-db/blob/master/docs/Accelerated%20Database%20Deployment%20Design%20Proposal.pdf

Go to the following link to obtain the bill of materials, which lists the required parts.

https://github.com/open-power-ref-design/accelerated-db/blob/master/docs/Accelerated%20Database%20Deployment%20BOM.pdf

If you do not already have the needed parts, go to the following link to 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: 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

Parameter	Description	Example
Domain name		Ibm.com
Upstream DNS servers	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.	*4.4.4.4, 8.8.8.8 as default public upstream DNS servers
Deployment node host name	The name of the deployment node.	depnode
Management network IP address	Management for the cluster takes place on its own internal network.	192.168.3.3.24
Data network IP address	Labeled *interconnect* in the config.yml file in the example below.	10.0.0.1/24
Management switch IP address	Labeled *ipaddr-mgmt-switch* in the config.yml file in the example below.	192.168.3.5
Data switch IP addresses	Labeled *ipaddr-data-switch* in the config.yml file in example below.	1.2.3.178
Default login data	Both IDs and passwords.	BMC network, OS Mgmt network

<table>
<tr>
<td><b>Data node hostnames and IPs addresses</b></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 IP</th><th>Data IP</th></tr>
<tr><td>Min-1</td><td>192.168.3.102</td><td>10.0.0.2</td></tr>
<tr><td>Min-2</td><td>192.168.3.104</td><td>10.0.0.4</td></tr>
<tr><td>Min-3</td><td>192.168.3.106</td><td>10.0.0.6</td></tr>
<tr><td>Min-4</td><td>192.168.3.108</td><td>10.0.0.8</td></tr>
</table>
</td>
</tr>
</table>

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

https://github.com/open-power-ref-design/accelerated-db/blob/master/accel-db.4compute.config.yml

Step 3: Prepare the deployer node

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 4: Rack and cable the hardware

GPU-accelerated databases are best optimized with the unique high-speed NVIDIA® NVLink™ bus between its CPU and GPUs. This design prescribes the IBM S822LC for the HPC server (product ID: 8335-GTB) to enable the best possible database performance.

Go to the following link for more information about the specific configuration.

```
http://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/redp54
05.html
```

This step describes how to cable and rack the servers in the rack so they are networked together correctly. It is not intended to be comprehensive. For example, it is assumed you can cable the servers to the power source as needed.

The OpenPOWER server's network adapters must be configured as shown in following figures for a system named MIN for this document.

