



# CDW Documentation

**nvidia\_bcm-intro**

---

[Back to NVIDIA page](#)

[Back to BCM page](#)

## Base Command Manager (BCM) Intro What NVIDIA Base Command Manager Is

NVIDIA Base Command Manager (BCM) is NVIDIA's cluster management platform for AI and HPC environments. Its purpose is to streamline **cluster provisioning, workload management, and infrastructure monitoring** from a centralized platform. \* NVIDIA describes BCM as providing the tools needed to **deploy and manage an AI data center**, not just a few isolated servers. BCM is designed for environments that need repeatable deployment, centralized administration, monitoring, and operational consistency across many nodes. ([NVIDIA Docs][1])

### Core Idea Behind BCM

BCM is meant to manage the **full cluster lifecycle**:

- initial installation
- node provisioning
- software image management
- workload manager integration
- monitoring
- health checks
- accounting and chargeback
- scaling and cloud extension

That is an important point: BCM is not just an installer. It is an ongoing **operations and lifecycle management framework** for clusters. ([NVIDIA Docs][2])

### What BCM Helps Administrators Do

BCM contains tools and applications to help administrators:

- install a cluster
- configure and maintain it
- monitor nodes and jobs
- keep the environment consistent over time

NVIDIA also positions BCM as a way to provide users with an optimized environment for running compute-intensive applications. In other words, BCM is not only about infrastructure control, but also about creating a usable and reliable compute platform for end users. ([NVIDIA Docs][2])

### Centralized Management Model

BCM centralizes cluster control rather than forcing administrators to manage each node individually. \* Two major management interfaces are provided:

- **Base View**: the graphical web interface
- **cmsh**: the command-line shell

These interfaces expose the same cluster management functionality, so administrators can choose GUI or CLI workflows depending on the task. \* Base View is installed by default and provides an encrypted web interface for cluster administration and monitoring. ([NVIDIA Docs][2])

## What Types of Systems BCM Can Manage

BCM is built to manage more than just standard compute nodes. NVIDIA documentation shows BCM can represent and manage device types such as:

- head nodes
- physical nodes
- virtual nodes
- cloud nodes
- GPU units
- switches

\* That matters because BCM is intended for real cluster environments, where compute, networking, GPUs, and hybrid resources all need to be managed together. ([NVIDIA Docs][2])

## Platform and Architecture Support Operating System Support

NVIDIA BCM 11 supports several Linux distributions, including:

- Rocky Linux 8 and 9
- SUSE Linux Enterprise Server 15
- Ubuntu 22.04 and 24.04

The installation documentation also notes support for:

- Red Hat Enterprise Linux derivatives
- DGX OS 7

BCM is built on top of major Linux distributions rather than being its own standalone operating system. ([NVIDIA Docs][1])

## Processor Architecture Support

BCM supports:

- **x86\_64**
- **arm64 / AArch64**

NVIDIA states that BCM aims to be largely independent of operating system and processor architecture, though actual support still depends on documented feature limitations and platform compatibility. ([NVIDIA Docs][3])

## Mixed Operating System Environments

BCM can create software images using different supported Linux distributions for compute nodes. \* NVIDIA recommends keeping compute nodes on the same distribution as the head node when possible, because mixed-distribution clusters are generally harder to manage. \* So the practical exam and operations takeaway is:

- BCM is flexible enough for mixed environments
- but same-OS deployments are simpler and usually preferred ([NVIDIA Docs][1])

## Head Node and Software Image Model Head Node Role

BCM is typically installed on:

- a single head node, or
- a high-availability head-node pair

The head node is the control point for provisioning and management. \* In common cluster designs, the head node also provides key cluster services needed for provisioning and operation. ([NVIDIA Docs][3])

## Software Images

One of the most important BCM concepts is the **software image**. NVIDIA describes a software image as a blueprint for a node's filesystem. It lives on the head node and is used to provision regular nodes. The default image is commonly named **default-image**. When a regular node boots for provisioning, BCM sets it up using a copy of the software image. This is a major reason BCM helps prevent drift: nodes are not all manually configured one by one; they are built from managed images. ([NVIDIA Docs][2])

## Why Software Images Matter

Software images make cluster administration more repeatable and scalable. \* Instead of logging into every server and making one-off changes, administrators can:

