PassQueen
a.k.a dumpsqueen.com

NVIDIA AI Infrastructure and Operations

NVIDIA NCA-AIIO

Version Demo

Total Demo Questions: 29

Total Premium Questions: 290

Buy Premium PDF

https://passqueen.com

support@passqueen.com

Topic Break Down

Topic	No. of Questions
Topic 1, AI Infrastructure Fundamentals	81
Topic 2, NVIDIA AI Infrastructure	110
Topic 3, AI Infrastructure Operations	74
Topic 4, AI Infrastructure Troubleshooting	25
Total	290

QUESTION NO: 1

When designing a data center specifically for AI workloads, which of the following factors is most critical to optimize for training large-scale neural networks?

A. Maximizing the number of storage arrays to handle data volumes

B. Deploying the maximum number of CPU cores available in each node

C. High-speed, low-latency networking between compute nodes

D. Ensuring the data center has a robust virtualization platform

ANSWER: C

Explanation:

High-speed, low-latency networking between compute nodes is the most critical factor for efficient large-scale neural-network training because distributed training requires GPUs across many servers to repeatedly exchange gradients, parameters, and other collective-communication data. During operations such as all-reduce, slow or oversubscribed networking leaves expensive GPUs waiting for data rather than performing computation, which directly increases time to train. A properly designed AI fabric therefore prioritizes high bandwidth, low latency, predictable congestion behavior, and a topology that provides sufficient bisection bandwidth as the cluster grows. NVIDIA AI infrastructure designs use high-performance InfiniBand or Ethernet fabrics with RDMA capabilities to support efficient GPU-to-GPU communication across nodes. This allows collective libraries such as NVIDIA NCCL to use the network efficiently and maintain strong scaling for multi-node workloads. NVIDIA's DGX SuperPOD reference architecture identifies the scale-out network as a core infrastructure component and details how the fabric is designed to carry distributed-training traffic efficiently. NVIDIA also documents that RDMA enables direct data movement with lower CPU overhead and latency, both important characteristics for GPU clusters. See the NVIDIA DGX SuperPOD Reference Architecture and NVIDIA RDMA Aware Networks Programming User Manual.

QUESTION NO: 2

Which of the following statements is true about Kubernetes orchestration?

A. It is bare-metal based but it supports containers.

B. It has advanced scheduling capabilities to assign jobs to available resources.

C. It has no inferencing capabilities.

D. It does load balancing to distribute traffic across containers.

ANSWER: B D

Explanation:

Kubernetes provides advanced scheduling capabilities to place workloads on suitable cluster nodes. Its scheduler evaluates pending Pods and selects nodes using factors such as requested CPU and memory resources, node constraints, affinity or anti-affinity rules, taints and tolerations, and available capacity. This allows an operations team to use cluster resources efficiently while ensuring that workloads are assigned according to their declared requirements. In an AI infrastructure, these scheduling mechanisms can also be used with node labels, resource requests, and device-plugin-exposed GPU resources to direct accelerated workloads to appropriate nodes.

Kubernetes also supports load distribution for application traffic through the Service abstraction. A Service supplies a stable network endpoint for a set of matching Pods and can distribute requests among the available, ready backend endpoints. This decouples clients from individual Pod instances and enables traffic to continue reaching an application as Pods are created, removed, or rescheduled. Together, scheduling and Service-based traffic distribution are core orchestration functions that help Kubernetes operate containerized workloads reliably at scale. See the Kubernetes documentation on scheduling and Services and networking.

QUESTION NO: 3

Your organization is planning to deploy an AI solution that involves large-scale data processing,

training, and real-time inference in a cloud environment. The solution must ensure seamless integration of data pipelines, model training, and deployment. Which combination of NVIDIA software components will best support the entire lifecycle of this AI solution?

A. NVIDIA RAPIDS + NVIDIA Triton Inference Server + NVIDIA NGC Catalog

B. NVIDIA TensorRT + NVIDIA DeepStream SDK

C. NVIDIA Triton Inference Server + NVIDIA NGC Catalog

D. NVIDIA RAPIDS + NVIDIA TensorRT

ANSWER: A

Explanation:

NVIDIA RAPIDS + NVIDIA Triton Inference Server + NVIDIA NGC Catalog best supports the stated end-to-end workflow because these components address data processing, training workflows, and production inference together. RAPIDS provides GPU-accelerated data science and data-processing libraries, including tools compatible with common Python workflows, helping organizations prepare and transform large datasets efficiently for AI development. The NGC Catalog supplies curated, GPU-optimized containers, frameworks, pretrained models, and deployment resources. This gives teams a consistent way to obtain and run validated software for model training and related cloud-native workflows. NVIDIA Triton Inference Server provides the production serving layer for deploying trained models. It supports multiple model frameworks and offers capabilities such as concurrent model execution, dynamic batching, and metrics integration, which are valuable for scalable, real-time inference services. Used together, RAPIDS accelerates pipeline preparation, NGC standardizes access to the software assets needed to develop and train models, and Triton operationalizes those models in a scalable deployment environment. This combination therefore aligns with the full lifecycle described rather than addressing only a specialized inference workload or only part of the development-to-deployment path. See the RAPIDS documentation, the Triton Inference Server User Guide, and the NVIDIA NGC Catalog.

QUESTION NO: 4

Your team is building an AI-powered application that requires the deployment of multiple models, each trained using different frameworks (e.g., TensorFlow, PyTorch, and ONNX). You need a deployment solution that can efficiently serve all these models in production, regardless of the framework they were built in. Which software component should you choose?