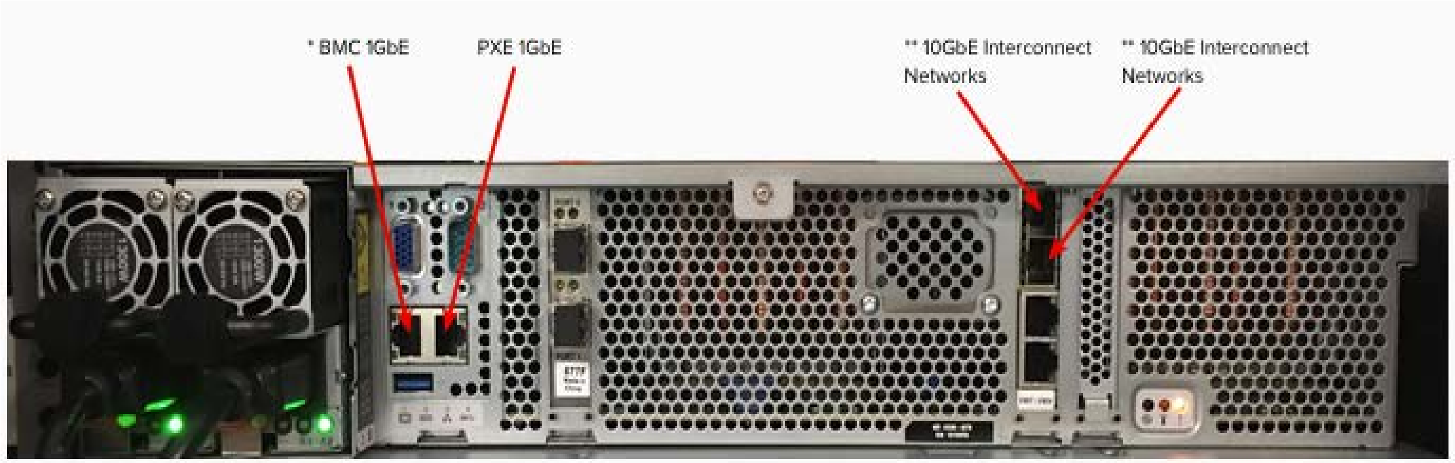

* BMC 1GbE
PXE 1GbE
** 10GbE Interconnect Networks
** 10GbE Interconnect Networks

Figure 1. Back view of server MIN

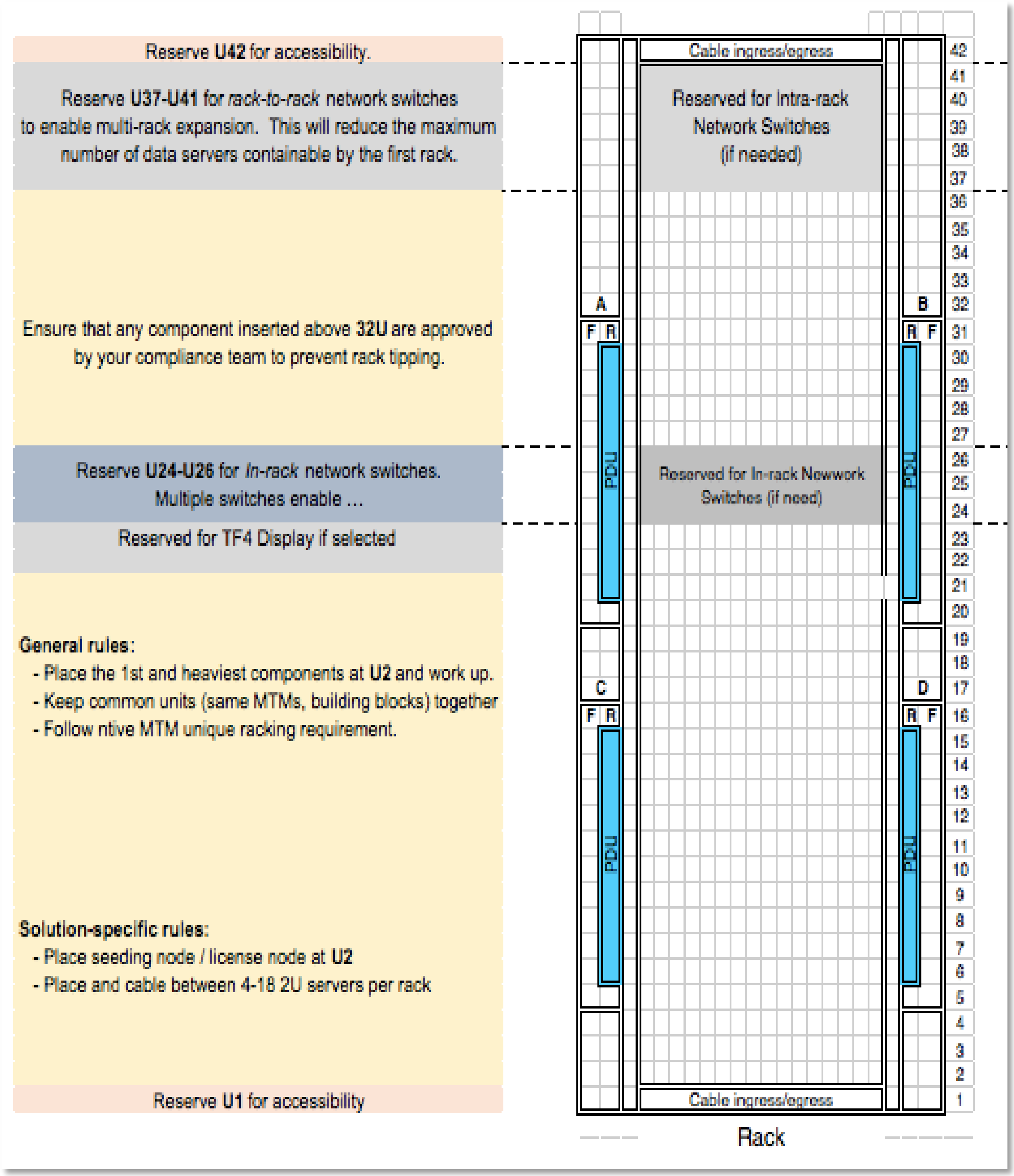
Note: While these servers are capable of sharing ports (multi-function ports), automation requires the port to be set up as a baseboard management controller (BMC) for data only.

