NVIDIA Base Command Manager 11

NVIDIA Mission Control Manual

Revision: 2505667c1

Date: Wed Jun 18 2025

Table Of Contents

Preface

Welcome to the *NVIDIA Mission Control Manual* for NVIDIA Base Command Manager 11.

0.1 About This Manual

The NVIDIA Mission Control features of NVIDIA Base Command Manager (BCM) are available on clusters that are running the NVIDIA Mission Control edition of BCM. The features are listed in chapter 1 of this manual.

This manual is aimed at helping cluster administrators install, understand, configure, and manage the NVIDIA Mission Control features and capabilities of NVIDIA Base Command Manager. The administrator is expected to be reasonably familiar with the BCM *Administrator Manual*. The *NVIDIA Mission Control Manual* is thus a supplementary manual to the BCM *Administrator Manual* for the NVIDIA Mission Control features.

0.2 About The Manuals In General

Regularly updated versions of the manuals are available on updated clusters by default at `/cm/shared/docs/cm`. The latest updates are always online at `https://docs.nvidia.com/base-command-manager`.

- The *Installation Manual* describes installation procedures for the basic cluster.
- The *Administrator Manual* describes the general management of the cluster.
- The *User Manual* describes the user environment and how to submit jobs for the end user.
- The *Cloudbursting Manual* describes how to deploy the cloud capabilities of the cluster.
- The *Developer Manual* has useful information for developers who would like to program with BCM.
- The *Edge Manual* explains how BCM can be used with edge sites.
- The *Containerization Manual* describes how to manage containers with BCM.

If the manuals are downloaded and kept in one local directory, then in most pdf viewers, clicking on a cross-reference in one manual that refers to a section in another manual opens and displays that section in the second manual. Navigating back and forth between documents is usually possible with keystrokes or mouse clicks.

For example: `<Alt>-<Backarrow>` in Acrobat Reader, or clicking on the bottom leftmost navigation button of xpdf, both navigate back to the previous document.

The manuals constantly evolve to keep up with the development of the BCM environment and the addition of new hardware and/or applications. The manuals also regularly incorporate feedback from administrators and users, who can submit comments, suggestions or corrections via the website

`https://enterprise-support.nvidia.com/s/create-case`

Section 14.2 of the *Administrator Manual* has more details on submitting an issue.

0.3 Getting Administrator-Level Support

If the reseller from whom BCM was bought offers direct support, then the reseller should be contacted.

Otherwise the primary means of support is via the website

`https://enterprise-support.nvidia.com/s/create-case`

Section 14.2 of the *Administrator Manual* has more details on working with support.

0.4 Getting Professional Services

The BCM support team normally differentiates between

- regular support (customer has a question or problem that requires an answer or resolution), and
- professional services (customer asks for the team to do something or asks the team to provide some service).

Professional services can be provided via the NVIDIA Enterprise Services page at:

`https://www.nvidia.com/en-us/support/enterprise/services/`

1 Introduction

1.1 NVIDIA Mission Control Requirements And Features

The NVIDIA Mission Control features are extended features that are only available in the NVIDIA Mission Control edition of NVIDIA Base Command Manager (BCM). NVIDIA Mission Control features are targeted at NVIDIA B200, GB200, and other OEM platforms. The NVIDIA sales team must be contacted in case an NVIDIA Mission Control edition is required.

The NVIDIA Mission Control features of the NVIDIA Mission Control edition of BCM can be enabled on a BCM cluster with an active BCM license. Activation of the BCM license is described in Chapter 4 of the *Installation Manual*.

1.1.1 NVIDIA Mission Control Features

The NVIDIA Mission Control edition is capable of running the following:

- NVIDIA-conforming BMS: a *Building Management System* software that aids in cluster management in the building.
- Leak detection: monitoring for leak detection in the fluid-cooled processors. Associated with this are the possible automated actions taken.
- Autonomous hardware recovery:: Automates some hardware management operations in order to keep the cluster up.
- NMX: Product suite name for HPC cluster network monitoring and management system that comprises NMX Telemetry, NMX Manager, NMX Controller and NMX Oasis.
- IMEX (Internode Memory Exchange Service): a secure service that facilitates the mapping of GPU memory over NVLink between the GPUs in an NVLlink domain. IMEX configuration is described in section 7.5.1 of the *Administrator Manual* as part of workload management configuration.
- GB200 rack management: Automated management of rack power up and power down sequence.
- GB200 firmware management: GB200 firmware and BIOS management.
- Power reservation steering: Predicting power consumption, and capping the power to active components based on that prediction, to reduce data center energy costs.
- Run:ai: Kubernetes-based workload management and orchestration via Run:ai.

1.1.2 Checking That The NVIDIA Mission Control Edition Is Running

Whether the NVIDIA Mission Control edition is running on the cluster can be checked:

- in `cmsh`:

- using the licenseinfo command within cmsh

  Example

  ```
  root@basecm11:~# cmsh -c 'main licenseinfo | egrep Version\|Edition'
  Version                              10 and above
  Edition                              NVIDIA Mission Control
  ```

  or

- using the verify-license utility from a command line:

  Example

  ```
  root@basecm11:~# verify-license info
  ========= Certificate Information ========
  Version:                    10
  Edition:                    NVIDIA Mission Control
  ...
  ```

  or

- using the editiondisabledfeatures command.

  * If the NVIDIA Mission Control edition is running, then the features are all available, as indicated by:

    Example

    ```
    root@basecm11:~# cmsh -c 'main editiondisabledfeatures'
    No disabled features
    ```

  * If the NVIDIA Mission Control edition is not running, then the features are not available, as indicated by:

    ```
    root@basecm11:~# cmsh -c 'main editiondisabledfeatures'
    LeakDetection
    DisplayRack
    RackPower
    NGG
    APS
    Firmware
    NVIDIA-conforming BMS
    Run:AI
    IMEX
    NMX
    PRS
    ```

- in Base View the status of the NVIDIA Mission Control features is indicated by the navigation path: `Cluster>License Information>NVIDIA Mission Control Attributes`

2 NMX Settings For NVLink Monitoring

2.1 Introduction

NMX is a product suite for HPC cluster network monitoring and management. The system is made up of:

- NMX Telemetry
- NMX Manager
- NMX Controller
- NMX Oasis

The NVIDIA Mission Control edition of BCM can monitor NMX telemetry services that run on:

- NVLinks (direct GPU interconnection within a server)
- NVLink switches (direct GPU interconnection across servers)

2.2 NMX Metrics In BCM

NMX service authentication settings can be configured from within `partition` mode, within the `nvmsettings` submode:

```
[basecm11->partition[base]->nmxmsettings]% show
Parameter                         Value
--------------------------------- --------------------------------------------
Revision
Server                            10.140.0.41
User name                         rw-user
Password                          ********
Port                              443
Verify SSL                        no
```

After BCM has provisioned the NMX Controller and the NMX Telemetry services on the NVSwitches, then the `sample_nmxm` data producer produces the NMX measurables. These can be viewed from `partition` mode:

Example

```
[basecm11->partition[base]]% latestmetricdata| head -2; latestmetricdata| grep nmx
Measurable                       Parameter       Type    Value  Age   State Info
-------------------------------- -------------- ------- ------ ----- ----- -----
nmxm_chassis_count                               NMX-M   9      30.9s
nmxm_compute_allocation_count    all             NMX-M   54     30.9s
nmxm_compute_allocation_count    free            NMX-M   48     30.9s
nmxm_compute_allocation_count    full            NMX-M   6      30.9s
nmxm_compute_allocation_count    partial         NMX-M   0      30.9s
nmxm_compute_health_count        degraded        NMX-M   0      30.9s
nmxm_compute_health_count        healthy         NMX-M   0      30.9s
nmxm_compute_health_count        unhealthy       NMX-M   6      30.9s
nmxm_compute_health_count        unknown         NMX-M   63     30.9s
nmxm_compute_nodes_count                         NMX-M   69     30.9s
nmxm_domain_health_count         degraded        NMX-M   0      30.9s
nmxm_domain_health_count         healthy         NMX-M   0      30.9s
nmxm_domain_health_count         unhealthy       NMX-M   0      30.9s
nmxm_domain_health_count         unknown         NMX-M   3      30.9s
nmxm_gpu_health_count            degraded        NMX-M   0      30.9s
nmxm_gpu_health_count            degraded_bw     NMX-M   0      30.9s
nmxm_gpu_health_count            healthy         NMX-M   0      30.9s
nmxm_gpu_health_count            nonvlink        NMX-M   24     30.9s
nmxm_gpu_health_count            unknown         NMX-M   16     30.9s
nmxm_gpus_count                                  NMX-M   40     30.9s
nmxm_ports_count                                 NMX-M   152    30.9s
nmxm_switch_health_count         healthy         NMX-M   0      30.9s
nmxm_switch_health_count         missing_nvlink  NMX-M   8      30.9s
nmxm_switch_health_count         unhealthy       NMX-M   0      30.9s
nmxm_switch_health_count         unknown         NMX-M   0      0.9s
nmxm_switch_nodes_count                          NMX-M   46     30.9s
```

2.3 NMX Manager Metrics Forwarding Via BCM And `prometheusmetricforwarders`

Prometheus servers can be used to pick up NMX Manager metrics via Prometheus exporters. However in a typical BCM configuration, such as in a BCM type 1 network (section 3.3.9 of the *Installation Manual*), the Prometheus exporter is behind the BCM head node firewall. This means that by default an external Prometheus server cannot reach the exporter endpoint.

In such a case the `prometheusmetricforwarders` submode can be used to define the endpoint URL and HTTP method that the exporter uses. This forwards the exporter endpoint to the external network interface of the head node. The external Prometheus server can then reach the forwarded exporter endpoint.

Example

```
[basecm11->partition[base]->nmxmsettings->prometheusmetricforwarders[all]]% show
Parameter                        Value
-------------------------------- ------------------------------------------------
Name                             all
Revision
Urls                             http://10.140.0.41:9091/metrics
Method                           GET
```

3 Rack Management

3.1 Introduction

Rack management is the process of using tools and software to organize and manage the infrastructure components and racks in a data center.

A DGX GB200 system can be thought of as a "disaggregated supercomputer in a rack". The cluster administrator can manage the individual nodes in the traditional way, but BCM rack management features introduced in BCM version 11 allow the GB200 nodes to be managed more conveniently as unified NVL domains.

3.2 Basic Rack Management

Basic rack management is covered in section 3.15 of the *Administrator Manual*, The commands in `rack` mode that are discussed there include:

- The `list` command, which can give useful information on how the devices in racks are organized, if the attributes of racks in `rack` mode have been set up in a meaningful way, An example is given in section 4.8.1.
- The `display` command, which is useful for seeing the position of where devices are located in the rack, if the cluster administrator has recorded the device positions in the rack, An example is given in section 4.8.3.
- The `addrackposition` command (page 172 of the *Administrator Manual*) was introduced in BCM version 11. It allows rack positions in a cluster to be filled more easily than with a looping script.
- The `rackoverview` command (page 168 of the *Administrator Manual*) was also introduced in BCM version 11. It allows rack positions in a cluster to be overviewed conveniently. It shows information about the types of entities in a rack. It also then lists some more detailed information about some of the entity types.

For a DGX GB200 platform that has been configured, the output of the `rackoverview` command looks similar to the following for a specified rack (here it is a rack named `b05` that has 18 nodes, 9 switches, and 8 power shelves):

Example

```
[a03-p1-head-01->rack]% rackoverview b05
Type              Up               Down             Closed           Total
----------------- ---------------- ---------------- ---------------- -----------
Nodes             18               0                0                18
DPU nodes         0                0                0                0
Managed switches  0                0                0                0
NVLink switches   9                0                0                9
Power shelves     8                0                0                8
```

```
Devices          0                0                0                0
Cores            2,592            -                -                2,592
GPUs             72               -                -                72

Name                         Value
---------------------------- -----------------
User CPU                     0.03%
System CPU                   0.07%
Idle CPU                     99.9%
Other CPU                    0.0%
Memory used                  1.09 TiB (3.75%)
Memory unused                28.4 TiB (97.9%)
Memory total                 29.0 TiB
Total GPU utilization        0 W
Total GPU power usage        11.6 KW
Total GPU NVlink bandwidth   0 W
Average GPU temperature      30.9028 C
Total CPU power usage        1.60 KW
Average CPU temperature      48.6361 C

Node                GPU      Utilization  Temperature  Power usage  Memory used...
------------------- -------- ------------ ------------ ------------ -----------...
b05-p1-dgx-05-c01   gpu0     0.0%         30 C         167.184 W    0 B        ...
b05-p1-dgx-05-c01   gpu1     0.0%         31 C         154.726 W    0 B        ...
b05-p1-dgx-05-c01   gpu2     0.0%         32 C         154.894 W    0 B        ...
b05-p1-dgx-05-c01   gpu3     0.0%         31 C         170.649 W    0 B        ...
b05-p1-dgx-05-c02   gpu0     0.0%         31 C         162.994 W    0 B        ...
b05-p1-dgx-05-c02   gpu1     0.0%         31 C         151.328 W    0 B        ...
b05-p1-dgx-05-c02   gpu2     0.0%         31 C         151.149 W    0 B        ...
b05-p1-dgx-05-c02   gpu3     0.0%         31 C         170.537 W    0 B        ...
b05-p1-dgx-05-c03   gpu0     0.0%         31 C         166.664 W    0 B        ...
b05-p1-dgx-05-c03   gpu1     0.0%         31 C         154.725 W    0 B        ...
b05-p1-dgx-05-c03   gpu2     0.0%         31 C         151.047 W    0 B        ...
b05-p1-dgx-05-c03   gpu3     0.0%         31 C         170.125 W    0 B        ...
b05-p1-dgx-05-c04   gpu0     0.0%         31 C         170.642 W    0 B        ...
b05-p1-dgx-05-c04   gpu1     0.0%         31 C         155.034 W    0 B        ...
b05-p1-dgx-05-c04   gpu2     0.0%         31 C         152.078 W    0 B        ...
b05-p1-dgx-05-c04   gpu3     0.0%         31 C         166.551 W    0 B        ...
b05-p1-dgx-05-c05   gpu0     0.0%         31 C         170.516 W    0 B        ...
b05-p1-dgx-05-c05   gpu1     0.0%         31 C         158.7 W      0 B        ...
b05-p1-dgx-05-c05   gpu2     0.0%         31 C         158.7 W      0 B        ...
b05-p1-dgx-05-c05   gpu3     0.0%         31 C         174.413 W    0 B        ...
b05-p1-dgx-05-c06   gpu0     0.0%         30 C         166.44 W     0 B        ...
b05-p1-dgx-05-c06   gpu1     0.0%         30 C         154.628 W    0 B        ...
b05-p1-dgx-05-c06   gpu2     0.0%         31 C         154.865 W    0 B        ...
b05-p1-dgx-05-c06   gpu3     0.0%         31 C         170.423 W    0 B        ...
b05-p1-dgx-05-c07   gpu0     0.0%         31 C         164.078 W    0 B        ...
b05-p1-dgx-05-c07   gpu1     0.0%         31 C         158.303 W    0 B        ...
b05-p1-dgx-05-c07   gpu2     0.0%         31 C         150.958 W    0 B        ...
b05-p1-dgx-05-c07   gpu3     0.0%         31 C         170.537 W    0 B        ...
b05-p1-dgx-05-c08   gpu0     0.0%         31 C         166.661 W    0 B        ...
b05-p1-dgx-05-c08   gpu1     0.0%         31 C         155.875 W    0 B        ...
b05-p1-dgx-05-c08   gpu2     0.0%         31 C         154.726 W    0 B        ...
b05-p1-dgx-05-c08   gpu3     0.0%         32 C         170.311 W    0 B        ...
b05-p1-dgx-05-c09   gpu0     0.0%         31 C         166.331 W    0 B        ...
b05-p1-dgx-05-c09   gpu1     0.0%         31 C         155.469 W    0 B        ...
b05-p1-dgx-05-c09   gpu2     0.0%         30 C         154.751 W    0 B        ...
b05-p1-dgx-05-c09   gpu3     0.0%         32 C         175.429 W    0 B        ...
b05-p1-dgx-05-c10   gpu0     0.0%         30 C         174.181 W    0 B        ...
```

```
b05-p1-dgx-05-c10  gpu1     0.0%         31 C         158.523 W    0 B        ...
b05-p1-dgx-05-c10  gpu2     0.0%         31 C         152.808 W    0 B        ...
b05-p1-dgx-05-c10  gpu3     0.0%         31 C         166.551 W    0 B        ...
b05-p1-dgx-05-c11  gpu0     0.0%         30 C         166.704 W    0 B        ...
b05-p1-dgx-05-c11  gpu1     0.0%         31 C         154.726 W    0 B        ...
b05-p1-dgx-05-c11  gpu2     0.0%         31 C         155.034 W    0 B        ...
b05-p1-dgx-05-c11  gpu3     0.0%         30 C         170.679 W    0 B        ...
b05-p1-dgx-05-c12  gpu0     0.0%         31 C         170.424 W    0 B        ...
b05-p1-dgx-05-c12  gpu1     0.0%         31 C         155.601 W    0 B        ...
b05-p1-dgx-05-c12  gpu2     0.0%         31 C         150.892 W    0 B        ...
b05-p1-dgx-05-c12  gpu3     0.0%         31 C         166.806 W    0 B        ...
b05-p1-dgx-05-c13  gpu0     0.0%         31 C         166.766 W    0 B        ...
b05-p1-dgx-05-c13  gpu1     0.0%         30 C         151.894 W    0 B        ...
b05-p1-dgx-05-c13  gpu2     0.0%         31 C         151.22 W     0 B        ...
b05-p1-dgx-05-c13  gpu3     0.0%         31 C         170.205 W    0 B        ...
b05-p1-dgx-05-c14  gpu0     0.0%         31 C         172.778 W    0 B        ...
b05-p1-dgx-05-c14  gpu1     0.0%         31 C         162.152 W    0 B        ...
b05-p1-dgx-05-c14  gpu2     0.0%         31 C         150.857 W    0 B        ...
b05-p1-dgx-05-c14  gpu3     0.0%         31 C         166.33 W     0 B        ...
b05-p1-dgx-05-c15  gpu0     0.0%         31 C         166.518 W    0 B        ...
b05-p1-dgx-05-c15  gpu1     0.0%         31 C         154.729 W    0 B        ...
b05-p1-dgx-05-c15  gpu2     0.0%         31 C         154.1 W      0 B        ...
b05-p1-dgx-05-c15  gpu3     0.0%         31 C         170.537 W    0 B        ...
b05-p1-dgx-05-c16  gpu0     0.0%         30 C         174.067 W    0 B        ...
b05-p1-dgx-05-c16  gpu1     0.0%         30 C         158.385 W    0 B        ...
b05-p1-dgx-05-c16  gpu2     0.0%         31 C         151.157 W    0 B        ...
b05-p1-dgx-05-c16  gpu3     0.0%         31 C         166.77 W     0 B        ...
b05-p1-dgx-05-c17  gpu0     0.0%         31 C         166.161 W    0 B        ...
b05-p1-dgx-05-c17  gpu1     0.0%         31 C         158.62 W     0 B        ...
b05-p1-dgx-05-c17  gpu2     0.0%         31 C         151.058 W    0 B        ...
b05-p1-dgx-05-c17  gpu3     0.0%         31 C         169.704 W    0 B        ...
b05-p1-dgx-05-c18  gpu0     0.0%         31 C         166.441 W    0 B        ...
b05-p1-dgx-05-c18  gpu1     0.0%         31 C         155.164 W    0 B        ...
b05-p1-dgx-05-c18  gpu2     0.0%         31 C         150.758 W    0 B        ...
b05-p1-dgx-05-c18  gpu3     0.0%         31 C         169.752 W    0 B        ...

Switch           Utilization  Temperature Power usage  Fan speed Links active L...
---------------- ------------ ----------- ------------ --------- ------------ -...
B05-P1-NVSW-01  30.2%        0 C         0 W          0 RPM     0            0
B05-P1-NVSW-02  27.5%        0 C         0 W          0 RPM     0            0
B05-P1-NVSW-03  0.0%         0 C         0 W          0 RPM     0            0
B05-P1-NVSW-04  27.6%        0 C         0 W          0 RPM     0            0
B05-P1-NVSW-05  27.6%        0 C         0 W          0 RPM     0            0
B05-P1-NVSW-06  27.5%        0 C         0 W          0 RPM     0            0
B05-P1-NVSW-07  27.5%        0 C         0 W          0 RPM     0            0
B05-P1-NVSW-08  27.5%        0 C         0 W          0 RPM     0            0
B05-P1-NVSW-09  27.6%        0 C         0 W          0 RPM     0            0

Power shelf    Input power  Output power Temperature  Fan speed Active PSU  To...
-------------- ------------ ------------ ------------ --------- ----------- --...
B05-P1-PWR-01  3.7 KW       3.5 KW       43.3333 C    6.7 KRPM  6           6
B05-P1-PWR-02  3.11 KW      2.95 KW      42.4 C       6.7 KRPM  6           6
B05-P1-PWR-03  3.7 KW       3.5 KW       44.1667 C    6.7 KRPM  6           6
B05-P1-PWR-04  3.8 KW       3.6 KW       44 C         6.7 KRPM  6           6
B05-P1-PWR-05  3.6 KW       3.4 KW       45.5 C       6.7 KRPM  6           6
B05-P1-PWR-06  3.8 KW       3.6 KW       47.1667 C    6.7 KRPM  6           6
B05-P1-PWR-07  3.5 KW       3.4 KW       46 C         6.7 KRPM  6           6
B05-P1-PWR-08  3.6 KW       3.5 KW       47 C         6.7 KRPM  6           6
[a03-p1-head-01->rack]%
```