A. NVIDIA Clara Deploy SDK

B. NVIDIA TensorRT

C. NVIDIA DeepOps

D. NVIDIA Triton Inference Server

ANSWER: D

Explanation:

NVIDIA Triton Inference Server is the appropriate production deployment component because it provides a unified serving platform for models produced by diverse machine-learning frameworks. Its supported backends include TensorFlow, PyTorch, ONNX Runtime, TensorRT, and Python, allowing a team to place heterogeneous models in a common model repository and expose them through consistent HTTP/REST and gRPC inference APIs. This removes the need to build and operate a separate serving stack for each framework.

Triton is designed for efficient, scalable inference operations rather than merely running a single optimized model. It supports concurrent execution of multiple models, configurable model instances, dynamic batching, model versioning, health and readiness endpoints, Prometheus-compatible metrics, and ensemble models. These capabilities allow infrastructure and application teams to manage throughput, latency, observability, and lifecycle needs in production while retaining framework

flexibility. NVIDIA documents Triton as an open-source inference-serving software that supports multiple deep-learning and machine-learning frameworks and is intended to maximize GPU and CPU utilization for inference. See the NVIDIA Triton Inference Server User Guide and the NVIDIA Triton Inference Server product page.

QUESTION NO: 5

How many distinct network fabrics are in an AI cluster?

A. 3

B. 2

C. 4

D. 5

ANSWER: A

Explanation:

3 is correct because a production AI cluster is commonly designed around three logically distinct network fabrics, each serving a different traffic pattern and operational purpose. The management and client fabric carries administrative access, provisioning, monitoring, and user-facing or external connectivity. The storage fabric provides high-throughput access to the large datasets, checkpoints, and shared storage systems required during training and inference workflows. The compute fabric connects accelerated compute nodes with extremely low latency and high bandwidth, typically using InfiniBand or Ethernet with RDMA capabilities, to support distributed training communications such as collective operations and gradient synchronization. Separating these functions prevents storage and management traffic from contending with latency-sensitive GPU-to-GPU communication, while also allowing each fabric to be sized, secured, and operated according to its workload. NVIDIA reference architectures describe this separation of management, storage, and high-performance compute networking as a key design principle for scalable AI infrastructure. See NVIDIA's AI networking overview and the NVIDIA DGX H100 System User Guide.

QUESTION NO: 6

Which two software components are directly involved in the life cycle of AI development and deployment, particularly in model training and model serving? (Select two)

A. Prometheus

B. MLflow

C. Airflow

D. Apache Spark

E. Kubeflow

ANSWER: B E

Explanation:

MLflow and **Kubeflow** directly support the machine-learning lifecycle from development and training through deployment and serving. MLflow provides lifecycle capabilities for tracking experiments and their parameters, metrics, and artifacts; packaging trained models in standardized formats; registering and versioning models; and deploying models to supported serving targets. This creates traceability between training runs and production model releases, which is central to operational ML workflows.

Kubeflow is a Kubernetes-native platform for executing and managing end-to-end ML workflows. Its components support reproducible pipeline execution and scalable distributed training workloads on Kubernetes. Kubeflow's ecosystem also supports deployment of trained models for online inference, commonly through KServe, while NVIDIA GPU-enabled

environments can use NVIDIA Triton Inference Server as a high-performance serving runtime. Together, these capabilities connect the training process to controlled, scalable model deployment.

These tools are therefore directly relevant to both sides named in the question: producing and managing trained model artifacts, then operationalizing those artifacts for inference. NVIDIA AI infrastructure commonly integrates Kubernetes-based orchestration and GPU-accelerated training and inference services into this broader MLOps lifecycle. See the MLflow documentation and the Kubeflow documentation for their lifecycle, training, pipeline, and deployment capabilities.

QUESTION NO: 7

An enterprise platform team wants to standardize how its developers obtain NVIDIA-optimized software for GPU-accelerated AI workloads. Which two benefits does the NVIDIA NGC Catalog provide? (Select two)

A. Access to GPU-optimized containers

B. Access to pretrained models and related resources

C. Automatic replacement of Kubernetes scheduling policies

D. Elimination of NVIDIA driver compatibility requirements

E. Direct hardware partitioning of a GPU into MIG instances

ANSWER: A B

Explanation:

Access to GPU-optimized containers is a principal NGC Catalog benefit. NVIDIA publishes containers for common AI, data science, HPC, and visualization workloads that package tested application frameworks and dependencies. These containers help teams begin with a validated software environment rather than assembling every component independently.

Access to pretrained models and related resources is also provided through NGC. Organizations can use available model and framework resources to accelerate development, while still validating suitability, licensing, security requirements, and performance for their own deployment environment. NGC supports more consistent consumption of NVIDIA software across development and operations teams. See the NVIDIA NGC Catalog and the NGC Catalog User Guide.