Racking the components

The racking rules specify where to place the servers and switches and where to connect the cables.

The suggested racking rules, shown in *Figure 2* on page 5, focus on enabling:

- Rack modularity
- Consistency
- Expandability
- Ease of servicing, repurposing, shipping, and cooling



Reserve **U42** for accessibility.

Reserve **U37-U41** for *rack-to-rack* network switches to enable multi-rack expansion. This will reduce the maximum number of data servers containable by the first rack.

Ensure that any component inserted above **32U** are approved by your compliance team to prevent rack tipping.

Reserve **U24-U26** for *In-rack* network switches. Multiple switches enable …

Reserved for TF4 Display if selected

General rules:

- Place the 1st and heaviest components at **U2** and work up.
- Keep common units (same MTMs, building blocks) together
- Follow ntive MTM unique racking requirement.

Solution-specific rules:

- Place seeding node / license node at **U2**
- Place and cable between 4-18 2U servers per rack

Reserve **U1** for accessibility

Figure 2. Suggested Racking Rules

Additional racking rules

The following additional racking rules are for version 1.1 of the accelerated DB design, which supports in-server storage only. A future version will add support for external storage.

Place the intra-rack (leaf) network switches in slots U24 - U26 as follows:

- Place the 10G/40G/IB data plane switch in slot 26U (parts 8828-E36, 8831-NF2, or 7120-64C)
- Leave slot U25 open. Reserve this for later use of short-depth devices. For more information see the *Bill of Materials* document:

  https://github.com/open-power-ref-design/accelerated-db/blob/master/docs/Accelerated%20Database%20Deployment%20BOM.pdf

- Place the 1G management plane switch in slot 24U.

The Figure 3 on page 7 shows an example of these additional racking rules.