3.3 Advanced Features In Rack Management

Over time, clusters have become more powerful and their cooling requirements have become more demanding. Liquid cooling technology has therefore become more common over time.

The monitoring associated with liquid cooling has also developed. Monitoring of the electrical supply, fans, component temperatures, and coolant values such as flow and and pressure has become popular. Coolant monitoring in BCM includes an ability to detect leaks and then take actions.

This section describes some of the rack mode commands in BCM that can help the cluster administrator with an overview of monitoring values, including values associated with liquid cooling.

3.3.1 The electricaloverview Command

The electricaloverview command within rack mode gives a convenient overview of the electrical supply to the cluster. The power used and the power budget of a specified rack (B05 in the following example) can be seen:

Example

```
[basecm11->rack[B05]]% electricaloverview
Rack             Power used       Power budget     Isolation status
---------------- ---------------- ---------------- ----------------
B05              28.5 kW          135 kW           0
```

If no rack is specified, then the command displays the information for all racks.

3.3.2 The liquidcoolingoverview Command

The liquidcoolingoverview command within rack mode conveniently displays coolant metrics:

```
[basecm11->rack]% liquidcoolingoverview a05
      Supply         Return        Flow         Differential  Isolation
Rack  temperature    temperature   rate         pressure      status
----- -------------- ----------- ----------- ------------- -----------
A05   26.03 C        30.78 C       70.938 LPM   31 KPa        0
```

If no rack is specified, then the command displays the information for all racks.

3.3.3 The leakdetectionoverview Command

The leakdetectionoverview command within rack mode gives a convenient overview about sensor readings related to leak detection in BCM and BMS. BMS (Building Management System) is a software used to manage IT assets in a building, and is used by BCM internally.

The output from the command applied to a specific rack looks similar to the following:

```
[basecm11->rack]% leakdetectionoverview a05
                             Liquid    Liquid              Electrical Electrical
               Leak     Leak isolation isolation Liquid    isolation  isolation  Electrical
     Leak      tray     sensor request  request  isolation request    request    isolation
Rack detected detected fault  (BCM)     (BMS)    status    (BCM)      (BMS)      status
---- -------- -------- ------ --------- --------- --------- ---------- --------- ----------
A05  0        0        0      0         0         0         0          0          0
[basecm11->rack]%
```

If no rack is specified, then the command displays the information for all racks.

3.3.4 The `leakinfo` Command

The `leakinfo` command is related to leakage detection, but it is not a `rack` mode command. It is a `device` mode command. It is noted in this section for completeness. It can be run at `device` mode level to list all leaks detected in the cluster.

Typically, and ideally, running the command shows something similar to:

```
[basecm11->device]% leakinfo
Hostname         Timestamp Severity  Info
---------------- --------- --------- ----------------
[basecm11->device]%
```

If there are issues on the devices, then the output may look similar to::

```
[basecm11->device]% leakinfo
Hostname        Timestamp Severity  Info
--------------- --------- --------- --------------------------------------------
a05-dgx-01-c09  15:09:23  large     The state of resource `Chassis_0_LeakDete...
a07-dgx-03-c16  15:38:48  large     The state of resource `Chassis_0_LeakDete...
b05-dgx-05-c08  09:11:06  large     The state of resource `Chassis_0_LeakDete...
b05-dgx-05-c10  09:11:05  large     The state of resource `Chassis_0_LeakDete...
b06-dgx-06-c17  11:49:30  large     The state of resource `Chassis_0_LeakDete...
[basecm11->device]%
```

Running `help leakinfo` lists further command options.

3.3.5 The `isolationrequestinfo` Command

If a leak is detected, then BCM receives the report. BCM then:

- powers off the node that is leaking
- powers off the rack if multiple nodes are leaking
- isolates the rack to prevent other nodes suffering the same fate

Isolation means that power no longer runs through the leaking racks.

The isolation status of a rack can be viewed with the `isolationrequestinfo` command, which normally shows something similar to:

Example

```
[basecm11->rack]% isolationrequestinfo
Rack             Timestamp    Kind         Value    Info
---------------- ------------ ------------ -------- ----------------
[basecm11->rack]%
```

During a leakage event, the output may show something similar to:

Example

```
[basecm11->rack]% isolationrequestinfo
Rack  Timestamp Kind Value Info
----- --------- ---- ----- ----------------------------
B05   03:40:01       4     Default building policy/Rack
[basecm11->rack]%
```

3.3.6 Managing Leaks With rules In leakactionpolicies

The cluster administrator has the freedom to construct actions based on triggers based on leak detection measurables. While that is powerful, that can be quite labor-intensive for cluster administrators.

BCM provides a simpler way to deal with leaks, based on leak action policies. Leak action policies are a collection of rules that carry out actions to be followed with leaks are detected. Racks play an important part in these policies and rules.

The policies use predefined scopes, or groupings. The names of these groupings are rules. The groupings are listed next, in order from largest to smallest, along with their associated descriptions:

- `Building`: a building is made up of rooms
- `Room`: A room has rows of racks in it.
- `Row`: A row of racks has individual racks in it.
- `Rack`: An individual rack has devices in it.
- `Device`: The smallest possible grouping

The policies can be listed in the `leakactionpolicies` submode, under `partition` mode.

Example

```
[basecm11->partition[base]->leakactionpolicies]% list
Name (key)               Rules
------------------------ ----------------------------------
Default building policy  Building, Device, Rack, Room, Row
Default device policy    Device
Default rack policy      Device, Rack
Default room policy      Device, Rack, Room, Row
Default row policy       Device, Rack, Row
```

The rules used by each policy are also shown, under the `Rules` column in the preceding example.

Each rule in a policy is a copy of a template. By default the rules per policy are identical.

Just the rules used by a specific policy can also be viewed:

Example

```
[basecm11->partition[base]->leakactionpolicies]% use default building policy
[basecm11->partition[base]->leakactionpolicies[Default building policy]]% show
Parameter                        Value
-------------------------------- ------------------------------------------------
Name                             Default building policy
Revision
Rules                            Building, Device, Rack, Room, Row
```

Within a policy, the value of the settings for these rules can be listed:

Example

```
[basecm11->...leakactionpolicies[Default building policy]->rules]% list
                       Power  Power    Electrical Liquid     Minimal Maximal
Name (key)  Scope     off    off rack isolation  isolation  devices devices
----------- --------- ------ -------- ---------- ---------- ------- -------
Building    building  yes    yes      no         no         11      100000
Device      device    yes    no       no         no
Rack        rack      yes    yes      yes        yes        3       100
Room        room      yes    yes      no         no         7       100000
Row         row       yes    yes      no         no         5       100
```

The rules for each policy can be modified independently for each policy, so that, for example the `Rack` rule values may differ between `Default row policy` and `Default building policy`. However, a sensible cluster administrator is not expected to carry out such a modification.

Another way of phrasing this is: The values of the rules can be changed from the defaults. In that case, a sensible administrator is expected to keep the values synchronized across the rules in the leak action policies.

In the preceding example:

If the minimal devices criterion is met for leaking for each rule in a policy, then the values for the column header are applied to the rules of that policy.

So, for the policy rules in the preceding example, when leaking is detected:

- If only one or two devices are detected as leaking, then only the devices involved are powered off.
- If, in one rack, more than 3 devices, but less than 5 devices, are detected as leaking, then in addition to the device rules being applied, the entire rack is powered off. Isolation is also carried out for the electrical power and liquid coolant.
- If more devices meet the conditions, so a larger scope is affected by leaking, then the rules for the larger scope are also applied.

The rules for a specific policy can be removed. The `row` rule for `Default building policy` can be removed with, for example:

```
[basecm11->partition[base]->leakactionpolicies[Default building policy]]% rules
[basecm11->...leakactionpolicies[Default building policy]->rules]% remove <TAB><TAB>
building  device    rack      room      row
[basecm11->...leakactionpolicies[Default building policy]->rules]% remove row; commit
```

Custom Rules For `leakactionpolicies`

The names for the rules (`Device`, `Rack`, `Row` ...) are really just default labels, and a sensible cluster administrator is expected to have the description associated with the name match the physical reality. Creating a custom rule may help with this.

Custom rules can be added by the cluster administrator, by first adding another policy:

```
[basecm11->partition[base]->leakactionpolicies[Default building policy]]% add mypolicy
[basecm11->partition*[base*]->leakactionpolicies*[mypolicy*]]% list
Name (key)                Rules
------------------------- ----------------------------------
Default building policy   Building, Device, Rack, Room, Row
Default device policy     Device
Default rack policy       Device, Rack
Default room policy       Device, Rack, Room, Row
Default row policy        Device, Rack, Row
mypolicy                  none
[basecm11->partition*[base*]->leakactionpolicies*[mypolicy*]]% show
Parameter                         Value
--------------------------------- ------------------------------------------------
Name                              mypolicy
Revision
Rules                             none
```

By default, there are no rules in the new policy:

```
[basecm11->partition*[base*]->leakactionpolicies*[mypolicy*]]% rules
[basecm11->partition*[base*]->leakactionpolicies*[mypolicy*]->rules]% list

                                                 Electrical  Liquid      Minimal Maximal
Name (key)  Scope     Power off  Power off rack  isolation   isolation   devices devices
----------- --------- ---------- --------------- ---------- ----------- -------- ------
```

The cluster administrator may have a testing room in which to keep a rack. In that case, the following might be added:

```
[basecm11->partition*[base*]->leakactionpolicies*[mypolicy*]->rules]% add testroom
[basecm11->partition*[base*]->leakactionpolicies*[mypolicy*]->rules*[testroom*]]% show
Parameter                         Value
--------------------------------- ------------------------------------------------
Name                              testroom
Revision
Scope                             rack
Disabled                          no
Minimal devices                   1
Maximal devices                   100
Grace period                      0s
Correlation window                5m
Power off                         yes
Power off rack                    no
Electrical isolation              no
Liquid isolation                  no
Minimal severity                  1
Maximal severity                  100
```

Adding a rule adds a default with the preceding rack-based rule values.

The rule values can be modified using the `set` command.

Running `set` without any options provides a help text explaining what can be set:

```
[basecm11->partition*[base*]->leakactionpolicies*[mypolicy*]->rules]% set
```

3.4 Building Management System And CMDaemon

3.4.1 Introduction To How BMS And BCM Work Together

A Building Management System, (BMS) software that is capable of integration with BCM, is known as an *conforming BMS*. The conformance is to a reference design for data center planning (NVIDIA Controls and Monitoring Reference Design), available to some NVIDIA product users.

Since BCM version 11, BCM integrates with a conforming BMS for leak detection and control in the data center (DC). The implementation of the integration between BCM and a conforming BMS is also meant to serve as a guide to NVIDIA customers who may use different BMS solutions.

BCM communicates with a conforming BMS. Together they monitor the DC. The integration works with DGX GB200 systems that are liquid-cooled. as well as other systems.

In a DC, a conforming BMS is able to detect most of the detected leak-related events. The remaining detected leak-related events, from the in-tray BMCs (compute tray and switch tray), are dealt with by BCM.

BCM uses the MQTT protocol for two-way communication with the conforming BMS, using a BCM service `cm-mqtt`.

The BMS sends BCM information about the inventory of the DC, and about the leak-related events it knows about (non-tray events). MQTT topics are created, and structured in a way that BCM can parse.

BCM in turn sends the BMS instructions on liquid management or power management, according to defined policies on events such as leaks or power outages (section 4.7).

3.4.2 Building Management System Configuration

BMS integration can be configured in BCM in partition mode in `cmsh` and Base View.