QUESTION NO: 8

You have developed two different machine learning models to predict house prices based on various features like location, size, and number of bedrooms. Model A uses a linear regression approach, while Model B uses a random forest algorithm. You need to compare the performance of these models to determine which one is better for deployment. Which two statistical performance metrics would be most appropriate to compare the accuracy and reliability of these models? (Select two)

A. F1 Score

B. Learning Rate

C. Mean Absolute Error (MAE)

D. Cross-Entropy Loss

E. R-squared (Coefficient of Determination)

ANSWER: C E

Explanation:

House-price prediction is a regression task because the target is a continuous numeric value. Mean Absolute Error (MAE) is an appropriate metric because it calculates the mean magnitude of prediction errors without considering their direction. Its result is expressed in the same unit as the house-price target, making the typical prediction error straightforward to interpret

and compare between the linear-regression and random-forest models. R-squared (Coefficient of Determination) is also appropriate because it measures the proportion of variation in observed house prices explained by each model relative to a baseline that predicts the mean target value. Used on the same held-out test data or under the same cross-validation procedure, these metrics provide complementary evidence: MAE quantifies typical practical error, while R-squared summarizes goodness of fit and explanatory performance. This combination supports a meaningful comparison of predictive accuracy and reliability before deployment. For a robust assessment, calculate both metrics on data not used to fit or tune either model, and consider the business impact of the observed MAE in the target currency. The scikit-learn regression-metrics documentation describes both MAE and R-squared and their expected interpretation: scikit-learn regression metrics. NVIDIA RAPIDS cuML also provides GPU-accelerated machine-learning workflows compatible with evaluating regression models: RAPIDS cuML documentation.

QUESTION NO: 9

Which components are essential parts of the NVIDIA software stack in an AI environment? (Select two)

A. NVIDIA CUDA Toolkit

B. NVIDIA TensorRT

C. NVIDIA JetPack SDK

D. NVIDIA Nsight Systems

E. NVIDIA GameWorks

ANSWER: A B

Explanation:

NVIDIA CUDA Toolkit and NVIDIA TensorRT are essential components of NVIDIA's software stack for AI workloads. The CUDA Toolkit supplies the CUDA programming platform and core GPU-accelerated libraries that enable applications and AI frameworks to use NVIDIA GPUs. It includes the compiler toolchain, runtime, debugging and profiling capabilities, and libraries that provide a foundation for developing and executing accelerated computing workloads. AI frameworks commonly rely on CUDA and its associated libraries to perform GPU computation efficiently.

NVIDIA TensorRT is NVIDIA's SDK for optimizing and deploying trained neural networks for inference. It applies graph and kernel optimizations, precision calibration or conversion where appropriate, and hardware-specific execution planning to deliver high-throughput, low-latency inference on NVIDIA GPUs. This makes it a central deployment-layer component for production AI inference environments, including data center systems. Together, CUDA provides the fundamental accelerated-computing platform, while TensorRT supplies specialized inference optimization and runtime functionality. NVIDIA documents the CUDA platform at NVIDIA CUDA Toolkit and describes inference optimization and deployment capabilities at NVIDIA TensorRT.

QUESTION NO: 10

What is the primary advantage of using virtualized environments for AI workloads in a large enterprise setting?

A. Reduces the need for specialized hardware by running AI workloads on general-purpose CPUs

B. Allows for easier scaling of AI workloads across multiple physical machines

C. Ensures that AI workloads are always running on the same physical machine for consistency

D. Enables AI workloads to utilize cloud resources without requiring any changes to the underlying code

ANSWER: B

Explanation:

Allows for easier scaling of AI workloads across multiple physical machines is correct because virtualization abstracts compute, GPU, network, and storage resources from the applications consuming them. In a large enterprise, this enables

infrastructure teams to provision standardized virtual machines with appropriate GPU capacity for different teams, projects, and workload types without requiring every workload to be permanently tied to a particular physical server. Capacity can be allocated, adjusted, and managed centrally as demand changes, improving utilization of expensive accelerated infrastructure and making it easier to support multi-tenant environments.

NVIDIA vGPU technology is designed to virtualize NVIDIA GPUs so that multiple virtual machines can use a physical GPU securely and with predictable resource allocations. This supports flexible delivery of GPU-accelerated resources to enterprise users while retaining centralized administration. Virtualized infrastructure can also use orchestration and scheduling capabilities to place workloads on suitable available hosts, which helps organizations expand AI services across their data-center fleet and respond to changing demand. NVIDIA documents the supported virtualization architectures, GPU resource profiles, and management considerations in its NVIDIA vGPU documentation. For cloud-native environments, the NVIDIA GPU Operator documentation describes automated management of NVIDIA GPU software components across Kubernetes clusters.

QUESTION NO: 11

You are tasked with optimizing the performance of a deep learning model used for image recognition. The model needs to process a large dataset as quickly as possible while maintaining high accuracy. You have access to both GPU and CPU resources. Which two statements best describe why GPUs are more suitable than CPUs for this task? (Select two)

A. CPUs are better suited for handling the large dataset due to their superior memory bandwidth.

B. GPUs have a higher number of cores compared to CPUs, allowing for parallel processing of many operations simultaneously.

C. GPUs are optimized for matrix operations, which are common in deep learning algorithms.

D. GPUs have a lower latency than CPUs, making them faster for individual calculations.

E. CPUs consume less power than GPUs, making them more suitable for prolonged computations.

ANSWER: B C

Explanation:

“GPUs have a higher number of cores compared to CPUs, allowing for parallel processing of many operations simultaneously” is correct because image-recognition neural networks perform enormous numbers of independent or data-parallel calculations across pixels, channels, filters, and training examples. NVIDIA GPUs are designed to schedule and execute very large numbers of lightweight threads concurrently using the CUDA parallel-computing model. This high throughput is particularly valuable when processing large image datasets and batches.

