CDW Documentation

nvidia_bcm-bringing-up-cluster

Back to NVIDIA page

Back to BCM page

Topic: Bringing Up the Cluster (Post-Installation Tasks)

What This Unit Covers

* The complete post-installation workflow required to move a newly installed NVIDIA Base Command Manager cluster from a basic head-node installation into a usable, provisioned cluster * What must be done on the head node before compute nodes can be safely powered on and provisioned * Licensing, software updates, software image preparation, category design, node assignment, identification, and provisioning flow * Common operational mistakes that slow down bring-up, especially around licensing, image changes, and node identification

1. **What “Bringing Up the Cluster” Means**

* In NVIDIA BCM, cluster bring-up is the work that happens after the base installation completes and the head node reboots successfully * The installation gives you a working head node, but it does not mean the full cluster is ready yet * At this stage, the administrator still needs to install a valid cluster license, update software, prepare or protect software images, define how nodes should inherit configuration, and then identify and provision the physical nodes * A practical way to think about bring-up is: turning a newly installed BCM head node into a trusted source of software, identity, and configuration for the rest of the cluster * NVIDIA deployment guidance consistently treats this as a structured sequence: access the head node, license the cluster, back up or clone images, assign categories and software images, define interfaces and MAC identity, then power on and provision nodes

High-level bring-up workflow

* Log in to the head node * Install and verify the license * Update the head node * Update the node software image * Clone or back up the software image before major changes * Clone or create the node category you want to use * Set the category’s **softwareimage** property * Move nodes into the correct category * Power on nodes * Ensure they PXE boot * Identify the nodes correctly * Let BCM provision them with the assigned image and configuration ([NVIDIA Docs][1])

2. **Step 1: Log Into the Head Node**

* After installation and reboot, BCM presents a login prompt on the head node * NVIDIA’s installation guidance states that the administrator logs in as **root** using the password created during installation * NVIDIA also states that after reboot, the cluster should be updated with the latest packages before further configuration continues * That means successful login is not the end of installation — it is the handoff point into post-install configuration and bring-up

3. **Step 2: Install the Cluster License**

* Licensing is a required bring-up step because BCM must have a valid cluster license before normal cluster management can proceed * NVIDIA documents the license file as the cluster certificate or head-node certificate, and identifies it as the head node’s **X509v3 certificate** * The certificate contains important licensing attributes such as start date, expiration date, MAC address or Cloud ID, licensed token count, and license type * One of the most important fields is **Licensed tokens**. NVIDIA states that this is the maximum number of GPUs BCM may manage, and also the maximum number of nodes the license covers. Head nodes count toward that node limit as well * The license is tied to

the head node’s MAC address. In HA configurations, the license can be tied to two MAC addresses so that licensing survives head-node failover * NVIDIA’s current deployment guides also note that for HA, you should use a stable onboard NIC MAC rather than a removable NIC if possible, because replacing that NIC later can break license detection

Default / initial license behavior

* NVIDIA’s BCM installation manual says evaluation ISO downloads include a **temporary built-in license** * That temporary built-in license allows a **2-node cluster** to be tried out * In practice, that is a minimal evaluation state and is not the target end state for a production cluster bring-up

How license installation works

* The standard method is to run **request-license** from the head node and provide the product key * If the cluster has direct internet access, the request can be submitted online and the license can be obtained and activated directly * If the cluster is air-gapped or does not have direct web access, NVIDIA documents an off-cluster workflow: answer **No** to online submission, generate the CSR, and then process that CSR from an off-cluster browser * NVIDIA says the CSR is saved to **/cm/local/apps/cmd/etc/cluster.csr.new**, while the private key remains on the cluster and should not leave it * If **request-license** is allowed to install immediately, it automatically runs **install-license**; otherwise, the administrator can install the returned certificate manually with **install-license /cm/local/apps/cmd/etc/cluster.pem.new**