The BMS software type used can be configured in `partition` mode:

Example

```
root@basecm11:~# cmsh
[basecm11]% partition
[basecm11->partition[base]]% set bms nvidia
conforming
bms; confirm
```

The value set for `bms` can also be set to the URL of a BMS server, for example `http://bmserver/path`, and credentials for it can then be set:

Example

```
root@basecm11:~# cmsh
[basecm11]% partition
[basecm11->partition[base]]% set bms url
[basecm11->partition*[base*]]% set bmscertificate
[basecm11->partition*[base*]]% set bmspath http://bmsserver/path
[basecm11->partition*[base*]]% set bmsprivatekey
[basecm11->partition*[base*]]% commit
```

In the following example, an `mqtt` role is assigned in the `mqtt` configuration overlay. The overlay specifies where the `cm-mqtt` service should run. In this case it runs on all the head nodes:

Example

```
root@basecm11:~# cmsh
[basecm11->partition[base]]% configurationoverlay
[basecm11->configurationoverlay]% add mqtt
[basecm11->configurationoverlay*[mqtt*]]% set allheadnodes yes
```

Within the role assignment, the administrator specifies the hostname or IP address of where the BMS server runs, and where the `cm-mqtt` service pulls data from:

```
[basecm11->configurationoverlay*[mqtt*]]% roles
[basecm11->configurationoverlay*[mqtt*]->roles]% assign mqtt
[basecm11->configurationoverlay*[mqtt*]->roles*[mqtt*]]% show
Parameter                        Value
-------------------------------- ----------------------------------------------------------
Name                             mqtt
Revision
Type                             MQTTRole
Add services                     yes
Servers                          <0 in submode>
CA certificate path              /cm/local/apps/cmd/pythoncm/lib/python3.12/site-packages/\
                                 pythoncm/etc/cacert.pem
Private key path                 /cm/local/apps/cmd/cm-mqtt/etc/mqtt.key
Certificate path                 /cm/local/apps/cmd/cm-mqtt/etc/mqtt.pem
Write named pipe path            /var/spool/cmd/mqtt.pipe
[basecm11->configurationoverlay*[mqtt*]->roles*[mqtt*]]% servers
[basecm11->configurationoverlay*[mqtt*]->roles*[mqtt*]->servers]% list
Server (key)     Port       Disabled
---------------- ---------- ----------
[basecm11->configurationoverlay*[mqtt*]->roles*[mqtt*]->servers]% add mqhost
[basecm11->configurationoverlay*[mqtt*]->roles*[mqtt*]->servers*[mqhost*]]% show
Parameter                        Value
-------------------------------- ------------------------------------------------
Revision
Server                           mqhost
Port                             1883
Topic                            BCM/#
Disabled                         no
Username
Password                         < not set >
Transport                        tcp
Protocol                         v3.1.1
Certificate required             yes
CA certificate
Certificate
Private key
[basecm11->configurationoverlay*[mqtt*]->roles*[mqtt*]->servers*[mqhost*]]% set certificate
...
```

A username, password, and service certificate can be set. These allow BCM to communicate with the BMS securely:

```
[basecm11->configurationoverlay[mqtt]->roles[mqtt]->servers[mqhost]]% show
Parameter                        Value
-------------------------------- ------------------------------------------------
Revision
Server                           mqhost
Port                             1883
Topic                            BCM/#
Disabled                         no
```

```
Username                          cronus
Password                          ********
Transport                         tcp
Protocol                          v3.1.1
Certificate required              yes
CA certificate
Certificate
Private key
[basecm11->configurationoverlay[mqtt]->roles[mqtt]->servers[mqhost]]%
```

If the cluster manager has correctly established communication with the BMS, then the status of the BMS (metrics for power circuit, CDUs, electrical, and liquid cooling) can be seen from within cmsh (section 4.7).

Example

```
[basecm11]% powercircuit overview
Power circuit  Power     Current     Current limit   Current phase 1  Current phase 2 ...
-------------- --------- ----------- --------------- ---------------- ----------------...
RPP-B12-3      14.0 KW   26.1118 A   0 A             17.7822 A        26.1118 A       ...
RPP-B14-3      15.3 KW   29.6178 A   0 A             17.7419 A        29.6178 A       ...
RPP-B21-5      7.5 KW    11.2255 A   0 A             10.4872 A        10.5412 A       ...
...
```

4 BCM Power Shelf Integration

The NVIDIA GB200 server platforms use power shelves to distribute power in racks. Several power shelves are used per rack. Each power shelf takes up one rack unit. Each power shelf is comprised of several PSUs (Power Supply Units).

The reference design for an NVIDIA GB200 rack has 6 power shelves, and uses 6 PSUs in each power shelf. Each PSU is rated up to 5.5kW, so that each power shelf is rated up to 33kW.

This chapter considers an NVIDIA GB200 server platform that has been configured to work with BCM.

4.1 Power Shelf Listing And Overview

The list -t powershelf command in device mode can be used to list the power shelves of a configured cluster

Power shelves are configured as devices on an IPMI network, as indicated by the following example (some rows ellipsized):

Example

```
[basecm11->device]% list -t powershelf
Type        Hostname (key)   MAC                IP          Network    Status
----------- ---------------- ------------------ ----------- ---------- ----------
PowerShelf  B05-P1-PWR-01    00:18:23:0C:1E:90  7.241.1.47  rfnet      [   UP    ]
...
PowerShelf  B05-P1-PWR-08    00:18:23:0C:1E:85  7.241.1.54  rfnet      [   UP    ]
PowerShelf  a05-p1-pwr-01    00:18:23:0C:1E:84  7.241.1.8   rfnet      [   UP    ]
...
PowerShelf  a05-p1-pwr-08    00:18:23:0C:1E:8F  7.241.1.80  rfnet      [   UP    ]
PowerShelf  b06-p1-pwr-01    00:18:23:0C:40:2A  7.241.1.55  rfnet      [   UP    ]
...
PowerShelf  b06-p1-pwr-08    00:18:23:0C:40:99  7.241.1.62  rfnet      [   UP    ]
```

The powershelfoverview command displays an overview of a particular power shelf, showing the measurements from its PSUs (6 in the following example):

Example

```
[basecm11->device*[B05-P1-PWR-01]]% powershelfoverview
       Input  Input   Input     Rail  Rail    Rail                 Fan
Supply power  current voltage   power current voltage Temperature speed    Healthy
------ ------ ------- --------  ----- ------- ------- ----------- -------- -------
1      679  W  2.98 A  240.75 V  652  W 13.06 A 49.85 V 44 C        6.7 KRPM PASS
```

```
2       632  W   2.8  A   240.75  V   607  W  12.18  A  49.87  V  44  C        6.6  KRPM  PASS
3       599  W   2.64  A  240.75  V   575  W  11.48  A  49.87  V  43  C        6.8  KRPM  PASS
4       599  W   2.63  A  240.75  V   575  W  11.51  A  49.89  V  43  C        6.8  KRPM  PASS
5       617  W   2.76  A  240.5  V    574  W  11.5  A   49.87  V  43  C        6.9  KRPM  PASS
6       609  W   2.64  A  240.5  V    584  W  11.7  A   49.87  V  43  C        6.6  KRPM  PASS
```

4.2 Power Shelves Networking Configuration

The network interface settings for a specific power shelf can be managed with its Redfish interface (section 3.7.1 of the *Administrator Manual*). For example, a power shelf, here named `B05-P1-PWR-01`, can have the following values set with the `set` command:

Example

```
[basecm11->device[B05-P1-PWR-01]->interfaces[rf0]]% show
Parameter                         Value
--------------------------------- ------------------------------------------------
Revision
Type                              bmc
Network device name               rf0
Network                           rfnet
IP                                7.241.1.47
DHCP                              no
Alternative Hostname
Additional Hostnames
Switch ports
Start if                          always
BringUpDuringInstall              no
On network priority               60
Bootable                          no
MAC                               00:18:23:0C:1E:90
Speed
Card Type
```

Redfish metrics, which are the ones prefixed by `RF_` (as seen in section 6.8), are sampled only when the BMC interface starts with `rf0`, `rf1`, ...

4.3 Access Configuration For The Power Management Controller

Access to the Power Management Controller (PMC) is configured in the BMC settings:

Example

```
[basecm11->device[B05-P1-PWR-01]->bmcsettings]% show
Parameter                         Value
--------------------------------- ------------------------------------------------
Revision
User name                         root
Password                          ********
User ID                           -1
Power reset delay                 0s
Extra arguments
Privilege                         administrator
```

Authentication allows NVIDIA Mission Control to monitor the power shelves using Redfish sampling.

4.4 Power Shelf Settings Overview

An overview of the settings for a particular power shelf can be viewed with the show command:

Example

```
[basecm11->device[B05-P1-PWR-01]]% show
Parameter                         Value
--------------------------------- ------------------------------------------------
Hostname                          B05-P1-PWR-01
IP                                7.241.1.47
Network                           rfnet
Revision
Type                              PowerShelf
Mac                               00:18:23:0C:1E:90
Model
Activation                        Tue, 11 Feb 2025 16:45:39 PST
Rack                              B05:6
Chassis                           < not set >
Access Settings                   <submode>
Management network                rfnet
Power control                     none
Custom power script
Custom power script argument
Power distribution units
Default gateway metric            0
Switch ports                      B04-P1-OOB-02:1
Interfaces                        <1 in submode>
PMC Settings                      <submode>
Userdefined1
Userdefined2
User defined resources
Supports GNSS                     no
Custom ping script
Custom ping script argument
Partition                         base
Part number
Serial number
Notes                             <0B>
Prometheus metric forwarders      <0 in submode>
```

4.5 Power Shelf Metrics Overview

A listing of the power shelf metrics can be seen with some grepping (the output for PSUs 2 to 5 have been ellipsized for readablity in this manual):

```
[basecm11->monitoring->measurable]% list -f name:40,parameter:9,producer
|head -2; list -f name:40,parameter:9,producer| egrep -i
'(powershel|redfish_power_shelf)'
name (key)                               parameter producer
---------------------------------------- --------- --------------------
AveragePowerShelfTemperature                       AggregatePowerShelf
PowerShelvesClosed                                 ClusterTotal
PowerShelvesDown                                   ClusterTotal
PowerShelvesTotal                                  ClusterTotal
PowerShelvesUp                                     ClusterTotal
RF_Active_PSU                                      redfish_power_shelf
RF_PowerShelf_Status                     1         redfish_power_shelf
...
```

```
RF_PowerShelf_Status                     6         redfish_power_shelf
RF_PowerShelf_Status                     total     redfish_power_shelf
RF_Power_Supply_Energy                   1         redfish_power_shelf
...
RF_Power_Supply_Energy                   6         redfish_power_shelf
RF_Power_Supply_Energy                   total     redfish_power_shelf
RF_Power_Supply_FanSpeed                 1         redfish_power_shelf
...
RF_Power_Supply_FanSpeed                 6         redfish_power_shelf
RF_Power_Supply_FanSpeed                 average   redfish_power_shelf
RF_Power_Supply_InputCurrent             1         redfish_power_shelf
...
RF_Power_Supply_InputCurrent             6         redfish_power_shelf
RF_Power_Supply_InputCurrent             total     redfish_power_shelf
RF_Power_Supply_InputPower               1         redfish_power_shelf
...
RF_Power_Supply_InputPower               6         redfish_power_shelf
RF_Power_Supply_InputPower               total     redfish_power_shelf
RF_Power_Supply_InputVoltage             1         redfish_power_shelf
...
RF_Power_Supply_InputVoltage             6         redfish_power_shelf
RF_Power_Supply_LifetimeReading_Energy   1         redfish_power_shelf
...
RF_Power_Supply_LifetimeReading_Energy   6         redfish_power_shelf
RF_Power_Supply_OutputPower              1         redfish_power_shelf
...
RF_Power_Supply_OutputPower              6         redfish_power_shelf
RF_Power_Supply_OutputPower              total     redfish_power_shelf
RF_Power_Supply_PowerFactor_Input        1         redfish_power_shelf
...
RF_Power_Supply_PowerFactor_Input        6         redfish_power_shelf
RF_Power_Supply_RailCurrent              1         redfish_power_shelf
...
RF_Power_Supply_RailCurrent              6         redfish_power_shelf
RF_Power_Supply_RailCurrent              total     redfish_power_shelf
RF_Power_Supply_RailPower                1         redfish_power_shelf
...
RF_Power_Supply_RailPower                6         redfish_power_shelf
RF_Power_Supply_RailPower                total     redfish_power_shelf
RF_Power_Supply_RailVoltage              1         redfish_power_shelf
...
RF_Power_Supply_RailVoltage              6         redfish_power_shelf
RF_Power_Supply_Temperature              1         redfish_power_shelf
...
RF_Power_Supply_Temperature              6         redfish_power_shelf
RF_Power_Supply_Temperature              average   redfish_power_shelf
RF_Power_Supply_code_error               1         redfish_power_shelf
...
RF_Power_Supply_code_error               6         redfish_power_shelf
RF_Power_Supply_message_error            1         redfish_power_shelf
...
RF_Power_Supply_message_error            6         redfish_power_shelf
RF_Summary_Metrics_Power                           redfish_power_shelf
RF_Summary_Metrics_Temperature                     redfish_power_shelf
RF_Total_PSU                                       redfish_power_shelf
TotalPowerShelfInputCurrent                        AggregatePowerShelf
TotalPowerShelfInputPower                          AggregatePowerShelf
TotalPowerShelfRailCurrent                         AggregatePowerShelf
TotalPowerShelfRailPower                           AggregatePowerShelf
```

The descriptions for these metrics are as indicated in table 6.4.

4.6 Power Shelf Firmware

Power shelf firmware is managed by the firmware command options as described in Section 14.5.3 of the *Administrator Manual*.

4.7 Power Shelf Circuit Monitoring

A power circuit typically supplies several power shelves, based on the rated power of the circuit and power shelves. Administration and wiring convenience may also play a role in allocating a circuit to a power shelf.

So, for example, the administrator may configure a rack with 6 power shelves so that it is supplied power from one power circuit.

The power circuit breakers used by each power circuit, typically provided by a remote power panel (RPP), can have their locations noted for administrative convenience.

Example

```
[basecm11->powercircuit]% list
Name (key) Row  Room   Building     Location    Racks
---------- ---- ------ ------------ ----------- -----
RPP-B12-3  1    G      Watchtower   Trantor
RPP-B14-3
RPP-B21-5
RPP-B21-6
RPP-B22-5
...
```

The cluster may have been set up to monitor the power circuit breaker for the power reaching BCM (section 3.4). In that case, the overview command typically displays the power consumption and current per phase.
Example

```
[basecm11->powercircuit]% overview RPP-B14-3
                                 Current  Current  Current   Current
Power circuit Power   Current   limit    phase 1  phase 2   phase 3   Racks
------------- ------- --------- -------- -------- --------- --------- -----
RPP-B14-3     15.6 KW 30.137 A  0 A      18.28 A  30.119 A  28.134 A  1
```

A value of 0 A means that the current limit is not reported.

4.8 Power Shelves And Rack Mode Commands

This section discusses some of the commands that can be useful for setting up and viewing devices from within the `rack` mode of `cmsh` (section 3.15 of the *Administrator Manual*), but with a focus on power shelf management.

4.8.1 The `list` Command

If the attributes of racks in `rack` mode have been set up in a meaningful way, then the list command can give useful information on how the power shelves in racks are organized:

Example

```
[basecm11->rack]% list
                     x-Coord-   y-Coord
Name (key)   Room   inate      inate   Height   Devices
----------- ----- --------- ------ ------- ------------------------------------
A01          T      1          1       48
A02          T      1          2       48
A03          T      1          3       48
A04          T      1          4       48
A05          T      1          5       48        a05-p1-pwr-01..a05-p1-pwr-08,a05...
A06          T      1          6       48
A07          T      1          7       48
A08          T      1          8       48
A09          T      1          9       48
A10          T      1          10      48
A11          T      1          11      48
A12          T      1          12      48
...
```

The help set command describes how these values can be set.

In the preceding example, the administrator has noted that a rack, for example `A05`, is in a room T, and is at an (x,y) coordinate of (1,5) in that room.

The power shelf devices have also been given meaningful names: `a05-p1-pwr-01`, `a05-p1-pwr-02`..., which here has a prefix that is the rack it is in, and uses `pwr` to indicate it is a power shelf.

4.8.2 The `rackoverview` Command

The rackoverview command displays information about the types of entities in a rack. It also then lists some more detailed information about some of the entity types.

The power shelf one of the types that has more detailed information listed (some output truncated and ellipsized, because this section is focussing on the power shelves):

Example

```
[basecm11->rack]% rackoverview <TAB><TAB>
a01  a02  a03  a04  a05  a06  a07  a08  a09  a10  a11  a12  b01  b02  b03  b04...
[basecm11->rack]% rackoverview a05
Type                 Up                Down              Closed            Total
----------------- --------------- --------------- --------------- ---------------
Nodes                18                0                 0                 18
DPU nodes            0                 0                 0                 0
Managed switches     0                 0                 0                 0
NVLink switches      0                 9                 0                 9
Power shelves        8                 0                 0                 8
Devices              0                 0                 0                 0
Cores                2,592             -                 -                 2,592
GPUs                 72                -                 -                 72

...

Power              Input      Output                 Fan        Active   Total
shelf              power      power      Temperature  speed      PSU      PSU
--------------- -------- -------- ----------- --------- ------- ------
a05-p1-pwr-01     0 W        0 W        0 C           0 RPM      0        6
a05-p1-pwr-02     0 W        0 W        0 C           0 RPM      0        6
a05-p1-pwr-03     3.7 KW     3.6 KW     43.3333 C     6.7 KRPM   6        6
a05-p1-pwr-04     2.51 KW    2.38 KW    42.75 C       6.7 KRPM   6        6
```

```
a05-p1-pwr-05   2.83 KW  2.69 KW  49 C          6.8 KRPM  6       6
a05-p1-pwr-06   3.7 KW   3.5 KW   45 C          6.7 KRPM  6       6
a05-p1-pwr-07   3.7 KW   3.5 KW   46.3333 C     6.7 KRPM  6       6
a05-p1-pwr-08   3.8 KW   3.7 KW   47.3333 C     6.7 KRPM  6       6
[basecm11->rack]%
```

In the preceding, the A05 rack is seen to have some active power shelves, which show some power-related values.