“GPUs are optimized for matrix operations, which are common in deep learning algorithms” is also correct. Convolutions, fully connected layers, attention-related operations, and other neural-network computations are implemented primarily as matrix and tensor operations. NVIDIA GPUs provide specialized hardware and highly optimized software, including Tensor Cores where supported and CUDA-X libraries such as cuBLAS and cuDNN, to accelerate these operations while preserving the numerical precision required by the chosen training or inference workflow. This combination of massively parallel execution, high memory bandwidth, and tensor-math acceleration generally delivers substantially greater deep-learning throughput than CPUs for this workload. NVIDIA describes the CUDA execution model in the CUDA C++ Programming Guide and details GPU acceleration of deep-learning matrix multiplication in its Matrix Multiplication Background User's Guide.

QUESTION NO: 12

In an AI environment, the NVIDIA software stack plays a crucial role in ensuring seamless operations across different stages of the AI workflow. Which components of the NVIDIA software stack would you use to accelerate AI model training and deployment? (Select two)

A. NVIDIA cuDNN (CUDA Deep Neural Network library)

B. NVIDIA TensorRT

C. NVIDIA DGX-1

D. NVIDIA Nsight

E. NVIDIA DeepStream SDK

ANSWER: A B

Explanation:

NVIDIA cuDNN (CUDA Deep Neural Network library) and NVIDIA TensorRT are the NVIDIA software-stack components that directly accelerate the two stated AI workflow stages. NVIDIA cuDNN provides GPU-optimized primitives for deep neural networks, including operations such as convolutions, normalization, attention, and recurrent-network functions. Deep learning frameworks use these highly tuned GPU routines to execute training workloads efficiently, improving training performance on NVIDIA GPUs.

NVIDIA TensorRT is designed to optimize and run trained models efficiently in production inference environments. It applies graph and kernel optimizations, including layer fusion, precision selection and calibration for formats such as FP16 and INT8, and GPU-specific tactic selection. Its runtime is intended to improve inference throughput while reducing latency, making it a core deployment acceleration component. Together, these tools address the general training and deployment lifecycle: cuDNN accelerates the compute primitives used during neural-network training, while TensorRT optimizes trained networks for high-performance inference deployment.

NVIDIA documents cuDNN as a library of GPU-accelerated primitives for deep neural networks and TensorRT as an SDK for high-performance deep-learning inference. See the NVIDIA cuDNN product page and the NVIDIA TensorRT product page.

QUESTION NO: 13

Your AI-driven data center experiences occasional GPU failures, leading to significant downtime for critical AI applications. To prevent future issues, you decide to implement a comprehensive GPU health monitoring system. You need to determine which metrics are essential for predicting and preventing GPU failures. Which of the following metrics should be prioritized to predict potential GPU failures and maintain GPU health?

A. GPU Clock Speed

B. GPU Temperature

C. CPU Utilization

D. Error Rates (e.g., ECC errors)

ANSWER: D

Explanation:

Error Rates (e.g., ECC errors) should be prioritized because GPU error telemetry is a direct indicator of memory and hardware reliability. NVIDIA Data Center GPU Manager (DCGM) exposes detailed health and diagnostic information, including ECC error counters. Correctable ECC errors indicate that GPU memory errors have occurred but were repaired by ECC; monitoring their rate and trend helps operations teams identify a GPU that may be degrading before it affects production workloads. Uncorrectable ECC errors are especially important because they indicate an error that could not be corrected and can lead to application failure, workload interruption, or a need for GPU remediation or replacement. A robust monitoring system should alert on these errors and retain historical trends so that increasing error activity can be correlated with specific devices and maintenance actions. NVIDIA also provides ECC and related reliability information through NVIDIA System Management Interface telemetry, enabling administrators to inspect device-level error status in operational environments. These signals are therefore highly actionable for proactively protecting critical AI services and maintaining fleet reliability. See the NVIDIA DCGM feature overview and the NVIDIA System Management Interface documentation.

QUESTION NO: 14

You are tasked with optimizing an AI-driven financial modeling application that performs both complex mathematical calculations and real-time data analytics. The calculations are CPU-intensive, requiring precise sequential processing, while the data analytics involves processing large datasets in parallel. How should you allocate the workloads across GPU and CPU architectures?

A. Use CPUs for data analytics and GPUs for mathematical calculations

B. Use GPUs for mathematical calculations and CPUs for managing I/O operations

C. Use CPUs for mathematical calculations and GPUs for data analytics

D. Use GPUs for both the mathematical calculations and data analytics

ANSWER: C

Explanation:

Use CPUs for mathematical calculations and GPUs for data analytics. The workload description identifies two fundamentally different compute patterns. The mathematical portion has precise sequential dependencies and is explicitly CPU-intensive. CPUs are designed for strong low-latency execution of serial and control-heavy tasks, with sophisticated branch handling and large caches that support such processing effectively. Keeping this work on the CPU also avoids unnecessary host-device transfer and coordination overhead for code that cannot be efficiently parallelized.

The analytics component processes large datasets in parallel, which maps well to the GPU architecture. NVIDIA GPUs provide many parallel processing cores and high memory bandwidth, enabling high-throughput execution for data-parallel operations such as filtering, transformations, aggregations, and matrix-oriented analytics. NVIDIA RAPIDS is an example of this model: it uses GPU acceleration for dataframe and analytics workflows while fitting into CPU-hosted application environments. This heterogeneous allocation uses each processor for the class of work it is best suited to perform, improving overall throughput without compromising the sequential behavior required by the financial-modeling calculations. See NVIDIA RAPIDS and NVIDIA's guidance on GPUs for data-science workloads.