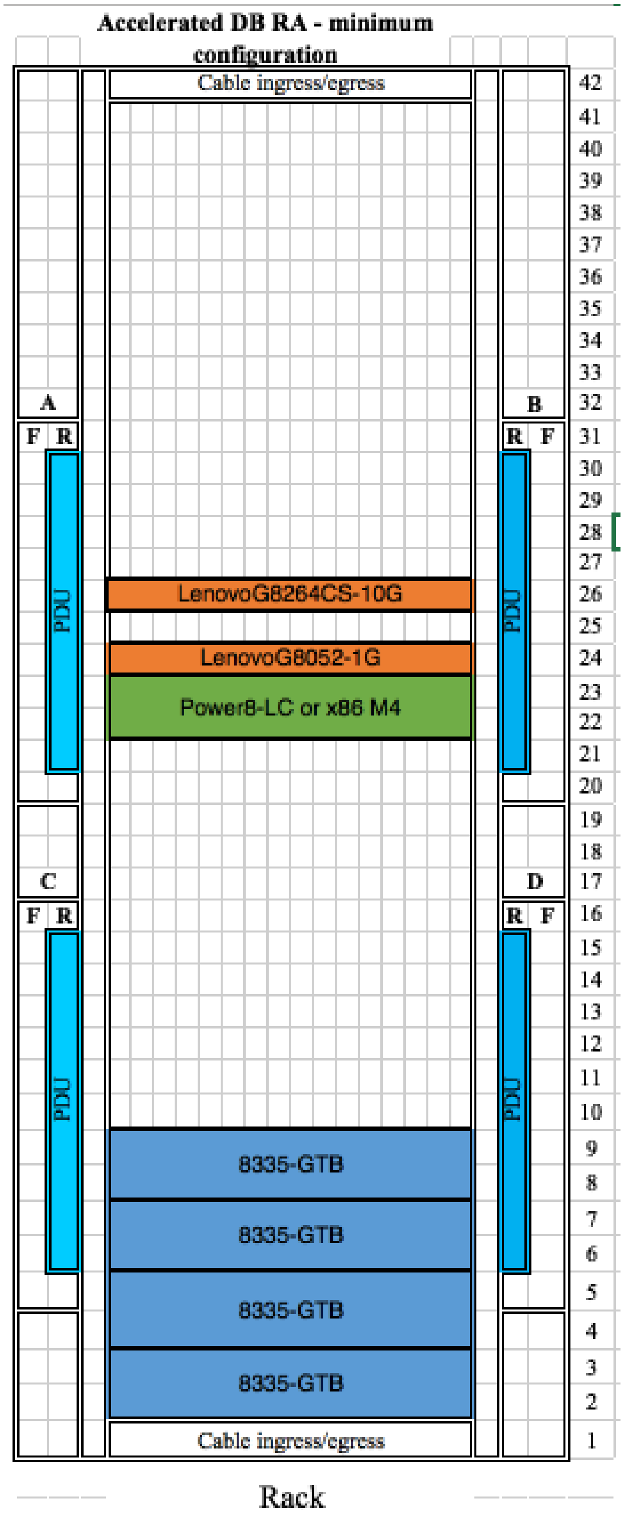


Figure 3. Minimum four 8335-GTB system cluster

Cable the components together

To follow the cabling rules, cable the like labels on the servers to the applicable network switches. *Figure 4* shows an example approach for a four-server cluster configuration. The rear server view is shown with labels for four servers named MIN1, MIN2, MIN3, and MIN4.

