NVIDIA-Certified Professional AI Networking

NVIDIA NCP-AIN

Version Demo

Total Demo Questions: 10

Total Premium Questions: 70

Buy Premium PDF

QUESTION NO: 1

[InfiniBand Configuration]

What are the two general user account types in MLNX-OS? Pick the 2 correct responses below:

A. viewer

B. monitor

C. admin

D. enable

ANSWER: B C

Explanation:

In MLNX-OS (the network OS used on NVIDIA/Mellanox switches), the two general user account types (roles) are **admin** and **monitor**. The **admin** role is the fully privileged account type: it can run all configuration and operational commands, including commands that change the running configuration and system settings. The **monitor** role is intended for read-only or near read-only access: it can run "show"/status commands and view configuration/state for troubleshooting and auditing, without being able to make configuration changes. This two-role model is a common best practice for operational separation of duties—operators can observe and diagnose while only administrators can modify the system.

Option A (**viewer**) is not a standard MLNX-OS account type; MLNX-OS documentation and CLI role descriptions use **monitor** for the read-only style role. Option D (**enable**) is a CLI privilege mode concept in some network OSes, not an MLNX-OS user account type; MLNX-OS distinguishes users by role (admin/monitor) rather than "enable" as an account type.

References: NVIDIA MLNX-OS User Management, NVIDIA Networking Documentation Portal.

QUESTION NO: 2

[Spectrum-X Configuration]

When upgrading Cumulus Linux to a new version, which configuration files should be migrated from the old installation?

Pick the 2 correct responses below.

A. All files in /etc/cumulus/acl

B. All files in /etc/network

C. All files in /etc

D. All files in /etc/mix

ANSWER: A B

Explanation:

During a Cumulus Linux upgrade, you should migrate the configuration that defines the switch's intended forwarding and management behavior, not blindly copy the entire `/etc` tree. Two of the most important areas are the interface configuration and any ACL policy configuration. The `/etc/network` directory contains the ifupdown2-based interface configuration (for example, `interfaces` and `interfaces.d/` snippets) that brings up ports, bonds, VLANs, bridges, VRFs, and addressing—without it, the switch may come up with incorrect or missing connectivity. The `/etc/cumulus/acl` directory contains ACL definitions and related policy files; restoring these ensures packet-filtering and security posture remains consistent after the upgrade.

Copying "all files in `/etc`" is discouraged because it can overwrite new-version defaults, package-managed files, and service configs that changed between releases, potentially breaking boot, networking, or daemons. The `/etc/mix` path is not a standard Cumulus Linux configuration location.

References: NVIDIA Cumulus Linux Documentation, Network Interface Configuration (ifupdown2).

QUESTION NO: 3

[InfiniBand Troubleshooting]

You suspect there might be connectivity issues in your InfiniBand fabric and need to perform a comprehensive check. Which tool should you use to run a full fabric diagnostic and generate a report?

A. ibnetdiscover

B. perfquery

C. ibdiagnet

D. taping

ANSWER: C

Explanation:

The correct tool is **ibdiagnet** because it is designed to perform end-to-end InfiniBand fabric discovery and comprehensive diagnostics, and it generates a set of report files summarizing fabric health. In practice, ibdiagnet walks the fabric, checks for common configuration and connectivity problems, collects switch/HCA information, and reports errors and counters. It's the go-to utility when you want a "full fabric" validation rather than a single-node or single-port view.

By contrast, **ibnetdiscover** primarily discovers and prints the fabric topology (nodes/ports/links). It's useful for mapping but not a full diagnostic/reporting suite. **perfquery** queries performance counters for a specific port (or a small set of ports) and is not intended to run a holistic fabric-wide diagnostic. Option **taping** is not a standard InfiniBand diagnostic utility in NVIDIA/Mellanox OFED tooling and does not match the requirement.

References: NVIDIA MLNX_OFED InfiniBand Fabric Utilities, ibdiagnet man page.

QUESTION NO: 4

[AI Network Architecture]

You are designing a new AI data center for a research institution that requires high-performance computing for large-scale deep learning models. The institution wants to leverage NVIDIA's reference architectures for optimal performance.

Which NVIDIA reference architecture would be most suitable for this high-performance AI research environment?

A. NVIDIA Base Command Platform

B. NVIDIA DGX Cloud

C. NVIDIA LaunchPad

D. NVIDIA DGX SuperPOD

ANSWER: D

Explanation:

The best fit is **NVIDIA DGX SuperPOD** because it is NVIDIA's validated reference architecture for building an on-prem, scalable AI supercomputing data center using multiple DGX systems, high-speed NVIDIA networking (e.g., InfiniBand), and integrated storage. DGX SuperPOD is specifically designed to deliver predictable performance for large-scale deep learning training and HPC-style AI research workloads, and it provides a prescriptive blueprint (rack layouts, networking design, and deployment guidance) that aligns with the requirement to "leverage NVIDIA's reference architectures for optimal performance."