QUESTION NO: 15

What NVIDIA tool should a data center administrator use to monitor NVIDIA GPUs?

A. NVIDIA System Monitor

B. NetQ

C. DCGM

ANSWER: C

Explanation:

DCGM is the correct tool because NVIDIA Data Center GPU Manager is specifically designed to monitor, manage, and diagnose NVIDIA GPUs deployed in data-center environments. It gathers operational telemetry such as GPU utilization, memory usage, temperature, power consumption, clock rates, error counts, and health status. DCGM also supports automated diagnostics and policy-based monitoring, helping administrators identify hardware or operational issues before they affect AI, HPC, or accelerated-computing workloads. Its architecture supports both individual hosts and fleet-scale deployments, and it can integrate with common infrastructure-monitoring stacks through DCGM Exporter, which exposes GPU metrics for Prometheus collection and Grafana visualization. This makes DCGM the NVIDIA-supported foundation for observability and health monitoring of production data-center GPUs. NVIDIA documents DCGM's monitoring, diagnostics, and management capabilities in the NVIDIA DCGM User Guide. For integration into metrics-based monitoring environments, see the NVIDIA DCGM Exporter project.

QUESTION NO: 16

Your organization has deployed a large-scale AI data center with multiple GPUs running complex deep learning workloads. You've noticed fluctuating performance and increasing energy consumption across several nodes. You need to optimize the

data center's operation and improve energy efficiency while ensuring high performance. Which of the following actions should you prioritize to achieve optimized AI data center management and maintain efficient energyconsumption?

A. Disable power management features on all GPUs to ensure maximum performance

B. Implement GPU workload scheduling based on real-time performance metrics

C. Install additional GPUs to distribute the workload more evenly

D. Increase the number of active cooling systems to reduce thermal throttling

ANSWER: B

Explanation:

Implement GPU workload scheduling based on real-time performance metrics is the priority because it addresses both inconsistent application performance and unnecessary energy use at their operational source: inefficient allocation of GPU resources. NVIDIA Data Center GPU Manager (DCGM) provides telemetry on GPU utilization, power usage, temperatures, clocks, memory, and health. Operations teams and schedulers can use these metrics to place workloads on suitable GPUs, balance utilization across nodes, identify underused capacity, and avoid scheduling work onto devices that are thermally constrained or experiencing reliability issues. This improves sustained throughput by reducing contention and performance variability, while improving performance per watt by limiting idle powered-on capacity and avoiding inefficient operating conditions. DCGM is designed to support monitoring and management of NVIDIA GPUs in clustered and data-center environments, including integration with orchestration and monitoring systems. Using real-time measurements also enables policy-driven decisions, such as consolidating lower-priority work, selecting nodes with sufficient thermal and power headroom, and responding promptly to utilization or health changes. These capabilities make metric-informed scheduling a foundational AI infrastructure operations practice for maintaining high performance efficiently at scale. See the NVIDIA DCGM feature overview and DCGM user guide overview.

QUESTION NO: 17

Which two components are included in GPU Operator? (Choose two.)

A. Drivers

B. PyTorch

C. DCGM

D. TensorFlow

ANSWER: A C

Explanation:

Drivers and **DCGM** are included in the NVIDIA GPU Operator’s Kubernetes-managed software stack. The GPU Operator automates deployment and lifecycle management of the NVIDIA components needed to provision GPUs for container workloads. Its NVIDIA driver component installs the kernel-mode driver and user-space software required for Kubernetes nodes to recognize and use NVIDIA GPUs. This allows GPU-enabled workloads to access the hardware without requiring manual driver installation on every node.

The operator also deploys NVIDIA Data Center GPU Manager (DCGM) components. DCGM provides GPU health and telemetry capabilities, including metrics used to observe utilization, temperature, power, memory, errors, and other operational characteristics. In a typical GPU Operator deployment, DCGM Exporter exposes these metrics in a Prometheus-compatible format for monitoring systems. Together, the driver and DCGM support the foundational goals of reliable GPU provisioning and observability in Kubernetes clusters.

NVIDIA documents the managed stack and its operands in the GPU Operator overview. Details of DCGM’s monitoring and management capabilities are available in the DCGM feature overview.

QUESTION NO: 18

A Kubernetes platform team needs to run several independent inference workloads on supported NVIDIA GPUs. Each workload requires predictable performance and fault isolation, but does not require a complete physical GPU. Which two characteristics of NVIDIA Multi-Instance GPU (MIG) make it suitable for this requirement? (Select two)

A. Hardware-level isolation of compute and memory resources

B. Exposure of GPU instances as schedulable resources

C. Automatic sharing of a single GPU memory pool by all Pods

D. Support for combining GPU partitions across separate physical GPUs into one instance

E. Elimination of the need for NVIDIA GPU drivers

ANSWER: A B

Explanation:

Hardware-level isolation of compute and memory resources is a core MIG capability. MIG partitions a supported physical GPU into isolated GPU instances, each with dedicated compute capacity and memory resources. This helps provide predictable behavior for separate workloads and reduces the risk that one workload will consume resources assigned to another instance.

Exposure of GPU instances as schedulable resources also makes MIG appropriate for Kubernetes environments. With the NVIDIA device plugin and GPU Operator configured for MIG, Kubernetes can discover MIG instances and schedule Pods that request the corresponding GPU resources. This allows smaller workloads to be placed on isolated GPU partitions instead of requiring an entire accelerator. See the NVIDIA MIG User Guide and the NVIDIA GPU Operator MIG documentation.