4.8.3 The `display` Command

If the cluster administrator has recorded the device positions in the rack, then the `display` command in `rack` mode is useful for seeing the position of where power shelves are located in the rack:

Example

```
[basecm11->rack]% display |less -R
...
            A05                       B05

48                        48
47                        47
46                        46
45                        45
44                        44
43                        43
42   a05-p1-pwr-08        42   B05-P1-PWR-08
41   a05-p1-pwr-07        41   B05-P1-PWR-07
40   a05-p1-pwr-06        40   B05-P1-PWR-06
39   a05-p1-pwr-05        39   B05-P1-PWR-05
38                        38
37   a05-p1-dgx-01-c18    37   b05-p1-dgx-05-c18
36   a05-p1-dgx-01-c17    36   b05-p1-dgx-05-c17
35   a05-p1-dgx-01-c16    35   b05-p1-dgx-05-c16
34   a05-p1-dgx-01-c15    34   b05-p1-dgx-05-c15
33   a05-p1-dgx-01-c14    33   b05-p1-dgx-05-c14
32   a05-p1-dgx-01-c13    32   b05-p1-dgx-05-c13
31   a05-p1-dgx-01-c12    31   b05-p1-dgx-05-c12
30   a05-p1-dgx-01-c11    30   b05-p1-dgx-05-c11
29   a05-p1-dgx-01-c10    29   b05-p1-dgx-05-c10
28   a05-p1-dgx-01-c09    28   b05-p1-dgx-05-c09
27   a05-p1-nvsw-09       27   B05-P1-NVSW-09
26   a05-p1-nvsw-08       26   B05-P1-NVSW-08
25   a05-p1-nvsw-07       25   B05-P1-NVSW-07
24   a05-p1-nvsw-06       24   B05-P1-NVSW-06
23   a05-p1-nvsw-05       23   B05-P1-NVSW-05
22   a05-p1-nvsw-04       22   B05-P1-NVSW-04
21   a05-p1-nvsw-03       21   B05-P1-NVSW-03
20   a05-p1-nvsw-02       20   B05-P1-NVSW-02
19   a05-p1-nvsw-01       19   B05-P1-NVSW-01
18   a05-p1-dgx-01-c08    18   b05-p1-dgx-05-c08
17   a05-p1-dgx-01-c07    17   b05-p1-dgx-05-c07
16   a05-p1-dgx-01-c06    16   b05-p1-dgx-05-c06
15   a05-p1-dgx-01-c05    15   b05-p1-dgx-05-c05
14   a05-p1-dgx-01-c04    14   b05-p1-dgx-05-c04
13   a05-p1-dgx-01-c03    13   b05-p1-dgx-05-c03
12   a05-p1-dgx-01-c02    12   b05-p1-dgx-05-c02
11   a05-p1-dgx-01-c01    11   b05-p1-dgx-05-c01
10                        10
```

```
09   a05-p1-pwr-04       09   B05-P1-PWR-04
08   a05-p1-pwr-03       08   B05-P1-PWR-03
07   a05-p1-pwr-02       07   B05-P1-PWR-02
06   a05-p1-pwr-01       06   B05-P1-PWR-01
05                       05
04                       04
03                       03
02                       02
01                       01
```

5 NVIDIA Autonomous Hardware Recovery

5.1 Introduction

NVIDIA Mission Control can provide autonomous hardware recovery. This capability can be installed on a cluster managed by BCM.

It is designed to allow a user to carry out:

- automated baseline tests to validate hardware
- automated health checks to detect hardware failures at the track, rack, node, and system levels
- automated break/fix workflows (guided recovery on failure)

Autonomous hardware recovery can be integrated with an NVIDIA Base Command Manager (BCM) that is running a Kubernetes instance. The integration is carried out with the BCM script `nvidia-mission-control-setup` (section 5.3).

The script deploys the autonomous hardware recovery *agents* on all nodes. These agents communicate with the autonomous hardware recovery *backends* that are deployed within Kubernetes worker nodes.

BCM integration allows API access to the agents and provides a GUI to carry out *runbooks*.

More details on autonomous hardware recovery can be found in its dedicated documentation.

The scope of this chapter of the manual is to give guidance on autonomous hardware recovery deployment, and how to access its API and GUI.

5.2 Prerequisites To Install Autonomous Hardware Recovery

Before running the `nvidia-mission-control-setup` script to install autonomous hardware recovery, the following prerequisites must be met:

- A Kubernetes instance that is integrated with BCM (section 4.2 of the *Containerization Manual*) must be running.
- The Kubernetes instance must have the Kubernetes NGINX Ingress controller add-on for route-based access control installed.
- At the time of writing of this section (May 2025), Kyverno must not be installed on the Kubernetes instance, or the installation fails. This is expected to change later on.

As a rough guide to the physical hardware requirements for autonomous hardware recovery, it is expected that a cluster with hundreds of nodes should use a backend node with at least 8 CPUs per node, at least 2 local hard drives, and at least 16 GB RAM per node. More precise physical hardware requirements are best determined with the help of NVIDIA engineers.

5.3 Installation Of NVIDIA Autonomous Hardware Recovery Using `nvidia-mission-control-setup`

Running `nvidia-mission-control-setup` as the root user starts up the TUI installation (figure 5.1):

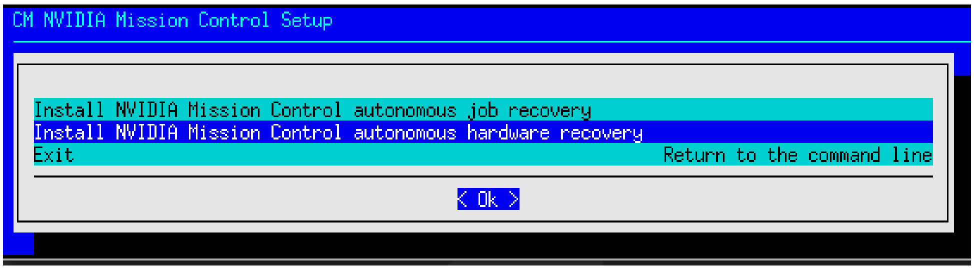

Figure 5.1: Selection from `cm-mission-control-setup` of autonomous hardware recovery

Selecting `NVIDIA Mission Control autonomous hardware recovery` brings up the Helm registry credentials screen (figure 5.2):

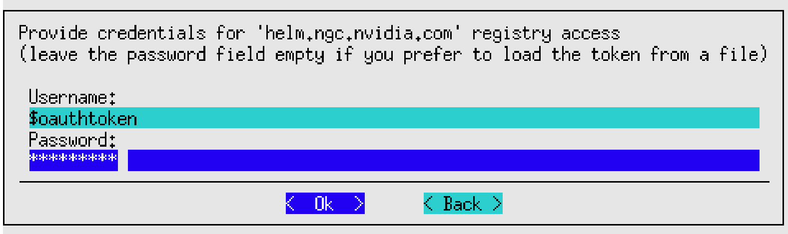

Figure 5.2: Setting Helm credentials for autonomous hardware recovery

If the Helm registry password is accepted, then the NVCR registry credentials screen (figure 5.3) comes up:

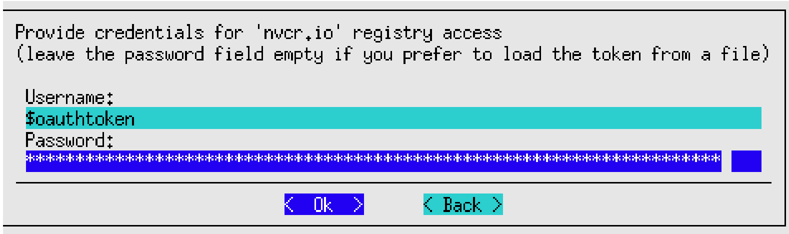

Figure 5.3: Setting NVCR credentials for autonomous hardware recovery

If the NVCR registry password is accepted, then the API registry credentials screen (figure 5.4) comes up:

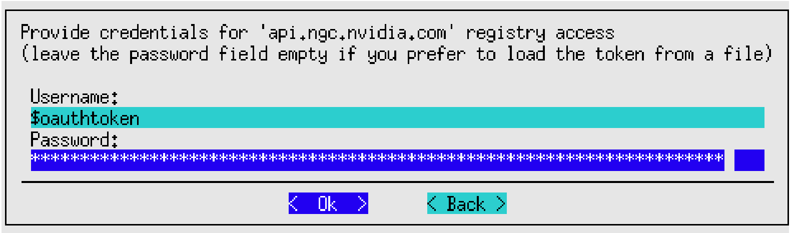

Figure 5.4: Setting API credentials for autonomous hardware recovery

If the password for API registry access is accepted, then a TLS certificates screen (figure 5.5) comes up:

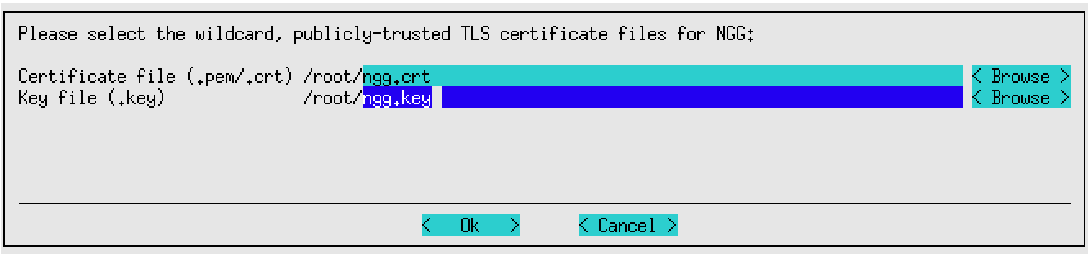

Figure 5.5: Setting TLS certificate files for autonomous hardware recovery

After the path to the TLS certificates are entered, a backend node selection screen (figure 5.6) comes up:

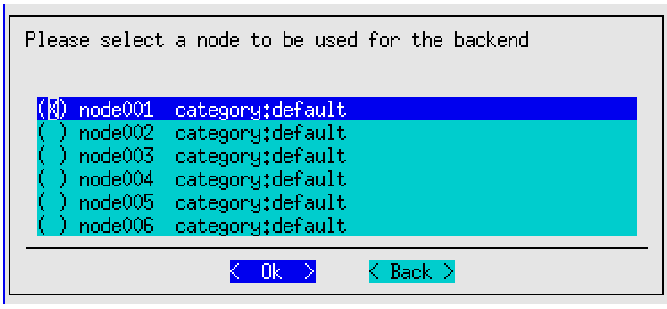

Figure 5.6: Setting a backend node for autonomous hardware recovery

After a backend node is selected, a failover node selection screen (figure 5.7) comes up. Selection is optional:

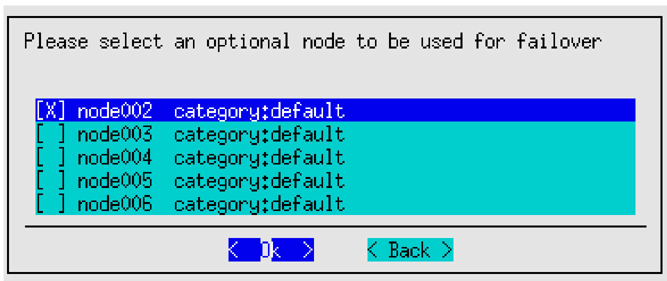

Figure 5.7: Setting an optional failover node for autonomous hardware recovery

After that an agent category selection screen (figure 5.8) comes up:

Figure 5.8: Setting an agent category for autonomous hardware recovery

After that a backend customizations screen (figure 5.9) comes up:

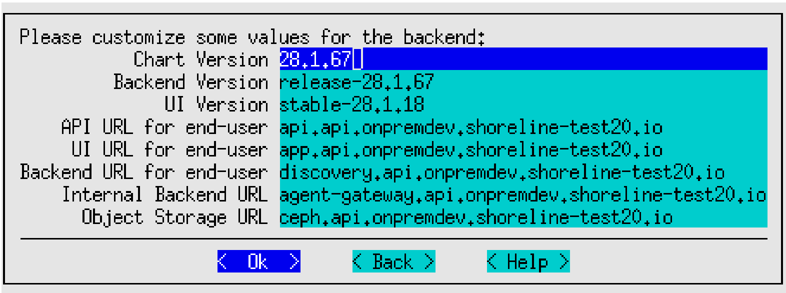

Figure 5.9: Setting backend customizations for autonomous hardware recovery

The endpoints set in the backend should resolve either to the active head node, or to the backend pod node.

A warnings screen appears if the domains cannot be resolved (figure 5.10):

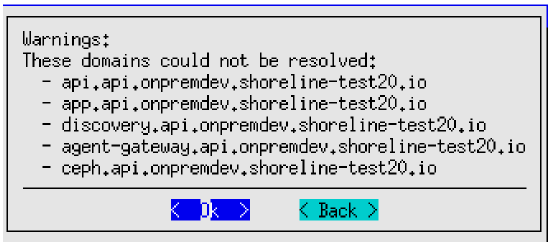

Figure 5.10: Configuration warnings for autonomous hardware recovery

After that a storage customizations screen for the backend appears (figure 5.11) with suggested default values:

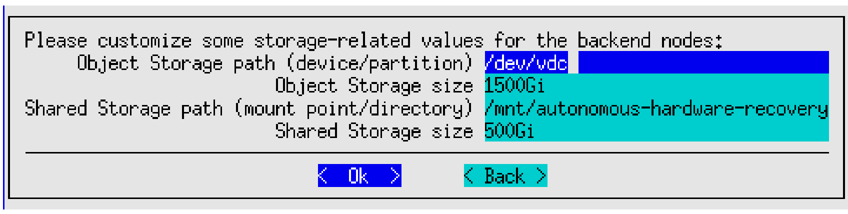

Figure 5.11: Storage customizations for the backend for autonomous hardware recovery

The TUI then asks if monitoring should be set up.

If monitoring is to be configured, then values for the Prometheus and Grafana Tempo endpoints, and for the Prometheus and Grafana Tempo API keys must be entered. The HTTP user ID and password for Loki logging must also be entered (figure 5.12):

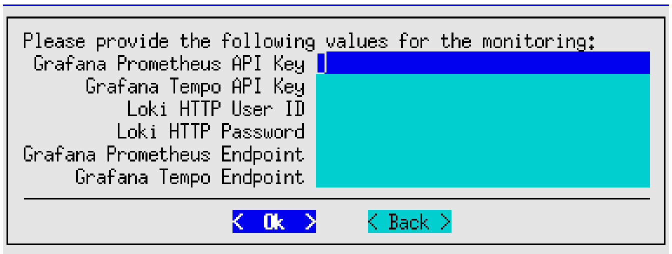

Figure 5.12: Monitoring values configuration for autonomous hardware recovery

The agent version should be set (figure 5.13):

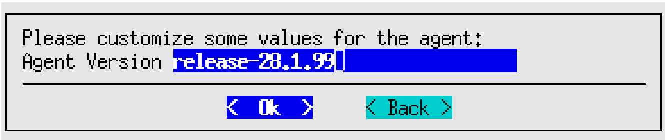

Figure 5.13: Agent version setting for autonomous hardware recovery

The configuration from the wizard can be saved and the configuration can be deployed (figure 5.14):

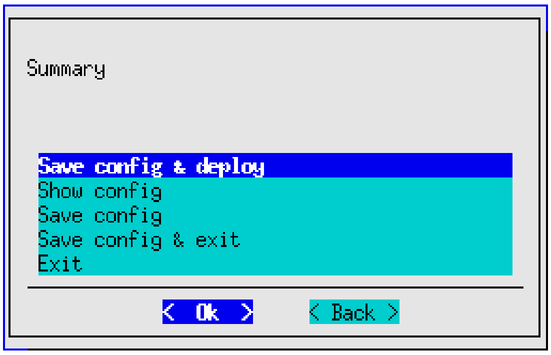

Figure 5.14: Saving the configuration and deploying it for autonomous hardware recovery

The API (described in `https://ssapi.shorelinesoftware.net/`) can be tested to verify that is working.

6 NVIDIA Mission Control DGX GB200 Measurables

Measurables are metrics, health checks, and enummetrics (appendix G of the *Administrator Manual*).

This section describes the extra DGX GB200 measurables and their parameters that may not have been described in appendix G of the *Administrator Manual*. The DGX GB200 measurables are grouped in this chapter according to the following themes:

- circuit-related metrics (section 6.1)
- leak detection measurables (section 6.2)
- NVLink measurables (section 6.3)
- power shelf measurables (section 6.4)
- cooling distribution unit metrics (section 6.5)
- GPU measurables (section 6.6)
- Prometheus metrics (section 6.7)
- Redfish metrics (section 6.8)
- Redfish enums (section 6.9)
- health checks (section 6.10)

6.1 Circuit-Related DGX GB200 Metrics

Electrical power in the data center is supplied to the DGX GB200 racks from a circuit that is typically from a remote power panel (RPP) or an overhead busway.

Circuit-related metrics for the DGX GB200 platform are shown in table 6.1.