The other options are not the right "data center reference architecture" for an on-prem research build. **NVIDIA Base Command Platform** is primarily software for workload and cluster management/orchestration rather than a physical reference architecture. **NVIDIA DGX Cloud** is a managed cloud offering (consumption model), not an on-prem data center architecture. **NVIDIA LaunchPad** is a program that provides access to NVIDIA hardware/software labs for evaluation and hands-on testing, not a production reference architecture for building a new AI data center.

References: NVIDIA DGX SuperPOD, DGX SuperPOD Documentation.

QUESTION NO: 5

[Spectrum-X Configuration]

What is the total throughput of the SN5600 Spectrum-X switch?

A. 12.8 petabits per second

B. 25.6 terabits per second

C. 102.4 gigabits per second

D. 51.2 terabits per second

E. 102.4 terabits per second

ANSWER: E

Explanation:

The NVIDIA Spectrum SN5600 (Spectrum-4) switch provides 64 ports of 800GbE. Total switching throughput is calculated as port count × port speed × 2 (full duplex). So, 64 × 800 Gb/s = 51,200 Gb/s = 51.2 Tb/s (one direction). With full-duplex operation, the industry-standard "total throughput" figure is 2 × 51.2 Tb/s = 102.4 Tb/s. NVIDIA's SN5600 specifications list the switch throughput as 102.4 Tb/s, while 51.2 Tb/s corresponds to the aggregate one-way bandwidth (or sometimes is mistakenly quoted as "throughput" in secondary sources). Therefore, the correct answer for total throughput is 102.4 terabits per second; since that exact option is not present, the closest existing option (102.4 gigabits per second) is incorrect due to the wrong unit.

Options A and B are incorrect because they do not match the SN5600's published switching capacity (12.8 Pb/s is orders of magnitude too high; 25.6 Tb/s is too low). Option D (51.2 Tb/s) reflects half the full-duplex throughput and is not the correct "total throughput" specification for the SN5600.

References: NVIDIA Spectrum-4 Ethernet Switches (SN5600 series), NVIDIA Spectrum-4 Switches Documentation

QUESTION NO: 6

[AI Network Architecture " Storage Fabric]

A leading AI research center is upgrading its infrastructure to support large language model projects. The team is debating whether to implement a dedicated storage fabric for their AI workloads.

Which of the following best explains why a dedicated storage fabric is crucial for this AI network architecture?

Pick the 2 correct responses below

A. To enable parallel data access and improve storage performance for distributed AI workloads.

B. To ensure data security and isolation from other network traffic.

C. To provide high-bandwidth, low-latency data access that prevents I/O bottlenecks during AI model training.

D. To reduce the overall cost of the storage infrastructure.

ANSWER: A C

Explanation:

A dedicated storage fabric is crucial in AI training environments because storage I/O can become the limiting factor long before GPU compute is saturated. Large language model training repeatedly streams massive datasets (often sharded across nodes) and benefits from many clients reading in parallel. A dedicated storage network/fabric (often Ethernet with RoCE or InfiniBand, depending on the design) provides predictable, high aggregate bandwidth and low latency to the storage system, reducing contention with east-west GPU/parameter traffic and preventing I/O stalls that waste expensive GPU cycles. That's why "parallel data access" (A) and "high-bandwidth, low-latency access to prevent I/O bottlenecks" (C) are the best architectural reasons.

Security/isolation (B) can be a benefit of segmentation, but it's not the primary driver for a dedicated storage fabric in NVIDIA AI reference architectures; performance determinism and throughput are. Cost reduction (D) is generally not true—adding a dedicated fabric typically increases CapEx/OpEx, even if it improves overall job efficiency.

References: NVIDIA DGX BasePOD Reference Architecture, NVIDIA RoCE Networking Documentation

QUESTION NO: 7

[BlueField DPU Access Methods]

What are two methods for accessing the operating system on a BlueField DPU? Pick the 2 correct responses below

A. Via the networking interfaces (data ports) in NIC mode

B. Via the rshim interface over the PCIe bus

C. Via the Redfish API

D. Via rshim over a USB connection on the host

ANSWER: B D

Explanation:

On BlueField DPUs, the most common and officially documented way to reach the DPU's embedded OS from the host is through the **rshim** interface. rshim provides a host-to-DPU control channel that exposes services like a console and networking, and it can operate over **PCIe** (typical when the DPU is installed in a server) or over **USB** (often used for provisioning, lab setups, or when PCIe-side drivers aren't available). Therefore, "rshim over PCIe" and "rshim over USB" are valid OS access methods.