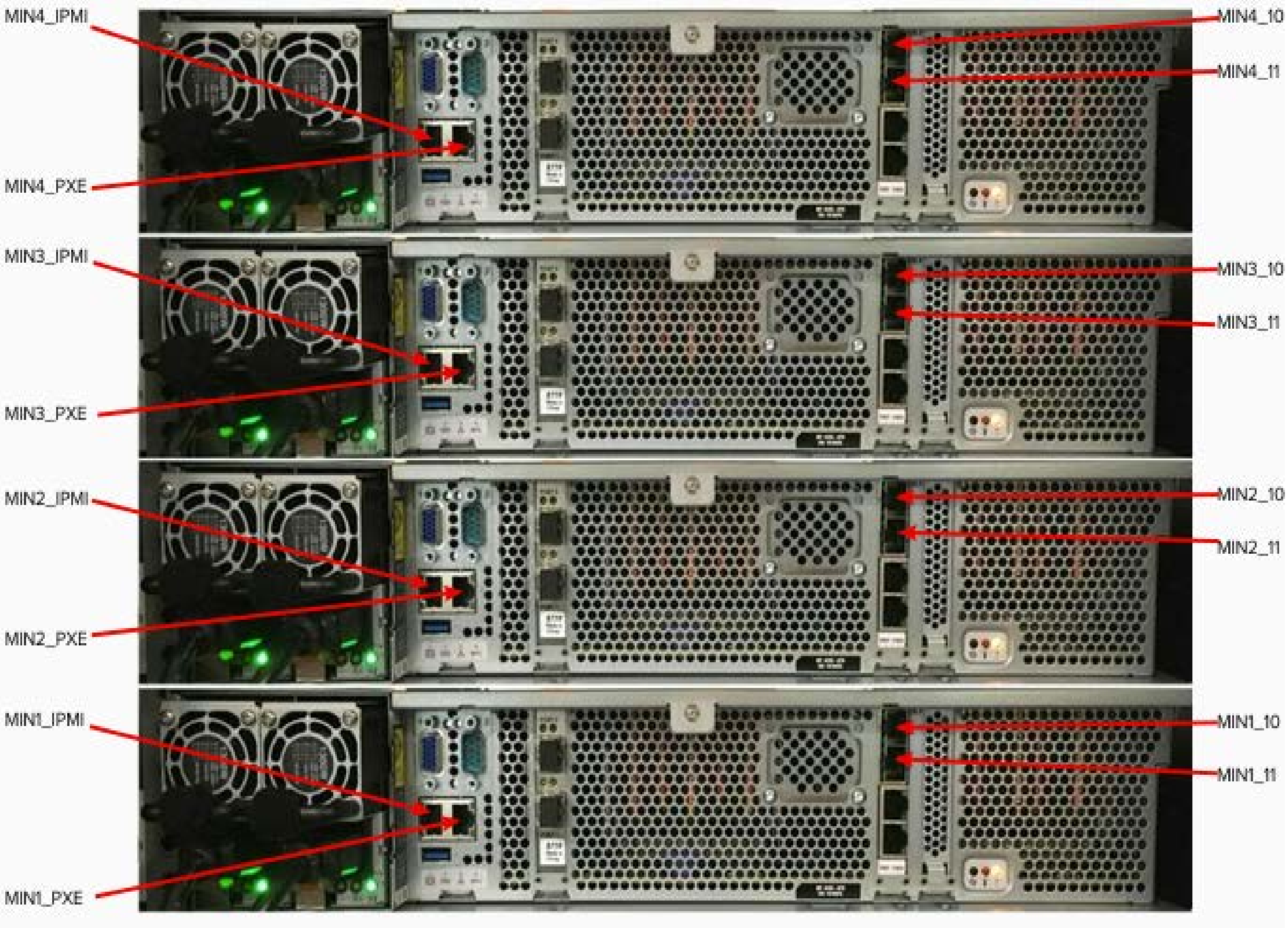


Figure 4. Example cabling for a minimum four 8335-GTB system cluster network

Figure 5 and *Figure 6* show a network-switch port view. The labels on the ports correspond to the server labels above and also to the information in the *config.yml* and *inventory.yml* files. In the first example, 40GbE-to-4x10GbE fanout cables are used.