Table 6.1: Circuit-related DGX GB200 metrics

DGX GB200 Circuit Metric	Parameter	Description
CircuitCurrent		
CircuitPhaseCurrent	phase=1	
CircuitPhaseCurrent	phase=2	
CircuitPhaseCurrent	phase=3	
CircuitPower		
RackCircuitCurrent*	circuit=RPP-B12-3	
RackCircuitCurrentLimit*	circuit=RPP-B12-3	
RackCircuitPhaseCurrent*,**	circuit=RPP-B12-3;phase=3	
RackCircuitPower*	circuit=RPP-B12-3	
TotalCircuitCurrent		Total
TotalCircuitCurrentLimit		Total
TotalCircuitPhaseCurrent		Total
TotalCircuitPhaseCurrent	phase=1	Total
TotalCircuitPhaseCurrent	phase=2	Total
TotalCircuitPhaseCurrent	phase=3	Total
TotalCircuitPower		Total

* The parameter set here depends on the listed power circuit name (section 4.7)
** The phase value can be 1, 2, or 3

6.2 Leak Detection DGX GB200 Measurables

Liquid cooling is used in the DGX GB200. The ability to detect leaking coolant is an important enough concern that there are several measurables associated with it.

Leak detection measurables for the DGX GB200 platform are shown in table 6.2:

Table 6.2: Leak detection DGX GB200 measurables

DGX GB200 leak measurable	Parameter	Description
LeakResponseRackElectrical\IsolationStatus		
LeakResponseRackLiquidIso\lationStatus		
LeakSensorFaultRack		
DevicesWithLeaks	Total	Number of devices that have detected a leak
RF_Chassis_0_LeakDetector_0_ColdPlate*		
RF_Chassis_0_LeakDetector_0_Manifold*		
RF_Chassis_0_LeakDetector_0_	state	Enum, not metric

...continues

Table 6.2: Leak detection DGX GB200 measurables...continued

DGX GB200 leak measurable	Parameter	Description
ColdPlate*		
RF_Chassis_0_LeakDetector_0_Manifold*	state	Enum, not metric
RF_LD_LeakDetection		Enum
RF_LeakDetector	Chassis_0_LeakDetector_0_ColdPlate*	Enum
RF_LeakDetector	Chassis_0_LeakDetector_0_Manifold*	Enum

* The substring LeakDetector_0 can also take the value LeakDetector_1

6.3 DGX GB200 NVLink Measurables

NVLink-related measurables for the DGX GB200 platform are shown in table 6.3.

The measurables without the RF_ prefix are provided by the ClusterTotal data producer. The measurables with the RF_ prefix are provided via Redfish.

BCM picks up the Redfish measurables and their descriptions automatically. Many measurables still have no associated descriptions. For now, the names of the measurables can be a guide on what they are about.

Table 6.3: DGX GB200 NVLink measurables

DGX GB200 NVLink measurables	Parameter	Description
NVLinkSwitchesClosed		Number of NVLink switches not marked as UP or DOWN
NVLinkSwitchesDown		Number of NVLink switches marked as DOWN
NVLinkSwitchesTotal		Total number of NVLink switches
NVLinkSwitchesUp		Number of NVLink switches marked as UP
RF_NVLink_Port_Nvidia_BitErrorRate*		
RF_NVLink_Port_Nvidia_LinkDowned*		
RF_NVLink_Port_Nvidia_RXNoProtocol*		
RF_NVLink_Port_Nvidia_TXNoProtocol*		
RF_NVLink_Port_Nvidia_TXWait*		
RF_NVLink_Port_RX*		
RF_NVLink_Port_RXFrames_Networking*		
RF_NVLink_Port_RX_Total		
RF_NVLink_Port_TX*		
RF_NVLink_Port_TXFrames_Networking*		

...continues

Table 6.3: DGX GB200 NVLink measurables...continued

DGX GB200 NVLink measurables	Parameter	Description
RF_NVLink_Port_TX_Total		
RF_NVLink_ResourceHealthRollup_Status		
RF_NVLink_ResourceHealth_Status		
RF_NVLink_ResourceState_Status*		
RF_NVLink_Resource_CurrentSpeed*		
RF_NVLink_Resource_LinkState*		
RF_NVLink_Resource_LinkStatus*		
RF_NVLink_Resource_MaxSpeed*		
RF_NVLink_Resource_Nvidia_RXWidth*		
RF_NVLink_Resource_Nvidia_TXWidth*		

* The parameter here is one of 17 ports from the node to the switch.
It can take a value of acp0 to acp17

6.4 DGX GB200 Power Shelf Measurables

The list of power shelf measurables on a DGX GB200 platform are displayed in table 6.4.

These are produced by the `redfish_power_shelf`, `ClusterTotal`, and `AggregatePowerShelf` data producers.

Table 6.4: Power shelf measurables

Power Shelf Measurable	Parameter	Description
AveragePowerShelfTemperature*		Average power shelf temperature
PowerShelvesClosed**		Number of power shelves not marked as UP or DOWN
PowerShelvesDown**		Number of power shelves marked as DOWN
PowerShelvesTotal**		Total number of power shelves
PowerShelvesUp**		Number of power shelves marked as UP
RF_Active_PSU		Number of PSUs of power shelf active
RF_Health_PSU‡		Are all PSUs of power shelf active?
RF_PowerShelf_Status†	1	Status of power shelf PSU 1
RF_PowerShelf_Status	total	Status of power shelf PSU, total
RF_Power_Supply_Energy†	1	Energy consumption of PSU 1 since boot
RF_Power_Supply_Energy	total	Energy consumption of all PSUs of power shelf since boot, total
RF_Power_Supply_FanSpeed†	1	Fan speed of PSU 1
RF_Power_Supply_FanSpeed	average	Fan speed of all PSUs of power shelf, averaged
RF_Power_Supply_Id†	1	
RF_Power_Supply_InputCurrent†	1	Input current for PSU 1
RF_Power_Supply_InputCurrent	total	Input current for all PSUs of power shelf, total

...continues

Table 6.4: Power shelf measurables...continued

Power Shelf Measurable	Parameter	Description
RF_Power_Supply_InputPower†	1	Input power for PSU 1
RF_Power_Supply_InputPower	total	Input power for all PSUs of power shelf, total
RF_Power_Supply_InputVoltage†	1	Input voltage for PSU 1
RF_Power_Supply_LifetimeReading_Energy†	1	Input lifetime reading for energy for PSU 1
RF_Power_Supply_Name†	1	
RF_Power_Supply_OutputPower†	1	Output power for PSU 1
RF_Power_Supply_OutputPower	total	Output power for all PSUs of power shelf, total
RF_Power_Supply_PowerFactor_Input†	1	
RF_Power_Supply_RailCurrent†	1	Rail Current for PSU 1
RF_Power_Supply_RailCurrent	total	Rail Current for all PSUs of power shelf, total
RF_Power_Supply_RailPower†	1	Rail power for PSU 1
RF_Power_Supply_RailPower	total	Rail power for all PSUs of power shelf, total
RF_Power_Supply_RailVoltage†	1	Rail voltage for PSUs 1
RF_Power_Supply_Temperature†	1	Temperature for PSU 1
RF_Power_Supply_Temperature	average	Temperature for all PSUs of power shelf, averaged
RF_Power_Supply_code_error†	1	Code error for PSU 1
RF_Power_Supply_message_error†	1	Message error for PSU 1
RF_Summary_Metrics_Id		
RF_Summary_Metrics_Name		
RF_Summary_Metrics_Power		Power consumption (Watts)
RF_Summary_Metrics_Temperature		Temperature (Celsius)
RF_Total_PSU		
TotalPowerShelfActivePSU*		Total power shelf PSUs active
TotalPowerShelfAvailablePSU*		Total power shelf PSUs available
TotalPowerShelfCriticalPSU‡		Are all active PSUs below critical threshold?
TotalPowerShelfDegradedPSU‡		Are all active PSUs below degraded threshold?
TotalPowerShelfHealthyPSU‡		Are all PSUs active?
TotalPowerShelfInputCurrent*		Total power shelf input current
TotalPowerShelfInputPower*		Total power shelf input power
TotalPowerShelfRailCurrent*		Total power shelf rail current
TotalPowerShelfRailPower*		Total power shelf rail power
TotalPowerShelfPSU*		Total power shelf PSU

* From AggregatePowerShelf data producer
** From ClusterTotal data producer
† Takes the PSU number as a parameter. On the DGX GB200 it can be from 1 to 6
‡ Health check

6.5 Cooling Distribution Unit Metrics On The DGX GB200

Metrics for the cooling distribution units (CDUs) of the DGX GB200 platform are shown in table 6.5. The data producers for these are:

- MonitoringSystem for the metrics in the table beginning with the string CDU
- AggregateCDU for the metrics in the table that begin with the strings Average or Total

Table 6.5: DGX GB200 metrics for CDUs

DGX GB200 CDU Metric	Description
AverageCDULiquidDifferentialPressure	Average CDU liquid cooling differential pressure (Pa)
AverageCDULiquidFlow	Average CDU liquid cooling flow (LPM)
AverageCDULiquidReturnTemperature	Average CDU liquid cooling return temperature (°C)
AverageCDULiquidSupplyTemperature	Average CDU liquid cooling supply temperature (°C)
AverageCDULiquidSystemPressure	Average CDU liquid system pressure (Pa)
CDUAvailable	CDU available
CDULiquidDifferentialPressure	CDU liquid differential pressure (Pa)
CDULiquidFlow	CDU liquid flow (LPM)
CDULiquidReturnTemperature	CDU liquid return temperature (°C)
CDULiquidSupplyTemperature	CDU liquid supply temperature (°C)
CDULiquidSystemPressure	CDU liquid system pressure (Pa)
CDUStatus	CDU status (I/s)
TotalCDUAvailable	Total CDU available
TotalCDUStatus	Total CDU status

6.6 GPU Measurables For The DGX GB200

Some basic GPU measurables are covered in appendix G of the *Administrator Manual*.

Extra GPU measurables available on the DGX GB200 platform are:

Table 6.6:

GPU Measurables For The DGX GB200	Parameter	Description
gpu_board_limit_violation*	gpu0	Board violation limit
gpu_board_limit_violation	total	Throttling duration due to board limits
gpu_c2c_link_bandwidth*	gpu0	GPU C2C link bandwidth
gpu_c2c_link_count*	gpu0	GPU C2C link count
gpu_c2c_link_status*	gpu0	GPU C2C link status
gpu_correctable_remapped_rows*	gpu0	Number of remapped rows for correctable errors
gpu_dec_utilization	average	Average GPU decoder utilization
gpu_dec_utilization	gpu0	GPU decoding usage
gpu_ecc_dbe_agg*	gpu0	Total double bit aggregate ECC errors
gpu_ecc_dbe_vol*	gpu0	Total double bit volatile ECC errors
gpu_ecc_sbe_agg*	gpu0	Total single bit aggregate ECC errors

...continues

Table 6.6: …continued

GPU Measurables For The DGX GB200	Parameter	Description
gpu_ecc_sbe_vol*	gpu0	Total single bit volatile ECC errors
gpu_enc_utilization	average	Average GPU encoder utilization
gpu_enc_utilization*	gpu0	GPU encoding usage
gpu_enforced_power_limit*	gpu0	GPU enforced power limit
gpu_enforced_power_profile_mask*	gpu0	Enforced workload power profile mask
gpu_fabric_status*	gpu0	Status of the nvlink link domain (Enum)
gpu_fan_speed*	gpu0	GPU fan speed percentage
gpu_hbm_memory_temperature*	gpu0	High bandwidth memory (GPU memory) temperature
gpu_health_driver*,**	gpu0	Driver-related status
gpu_health_hostengine**		Host engine status
gpu_health_inforom*,**	gpu0	Inforom status
gpu_health_mcu*,**	gpu0	Microcontroller unit status
gpu_health_mem*,**	gpu0	Memory status
gpu_health_nvlink*,**	gpu0	NVLINK system status
gpu_health_nvswitch_fatal*,**	gpu0	NV switch fatal errors
gpu_health_nvswitch_non_fatal*,**	gpu0	NV switch non fatal errors
gpu_health_overall**		Overall system status
gpu_health_overall*,**	gpu0	Overall system status for the specified GPU
gpu_health_pcie*,**	gpu0	PCIe system status
gpu_health_pmu*,**	gpu0	Power management unit status
gpu_health_power*,**	gpu0	Power status
gpu_health_sm*,**	gpu0	Streaming multiprocessor status
gpu_health_thermal*,**	gpu0	Temperature status
gpu_low_util_violation*	gpu0	Low utilization violation limit
gpu_low_util_violation	total	Throttling duration due to low utilization limits
gpu_mem_clock*	gpu0	GPU memory clock
gpu_mem_copy_utilization	average	Average GPU memory copu utilization
gpu_mem_copy_utilization*	gpu0	Percentage of the GPU memory copy used
gpu_mem_free*	gpu0	Amount of GPU free memory
gpu_mem_total*	gpu0	GPU framebuffer size
gpu_mem_total	total	combined GPU total memory
gpu_mem_used*	gpu0	Amount of GPU used memory
gpu_mem_used	total	combined GPU memory usage
gpu_mem_utilization	average	Average GPU memory utilization
gpu_mem_utilization*	gpu0	Percentage of GPU memory used
gpu_memory_temp	average	Average GPU memory temperature
gpu_memory_temp*	gpu0	GPU memory temperature
gpu_nvlink_crc_data_errors*	gpu0	Total nvlink data CRC errors over all lanes
gpu_nvlink_crc_flit_errors*	gpu0	Total nvlink flit CRC errors over all lanes
gpu_nvlink_total_bandwidth*	gpu0	Total nvlink bandwidth used
gpu_nvlink_total_bandwidth	total	combined GPU nvlink bandwidth

…continues

Table 6.6: ...continued

GPU Measurables For The DGX GB200	Parameter	Description
gpu_performance_state*	gpu0	GPU performance state (0=highest)
gpu_power_management_limit*	gpu0	GPU power management limit
gpu_power_usage*	gpu0	GPU power usage
gpu_power_usage	total	combined GPU power usage
gpu_power_violation*	gpu0	Throttling duration due to power constraints
gpu_power_violation	total	Throttling duration due to power constraints
gpu_reliability_violation*	gpu0	Reliability violation limit
gpu_reliability_violation	total	Throttling duration due to reliability limits
gpu_requested_power_profile_mask*	gpu0	Requested workload power profile mask
gpu_row_remap_failure*	gpu0	Remapping of rows failures
gpu_shutdown_temp*	gpu0	GPU shutdown temperature
gpu_slowdown_temp*	gpu0	GPU slowdown temperature
gpu_sm_clock*	gpu0	GPU shader multiprocessor clock
gpu_sync_boost_violation*	gpu0	Throttling duration due to sync boost constraints
gpu_sync_boost_violation	total	Throttling duration due to sync boost constraints
gpu_temperature	average	Average GPU temperature
gpu_temperature*	gpu0	GPU temperature
gpu_thermal_violation*	gpu0	Throttling duration due to thermal constraints
gpu_thermal_violation	total	Throttling duration due to thermal constraints
gpu_total_app_clocks_violation*	gpu0	App clock violation limit
gpu_total_app_clocks_violation	total	Throttling duration due to application clocks limits
gpu_total_base_clocks_violation*	gpu0	Base clock violation limit
gpu_total_base_clocks_violation	total	Throttling duration due to base clocks limits
gpu_uncorrectable_remapped_rows*	gpu0	Number of remapped rows for uncorrectable errors
gpu_utilization	average	Average GPU utilization
gpu_utilization*	gpu0	GPU utilization percentage
gpu_xid_error*	gpu0	The value is the specific XID error

* parameter can be gpu0, gpu1...
** health check

6.7 Prometheus Metrics For The DGX GB200

Extra Prometheus metrics available on the DGX GB200 platform are shown in table 6.7.

The data producer for the metrics with a prefix of `job_metadata_` is `JobMetadataSampler`. The remaining metrics are produced by `JobSampler`.

BCM picks up the measurables and their descriptions automatically. Many measurables still have no associated descriptions. For now, the names of the measurables can be a guide on what they are about.

Table 6.7:

Prometheus Metrics For The DGX GB200	Description
job_cpu_power_limit	CPU power limit
job_cpu_power_usage	
job_cpu_temperature	
job_cpuacct_usage_seconds	Total CPU time consumed by processes
job_gpu_board_limit_violation	
job_gpu_c2c_link_count	
job_gpu_c2c_link_status	
job_gpu_correctable_remapped_rows	
job_gpu_dec_utilization	
job_gpu_ecc_dbe_agg	
job_gpu_ecc_dbe_vol	
job_gpu_ecc_sbe_agg	
job_gpu_ecc_sbe_vol	
job_gpu_enc_utilization	
job_gpu_enforced_power_limit	
job_gpu_enforced_power_profile_mask	
job_gpu_fabric_status	
job_gpu_fan_speed	
job_gpu_low_util_violation	
job_gpu_mem_clock	
job_gpu_mem_copy_utilization	
job_gpu_mem_free	
job_gpu_mem_total	
job_gpu_mem_used	
job_gpu_mem_utilization	
job_gpu_memory_temp	
job_gpu_nvlink_crc_data_errors	
job_gpu_nvlink_crc_flit_errors	
job_gpu_nvlink_total_bandwidth	Total nvlink bandwidth used
job_gpu_performance_state	
job_gpu_power_management_limit	
job_gpu_power_usage	
job_gpu_power_violation	
job_gpu_reliability_violation	
job_gpu_requested_power_profile_mask	
job_gpu_row_remap_failure	
job_gpu_shutdown_temp	
job_gpu_slowdown_temp	

...continues

Table 6.7: …continued

Prometheus Metrics For The DGX GB200	Description
job_gpu_sm_clock	
job_gpu_sync_boost_violation	
job_gpu_temperature	
job_gpu_thermal_violation	
job_gpu_total_app_clocks_violation	
job_gpu_total_base_clocks_violation	
job_gpu_uncorrectable_remapped_rows	
job_gpu_utilization	
job_gpu_wasted	The amount of the allocated GPUs that is not used
job_gpu_workload_power_profile	GPU workload power profile status
job_gpu_xid_error	
job_memory_cache_bytes	Page cache, including tmpfs (shmem)
job_memory_failcnt	Number of times that the memory limit has reached the value set in memory.limit_in_bytes
job_memory_mapped_file_bytes	Size of memory-mapped mapped files, including tmpfs
job_memory_memsw_failcnt	Number of times that the memory plus swap space limit has reached the value set in memory.memsw.limit_in_bytes
job_memory_memsw_usage_bytes	Maximum amount of memory and swap space used by processes
job_memory_rss_bytes	Anonymous and swap cache, not including tmpfs
job_memory_swap_bytes	Swap memory usage
job_memory_usage_bytes	Total memory usage processes
job_metadata_allocated_cpu_cores	
job_metadata_allocated_gpus	
job_metadata_is_running	
job_metadata_is_waiting	
job_metadata_num_cpus	
job_metadata_num_nodes	
job_metadata_pending_jobs	
job_metadata_running_jobs	
job_metadata_running_seconds	
job_metadata_waiting_seconds	

6.8 Redfish Metrics For The DGX GB200

Redfish monitoring can be carried out for devices with a Redfish subscription. The `redfishsubscriptions` command can be used to check if a device has a subscription.