- update the image
- reprovision or resynchronize nodes
- keep systems aligned with a known desired state

NVIDIA documents that filesystem changes from the image can be redistributed to nodes through synchronization, and reprovisioning installs the latest image onto a node. ([NVIDIA Docs][2])

## Installation and Initial Cluster Bring-Up Installation Flow

BCM installation can begin through either:

- a graphical installer, or
- a text-based installer

During installation, administrators can define core cluster settings such as:

- nameservers
- domain information
- workload manager selection
- networking behavior
- BMC configuration

This is why BCM installation is more than just putting software on one box. It is the start of building the management plane for the whole cluster. ([NVIDIA Docs][3])

## Services Commonly Provided by the Head Node

NVIDIA documentation shows BCM commonly manages or generates configuration for services such as:

- DHCP

- DNS
- TFTP / network boot support
- NFS
- LDAP-related services

In the default service role set, services such as `dhcpd`, `named`, `slapd`, `nfs`, and `cuda-dcgm` appear as part of BCM's managed cluster service model. This is a key exam concept: the head node is not just a login box. It is often the operational backbone of provisioning and cluster services. ([NVIDIA Docs][2])

## Network Boot and Provisioning

BCM uses network boot workflows for provisioning nodes. \* NVIDIA documents DHCP and TFTP as part of the boot and provisioning path. \* In a common Type 1 network setup, the head node provides DHCP and PXE-related services to an isolated worker-node network. \* This is what enables BCM to discover and provision systems in a more automated way than manual per-node installs. ([NVIDIA Docs][3])

## Why BCM Exists: Operational Value

### 1. Faster, More Repeatable Provisioning

BCM reduces manual cluster build work by using:

- centralized installation
- managed services
- software images
- automated node provisioning

The practical benefit is that a cluster can be built and expanded with much more consistency than a manual "install every server separately" model. ([NVIDIA Docs][1])

### 2. Reduced Administrative Complexity

BCM gives administrators one control plane for many tasks that would otherwise be scattered across scripts, separate tools, and manual node access. Instead of treating every node as a special case, BCM encourages image-based, policy-driven, centralized administration. Base View and cmsh both support this centralized model. ([NVIDIA Docs][2])

### 3. Better Consistency Across the Cluster

Because nodes are provisioned from centrally managed images, BCM helps reduce configuration drift. Image locking, synchronization, and reprovisioning all support the idea of maintaining known-good cluster state over time. ([NVIDIA Docs][2])

### 4. Better Visibility Into Cluster Health and Usage

BCM is not blind after deployment. It includes cluster monitoring, job monitoring, health checks, accounting, and chargeback capabilities. That means it supports both infrastructure operations and operational reporting. ([NVIDIA Docs][2])

## Monitoring, Health Checks, and Visibility

## Cluster Monitoring

BCM includes built-in monitoring capabilities for cluster resources. Base View includes dashboards and monitoring views, and NVIDIA documents support for PromQL queries and drilldowns. BCM can also interact with Prometheus-style monitoring workflows through CMDaemon exporter capabilities. ([NVIDIA Docs][2])

## Job Monitoring

BCM supports **job-centric monitoring**, which means monitoring can be tied to actual workload execution rather than only raw node status. \* NVIDIA documents that BCM begins updating job information in its database as soon as it detects a new job. \* Job monitoring includes visibility into resource usage such as CPU, memory, storage, and other job-related metrics. ([NVIDIA Docs][2])

## Health Checks

Health checks are an important BCM capability. \* NVIDIA documents that health checks can run as:

- scheduled checks
- pre-job checks
- post-job checks

A very important exam point:

- if a **pre-job** health check fails, the job can be prevented from running on that node
- a common corrective action is to drain that node

BCM health checks can also launch actions and support notification behavior. NVIDIA also provides example health check scripts, which shows the platform supports extension and customization. ([NVIDIA Docs][2])

## Workload Managers and Orchestration Workload Manager Integration

BCM supports workload management setup from the cluster management plane. NVIDIA documents setup tooling such as `cm-wlm-setup`. The Administrator Manual includes workload manager chapters for:

