DUMPSBOSS.

NVIDIA AI Operations

NVIDIA NCP-AIO

Version Demo

Total Demo Questions: 9

Total Premium Questions: 90

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, AI Infrastructure	39
Topic 2, AI Workloads	7
Topic 3, AI Operations	40
Topic 4, NVIDIA AI Enterprise	4
Total	90

QUESTION NO: 1

Which two (2) ways does the pre-configured GPU Operator in NVIDIA Enterprise Catalog differ from the GPU Operator in the public NGC catalog? (Choose two.)

A. It is configured to use a prebuilt vGPU driver image.

B. It supports Mixed Strategies for Kubernetes deployments.

C. It automatically installs the NVIDIA Datacenter driver.

D. It is configured to use the NVIDIA License System (NLS).

E. It additionally installs Network Operator.

ANSWER: A D

Explanation:

The pre-configured NVIDIA Enterprise Catalog GPU Operator is tailored for NVIDIA AI Enterprise deployments that use virtual GPUs and enterprise licensing. “It is configured to use a prebuilt vGPU driver image.” is correct because vGPU environments require a compatible NVIDIA vGPU software driver rather than the standard driver-installation approach generally used by the public GPU Operator catalog offering. Providing the vGPU driver image as part of the preconfigured enterprise deployment helps align the operator with the validated vGPU software stack required for virtualized GPU workloads.

“It is configured to use the NVIDIA License System (NLS).” is also correct because NVIDIA AI Enterprise and vGPU deployments use NVIDIA’s licensing infrastructure to obtain and manage feature entitlements. The Enterprise Catalog configuration includes the licensing-oriented settings needed for that environment, including use of the NVIDIA License System. Together, the prebuilt vGPU driver and NLS configuration distinguish this enterprise-focused operator packaging from the general public NGC GPU Operator offering. NVIDIA’s GPU Operator documentation describes support for vGPU deployments and the required driver configuration, while the NVIDIA AI Enterprise documentation describes licensing through NLS. See NVIDIA GPU Operator platform documentation and NVIDIA AI Enterprise User Guide.

QUESTION NO: 2

You are an administrator managing a large-scale Kubernetes-based GPU cluster using Run:AI.

To automate repetitive administrative tasks and efficiently manage resources across multiple nodes, which of the following is essential when using the Run:AI Administrator CLI for environments where automation or scripting is required?

A. Use the runai-adm command to directly update Kubernetes nodes without requiring kubectl.

B. Use the CLI to manually allocate specific GPUs to individual jobs for better resource management.

C. Ensure that the Kubernetes configuration file is set up with cluster administrative rights before using the CLI.

D. Install the CLI on Windows machines to take advantage of its scripting capabilities.

ANSWER: C

Explanation:

Ensure that the Kubernetes configuration file is set up with cluster administrative rights before using the CLI. The Run:AI Administrator CLI operates against the Kubernetes cluster through the credentials and API-server connection details supplied by the active kubeconfig context. For scripted or automated administration, that context must authenticate successfully and have authorization broad enough to perform the cluster-scoped actions the automation requires, such as administering Run:AI components and managing relevant cluster resources. Cluster-administrative rights provide the required scope for administrative workflows and prevent automation from failing partway through because the identity lacks permission for a required Kubernetes API operation. In practice, administrators should use a dedicated, protected kubeconfig or service identity, select the intended context explicitly in automation, and apply least-privilege controls where operational requirements permit. Kubernetes authorization is enforced through RBAC, so the permissions associated with the kubeconfig identity—not merely the presence of the CLI binary—determine whether commands can complete. NVIDIA

Run:AI documentation provides the administrative documentation and CLI guidance at NVIDIA Run:AI Documentation. For the Kubernetes authorization model underlying those credentials, see Kubernetes RBAC documentation.

QUESTION NO: 3

A Fleet Command system administrator wants to create an organization user that will have the following rights:

For locations - read only

For Applications - read/write/admin

For Deployments - read/write/admin

For Dashboards - read only

What role should the system administrator assign to this user?