Example

```
[basecm11->device]% redfishsubscriptions node001
hostname      ...initialized ip          kind     port     ssl      subscribed...
------------- ...------------ ----------- -------- -------- -------- ----------
node001       ...true         7.241.2.13  GB200    443      true     true
```

Redfish firmware metrics that may be available on the DGX GB200 platform are shown in table 6.8.

As is usual with hardware, the metrics can change according to hardware and especially with firmware changes. The table should therefore be regarded as an indication of what metrics are available, and not a list of what metrics must exist.

BCM picks up the metrics and their descriptions automatically. Many metrics have no associated descriptions. For now, the names of the metrics can be a guide on what they are about.

The list of Redfish metrics available can be found by running:

Example

```
basecm11->monitoring->measurable]% list metric | grep RF_
```

Table 6.8:

Redfish Metrics For The DGX GB200	Description
RF_AOC_NIC1Temp[1]	AOC NIC 1 Temperature
RF_AOC_PORTATemp	AOC PORT A Temperature
RF_AOC_PORTBTemp	AOC PORT B Temperature
RF_AOC_PORTCTemp	AOC PORT C Temperature
RF_Active_PSU[2]	
RF_Average_Temp	
RF_BF3_Slot_1_NIC_Temp_0	BF3 Slot 1 NIC Temperature 0
RF_BF3_Slot_2_NIC_Temp_0	BF3 Slot 2 NIC Temperature 0
RF_BF3_Slot_1_Port_0_Temp	BF3 Slot 1 Port 0 Temperature
RF_BF3_Slot_1_Port_1_Temp	BF3 Slot 1 Port 1 Temperature
RF_BF3_Slot_2_Port_0_Temp	BF3 Slot 2 Port 0 Temperature
RF_BF3_Slot_2_Port_1_Temp	BF3 Slot 2 Port 1 Temperature
RF_BF3_Slot_1_Temp	BF3 Slot 1 Temperature
RF_BF3_Slot_2_Temp	BF3 Slot 2 Temperature
RF_BMCTemp	BMC Temperature
RF_BMC_0_DCSCM_Temp_0	BMC 0 DCSCM Temperature 0
RF_C2CPU1Temp	C2 CPU1 Temperature
RF_C2CPU2Temp	C2 CPU2 Temperature
RF_CPU1RearTemp	CPU1 Rear Temperature
RF_CPU1Temp	CPU1 Temperature
RF_CPU1_VCCHV	CPU1_VCCHV

...continues

Table 6.8: ...continued

Redfish Metrics For The DGX GB200	Description
RF_CPU1_VCCIN	CPU1_VCCIN
RF_CPU1_VCCON	CPU1_VCCON
RF_CPU1_VRMHVTemp	CPU1_VRMHV Temperature
RF_CPU1_VRMINTemp	CPU1_VRMIN Temperature
RF_CPU1_VRMONTemp	CPU1_VRMON Temperature
RF_CPU2RearTemp	CPU2 Rear Temperature
RF_CPU2Temp	CPU2 Temperature
RF_CPU2_VCCHV	CPU2_VCCHV
RF_CPU2_VCCIN	CPU2_VCCIN
RF_CPU2_VCCON	CPU2_VCCON
RF_CPU2_VRMHVTemp	CPU2_VRMHV Temperature
RF_CPU2_VRMINTemp	CPU2_VRMIN Temperature
RF_CPU2_VRMONTemp	CPU2_VRMON Temperature
RF_CPU_0_CoreUtil_0[3]	CPU_0_CoreUtil_0
In the preceding row, the substring CoreUtil_0 *can also take values of* CoreUtil_1, CoreUtil_2..., up to CoreUtil_71	
RF_CPU_0_CpuFreq_0[3]	CPU_0_CpuFreq_0
RF_CPU_0_Energy_0[3]	CPU_0_Energy_0
RF_CPU_0_Power_0[3]	CPU_0_Power_0
RF_CPU_0_Processor_code_error[3]	
RF_CPU_0_Processor_message_error[3]	
RF_CPU_0_Processor_Metrics__Nvidia_AccumulatedGPUContextUtilizationDuration[3]	
RF_CPU_0_Processor_Metrics__Nvidia_AccumulatedSMUtilizationDuration[3]	
RF_CPU_0_Processor_Metrics__Nvidia_EDPViolationState[3]	
RF_CPU_0_Processor_Metrics__Nvidia_GlobalSoftwareViolationThrottleDuration[3]	
RF_CPU_0_Processor_Metrics__Nvidia_HardwareViolationThrottleDuration[3]	
RF_CPU_0_Processor_Metrics__Nvidia_MemoryPageRetirement[3]	
RF_CPU_0_Processor_Metrics__Nvidia_MemorySpareChannelPresence[3]	
RF_CPU_0_Processor_Metrics__Nvidia_PCIeRXBytes[3]	
RF_CPU_0_Processor_Metrics__Nvidia_PCIeTXBytes[3]	
RF_CPU_0_Processor_Metrics__Nvidia_PerformanceState[3]	
RF_CPU_0_Processor_Metrics__Nvidia_	

...continues

Table 6.8: ...continued

Redfish Metrics For The DGX GB200	Description
PowerBreakPerformanceState[3]	
RF_CPU_0_Processor_Metrics_PowerLimitThrottleDuration[3]	
RF_CPU_0_Processor_Metrics_ThermalLimitThrottleDuration[3]	
RF_CPU_0_TempAvg_0[3]	CPU_0_TempAvg_0
RF_CPU_0_TempLimit_0[3]	CPU_0_TempLimit_0
RF_Chassis_0_FAN_1_FRONT[4]	Chassis 0 FAN 1 FRONT
RF_Chassis_0_FAN_1_PWM[4]	Chassis 0 FAN 1 PWM
RF_Chassis_0_FAN_1_REAR[4]	Chassis 0 FAN 1 REAR
RF_Chassis_0_Front_IO_Temp_0	Chassis 0 Front IO Temp 0
RF_Chassis_0_LeakDetector_0_ColdPlate[5]	Chassis 0 LeakDetector 0 ColdPlate
RF_Chassis_0_LeakDetector_0_Manifold[5]	Chassis 0 LeakDetector 0 Manifold
RF_Chassis_0_LeakDetector_1_ColdPlate[5]	Chassis 0 LeakDetector 1 ColdPlate
RF_Chassis_0_LeakDetector_1_Manifold[5]	Chassis 0 LeakDetector 1 Manifold
RF_Chassis_0_TotalHSC_Power_0	Chassis 0 TotalHSC Power 0
RF_Chassis_Intru_Reading	
RF_Chassis_Intru_ReadingRangeMax	
RF_Chassis_Intru_ReadingRangeMin	
RF_DIMM_SlotPartLocationContext_Location	
RF_DIMM_SlotState_Status	
RF_DIMM_Slot_BusWidthBits	
RF_DIMM_Slot_Capacity	
RF_DIMM_Slotcode_error	
RF_DIMM_Slotmessage_error	
RF_DIMM_Slot_OperatingSpeed	
RF_DIMM_Slot_DataWidthBits	
RF_DIMM_Slot_FirmwareRevision	
RF_DIMM_Slot_Nvidia_RowRemappingFailed	
RF_DIMM_Slot_Nvidia_RowRemappingPending	
RF_DIMM_Slot_OperatingSpeed	
RF_DIMM_Slot_Rank	
RF_Energy_0	Energy_0
RF_FPGATemp	FPGA Temp

...continues

Table 6.8: …continued

Redfish Metrics For The DGX GB200	Description
RF_FPGA_ARTTemp	FPGA_ART Temp
RF_GPU_0_DRAM_0_Memory_MetricsCorrectableE\ CCErrorCount_LifeTime[6]	
RF_GPU_0_DRAM_0_Memory_MetricsUncorrectabl\ eECCErrorCount_LifeTime[6]	
RF_GPU_0_DRAM_0_Memory_Metrics_ BandwidthUtilization[6]	
RF_GPU_0_DRAM_0_Memory_Metrics_ CapacityUtilizationPercent[6]	
RF_GPU_0_DRAM_0_Memory_Metrics_ OperatingSpeed[6]	
RF_GPU_0_Processor_MetricsCorrectableErrorCo\ unt_PCIeErrors[6]	
RF_GPU_0_Processor_MetricsFatalErrorCount_ PCIeErrors[6]	
RF_GPU_0_Processor_MetricsL0ToRecoveryCount_ PCIeErrors[6]	
RF_GPU_0_Processor_MetricsNAKReceivedCount_ PCIeErrors[6]	
RF_GPU_0_Processor_MetricsNAKSentCount_ PCIeErrors[6]	
RF_GPU_0_Processor_MetricsNonFatalErrorCount_ PCIeErrors[6]	
RF_GPU_0_Processor_MetricsReplayCount_ PCIeErrors[6]	
RF_GPU_0_Processor_MetricsReplayRolloverCoun\ t_PCIeErrors[6]	
RF_GPU_0_Processor_MetricsUnsupportedRequest\ Count_PCIeErrors[6]	
RF_GPU_0_Processor_Metrics_ BandwidthUtilization[6]	
RF_GPU_0_Processor_Metrics_ CoreVoltage[6]	
RF_GPU_0_Processor_Metrics_ LifeTime_CorrectableECCError[6]	
RF_GPU_0_Processor_Metrics_ LifeTime_UncorrectableECCError[6]	
RF_GPU_0_Processor_Metrics_Nvidia_ AccumulatedGPUContextUtilizationDuration[6]	
RF_GPU_0_Processor_Metrics_Nvidia_ AccumulatedSMUtilizationDuration[6]	
RF_GPU_0_Processor_Metrics_Nvidia_ DMMAUtilizationPercent[6]	
RF_GPU_0_Processor_Metrics_Nvidia_ FP16ActivityPercent[6]	

…continues

Table 6.8: ...continued

Redfish Metrics For The DGX GB200	Description
RF_GPU_0_Processor_Metrics_Nvidia_FP32ActivityPercent[6]	
RF_GPU_0_Processor_Metrics_Nvidia_FP64ActivityPercent[6]	
RF_GPU_0_Processor_Metrics_Nvidia_GlobalSoftwareViolationThrottleDuration[6]	
RF_GPU_0_Processor_Metrics_Nvidia_GraphicsEngineActivityPercent[6]	
RF_GPU_0_Processor_Metrics_Nvidia_HMMAUtilizationPercent[6]	
RF_GPU_0_Processor_Metrics_Nvidia_HardwareViolationThrottleDuration[6]	
RF_GPU_0_Processor_Metrics_Nvidia_IMMAUtilizationPercent[6]	
RF_GPU_0_Processor_Metrics_Nvidia_IntegerActivityUtilizationPercent[6]	
RF_GPU_0_Processor_Metrics_Nvidia_NVDecUtilizationPercent[6]	
RF_GPU_0_Processor_Metrics_Nvidia_NVJpgUtilizationPercent[6]	
RF_GPU_0_Processor_Metrics_NVLinkDataRxBandwidth[6]	
RF_GPU_0_Processor_Metrics_Nvidia_NVLinkDataTxBandwidth[6]	
RF_GPU_0_Processor_Metrics_Nvidia_NVLinkRawRxBandwidth[6]	
RF_GPU_0_Processor_Metrics_Nvidia_NVLinkRawTxBandwidth[6]	
RF_GPU_0_Processor_Metrics_Nvidia_NVOfaUtilizationPercent[6]	
RF_GPU_0_Processor_Metrics_Nvidia_PCIeRXBytes[6]	
RF_GPU_0_Processor_Metrics_Nvidia_PCIeRawRxBandwidth[6]	
RF_GPU_0_Processor_Metrics_Nvidia_PCIeRawTxBandwidth[6]	
RF_GPU_0_Processor_Metrics_Nvidia_PCIeTXBytes[6]	
RF_GPU_0_Processor_Metrics_Nvidia_SMActivityPercent[6]	
RF_GPU_0_Processor_Metrics_Nvidia_SMOccupancyPercent[6]	
RF_GPU_0_Processor_Metrics_Nvidia_SMUtilizationPercent[6]	
RF_GPU_0_Processor_Metrics_Nvidia_	

...continues

Table 6.8: ...continued

Redfish Metrics For The DGX GB200	Description
SRAMECCErrorThresholdExceeded[6]	
RF_GPU_0_Processor_Metrics_Nvidia_TensorCoreActivityPercent[6]	
RF_GPU_0_Processor_Metrics_OperatingSpeed[6]	
RF_GPU_0_Processor_Metrics_PowerLimitThrottleDuration[6]	
RF_GPU_0_Processor_Metrics_ThermalLimitThrottleDuration[6]	
RF_GPU_0_Processor_Metricscode_error[6]	
RF_GPU_0_Processor_Metricsmessage_error[6]	
RF_GPU_0_Processor_Reset_Metrics_ConventionalResetEntry[6]	
RF_GPU_0_Processor_Reset_Metrics_ConventionalResetExit[6]	
RF_GPU_0_Processor_Reset_Metrics_FundamentalResetEntry[6]	
RF_GPU_0_Processor_Reset_Metrics_FundamentalResetExit[6]	
RF_GPU_0_Processor_Reset_Metrics_IRoTResetExit[6]	
RF_GPU_0_Processor_Reset_Metrics_PF_FLR_ResetEntry[6]	
RF_GPU_0_Processor_Reset_Metrics_PF_FLR_ResetExit[6]	
RF_HGX_BMC_0_Temp_0	HGX BMC 0 Temp 0
RF_HGX_Baseboard_0CoreCount_ProcessorSummary	
RF_HGX_Baseboard_0Count_ProcessorSummary	
RF_HGX_Baseboard_0Model_ProcessorSummary	
RF_HGX_Baseboard_0State_Status	
RF_HGX_Baseboard_0TotalSystemMemoryGiB_MemorySummary	
RF_HGX_Baseboard_0_Description	
RF_HGX_Baseboard_0_LastResetTime	
RF_HGX_Baseboard_0_Nvidia_ISTModeEnabled	
RF_HGX_Baseboard_0_PowerMode	
RF_HGX_Baseboard_0_PowerState	
RF_HGX_CPU_0State_Status[3]	
RF_HGX_CPU_0_AssetTag[3]	

...continues

Table 6.8: ...continued

Redfish Metrics For The DGX GB200	Description
RF_HGX_CPU_0_PartLocation_ServiceLabel[3]	
RF_HGX_CPU_0_SKU[3]	
RF_HGX_CPU_0code_error[3]	
RF_HGX_CPU_0message_error[3]	
RF_HGX_Chassis_0_TotalGPU_Power_0	HGX Chassis 0 TotalGPU Power 0
RF_HGX_GPU_0State_Status[6]	
RF_HGX_GPU_0_DRAM_0_Power_0[6]	HGX GPU 0 DRAM 0 Power 0
RF_HGX_GPU_0_DRAM_0_Temp_0[6]	HGX GPU 0 DRAM 0 Temp 0
RF_HGX_GPU_0_Energy_0[6]	HGX GPU 0 Energy 0
RF_HGX_GPU_0_MaxPower[6]	
RF_HGX_GPU_0_MinPower[6]	
RF_HGX_GPU_0_Power_0[6]	HGX GPU 0 Power 0
RF_HGX_GPU_0_SKU[6]	
RF_HGX_GPU_0_TEMP_0[6]	HGX GPU 0 TEMP 0
RF_HGX_GPU_0_TEMP_1[6]	HGX GPU 0 TEMP 1
RF_HGX_GPU_0_Voltage_0[6]	HGX GPU 0 Voltage 0
RF_HGX_ProcessorModule_0_Exhaust_Temp_0[12]	HGX ProcessorModule 0 Exhaust Temp 0
RF_HGX_ProcessorModule_0_Inlet_Temp_0[12]	HGX ProcessorModule 0 Inlet Temp 0
RF_HGX_ProcessorModule_0_Inlet_Temp_1[12]	HGX ProcessorModule 0 Inlet Temp 0
RF_HotswapTemp	Hotswap Temp
RF_IO_Board_0_CX7_0_Port_0_Temp	IO Board 0 CX7 0 Port 0 Temp
RF_IO_Board_0_CX7_0_Temp	IO Board 0 CX7 0 Temp
RF_IO_Board_0_CX7_0_Temp_0	IO Board 0 CX7 0 Temp 0
RF_IO_Board_0_CX7_1_Port_0_Temp	IO Board 0 CX7 1 Port 0 Temp
RF_IO_Board_0_CX7_1_Temp	IO Board 0 CX7 1 Temp
RF_IO_Board_0_CX7_1_Temp_0	IO Board 0 CX7 1 Temp 0
RF_IO_Board_1_CX7_0_Port_0_Temp	IO Board 1 CX7 0 Port 0 Temp
RF_IO_Board_1_CX7_0_Temp	IO Board 1 CX7 0 Temp
RF_IO_Board_1_CX7_0_Temp_0	IO Board 1 CX7 0 Temp 0