QUESTION NO: 19

A large enterprise is deploying a high-performance AI infrastructure to accelerate its machine

learning workflows. They are using multiple NVIDIA GPUs in a distributed environment. To optimize the workload distribution and maximize GPU utilization, which of the following tools or frameworks should be integrated into their system? (Select two)

A. NVIDIA CUDA

B. NVIDIA NGC (NVIDIA GPU Cloud)

C. TensorFlow Serving

D. NVIDIA NCCL (NVIDIA Collective Communications Library)

E. Keras

ANSWER: A D

Explanation:

NVIDIA CUDA and **NVIDIA NCCL (NVIDIA Collective Communications Library)** are the appropriate technologies for an NVIDIA multi-GPU distributed AI environment. CUDA is NVIDIA's parallel-computing platform and programming model. It supplies the GPU runtime, driver interface, compiler toolchain, and accelerated library ecosystem that enable machine-learning frameworks to execute compute-intensive operations efficiently on NVIDIA GPUs. A CUDA-enabled software stack is therefore foundational to achieving high utilization of each GPU during model training and related AI workloads.

NCCL complements CUDA by providing highly optimized, topology-aware collective communication operations for GPUs. Distributed data-parallel training repeatedly exchanges or aggregates gradients across devices; NCCL supplies operations such as all-reduce, all-gather, reduce-scatter, and broadcast for this purpose. It is designed to use high-speed interconnects, including NVLink and InfiniBand/RoCE-capable networks where available, and is integrated by distributed frameworks such

as PyTorch Distributed Data Parallel. Efficient collective communication reduces synchronization overhead, allowing the distributed job to scale across GPUs and nodes while maintaining productive GPU compute time.

Together, CUDA provides the execution foundation and NCCL provides the communication foundation needed by distributed training frameworks. See the NVIDIA CUDA documentation and the NVIDIA NCCL User Guide.

QUESTION NO: 20

Your company is developing an AI application that requires seamless integration of data processing, model training, and deployment in a cloud-based environment. The application must support realtime inference and monitoring of model performance. Which combination of NVIDIA software components is best suited for this end-to-end AI development and deployment process?

A. NVIDIA DeepOps + NVIDIA RAPIDS

B. NVIDIA Clara Deploy SDK + NVIDIA Triton Inference Server

C. NVIDIA RAPIDS + NVIDIA TensorRT

D. NVIDIA RAPIDS + NVIDIA Triton Inference Server + NVIDIA DeepOps

ANSWER: D

Explanation:

NVIDIA RAPIDS + NVIDIA Triton Inference Server + NVIDIA DeepOps is the best fit because it covers the principal stages of an operational cloud AI workflow. RAPIDS provides GPU-accelerated data science and machine-learning libraries, including cuDF for dataframe processing and cuML for machine-learning workflows. This enables data preparation and training-related pipeline stages to remain accelerated on NVIDIA GPUs rather than requiring data to move to a separate CPU-oriented processing path.

NVIDIA Triton Inference Server provides the production inference layer needed for real-time deployment. It supports models from multiple frameworks and provides serving capabilities such as concurrent model execution, dynamic batching, model repositories and versioning, and metrics that can be collected by Prometheus-compatible monitoring systems. Those capabilities directly support responsive inference and ongoing visibility into serving performance.

NVIDIA DeepOps supplies infrastructure automation and deployment practices for GPU-accelerated environments, including Kubernetes-oriented operations. Together, these components combine accelerated data and ML processing, scalable model serving, monitoring telemetry, and repeatable cloud infrastructure operations. NVIDIA documents RAPIDS at RAPIDS Documentation and describes Triton's deployment, scheduling, and metrics capabilities in the NVIDIA Triton Inference Server User Guide.

QUESTION NO: 21

You are responsible for managing an AI infrastructure that runs a critical deep learning application. The application experiences intermittent performance drops, especially when processing large datasets. Upon investigation, you find that some of the GPUs are not being fully utilized while others are overloaded, causing the overall system to underperform. What would be the most effective solution to address the uneven GPU utilization and optimize the performance of the deep learning application?

A. Reduce the size of the datasets being processed.

B. Increase the clock speed of the GPUs.

C. Add more GPUs to the system.

D. Implement dynamic load balancing for the GPUs.

ANSWER: D

Explanation:

Implement dynamic load balancing for the GPUs. The observed condition is a workload-placement and distribution problem: some devices receive more work than they can process promptly, while available capacity on other GPUs remains unused. Dynamic load balancing improves throughput and reduces latency by assigning or redistributing requests, batches, or tasks according to current device availability, queue depth, processing capacity, and observed demand. This prevents individual GPUs from becoming bottlenecks and lets the application use the aggregate compute capacity of the infrastructure more consistently, including during periods when large datasets create bursty or uneven demand.

In an NVIDIA AI deployment, this can be implemented at multiple layers. An inference service can use multiple model instances and dynamic batching to keep available GPU resources busy while serving variable request loads. At the platform layer, an orchestrator can schedule GPU-backed workloads based on declared resource requirements and available devices. NVIDIA Triton Inference Server documents model-instance configuration and dynamic batching as mechanisms for increasing utilization and throughput, while the NVIDIA GPU Operator provides the operational foundation for managed GPU workloads in Kubernetes. See the NVIDIA Triton model configuration documentation and the NVIDIA GPU Operator documentation.