- Slurm
- PBS
- LSF

That means workload orchestration is not an afterthought. It is built into BCM's operating model. ([NVIDIA Docs][3])

## Kubernetes Integration

BCM also supports Kubernetes-related integration. NVIDIA documentation references `cm-kubernetes-setup` for streamlining Kubernetes-related deployment flows. This matters because BCM is meant to support modern AI/HPC environments that may use both traditional schedulers and container orchestration platforms. ([NVIDIA Docs][4])

## Auto Scaling and Dynamic Resource Use

### cm-scale / Auto Scaler

NVIDIA documents **cm-scale** as BCM's mechanism for dynamic scaling based on workload demand. \* It can work with:

- static node providers
- dynamic node providers

BCM can use this to:

- power nodes on or off
- clone or remove dynamic nodes
- repurpose nodes for different categories or overlays

This is a major concept: BCM can adjust cluster resources according to demand instead of forcing a permanently fixed hardware role for every node. ([NVIDIA Docs][2])

## Reprovisioning and Repurposing

NVIDIA documents that when job demand changes, BCM can:

- drain a node
- stop it
- change its assigned category
- move it to different overlays
- reboot or boot it into a new role

In practical terms, that means the same physical infrastructure can be repurposed to serve different workload needs. ([NVIDIA Docs][2])

## Kubernetes and Queue Awareness

BCM's scaling model is not limited to just one scheduler type. NVIDIA documents queue-based thresholds for cloudbursting and also notes namespace tracking support for Kubernetes jobs and pods. So the platform is designed to react to workload pressure, not just machine states. ([NVIDIA Docs][2])

## Cloud Capabilities

### Cluster On Demand

BCM cloud documentation for current releases includes **Cluster On Demand** workflows for multiple cloud providers, including:

- AWS
- Azure
- OCI
- GCP

This allows BCM-managed cluster capacity to be created in cloud environments using BCM tooling rather than treating cloud as a completely separate management domain. ([NVIDIA Docs][5])

### Cluster Extension

BCM also supports **Cluster Extension**, where an existing cluster extends capacity into the cloud. \* NVIDIA specifically documents cluster extension to **AWS** and **Azure**. \* An important distinction:

- in Cluster Extension, the **head node remains outside the cloud**
- cloud nodes are added as an extension of the managed cluster

That makes Cluster Extension a hybrid-cloud capacity model rather than a fully cloud-native isolated cluster. ([NVIDIA Docs][5])

## Edge Capabilities

### BCM at the Edge

BCM includes edge computing capabilities documented in a dedicated Edge Manual. \* NVIDIA describes edge computing as useful when local processing is needed because of:

- low latency requirements
- high bandwidth demands
- high CPU-cycle needs

BCM's edge model allows a central cluster to manage resources distributed across multiple locations. ([NVIDIA Docs][6])

### One Cluster, Multiple Locations

NVIDIA explicitly describes the edge model as **one cluster, multiple locations**. \* In this design:

- a central head node manages the broader cluster
- edge sites include an edge director and edge nodes
- edge nodes can PXE boot from the edge director

NVIDIA also notes that direct connectivity between the central head node and edge nodes is not required in the same way, because the edge director acts as the local control point. ([NVIDIA Docs][6])

## NVIDIA-Specific GPU Integration

### DCGM and GPU Awareness

BCM has direct NVIDIA GPU integration through `cuda-dcgm`. \* NVIDIA documentation states that the `cuda-dcgm` package is required to access NVIDIA GPU metrics in BCM. \* That is important because BCM is not merely cluster-aware; it is GPU-aware, which is critical for AI and accelerated computing environments. ([NVIDIA Docs][2])

### GPU Metrics

BCM can collect and expose GPU-related monitoring data, including GPU usage and health-related metrics. \* NVIDIA documentation shows examples of GPU metrics and GPU-related monitoring fields such as utilization, ECC-related values, power-related values, and other GPU telemetry. ([NVIDIA Docs][2])

### GPU Health Checks

NVIDIA documents multiple GPU health checks, including checks related to:

- overall GPU health
- PCIe

- power
- SM
- thermal
- PMU

This makes BCM especially relevant for DGX and other GPU cluster environments where hardware acceleration health directly affects workload success. ([NVIDIA Docs][2])