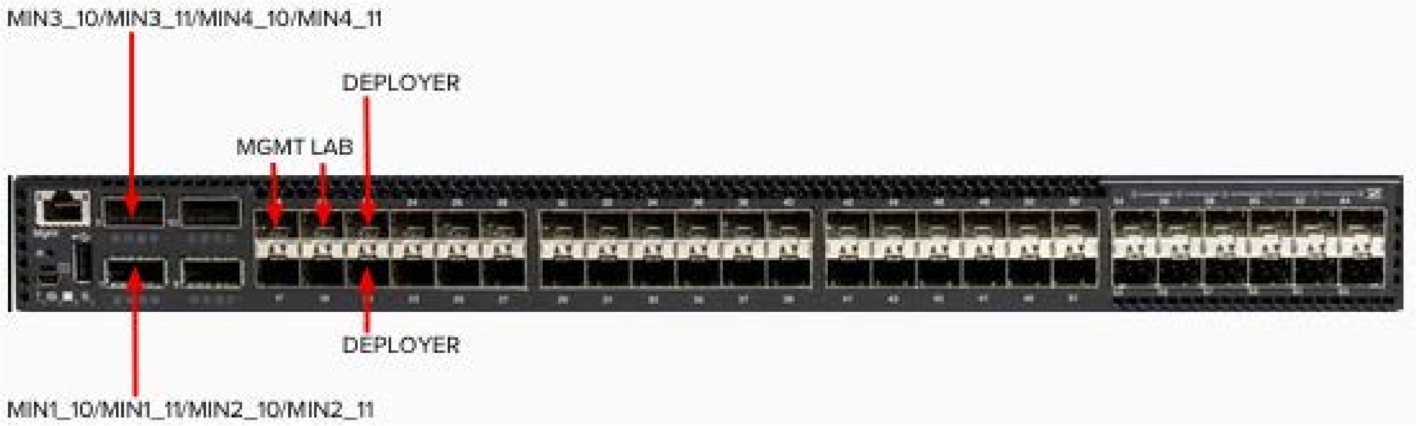

MIN3_10/MIN3_11/MIN4_10/MIN4_11
DEPLOYER
MGMT LAB
DEPLOYER
MIN1_10/MIN1_11/MIN2_10/MIN2_11

Figure 5. Lenovo G8264CS – 10 GbE data network switch cabling scheme

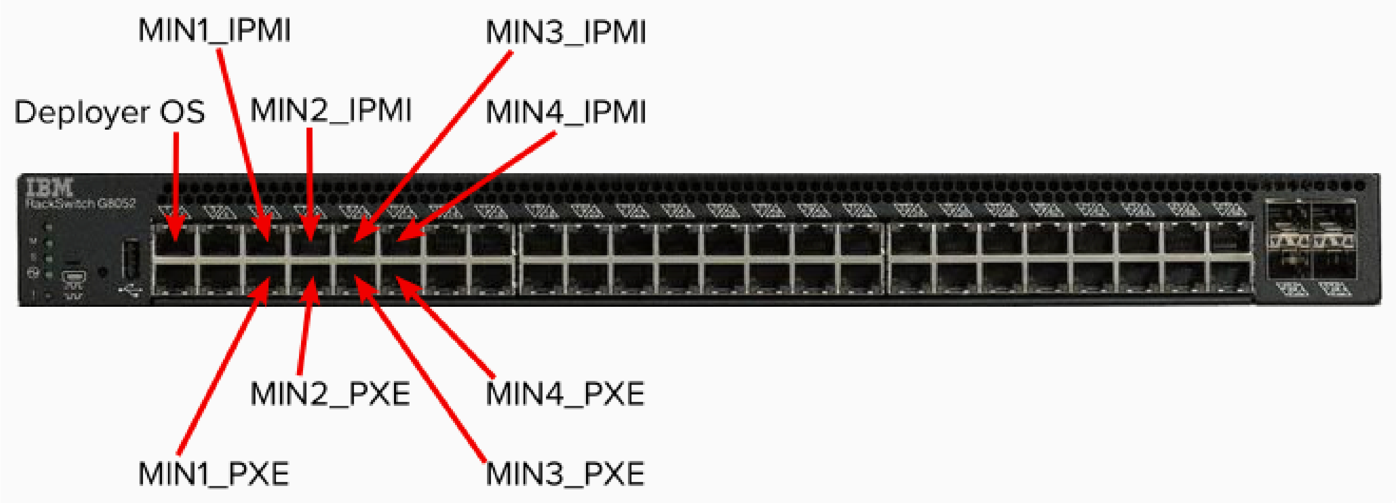


Figure 6. Lenovo G8052 – 1 GbE management network switch cabling scheme

Step 5: 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.com/open-power-ref-design-toolkit/cluster-genesis/blob/master/README.rst

Go to the following link for more information about the procedure overview and deployment automation procedures found in the Accelerated Database README file.

https://github.com/open-power-ref-design/accelerated-db/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.

Go to the following link for a copy of the OpenPOWER Accelerated Database Small Cluster configuration file.

https://github.com/open-power-ref-design/accelerated-db/blob/master/accel-db.4compute.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 2*: CHOOSE THE BASIC CONFIGURATION PARAMETERS, replacing the red text with your data. The following excerpt focuses on the lines to edit.

```
~ ~ ~ ~ ~ ~ ~  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: min                ← 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/accelerated-db/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 inven-tory 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: min-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: 8335-GTB
    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
reference-architecture:
  gpudb_nvidia_playbook:
    description: playbook for installing nvidia for gpudb
    cuda_deb: /tmp/cuda-repo-ubuntu1604-8-0-local_8.0.35-1_ppc64el.deb
    driver_level: nvidia-361
    dkms_deb: /tmp/dkms_2.2.0.3-2ubuntu14_all.deb
```

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. These dependencies include NVIDIA CUDA, a few specific Ubuntu packages that are required during the automated deployment, and the operations manager.

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

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

Step 6: Complete the post-Cluster genesis configuration

For this solution, the following post-genesis tasks must be completed. These steps have been automated in the *deploy.sh* script referenced in the Accelerated Database Read-me file and in *Perform the deployment of* **CLUSTER GENESIS** *on page 14*.