...continues

Table 6.8: ...continued

Redfish Metrics For The DGX GB200	Description
RF_IO_Board_1_CX7_1_Port_0_Temp	IO Board 1 CX7 1 Port 0 Temp
RF_IO_Board_1_CX7_1_Temp	IO Board 1 CX7 1 Temp
RF_IO_Board_1_CX7_1_Temp_0	IO Board 1 CX7 1 Temp 0
RF_InletTemp	Inlet Temp
RF_InterswitchPort_0_Port_MetricsRXFrames_Networking	
RF_InterswitchPort_0_Port_MetricsRXMulticastFrames_Networking	
RF_InterswitchPort_0_Port_MetricsRXUnicastFrames_Networking	
RF_InterswitchPort_0_Port_MetricsTXDiscards_Networking	
RF_InterswitchPort_0_Port_MetricsTXFrames_Networking	
RF_InterswitchPort_0_Port_MetricsTXMulticastFrames_Networking	
RF_InterswitchPort_0_Port_MetricsTXUnicastFrames_Networking	
RF_InterswitchPort_0_Port_Metrics_Nvidia_BitErrorRate	
RF_InterswitchPort_0_Port_Metrics_Nvidia_LinkDowned	
RF_InterswitchPort_0_Port_Metrics_Nvidia_LinkErrorRecovery	
RF_InterswitchPort_0_Port_Metrics_Nvidia_MalformedPackets	
RF_InterswitchPort_0_Port_Metrics_Nvidia_NeighborMTUDiscards	
RF_InterswitchPort_0_Port_Metrics_Nvidia_QP1Dropped	
RF_InterswitchPort_0_Port_Metrics_Nvidia_RXRemotePhysicalErrors	
RF_InterswitchPort_0_Port_Metrics_Nvidia_RXSwitchRelayErrors	
RF_InterswitchPort_0_Port_Metrics_Nvidia_TXWait	
RF_InterswitchPort_0_Port_Metrics_Nvidia_VL15Dropped	
RF_InterswitchPort_0_Port_Metrics_Nvidia_VL15TXBytes	
RF_InterswitchPort_0_Port_Metrics_Nvidia_VL15TXPackets	

...continues

Table 6.8: …continued

Redfish Metrics For The DGX GB200	Description
RF_InterswitchPort_0_Port_Metrics_RXBytes	
RF_InterswitchPort_0_Port_Metrics_RXErrors	
RF_InterswitchPort_0_Port_Metrics_TXBytes	
RF_InterswitchPort_0_ResourceState_Status	
RF_InterswitchPort_0_Resource_CurrentSpeed	
RF_InterswitchPort_0_Resource_LinkState	
RF_InterswitchPort_0_Resource_LinkStatus	
RF_InterswitchPort_0_Resource_MaxSpeed	
RF_LD_Chassis_0_LeakDetector_0_ColdPlate[5]	Chassis 0 LeakDetector 0 ColdPlate
RF_LD_Chassis_0_LeakDetector_0_Manifold[5]	Chassis 0 LeakDetector 0 Manifold
RF_LD_Chassis_0_LeakDetector_1_ColdPlate[5]	Chassis 0 LeakDetector 1 ColdPlate
RF_LD_Chassis_0_LeakDetector_1_Manifold[5]	Chassis 0 LeakDetector 1 Manifold
RF_LD_LeakDetection[5]	Leak Detection Systems
RF_LeakDetection[5]	Leak Detection Systems
RF_M2_SSD1Temp	M2_SSD1 Temp
RF_M2_SSD2Temp	M2_SSD2 Temp
RF_MB1.8VCC	MB 1.8VCC
RF_MB1.8VSB	MB 1.8VSB
RF_MB12V	MB 12V
RF_MB12VSB	MB 12VSB
RF_MB3.3VCC	MB 3.3VCC
RF_MB3.3VSB	MB 3.3VSB
RF_MB5VCC	MB 5VCC
RF_MB5VSB	MB 5VSB
RF_MGX_NVSwitch_0_TEMP_0	MGX NVSwitch 0 TEMP 0
RF_MGX_NVSwitch_1_TEMP_0	MGX NVSwitch 1 TEMP 0
RF_MemCntl_0_Freq_0	MemCntl_0_Freq_0
RF_MemCntl_1_Freq_0	MemCntl_1_Freq_0
RF_NVLinkManagement_0_Port_MetricsRXFrames_Networking	

…continues

Table 6.8: …continued

Redfish Metrics For The DGX GB200	Description
RF_NVLinkManagement_0_Port_MetricsRXMulticastFrames_Networking	
RF_NVLinkManagement_0_Port_MetricsRXUnicastFrames_Networking	
RF_NVLinkManagement_0_Port_MetricsTXDiscards_Networking	
RF_NVLinkManagement_0_Port_MetricsTXFrames_Networking	
RF_NVLinkManagement_0_Port_MetricsTXMulticastFrames_Networking	
RF_NVLinkManagement_0_Port_MetricsTXUnicastFrames_Networking	
RF_NVLinkManagement_0_Port_Metrics_Nvidia_BitErrorRate	
RF_NVLinkManagement_0_Port_Metrics_Nvidia_LinkDowned	
RF_NVLinkManagement_0_Port_Metrics_Nvidia_LinkErrorRecovery	
RF_NVLinkManagement_0_Port_Metrics_Nvidia_MalformedPackets	
RF_NVLinkManagement_0_Port_Metrics_Nvidia_NeighborMTUDiscards	
RF_NVLinkManagement_0_Port_Metrics_Nvidia_QP1Dropped	
RF_NVLinkManagement_0_Port_Metrics_Nvidia_RXRemotePhysicalErrors	
RF_NVLinkManagement_0_Port_Metrics_Nvidia_RXSwitchRelayErrors	
RF_NVLinkManagement_0_Port_Metrics_Nvidia_TXWait	
RF_NVLinkManagement_0_Port_Metrics_Nvidia_VL15Dropped	
RF_NVLinkManagement_0_Port_Metrics_Nvidia_VL15TXBytes	
RF_NVLinkManagement_0_Port_Metrics_Nvidia_VL15TXPackets	
RF_NVLinkManagement_0_Port_Metrics_RXBytes	
RF_NVLinkManagement_0_Port_Metrics_RXErrors	
RF_NVLinkManagement_0_Port_Metrics_TXBytes	
RF_NVLinkManagement_0_ResourceState_Status	
RF_NVLinkManagement_0_Resource_	

…continues

Table 6.8: …continued

Redfish Metrics For The DGX GB200	Description
CurrentSpeed	
RF_NVLinkManagement_0_Resource_LinkState	
RF_NVLinkManagement_0_Resource_LinkStatus	
RF_NVLinkManagement_0_Resource_MaxSpeed	
RF_NVLink_0_Port_MetricsRXFrames_Networking[7]	
RF_NVLink_0_Port_MetricsRXMulticastFrames_Networking[7]	
RF_NVLink_0_Port_MetricsRXUnicastFrames_Networking[7]	
RF_NVLink_0_Port_MetricsTXDiscards_Networking[7]	
RF_NVLink_0_Port_MetricsTXFrames_Networking[7]	
RF_NVLink_0_Port_MetricsTXMulticastFrames_Networking[7]	
RF_NVLink_0_Port_MetricsTXUnicastFrames_Networking[7]	
RF_NVLink_0_Port_Metrics_Nvidia_BitErrorRate[7]	
RF_NVLink_0_Port_Metrics_Nvidia_EffectiveError[7]	
RF_NVLink_0_Port_Metrics_Nvidia_LinkDowned[7]	
RF_NVLink_0_Port_Metrics_Nvidia_LinkErrorRecovery[7]	
RF_NVLink_0_Port_Metrics_Nvidia_MalformedPackets[7]	
RF_NVLink_0_Port_Metrics_Nvidia_NVLinkDataRxBandwidth[7]	
RF_NVLink_0_Port_Metrics_Nvidia_NVLinkDataTxBandwidth[7]	
RF_NVLink_0_Port_Metrics_Nvidia_NVLinkRawRxBandwidth[7]	
RF_NVLink_0_Port_Metrics_Nvidia_NVLinkRawTxBandwidth[7]	
RF_NVLink_0_Port_Metrics_Nvidia_NeighborMTUDiscards[7]	
RF_NVLink_0_Port_Metrics_Nvidia_QP1Dropped[7]	
RF_NVLink_0_Port_Metrics_Nvidia_RXNoProtocolBytes[7]	

…continues

Table 6.8: …continued

Redfish Metrics For The DGX GB200	Description
RF_NVLink_0_Port_Metrics_Nvidia_RXRemotePhysicalErrors[7]	
RF_NVLink_0_Port_Metrics_Nvidia_RXSwitchRelayErrors[7]	
RF_NVLink_0_Port_Metrics_Nvidia_TXNoProtocolBytes[7]	
RF_NVLink_0_Port_Metrics_Nvidia_TXWait[7]	
RF_NVLink_0_Port_Metrics_Nvidia_VL15Dropped[7]	
RF_NVLink_0_Port_Metrics_Nvidia_VL15TXBytes[7]	
RF_NVLink_0_Port_Metrics_Nvidia_VL15TXPackets[7]	
RF_NVLink_0_Port_Metrics_RXBytes[7]	
RF_NVLink_0_Port_Metrics_RXErrors[7]	
RF_NVLink_0_Port_Metrics_TXBytes[7]	
RF_NVLink_0_ResourceState_Status[7]	
RF_NVLink_0_Resource_CurrentSpeed[7]	
RF_NVLink_0_Resource_LinkState[7]	
RF_NVLink_0_Resource_LinkStatus[7]	
RF_NVLink_0_Resource_MaxSpeed[7]	
RF_NVLink_0_Resource_Nvidia_RXWidth[7]	
RF_NVLink_0_Resource_Nvidia_TXWidth[7]	
RF_NVME_M2_0_Temp_0	NVME M2 0 Temp 0
RF_NVMe_SSDATemp	NVMe_SSDA Temp
RF_NVSwitch_0_ResourceState_Status	
RF_NVSwitch_0_Resource_#Switch.Reset_target	
RF_NVSwitch_0_Resource_CurrentBandwidth	
RF_NVSwitch_0_Resource_Enabled	
RF_NVSwitch_0_Resource_FirmwareVersion	
RF_NVSwitch_0_Resource_MaxBandwidth	
RF_NVSwitch_0_Resource_Nvidia_SwitchIsolationMode	
RF_NVSwitch_1_ResourceState_	

…continues

Table 6.8: ...continued

Redfish Metrics For The DGX GB200	Description
Status	
RF_NVSwitch_1_Resource_#Switch.Reset_target	
RF_NVSwitch_1_Resource_CurrentBandwidth	
RF_NVSwitch_1_Resource_Enabled	
RF_NVSwitch_1_Resource_FirmwareVersion	
RF_NVSwitch_1_Resource_MaxBandwidth	
RF_NVSwitch_1_Resource_Nvidia_SwitchIsolationMode	
RF_P1_DIMMA DTemp	P1_DIMMA D Temp
RF_P1_DIMME HTemp	P1_DIMME H Temp
RF_P2_DIMMA DTemp	P2_DIMMA D Temp
RF_P2_DIMME HTemp	P2_DIMME H Temp
RF_PCB_Inlet_Temp	PCB Inlet Temp
RF_PCH1.05V	PCH 1.05V
RF_PCHPVNN	PCH PVNN
RF_PCHTemp	PCH Temp
RF_PCIe_0_ResourceState_Status	
RF_PCIe_0_Resource_ActiveWidth	
RF_PCIe_0_Resource_CurrentSpeed	
RF_PCIe_0_Resource_MaxSpeed	
RF_PCIe_0_Resource_Width	
RF_PCIe_DeviceLanesInUse_PCIeInterface	
RF_PCIe_DeviceMaxLanes_PCIeInterface	
RF_PCIe_DeviceMaxPCIeType_PCIeInterface	
RF_PCIe_DevicePCIeType_PCIeInterface	
RF_PCIe_DeviceState_Status	
RF_PCIe_Device_Nvidia_AERCorrectableErrorStatus	
RF_PCIe_Device_Nvidia_AERUncorrectableErrorStatus	
RF_PCIe_Device_Nvidia_NVLinkReferenceClockEnabled	
RF_PCIe_Device_Nvidia_PCIeReferenceClockEnabled	
RF_PCIe_Devicecode_error	
RF_PCIe_Devicemessage_error	
RF_PDB_0_HSC_0_Cur_0	PDB 0 HSC 0 Cur 0
RF_PDB_0_HSC_0_Pwr_0	PDB 0 HSC 0 Pwr 0
RF_PDB_0_HSC_0_Temp_0	PDB 0 HSC 0 Temp 0
RF_PDB_0_HSC_0_Volt_In_0	PDB 0 HSC 0 Volt In 0
RF_PDB_0_HSC_0_Volt_Out_0	PDB 0 HSC 0 Volt Out 0

...continues

Table 6.8: ...continued

Redfish Metrics For The DGX GB200	Description
RF_PDB_0_HSC_1_Cur_0	PDB 0 HSC 1 Cur 0
RF_PDB_0_HSC_1_Pwr_0	PDB 0 HSC 1 Pwr 0
RF_PDB_0_HSC_1_Temp_0	PDB 0 HSC 1 Temp 0
RF_PDB_0_HSC_1_Volt_In_0	PDB 0 HSC 1 Volt In 0
RF_PDB_0_HSC_1_Volt_Out_0	PDB 0 HSC 1 Volt Out 0
RF_PDB_0_Inlet_Temp_0	PDB 0 Inlet Temp 0
RF_PDB_0_Vreg_0_Temp_0	PDB 0 Vreg 0 Temp 0
RF_PDB_0_Vreg_0_Temp_1	PDB 0 Vreg 0 Temp 1
RF_PDB_0_Vreg_1_Temp_0	PDB 0 Vreg 1 Temp 0
RF_PDB_0_Vreg_1_Temp_1	PDB 0 Vreg 1 Temp 1
RF_PDB_TOTAL_CURRENT	
RF_PDB_TOTAL_PWR	
RF_PDB_TOTAL_VOLT_IN	
RF_PDB_TOTAL_VOLT_OUT	
RF_PS1_Status_Reading	
RF_PS1_Status_ReadingRangeMax	
RF_PS1_Status_ReadingRangeMin	
RF_PWConsumption	PW Consumption
RF_PeripheralTemp	Peripheral Temp
RF_PowerShelf_Status[2],[8]	Power shelf status of PSU
RF_PowerShelf_Status[2],[8]	Power shelf status total
RF_Power_0	Power_0
RF_Power_PowerControl	
RF_Power_PowerSupplies	
RF_Power_Redundancy	
RF_Power_Supply_Energy[2],[8]	Energy consumption since boot of PSU
RF_Power_Supply_Energy[9]	Energy consumption since boot of all PSUs
RF_Power_Supply_FanSpeed[2],[8]	Fan speed of PSU
RF_Power_Supply_FanSpeed[10]	Fan speed average of all PSUs
RF_Power_Supply_Id[2],[8]	ID of PSU
RF_Power_Supply_InputCurrent[2],[8]	Input current for PSU
RF_Power_Supply_InputCurrent[9]	Input current of all PSUs
RF_Power_Supply_InputPower[2],[8]	Input power for PSU
RF_Power_Supply_InputPower[9]	Input power total of all PSUs
RF_Power_Supply_InputVoltage[2],[8]	Input voltage for PSU
RF_Power_Supply_LifetimeReading_Energy[2],[8]	Energy consumption of PSU over lifetime