A. Fleet Command Operator

B. Fleet Command Admin

C. Fleet Command Supporter

D. Fleet Command Viewer

ANSWER: A

Explanation:

Fleet Command Operator is the appropriate organization role because it grants the operational permissions needed to manage the application and deployment lifecycle without granting broad administration of the entire Fleet Command organization. An Operator can create, modify, and administer applications and deployments, which supports tasks such as onboarding application versions, configuring deployment settings, targeting systems, and monitoring deployment activity. At the same time, the role provides read-level access to locations and dashboards. This lets the user inspect available locations and use dashboard information for operational awareness while keeping location configuration and dashboard administration under more restricted roles. This permission combination follows the principle of least privilege: the user receives full control where day-to-day application deployment work is required, but only visibility where configuration changes are not needed. NVIDIA documents Fleet Command's organization roles and their resource-specific permissions in its user-management guidance. See the NVIDIA Fleet Command User Management documentation and the NVIDIA Fleet Command documentation for current role and access-control details.

QUESTION NO: 4

An administrator plans to change the MIG layout on a GPU node that is managed by Slurm. The node currently runs production jobs, and Slurm is configured to schedule the existing MIG instances.

What two (2) actions are required to prevent incorrect allocations after the MIG change? (Choose two.)

A. Drain the node and ensure that active jobs have completed or been handled before changing the MIG layout.

B. Update or validate Slurm GRES discovery and configuration against the newly exposed MIG resources.

C. Increase the priority of jobs that request MIG resources.

D. Change only the GPU count in each user job script after the layout change.

E. Disable Slurm accounting so that the old resource records are ignored.

ANSWER: A B

Explanation:

The node should first be drained so that Slurm does not start additional work on hardware whose resource layout is about to change. Existing jobs must be allowed to complete or handled according to the maintenance policy before the MIG configuration is modified.

After the new layout is created, Slurm resource discovery and GRES configuration must match the GPU and MIG instances actually exposed by the node. Otherwise, Slurm can advertise stale resources and allocate instances that no longer exist. Increasing job priority or changing only a user job script does not correct the scheduler-to-hardware mismatch.

QUESTION NO: 5

When troubleshooting Slurm job scheduling issues, a common source of problems is jobs getting stuck in a pending state indefinitely.

Which Slurm command can be used to view detailed information about all pending jobs and identify the cause of the delay?

A. scontrol

B. sacct

C. sinfo

D. `squeue --states=PENDING --format="%.18i %.2t %.30R"`

ANSWER: D

Explanation:

`squeue --states=PENDING --format="%.18i %.2t %.30R"` is the appropriate command because `squeue` queries the Slurm job queue and can filter the display to jobs in the pending state. The `--states=PENDING` filter limits the results to the jobs that are waiting to run, while the custom output format includes the job identifier, state, and the reason field. The reason field is particularly useful for scheduling troubleshooting because Slurm reports why each job is pending, such as unavailable resources, priority, dependency, partition configuration, node state, or a job limit. This lets an administrator review all affected jobs in one command and quickly determine whether the delay is systemic or specific to a job or partition.

The Slurm `squeue` documentation describes state filtering and output formatting, including the `%R` format code for a pending job's reason: Slurm squeue documentation. Slurm also documents pending-job reason codes and their meanings in its scheduling documentation: Slurm job reason codes.

QUESTION NO: 6

You are tasked with deploying a deep learning framework container from NVIDIA NGC on a stand-alone GPU-enabled server.

What must you complete before pulling the container? (Choose two.)

A. Install Docker and the NVIDIA Container Toolkit on the server.

B. Set up a Kubernetes cluster to manage the container.

C. Install TensorFlow or PyTorch manually on the server before pulling the container.

D. Generate an NGC API key and log in to the NGC container registry using docker login.

ANSWER: A D

Explanation:

Installing Docker and the NVIDIA Container Toolkit on the server prepares the stand-alone GPU system to obtain and run an NGC deep learning container with NVIDIA GPU acceleration. Docker supplies the container engine and `docker` command used to retrieve images from the registry. The NVIDIA Container Toolkit configures the container runtime so that, when the

downloaded framework container is launched, it can discover and use the host NVIDIA driver and GPUs. NVIDIA's installation guidance identifies the toolkit as the supported mechanism for enabling GPU-accelerated containers on Docker hosts.

Generating an NGC API key and logging in to the NGC container registry using `docker login` establishes authenticated registry access before the image pull. NGC uses the API key as the registry login password with the special username `$oauthtoken`; this authentication authorizes Docker to download container images available to the user or organization. Completing the login first avoids an interrupted workflow and ensures that the required registry credentials are available to Docker for the pull operation. NVIDIA documents both the API-key-based registry login procedure and the host configuration required for GPU containers in its official NGC and Container Toolkit documentation: NVIDIA NGC User Guide: Generating an API Key and NVIDIA Container Toolkit Installation Guide.