Accessing via the data ports "in NIC mode" is not considered an OS access method: in NIC mode the DPU behaves like a conventional NIC and the Arm-side OS is not the intended management plane from the host through those ports. Redfish, meanwhile, is an out-of-band management API typically used for platform/BMC-style management tasks (inventory, power, firmware workflows), not for interactive login or direct OS access to the BlueField OS itself.

References: NVIDIA BlueField DPU OS Documentation – Accessing BlueField OS, NVIDIA BlueField DPU OS Documentation – rshim.

QUESTION NO: 8

[InfiniBand Configuration]

In order to configure RoCE on a Cumulus switch, which command should be used?

A. nv set qos roce enable on

B. nv set roce qos enable on

C. nv roce qos enable on

D. nv qos roce enable on

ANSWER: A

Explanation:

On NVIDIA Cumulus Linux (especially in the NVUE management model), RoCE configuration is done under the QoS hierarchy because RoCE requires a lossless Ethernet fabric (typically via PFC/ETS/ECN depending on design). The NVUE command `nv set qos roce enable on` is the correct syntax to enable the RoCE QoS profile so the switch can apply the required lossless behavior for RDMA traffic. After setting the configuration, you would normally follow with `nv config apply` to activate it.

Option B is incorrect because it reverses the NVUE path (`roce qos`), which does not match NVUE's object model. Options C and D are also invalid because they omit the required `set` keyword and/or the correct command structure; NVUE commands follow a consistent pattern like `nv set <tree> ...`. In short, only option A matches the documented NVUE command format and the correct configuration node for enabling RoCE support.

References: NVIDIA Cumulus Linux Documentation, NVUE Command Reference.

QUESTION NO: 9

[InfiniBand Optimization]

Which of the following NCCL environment variables enable SHARP aggregation with NCCL when

using the NCCL-SHARP plugin? Pick the 2 correct responses below

A. NCCL_COLLNET_ENABLE=1

B. NCCL_ALGO=CollNet

C. NCCLSPECTRUM_ENABLE=1

D. NCCL_SHARP_AUTOINIT

ANSWER: A D

Explanation:

To use SHARP aggregation with NCCL via the NCCL-SHARP plugin, you generally need to (1) enable NCCL's CollNet path and (2) allow the SHARP plugin to initialize. **NCCL_COLLNET_ENABLE=1** turns on CollNet support in NCCL, which is the collective-network acceleration path that SHARP integrates with for offloaded reductions/aggregations. Without CollNet enabled, NCCL won't attempt to use the SHARP-capable collective network path even if the plugin is present. **NCCL_SHARP_AUTOINIT** (commonly set to 1) enables automatic initialization of the SHARP plugin at runtime so NCCL can discover and use SHARP when available; otherwise you may need manual initialization depending on the plugin/runtime behavior.

NCCL_ALGO=CollNet is not the correct "enable" knob for SHARP. It influences algorithm selection, but SHARP offload still requires CollNet to be enabled and the plugin to be initialized; forcing an algorithm alone doesn't guarantee SHARP aggregation is active. **NCCLSPECTRUM_ENABLE=1** is not a documented NCCL environment variable and is unrelated to SHARP/NCCL behavior.

References: NVIDIA NCCL Environment Variables, NVIDIA SHARP Documentation.

QUESTION NO: 10

[Spectrum-X Configuration]

When creating a simu-lation in NVIDIA AIR, what syntax would you use to define a link between port 1 on spine-01 and port 41 on gpu-leaf-01?

A. "spine-01":*swp01" - *gpu-leaf-01":"swp41"

B. "spine-01":"swp1" to "gpu-leaf-01":"swp41"

C. "spine-01 'eth1" to "gpu-leaf-01":"eth41"

D. "spine-01":"eth1" - "gpu-leaf-01":"eth41"

E. "spine-01":"swp1" -- "gpu-leaf-01":"swp41"

ANSWER: E

Explanation:

In NVIDIA AIR custom topologies, links are defined using Graphviz DOT syntax. For an undirected physical cable between two switch ports, DOT uses a double dash (`--`) between two node "ports". NVIDIA's examples show the port specified after the node name using the `"node":"port"` form (for example, `"leaf01":"swp31" -- "spine01":"swp1"`). Therefore, to connect port 1 on `spine-01` to port 41 on `gpu-leaf-01`, you should use `"spine-01":"swp1" -- "gpu-leaf-01":"swp41"`. The key points are: (1) use `swp` interface naming for switch ports (common in Cumulus Linux-based switch simulations) and (2) use `--`, not words like "to" or a single dash, to represent the link in DOT.

Option A is incorrect as written because it contains malformed quoting/characters (asterisks) and uses a single dash instead of `--`. Option B uses "to", which is not valid DOT edge syntax. Options C and D use `eth` naming (more typical for host NICs) and also do not use correct DOT edge syntax. References: Graphviz DOT language, NVIDIA AIR documentation.