What happens after license installation

* Installing the license updates BCM with the new certificate and restarts CMDaemon so the license becomes active * The first time a product key is used and a new license is installed, NVIDIA says all nodes must be rebooted so they can pick up new certificates * NVIDIA explicitly notes that **pdsh -g computenode reboot** is the recommended parallel reboot command for those compute nodes * On later license changes, behavior depends on whether existing keys are reused * If existing keys are reused, compute-node reboots may not be required * If existing keys are not reused, existing certificates become invalid and node communication stops until the nodes are rebooted * This is why it is best to install the proper license early in bring-up, before the cluster is already busy running jobs

How to verify the license

* NVIDIA documents **verify-license info** as the main utility for printing license details * NVIDIA also documents **licenseinfo** in cmsh as another way to verify license attributes * In HA, NVIDIA notes that `cmsh -c “main licenseinfo”` can be used to verify that both MAC addresses are present * Useful attributes to verify include start date, expiration date, MAC binding, license type, and licensed token count

4. **Step 3: Update the Head Node OS**

* NVIDIA states that after rebooting from installation, the cluster should be updated with the latest packages * This matters because early bring-up problems can come from outdated BCM components, node-installer components, or OS packages * NVIDIA specifically recommends deploying available updates on the cluster in general * On the head node, NVIDIA says relevant software can be updated with the distribution package manager such as **yum**, **zypper**, or **apt** * This is not just hygiene — NVIDIA places software updates at the front of troubleshooting guidance for failed node boot and provisioning issues

5. **Step 4: Update the Software Image**

* NVIDIA states that the entries inside **/cm/images/** are the software images used to provision nodes * That means the software image is not just a reference copy of an OS — it is the file tree BCM uses when nodes are installed or synchronized * Because of that, updating the image is a major part of bring-up * For Ubuntu-based images, NVIDIA recommends entering the image with **cm-chroot-sw-img /cm/images/<image>** and then running **apt update; apt upgrade** * For yum-based images, NVIDIA documents updating the image through the image path with commands such as **yum update –installroot=/cm/images/<software image>** * Any changes to a software image can affect every node that uses it, so image work must be done carefully and deliberately

Important note about image changes

* NVIDIA also notes that when kernel modules or similar low-level components are changed in an image, BCM regenerates the initial ramdisk * That regeneration can take some time, and it is part of why administrators should treat image modification as a controlled operation rather than a casual edit

6. **Step 5: Clone the Software Image**

* Best practice is to avoid making major changes directly to the only working image you have * NVIDIA explicitly describes cloning images before modifying them, and also describes creating a backup image as a safety step * The reason is simple: software images in **/cm/images/** are the exact file trees used to provision nodes, so breaking one can directly affect future node installs and reprovisions * NVIDIA’s cluster bring-up and administrator guidance both support the idea of cloning or backing up a known-good image first, then working on the clone * This gives you a rollback point and preserves a clean baseline if your changes introduce package, driver, kernel, or configuration issues ([NVIDIA Docs][2])

7. **Step 6: Clone the Node Category**

* NVIDIA states that by default, all nodes are placed in the **default** category * Categories are important because they are one of the main ways BCM applies shared configuration to groups of nodes * The safest workflow is usually to leave **default** alone as a baseline and clone it when you want a new behavior set * NVIDIA shows this pattern directly in examples: clone the category, then set the cloned category’s software image * This is better than overloading the default category with experimental changes because it keeps a stable reference point and makes testing easier

8. **Step 7: Assign the Software Image to the Category**

* NVIDIA shows that after creating or cloning the target category, the next step is to set its **softwareimage** property * This is the mechanism that tells BCM which image nodes in that category should receive * NVIDIA’s examples state this explicitly: all nodes assigned to a given category will be provisioned with that category’s software image * Also important: BCM settings follow an inheritance model. Category settings apply broadly, but NVIDIA notes that individual node settings can override category settings when needed * That means categories define the normal rule, while node-level settings are exceptions ([NVIDIA Docs][3])

9. **Step 8: Assign Nodes to the Category**

* Since nodes start in **default** by default, they must be moved into the category that represents their intended role * NVIDIA shows multiple ways this can happen during cluster bring-up: * A node identity can be cloned from an existing node template * The node’s **category** can then be set to the new category * Additional nodes can later be cloned from the first validated node identity * This reflects an important operational pattern: validate one node first, then scale out from a known-good node object and category design * NVIDIA also notes that newly cloned categories have no nodes initially, which

reinforces that node assignment is a deliberate step and not automatic ([NVIDIA Docs][3])