QUESTION NO: 22

A team is deploying a trained computer vision model for high-throughput inference on NVIDIA GPUs. The team wants to reduce inference latency without unnecessarily reducing model accuracy. Which two TensorRT optimization approaches are appropriate when validated against representative data? (Select two)

A. Convert supported operations to FP16 precision

B. Use INT8 calibration with representative input data

C. Disable TensorRT layer fusion

D. Force all model operations to execute on the CPU

E. Increase model precision to FP64 for serving

ANSWER: A B

Explanation:

Converting supported operations to FP16 precision is an appropriate optimization because TensorRT can use lower-precision execution on supported NVIDIA GPUs, often enabling Tensor Core acceleration and reducing memory bandwidth requirements. FP16 should be validated to ensure that the model continues to meet its accuracy requirement.

Using INT8 calibration with representative input data is also appropriate when the accuracy target permits it. TensorRT calibration determines quantization scales from representative data so that the engine can execute suitable operations using INT8 efficiently. This can improve throughput and reduce latency, but the resulting model must be evaluated against production-relevant accuracy criteria. See the NVIDIA TensorRT documentation.

QUESTION NO: 23

You are assisting a senior researcher in analyzing the results of several AI model experiments conducted with different training datasets and hyperparameter configurations. The goal is to understand how these variables influence model overfitting and generalization. Which method would best help in identifying trends and relationships between dataset characteristics, hyperparameters, and the risk of overfitting?

A. Perform a time series analysis of accuracy across different epochs

B. Create a scatter plot comparing training accuracy and validation accuracy

C. Use a histogram to display the frequency of overfitting occurrences across datasets

D. Conduct a decision tree analysis to explore how dataset characteristics and hyperparameters affect overfitting

ANSWER: D

Explanation:

Conduct a decision tree analysis to explore how dataset characteristics and hyperparameters affect overfitting is the best method because it evaluates many experimental variables jointly against an outcome such as overfitting risk or a measured training-versus-validation performance gap. Each experiment can be represented as an observation, with inputs including dataset size, class balance, noise level, learning rate, batch size, model capacity, regularization settings, and augmentation choices. A decision tree can then identify threshold-based patterns and interactions among these inputs that are associated with differing generalization outcomes. For example, it can expose a combination of limited training data and a particular regularization setting that corresponds to a high likelihood of overfitting. This is especially useful for exploratory analysis because its split rules are interpretable and can help researchers prioritize variables for further controlled experimentation. NVIDIA RAPIDS cuML provides GPU-accelerated decision-tree implementations that can support analysis of large experiment-tracking datasets, while also fitting naturally into data science workflows. See the RAPIDS cuML documentation and the scikit-learn decision-tree guide for details on tree-based learning and interpretation.

QUESTION NO: 24

A platform team is preparing a GPU-accelerated training environment that must be reproducible across development, staging, and production clusters. Which two container practices most effectively support consistent deployment of the NVIDIA software stack? (Select two)

A. Pin container images to explicit version tags or immutable digests

B. Use NVIDIA GPU-optimized framework containers with compatible host drivers

C. Use the latest image tag for every production deployment

D. Install a separate GPU driver inside every application container

E. Allow each environment to select arbitrary CUDA versions

ANSWER: A B

Explanation:

Pinning container images to explicit version tags or immutable digests supports reproducibility because each environment uses the same tested application, framework, CUDA, and library versions. Mutable tags can change over time, causing an identical deployment manifest to retrieve different software.

Using NVIDIA GPU-optimized framework containers with compatible host drivers is also important. NVIDIA containers package tested combinations of frameworks, CUDA libraries, and supporting components, while the host driver supplies the GPU driver interface required by the containerized workload. Confirming compatibility avoids failures caused by unsupported driver and CUDA combinations. See the NVIDIA Container Toolkit documentation and the NVIDIA NGC Catalog.

QUESTION NO: 25

When using an InfiniBand network for an AI infrastructure, which software component is necessary for the fabric to function?

A. Verbs

B. MPI

C. OpenSM

ANSWER: C

Explanation:

OpenSM is the required subnet-management component for a standard InfiniBand fabric. Every InfiniBand subnet requires an active Subnet Manager to discover the fabric topology, assign local identifiers to ports, configure forwarding tables in switches, and establish the paths that let endpoints communicate. OpenSM is the open-source Subnet Manager commonly deployed with NVIDIA InfiniBand environments and can run on a host connected to the fabric or on a supported switch. Its

role is foundational: after the physical fabric is connected, the Subnet Manager brings the subnet into an operational state by performing discovery and programming routing information. In production AI clusters, administrators may use NVIDIA UFM to provide broader fabric monitoring and management; UFM includes Subnet Manager functionality. However, the essential functional requirement remains that a Subnet Manager be present, and OpenSM is the matching component among the supplied choices. NVIDIA's InfiniBand documentation describes the requirement for a Subnet Manager in an InfiniBand subnet, while the OpenSM documentation details its topology discovery, LID assignment, and routing responsibilities. See NVIDIA UFM Subnet Manager documentation and NVIDIA OpenSM documentation.

QUESTION NO: 26

Your AI team is running a distributed deep learning training job on an NVIDIA DGX A100 clusterusing multiple nodes. The training process is slowing down significantly as the model size increases. Which of the following strategies would be most effective in optimizing the training performance?