...continues

Table 6.8: ...continued

Redfish Metrics For The DGX GB200	Description
RF_Power_Supply_Name[2],[8]	Name of PSU
RF_Power_Supply_OutputPower[2],[8]	Output power of PSU
RF_Power_Supply_OutputPower[9]	Output power total of all PSUs
RF_Power_Supply_PowerFactor_Input[2],[8]	Input power factor of PSU
RF_Power_Supply_RailCurrent[2],[8]	Rail current of PSU
RF_Power_Supply_RailCurrent[9]	Rail current total of all PSUs
RF_Power_Supply_RailPower[2],[8]	Rail current of PSU
RF_Power_Supply_RailPower[9]	Rail power total of all PSUs
RF_Power_Supply_RailVoltage[2],[8]	Rail voltage of PSU
RF_Power_Supply_Temperature[2],[8]	Temperature of PSU
RF_Power_Supply_Temperature[10]	Temperature average of all PSUs
RF_Power_Supply_code_error[2],[8]	Code error of PSU
RF_Power_Supply_message_error[2],[8]	Message error of PSU
RF_Power_Voltages	
RF_ProcessorModule_0_CPU_0_CoreUtil_0[11]	ProcessorModule 0 CPU 0 CoreUtil 0. The substring CoreUtil_0 can also take values of CoreUtil_1, CoreUtil_2... up to CoreUtil_71
RF_ProcessorModule_0_CPU_0_CpuFreq_0[11]	ProcessorModule 0 CPU 0 CpuFreq 0
RF_ProcessorModule_0_CPU_0_Energy_0[11]	ProcessorModule 0 CPU 0 Energy 0
RF_ProcessorModule_0_CPU_0_EnforcedEDPc_0[11]	ProcessorModule 0 CPU 0 EnforcedEDPc 0
RF_ProcessorModule_0_CPU_0_EnforcedEDPp_0[11]	ProcessorModule 0 CPU 0 EnforcedEDPp 0
RF_ProcessorModule_0_CPU_0_Power_0[11]	ProcessorModule 0 CPU 0 Power 0
RF_ProcessorModule_0_CPU_0_TempAvg_0[11]	ProcessorModule 0 CPU 0 TempAvg 0
RF_ProcessorModule_0_CPU_0_TempLimit_0[11]	ProcessorModule 0 CPU 0 TempLimit 0
RF_ProcessorModule_0_MemCntl_0_Freq_0[11]	ProcessorModule 0 MemCntl 0 Freq 0
RF_ProcessorModule_0_Vreg_0_CpuPower_0[11]	ProcessorModule 0 Vreg 0 CpuPower 0
RF_ProcessorModule_0_Vreg_0_CpuVoltage_0[11]	ProcessorModule 0 Vreg 0 CpuVoltage 0
RF_ProcessorModule_0_Vreg_0_SocPower_0[11]	ProcessorModule 0 Vreg 0 SocPower 0
RF_ProcessorModule_0_Vreg_0_SocVoltage_0[11]	ProcessorModule 0 Vreg 0 SocVoltage 0

...continues

Table 6.8: ...continued

Redfish Metrics For The DGX GB200	Description
RF_SYS_ART1Temp	SYS_ART1 Temp
RF_SYS_ART2Temp	SYS_ART2 Temp
RF_StorageBackplane_0_SSD_0_Temp_0	StorageBackplane 0 SSD 0 Temp 0
RF_StorageBackplane_0_SSD_2_Temp_0	StorageBackplane 0 SSD 2 Temp 0
RF_StorageBackplane_1_SSD_0_Temp_0	StorageBackplane 1 SSD 0 Temp 0
RF_StorageBackplane_1_SSD_2_Temp_0	StorageBackplane 1 SSD 2 Temp 0
RF_Summary_Metrics_Id[2]	
RF_Summary_Metrics_Name[2]	
RF_Summary_Metrics_Power[2]	Power consumption (Watts)
RF_Summary_Metrics_Temperature[2]	Temperature (Celsius)
RF_SystemTemp	System Temp
RF_Total_PSU[2]	
RF_VBAT_Reading	
RF_VBAT_ReadingRangeMax	
RF_VBAT_ReadingRangeMin	
RF_Vreg_0_CpuPower_0	Vreg_0_CpuPower_0
RF_Vreg_0_CpuVoltage_0	Vreg_0_CpuVoltage_0
RF_Vreg_0_SocPower_0	Vreg_0_SocPower_0
RF_Vreg_0_SocVoltage_0	Vreg_0_SocVoltage_0
RF_Vreg_1_CpuPower_0	Vreg_1_CpuPower_0
RF_Vreg_1_CpuVoltage_0	Vreg_1_CpuVoltage_0
RF_Vreg_1_SocPower_0	Vreg_1_SocPower_0
RF_Vreg_1_SocVoltage_0	Vreg_1_SocVoltage_0

[1] The substring NIC1Temp can also be NIC2Temp, NIC3Temp...
[2] from redfish_power_shelf data producer
[3] The substring CPU_0 can also be CPU_1
[4] The substring FAN_1 can also be FAN_2, FAN_3...
[5] From redfish_leak data producer
[6] The substring GPU_0 can also be GPU_1, GPU_2...
[7] On the DGX GB200 the value of <*number*> in ...NVLink_<*number*>...can be from 0 to 71
[8] Takes the PSU number as a parameter. On the DGX GB200 it can be from 1 to 6
[9] Takes total as a parameter. On the DGX GB200 it means the total value of the 6 PSUs
[10] Takes average as a parameter. On the DGX GB200 it means the average value of the 6 PSUs
[11] On the DGX GB200 the value of <*number*> in ...ProcessorModule_<*number*>... can be 0 or 1
[12] The substring ProcessorModule_0 can also be ProcessorModule_1

6.9 Redfish Enums For The DGX GB200

Redfish monitoring can be carried out for devices with a Redfish subscription (page 46).

Redfish firmware enums that may be available on the DGX GB200 platform are shown in table 6.9.

As is usual with hardware, the metrics can change according to hardware and especially with firmware changes. The table should therefore be regarded as an indication of what metrics are available, and not a list of what metrics must exist.

The list of Redfish enums that are there on the DGX GB200 platform can be found by running:

Example

```
basecm11->monitoring->measurable]% list enum | grep RF_
```

Table 6.9:

Redfish Enums For The DGX GB200	Parameter	Description
RF_BF3_Slot_1_NIC_Temp_0	state	
RF_BF3_Slot_2_NIC_Temp_0	state	
RF_BF3_Slot_2_Port_0_Temp	state	
RF_BF3_Slot_2_Port_1_Temp	state	
RF_BF3_Slot_2_Temp	state	
RF_BMC_0_DCSCM_Temp_0	state	
RF_Chassis_0_FAN_1_FRONT[1]	state	
RF_Chassis_0_FAN_1_PWM	state	
RF_Chassis_0_FAN_1_REAR	state	
RF_CPU_0_Processor__Nvidia_EDPViolationState		
RF_CPU_0_Processor__Nvidia_PerformanceState		
RF_CPU_0_Processor__Nvidia_PowerBreakPerform\ anceState		
RF_CPU_1_Processor__Nvidia_EDPViolationState		
RF_CPU_1_Processor__Nvidia_PerformanceState		
RF_CPU_1_Processor__Nvidia_PowerBreakPerform\ anceState		
RF_DIMM_SlotHealthRollup_Status	state	
RF_DIMM_SlotHealth_Status	state	
RF_ERoT_BMC_0_PowerState		
RF_ERoT_BMC_0HealthRollup_Status	state	
RF_ERoT_BMC_0Health_Status	state	
RF_ERoT_BMC_0_PowerState		
RF_HGX_BMC_0_Temp_0	state	
RF_HGX_Baseboard_0HealthRollup_Status	state	
RF_HGX_Baseboard_0Health_Status	state	
RF_HGX_Baseboard_0_PowerState		
RF_HGX_CPU_0HealthRollup_Status	state	
RF_HGX_CPU_0Health_Status	state	
RF_HGX_CPU_0_PowerState		

...continues

Table 6.9: …continued

Redfish Enums For The DGX GB200	Parameter	Description
RF_HGX_CPU_1HealthRollup_Status	state	
RF_HGX_CPU_1Health_Status	state	
RF_HGX_CPU_1_PowerState		
RF_HGX_Chassis_0_TotalGPU_Power_0	state	
RF_HGX_ERoT_BMC_0HealthRollup_Status	state	
RF_HGX_ERoT_BMC_0Health_Status	state	
RF_HGX_ERoT_CPU_0HealthRollup_Status	state	
RF_HGX_ERoT_CPU_0Health_Status	state	
RF_HGX_ERoT_CPU_0_PowerState		
RF_HGX_ERoT_CPU_1HealthRollup_Status	state	
RF_HGX_ERoT_CPU_1Health_Status	state	
RF_HGX_ERoT_FPGA_0HealthRollup_Status	state	
RF_HGX_ERoT_FPGA_0Health_Status	state	
RF_HGX_ERoT_FPGA_1HealthRollup_Status	state	
RF_HGX_ERoT_FPGA_1Health_Status	state	
RF_HGX_CPU_0_PowerState		
RF_HGX_CPU_1_PowerState		
RF_HGX_GPU_0HealthRollup_Status[2]	state	
RF_HGX_GPU_0Health_Status[2]	state	
RF_HGX_GPU_0_DRAM_0_Power_0[2]	state	
RF_HGX_GPU_0_DRAM_0_Temp_0[2]	state	
RF_HGX_GPU_0_Energy_0[2]	state	
RF_HGX_GPU_0_PowerState[2]		
RF_HGX_GPU_0_Power_0[2]	state	
RF_HGX_GPU_0_TEMP_0[2]	state	
RF_HGX_GPU_0_TEMP_1[2]	state	
RF_HGX_GPU_0_Voltage_0[2]	state	
RF_HGX_IRoT_GPU_0HealthRollup_Status[2]	state	
RF_HGX_IRoT_GPU_0Health_Status[2]	state	
RF_HGX_IRoT_GPU_1_PowerState[2]		
RF_HGX_ProcessorModule_0_Exhaust_Temp_0	state	
RF_HGX_ProcessorModule_0_Inlet_Temp_0	state	
RF_HGX_ProcessorModule_0_Inlet_Temp_1	state	
RF_HGX_ProcessorModule_1_Exhaust_Temp_0	state	
RF_HGX_ProcessorModule_1_Inlet_Temp_0	state	
RF_HGX_ProcessorModule_1_Inlet_Temp_1	state	
RF_IO_Board_0_CX7_0_Port_0_Temp	state	
RF_IO_Board_0_CX7_0_Temp	state	
RF_IO_Board_0_CX7_0_Temp_0	state	

…continues

Table 6.9: ...continued

Redfish Enums For The DGX GB200	Parameter	Description
RF_IO_Board_0_CX7_1_Port_0_Temp	state	
RF_IO_Board_0_CX7_1_Temp	state	
RF_IO_Board_0_CX7_1_Temp_0	state	
RF_IO_Board_1_CX7_0_Port_0_Temp	state	
RF_IO_Board_1_CX7_0_Temp	state	
RF_IO_Board_1_CX7_0_Temp_0	state	
RF_IO_Board_1_CX7_1_Port_0_Temp	state	
RF_IO_Board_1_CX7_1_Temp	state	
RF_IO_Board_1_CX7_1_Temp_0	state	
RF_IRoT_CX7_0HealthRollup_Status	state	
RF_IRoT_CX7_0Health_Status	state	
RF_IRoT_CX7_1HealthRollup_Status	state	
RF_IRoT_CX7_1Health_Status	state	
RF_IRoT_CX7_2HealthRollup_Status	state	
RF_IRoT_CX7_2Health_Status	state	
RF_IRoT_CX7_3HealthRollup_Status	state	
RF_IRoT_CX7_3Health_Status	state	
RF_LD_LeakDetection[3]		
RF_LeakDetector[3]	Chassis_0_ LeakDetector_0_ ColdPlate	
RF_LeakDetector[3]	Chassis_0_ LeakDetector_0_ Manifold	
RF_LeakDetector[3]	Chassis_0_ LeakDetector_1_ ColdPlate	
RF_LeakDetector[3]	Chassis_0_ LeakDetector_1_ Manifold	
RF_NVLinkManagement_0_ResourceHealthRollup_ Status	state	
RF_NVLinkManagement_0_ResourceHealth_Status	state	
RF_NVLink_ResourceHealthRollup_Status	state	
RF_NVLink_ResourceHealth_Status	state	
RF_NVLink_ResourceState_Status[4]	acp0	
RF_NVLink_Resource_LinkState[4]	acp0	
RF_NVLink_0_ResourceHealthRollup_Status	state	
RF_NVLink_0_ResourceHealth_Status	state	
RF_NVME_M2_0_Temp_0	state	
RF_NVSwitch_0_ResourceHealthRollup_Status[5]	state	

...continues

Table 6.9: …continued

Redfish Enums For The DGX GB200	Parameter	Description
RF_NVSwitch_0_ResourceHealth_Status[5]	state	
RF_InterswitchPort_0_ResourceHealthRollup_Status	state	
RF_InterswitchPort_0_ResourceHealth_Status	state	
RF_PCIe_0_ResourceHealthRollup_Status	state	
RF_PCIe_0_ResourceHealth_Status	state	
RF_PCIe_DeviceHealthRollup_Status	state	
RF_PCIe_DeviceHealth_Status	state	
RF_PDB_0_HSC_0_Cur_0	state	
RF_PDB_0_HSC_0_Pwr_0	state	
RF_PDB_0_HSC_0_Temp_0	state	
RF_PDB_0_HSC_0_Volt_In_0	state	
RF_PDB_0_HSC_0_Volt_Out_0	state	
RF_PDB_0_HSC_1_Cur_0	state	
RF_PDB_0_HSC_1_Pwr_0	state	
RF_PDB_0_HSC_1_Temp_0	state	
RF_PDB_0_HSC_1_Volt_In_0	state	
RF_PDB_0_HSC_1_Volt_Out_0	state	
RF_PDB_0_Inlet_Temp_0	state	
RF_PDB_0_Vreg_0_Temp_0	state	
RF_PDB_0_Vreg_0_Temp_1	state	
RF_PDB_0_Vreg_1_Temp_0	state	
RF_PDB_0_Vreg_1_Temp_1	state	
RF_ProcessorModule_0_CPU_0_CpuFreq_0[6]	state	
RF_ProcessorModule_0_CPU_0_Energy_0[6]	state	
RF_ProcessorModule_0_CPU_0_EnforcedEDPc_0[6]	state	
RF_ProcessorModule_0_CPU_0_EnforcedEDPp_0[6]	state	
RF_ProcessorModule_0_CPU_0_Power_0[6]	state	
RF_ProcessorModule_0_CPU_0_TempAvg_0[6]	state	
RF_ProcessorModule_0_CPU_0_TempLimit_0[6]	state	
RF_ProcessorModule_0_MemCntl_0_Freq_0[6]	state	
RF_ProcessorModule_0_Vreg_0_CpuPower_0[6]	state	
RF_ProcessorModule_0_Vreg_0_CpuVoltage_0[6]	state	
RF_ProcessorModule_0_Vreg_0_SocPower_0[6]	state	
RF_ProcessorModule_0_Vreg_0_SocVoltage_0[6]	state	
RF_ProcessorModule_0_CPU_0_CoreUtil_0[6]	state	
In the preceding row, the substring CoreUtil_0 *can also take values of* CoreUtil_1, CoreUtil_2…, up to CoreUtil_71		
RF_StorageBackplane_0_SSD_0_Temp_0[7]	state	

…continues

Table 6.9: ...continued

Redfish Enums For The DGX GB200	Parameter	Description

[1] The number used in the substring `Chassis_0` and the substring `FAN_` can vary
[2] The substring `GPU_0` can be `GPU_1`, `GPU_2`...
[3] Produced by `redfish_leak` data producer
[4] The parameter substring `acp0` can be `acp1`, `acp2`...
[5] The substring `NVSwitch_0` can be `NVSwitch_1`, `NVSwitch_2`...
[6] The substring `ProcessorModule_0` can be `ProcessorModule_1`, `ProcessorModule_2`...
[7] The substring `SSD_0` can be `SSD_1`, `SSD_2`..., and
the substring `StorageBackplane_0` can be `StorageBackplane_1`, `StorageBackplane_2`...

6.10 Health Checks For The DGX GB200

Redfish health checks that may be additionally available on the DGX GB200 platform are shown in table 6.10.

As is usual with hardware, the health checks can change according to hardware and especially with firmware changes. The table should therefore be regarded as an indication of the health checks that are available, and not a list of the health checks that must exist.

The list of Redfish health checks that are there can be found by running:

Example

```
[basecm11->monitoring->measurable]% list healthcheck
```

Table 6.10: DGX GB200 Health Checks

DGX GB200 Health Check	Parameter	Description
`gpu_health_driver`	gpu0	Driver-related status
`gpu_health_hostengine`		Host engine status
`gpu_health_inforom`	gpu0	Inforom status
`gpu_health_mcu`	gpu0	Microcontroller unit status
`gpu_health_mem`	gpu0	Memory status
`gpu_health_nvlink`	gpu0	NVLINK system status
`gpu_health_nvswitch_fatal`	gpu0	NV switch fatal errors
`gpu_health_nvswitch_non_fatal`	gpu0	NV switch non-fatal errors
`gpu_health_overall`		Overall system status
`gpu_health_overall`	gpu0	Overall system status for GPU0
`gpu_health_pcie`	gpu0	PCIe system status
`gpu_health_pmu`	gpu0	Power management unit status
`gpu_health_power`	gpu0	Power status
`gpu_health_sm`	gpu0	Streaming multiprocessor status
`gpu_health_thermal`	gpu0	Temperature status
`nvsm_pci_health`		Checks health of PCI devices in a DGX system

...continues

Table 6.10: DGX GB200 Health Checks...continued

DGX GB200 Health Check	Parameter	Description
nvsm_show_health		Checks health of DGX system
NMXController		Is NMX Controller in the correct state?