10. **Step 9: Power On, Identify, and Provision Nodes**

* NVIDIA documents that nodes can be powered on through BCM power operations, BMC access, or IPMI-based methods * BCM supports powering individual nodes, lists of nodes, categories, groups, racks, and more * Once powered on, the node must attempt a network boot * For x86 nodes, this means PXE boot must be configured correctly in the BIOS or boot order * NVIDIA's troubleshooting guidance specifically says the boot interface should normally be first in the BIOS, and that wrong boot order, wrong cable, rogue DHCP, disabled node booting, or locked-down DHCP can all stop bring-up before provisioning even starts

Node identification is critical

* Before a node can be provisioned correctly, BCM must know which physical machine corresponds to which BCM node object * NVIDIA describes the device identification resource as assigning MAC addresses so node names are associated with MAC addresses * If a node is left with an unassigned MAC address, BCM treats it as a placeholder and the administrator must associate identity during the node configuration stage * NVIDIA documents that this can be handled through the node console, through the **newnodes** workflow in cmsh, or through switch-port-based detection and confirmation flows * This is why MAC identity is not a minor detail — it is how BCM knows which node object to serve during provisioning

Provisioning flow

* If the node's boot options are correct, NVIDIA says it should attempt PXE booting * The node loads the installer environment and then the Cluster Manager Node Installer environment * If the identity is recognized successfully, BCM can confirm the node configuration and continue automated provisioning * From there, the node installs or synchronizes from the assigned software image and configuration served by the head node * NVIDIA also notes that provisioning progress can be monitored from the head node through **cmsh** * If provisioning fails, NVIDIA's BasePOD bring-up guide points administrators to **/var/log/messages** and **/var/log/node-installer** for diagnosis ([NVIDIA Docs][2])

Advanced practical note

* In DGX BasePOD bring-up workflows that use bonded provisioning interfaces, NVIDIA enables **DeviceResolveAnyMAC=1** in CMDaemon to support provisioning and failover PXE booting on bonded downstream interfaces * That is more deployment-specific than the core generic BCM workflow, but it is a useful note if bonded provisioning is involved ([NVIDIA Docs][2])

Key Takeaways

* Bring-up is the structured post-install phase that turns a freshly installed head node into a provision-ready cluster * The head node should be accessed as **root**, then updated, licensed, and prepared before broad node rollout begins * The BCM license is a head-node certificate tied to one MAC address, or two in HA, and the **Licensed tokens** value governs the maximum GPUs and nodes BCM may manage * The evaluation state is only a temporary 2-node cluster license, so real cluster bring-up requires the proper installed license * **/cm/images/** contains the software images that are actually used to provision nodes, so image changes are high impact * Leave the baseline intact when possible: clone or back up working images, and clone categories instead of turning **default** into your experiment zone * Nodes should inherit their intended image through their category, with node-level overrides used only when necessary * If a node does not PXE boot or is not correctly identified,

provisioning will stall no matter how good the image and category design are * When provisioning problems happen, check BIOS boot order, DHCP behavior, PXE readiness, MAC identity, and the node-installer logs first

Send the next set whenever you’re ready.

[1]: https://docs.nvidia.com/dgx-basepod/deployment-guides/dgx-basepod-a100/latest/deployment-configure.html “Cluster Configuration — NVIDIA DGX BasePOD: Deployment Guide Featuring NVIDIA DGX A100 Systems” [2]: https://docs.nvidia.com/dgx-basepod/deployment-guide-dgx-basepod/latest/cluster-bringup.html “Cluster Bring Up — NVIDIA DGX BasePOD: Deployment Guide Featuring NVIDIA DGX H200/H100 Systems” [3]: https://docs.nvidia.com/dgx-basepod/deployment-guide-basepod-rhel/latest/head-node.html “Head Node Configuration — NVIDIA DGX BasePOD on RHEL: Deployment Guide Featuring NVIDIA DGX A100”

Back to NVIDIA page

Back to BCM page