Deploy.sh performs three post-genesis configuration tasks:

- Completes the networking configuration of each node to have a 10 Gb network interface for cluster interconnect and external connection.
- Installs the NVIDIA driver. Only the NVIDIA driver is installed, but the full CUDA is available locally on the deployment node in the *~/accelerated-db/playbooks/packages* directory.
- Installs the Operations Manager – Clones OpsMgr from GitHub, updates the Ansible inventory template, and deploys all OpsMgr components in the management node (deploys and configures all the auxiliary monitoring/collection services).

Step 7: Operations manager

The Operations Manager is a packaged collection of open-source management tools that is configured to manage this infrastructure (for example, health monitoring, logging and data collection and analysis, and performance metrics). These components are:

- Nagios Core
- Elasticsearch, Logstash, Kibana (ELK)

Operations manager: Access and use operations and applications

After the operations manager (OpsMgr) is installed, end users can access the Ops Portal by entering their deployer node IP into a web browser:

```
Error! Hyperlink reference not valid.>
```

Change the administrator superuser password

A default user ID and password for the administrator superuser is generated at deployment time. It can be found by executing the following command as root user in the deployment node:

```
grep "keystone_auth_admin_password"
/etc/openstack_deploy/user_secrets.yml
```

ELK and Nagios have the following default user names and passwords:

- Nagios
    - User: nagios
    - Password: nagios
- Kibana
    - User: kibana
    - Password: kibana

You are strongly encouraged to change passwords according to the instructions available at the following link:

```
https://github.com/open-power-ref-design-toolkit/opsmgr/blob/master/recipes/standalone/README.rst
```

Step 8: Install the Application Software

Because today's GPU-accelerated database tools are proprietary, this document does not describe how to automatically deploy them. Instead, pointers to the relevant sites are included. While Kinetica is the only database supported today, more will be added.

Kinetica

Kinetica, Inc. is the commercial provider of GPUDB. Go to the Kinetica web site or contact Kinetica for assistance in deploying their database onto this cluster. The following links might be useful:

- Kinetica's web site
- Kinetica's overview and architecture
- Kinetica's installation instructions
- Instructions on running Kinetica after it is installed.

References

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

- IBM Power System S822LC for High Performance Computing Introduction and Technical Overview
- More IBM Power System® S822LC (8335-GTB) reference material located in the IBM Knowledge Center

© Copyright International Business Machines Corporation 2017

Printed in the United States of America May 2017

IBM, the IBM logo, and ibm.com are trademarks or registered 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.

NVLink is a trademark of the NVIDIA Corporation in the United States, other countries, or both.

The OpenPOWER word mark and the OpenPOWER Logo mark, and related marks, are trademarks and service marks licensed by OpenPOWER.

Other company, product, and service names may be trademarks or service marks of others.

All information contained in this document is subject to change without notice. The products described in this document are NOT intended for use in applications such as implantation, life support, or other hazardous uses where malfunction could result in death, bodily injury, or catastrophic property damage. The information contained in this document does not affect or change IBM product specifications or warranties. Nothing in this document shall operate as an express or implied indemnity under the intellectual property rights of IBM or third parties. All 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.

This document is intended for the development of technology products compatible with Power Architecture®. You may use this document, for any purpose (commercial or personal) and make modifications and distribute; however, modifications to this document may violate Power Architecture and should be carefully considered. Any distribution of this document or its derivative works shall include this Notice page including but not limited to the IBM warranty disclaimer and IBM liability limitation. No other licenses (including patent licenses), expressed or implied, by estoppel or otherwise, to any intellectual property rights are granted by this document.

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED ON AN “AS IS” BASIS. IBM makes no representations or warranties, either express or implied, including but not limited to, warranties of merchantability, fitness for a particular purpose, or non-infringement, or that any practice or implementation of the IBM documentation will not infringe any third party patents, copyrights, trade secrets, or other rights. In no event will IBM be liable for damages arising directly or indirectly from any use of the information contained in this document.

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