A. Enable Mixed Precision Training

B. Use Data Parallelism Instead of Model Parallelism

C. Decrease the Number of Nodes

D. Increase Batch Size

ANSWER: A

Explanation:

Enable Mixed Precision Training is the most effective broadly applicable optimization for large-model distributed training on NVIDIA DGX A100 systems. A100 Tensor Cores are designed to accelerate reduced-precision matrix operations, including FP16 and BF16, while mixed-precision workflows retain suitable higher-precision operations where needed for numerical stability. This substantially increases available math throughput compared with FP32 execution and reduces the memory required for model activations, gradients, and some optimizer-related tensors. As model size grows, the memory reduction is particularly valuable because it can prevent memory pressure from constraining efficient batch sizes or parallelization choices. In a multi-node job, faster GPU computation improves step throughput, and reduced-precision gradient communication can also lower communication overhead when supported by the distributed-training configuration. NVIDIA's guidance describes mixed precision as using Tensor Cores to improve training performance while maintaining model accuracy through appropriate precision management. The A100 architecture also provides Tensor Core support for multiple precision formats, making this approach directly aligned with DGX A100 capabilities. See the NVIDIA Mixed Precision Training guide and the NVIDIA A100 Tensor Core GPU overview.

QUESTION NO: 27

A data center is running a cluster of NVIDIA GPUs to support various AI workloads. The operations team needs to monitor GPU performance to ensure workloads are running efficiently and to prevent potential hardware failures. Which two key measures should they focus on to monitor the GPUs effectively? (Select two)

A. Disk I/O rates

B. CPU clock speed

C. GPU temperature and power consumption

D. GPU memory utilization

E. Network bandwidth usage

ANSWER: C D

Explanation:

GPU temperature and power consumption are essential operational health measures. Temperature indicates whether a GPU is operating within its thermal envelope; excessive temperature can cause thermal throttling, which lowers application performance, and may require investigation of cooling, airflow, or job placement. Power telemetry provides visibility into whether each GPU is operating near its configured power limit and helps operators identify unexpected power behavior, capacity requirements, and performance states. NVIDIA exposes both measurements through NVIDIA Management Library-backed tooling, making them suitable for continuous collection and alerting across a GPU cluster.

GPU memory utilization is equally important for efficient AI workload operations. Training and inference workloads commonly require substantial device memory for model weights, activations, batches, and cached data. Monitoring memory use lets the team detect memory pressure before out-of-memory failures interrupt jobs, assess whether workloads are appropriately sized for their assigned GPUs, and identify potential underutilization. Together, thermal and power telemetry establish whether the hardware is operating safely, while memory utilization indicates whether AI jobs are using a critical GPU resource effectively. NVIDIA DCGM and `nvidia-smi` support collecting these GPU health and utilization metrics at scale. See NVIDIA System Management Interface and NVIDIA DCGM Feature Overview.

QUESTION NO: 28

Which of the following has been the most critical factor enabling the recent rapid improvements and adoption of AI in various sectors?

A. The rise of user-friendly AI frameworks and libraries.

B. The availability of large, annotated datasets for training AI models.

C. Increased investment in AI research and development by large tech companies.

D. The development and adoption of AI-specific hardware like GPUs and TPUs.

ANSWER: D

Explanation:

The development and adoption of AI-specific hardware like GPUs and TPUs has been a foundational enabler of the recent acceleration in AI capability and practical deployment. Modern deep-learning training and inference rely heavily on highly parallel matrix and tensor operations. GPUs are designed to execute very large numbers of such operations concurrently, while specialized units such as NVIDIA Tensor Cores accelerate the mixed-precision matrix calculations commonly used by neural networks. This makes it feasible to train larger models on more data in practical time frames and to run inference at the throughput and cost required for production services. NVIDIA's CUDA software platform also provides the programming and library ecosystem that lets frameworks and applications use this accelerated hardware effectively. The combination of scalable accelerated computing, high memory bandwidth, optimized AI libraries, and mature software tooling has therefore enabled the scale of today's computer vision, natural-language, recommendation, and generative-AI workloads. NVIDIA describes Tensor Cores as specialized GPU components that accelerate AI and high-performance-computing matrix operations, and its CUDA platform as the foundation for GPU-accelerated applications. See the NVIDIA Tensor Cores overview and the NVIDIA CUDA Zone.

QUESTION NO: 29

An enterprise is designing a multi-tenant GPU cluster for development and production workloads. The platform must provide fair access to GPU capacity and prevent one namespace from consuming all available accelerators. Which two Kubernetes controls are most appropriate? (Select two)

A. Configure ResourceQuotas for each namespace

B. Configure LimitRanges for each namespace

C. Grant every namespace unlimited GPU requests

D. Disable GPU resource limits on all Pods

E. Schedule all workloads in the default namespace

ANSWER: A B

Explanation:

ResourceQuotas are appropriate because they can limit aggregate resource consumption within a namespace, including extended resources such as GPUs. A quota prevents one tenant from creating GPU workloads beyond its assigned capacity.

LimitRanges are also useful because they can establish default, minimum, and maximum resource requirements for workloads in a namespace. This encourages Pods to declare resource needs consistently and helps prevent inappropriate requests or limits from being submitted by individual users. These controls work with Kubernetes scheduling and NVIDIA GPU resources exposed by the device plugin. See the Kubernetes Resource Quotas documentation and the Kubernetes Limit Ranges documentation.