QUESTION NO: 7

Which of the following correctly identifies the key components of a Kubernetes cluster and their roles?

A. The control plane consists of the kube-apiserver, etcd, kube-scheduler, and kube-controller-manager, while worker nodes run kubelet and kube-proxy.

B. Worker nodes manage the kube-apiserver and etcd, while the control plane handles all container runtimes.

C. The control plane is responsible for running all application containers, while worker nodes manage network traffic through etcd.

D. The control plane includes the kubelet and kube-proxy, and worker nodes are responsible for running etcd and the scheduler.

ANSWER: A

Explanation:

The control plane consists of the kube-apiserver, etcd, kube-scheduler, and kube-controller-manager, while worker nodes run kubelet and kube-proxy. This accurately describes the standard Kubernetes architecture. The control plane provides the cluster's management and decision-making functions: kube-apiserver exposes the Kubernetes API and acts as the central communication hub; etcd persistently stores cluster state and configuration; kube-scheduler selects suitable nodes for newly created Pods; and kube-controller-manager runs controllers that continuously work to bring the actual cluster state into alignment with the desired state. Worker nodes provide the execution environment for workloads. The kubelet agent on each node communicates with the API server and ensures that containers defined in assigned Pods are running. kube-proxy runs on nodes to implement Kubernetes Service networking, maintaining network rules that allow traffic to reach the appropriate Pods. In a typical deployment, worker nodes also run a container runtime, such as containerd, to execute containers. These component responsibilities are foundational to operating Kubernetes-based AI infrastructure, including environments used with NVIDIA software and accelerated workloads. Kubernetes documents the control-plane and node-component roles at Kubernetes Components, and further describes node operation at Nodes.

QUESTION NO: 8

If a Magnum IO-enabled application experiences delays during the ETL phase, what troubleshooting step should be taken?

A. Disable NVLink to prevent conflicts between GPUs during data transfer.

B. Reduce the size of datasets being processed by splitting them into smaller chunks.

C. Increase the swap space on the host system to handle larger datasets.

D. Ensure that GPUDirect Storage is configured to allow direct data transfer from storage to GPU memory.

ANSWER: D

Explanation:

Ensure that GPUDirect Storage is configured to allow direct data transfer from storage to GPU memory. ETL pipelines commonly spend substantial time reading, transforming, and staging data before it can be used by GPU-accelerated processing. GPUDirect Storage is part of NVIDIA's Magnum IO technology stack and provides a direct path between compatible storage and GPU memory. This minimizes intermediary copies through system memory and reduces CPU involvement, memory-bandwidth pressure, and data-movement latency. Consequently, verifying that the platform, storage path, drivers, and application are configured to use GPUDirect Storage is an appropriate first troubleshooting action when input or staging throughput is delaying the ETL phase. NVIDIA documents that GPUDirect Storage enables a direct data path between local or network storage and GPU memory, supporting higher throughput and lower CPU overhead for data-intensive workloads. Review the NVIDIA GPUDirect Storage overview and the GPUDirect Storage documentation to validate prerequisites, installation, configuration, and operational status.

QUESTION NO: 9

An HGX server with NVSwitches has completed a driver upgrade. After the reboot, multi-GPU workloads cannot use the expected NVLink/NVSwitch fabric, and Fabric Manager reports that the fabric is not initialized.

What two (2) actions should the administrator perform? (Choose two.)

A. Ensure that the Fabric Manager service is installed, enabled, and running after the reboot.

B. Verify that the Fabric Manager package version is compatible with the installed NVIDIA GPU driver.

C. Install the Kubernetes GPU Operator on the HGX server.

D. Replace NCCL with a TCP-based collective communication library.

E. Configure NVIDIA License System licensing for every physical GPU.

ANSWER: A B

Explanation:

Fabric Manager must be installed, enabled, and running to initialize and manage the NVSwitch/NVLink fabric on supported HGX systems. The Fabric Manager package must also be compatible with the installed NVIDIA GPU driver. A driver upgrade that leaves Fabric Manager stopped or at an incompatible version can prevent proper fabric bring-up.

The Kubernetes GPU Operator and vGPU licensing configuration do not initialize an NVSwitch fabric. Replacing NCCL is also not the first corrective action because NCCL depends on the underlying GPU fabric being initialized correctly.