## **CUDA and Software Environment Management**

### **Multiple CUDA Versions**

BCM supports management of installed CUDA versions and environment control through modules. \* NVIDIA documentation shows multiple CUDA versions being available in the environment and selected as needed. \* This matters in real AI/HPC environments because different applications may require different CUDA versions. BCM supports that kind of controlled multi-version environment rather than forcing one CUDA stack for everyone. ([NVIDIA Docs][3])

### **Environment Modules**

BCM supports environment modules, with Tcl Modules as the default and Lmod as an alternative. \* Administrators can configure default modules cluster-wide. \* This is an important usability feature because it standardizes how users load software stacks such as compilers, libraries, and CUDA-related components. ([NVIDIA Docs][3])

### **Accounting and Chargeback Job Accounting**

BCM supports job accounting, which NVIDIA distinguishes from raw job monitoring. \* Monitoring focuses on resource consumption per job as it runs. \* Accounting aggregates resource consumption by groupings such as:

- user
- group
- account
- job-related dimensions

This gives administrators and organizations a way to understand resource use at an ownership or cost-center level. ([NVIDIA Docs][2])

### **Chargeback**

BCM includes chargeback reporting capabilities. \* NVIDIA documents predefined chargeback models based on metrics such as:

- CPU second
- GPU second
- CPU core second
- slot second
- memory byte-second
- power watt-second

Chargeback reports can also be grouped by fields such as user, group, account, job name, and job ID. \* That makes BCM useful not just for technical operations, but also for internal cost allocation and

usage reporting. ([NVIDIA Docs][2])

## Takeaways

BCM is best understood as **end-to-end cluster lifecycle management**, not just provisioning software. \* It combines:

- installation
- provisioning
- software image control
- workload manager integration
- monitoring
- health checks
- scaling
- accounting
- GPU-aware operations ([NVIDIA Docs][1])

## What Makes BCM Strong

The biggest strengths of BCM are:

- centralized management
- repeatable software image-based provisioning
- integrated workload and infrastructure monitoring
- support for dynamic scaling
- cloud extension and hybrid models
- edge deployment support
- deep NVIDIA GPU integration ([NVIDIA Docs][7])

## High-Value Concepts to Remember

- **Head node** is the main management and provisioning control point.
- **Software image** is the blueprint used to provision nodes.
- **Base View** is the main GUI; **cmsh** is the CLI.
- **Pre-job health check failure** can block a job from running on a node.
- **cm-scale** supports dynamic scaling and node repurposing.
- **Cluster Extension** keeps the head node outside the cloud while adding cloud capacity.
- **cuda-dcgm** is important for NVIDIA GPU metrics and GPU-aware monitoring. ([NVIDIA Docs][2])

[1]: <https://docs.nvidia.com/base-command-manager/bcm-11-release-notes/overview.html>  
"https://docs.nvidia.com/base-command-manager/bcm-11-release-notes/overview.html"

[2]: <https://docs.nvidia.com/base-command-manager/manuals/11/admin-manual.pdf>  
"https://docs.nvidia.com/base-command-manager/manuals/11/admin-manual.pdf"

[3]: <https://docs.nvidia.com/base-command-manager/manuals/11/installation-manual.pdf>  
"https://docs.nvidia.com/base-command-manager/manuals/11/installation-manual.pdf"

[4]: <https://docs.nvidia.com/base-command-manager/manuals/11/containerization-manual.pdf>  
"https://docs.nvidia.com/base-command-manager/manuals/11/containerization-manual.pdf"

[5]: <https://docs.nvidia.com/base-command-manager/manuals/11/cloudbursting-manual.pdf>  
"https://docs.nvidia.com/base-command-manager/manuals/11/cloudbursting-manual.pdf"

[6]: <https://docs.nvidia.com/base-command-manager/manuals/11/edge-manual.pdf>

“<https://docs.nvidia.com/base-command-manager/manuals/11/edge-manual.pdf>”

[7]: <https://docs.nvidia.com/datacenter/dps/versions/latest/guides/concepts/base-command-manager/>

“<https://docs.nvidia.com/datacenter/dps/versions/latest/guides/concepts/base-command-manager/>”

[BCM - Back to BCM page](#)

[Back to NVIDIA page](#)