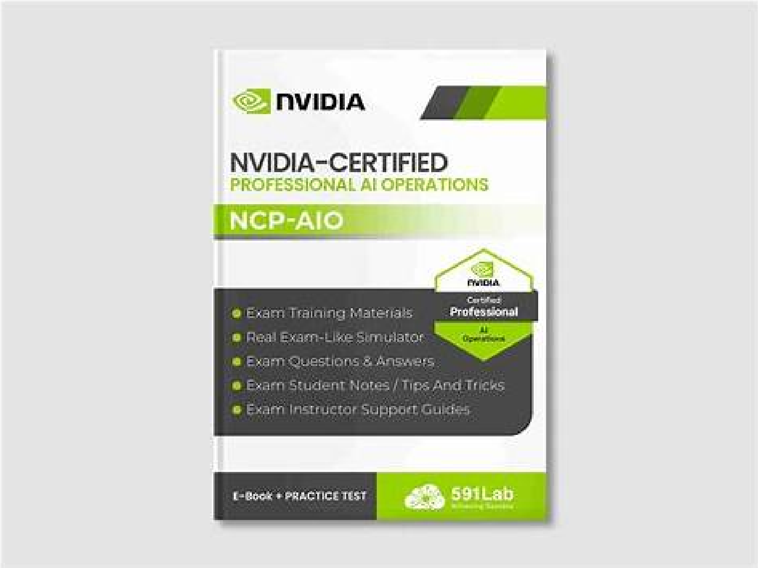
Pass Guaranteed 2026 NCP-AIO: Reliable Passing NVIDIA AI Operations Score Feedback



P.S. Free & New NCP-AIO dumps are available on Google Drive shared by TrainingQuiz: https://drive.google.com/open?id=1WnY5A-8HxYjRvRLzlSOnQWH0DqzltTIC

Our NCP-AIO learning prep boosts the self-learning, self-evaluation, statistics report, timing and test stimulation functions and each function plays their own roles to help the clients learn comprehensively. The self-learning and self-evaluation functions of our NCP-AIO guide materials help the clients check the results of their learning of the NCP-AIO Study Materials. The timing function of our NCP-AIO training quiz helps the learners to adjust their speed to answer the questions and keep alert and our study materials have set the timer.

NVIDIA NCP-AIO Exam Syllabus Topics:

Topic	Details
Topic 1	• Installation and Deployment: This section of the exam measures the skills of system administrators and addresses core practices for installing and deploying infrastructure. Candidates are tested on installing and configuring Base Command Manager, initializing Kubernetes on NVIDIA hosts, and deploying containers from NVIDIA NGC as well as cloud VMI containers. The section also covers understanding storage requirements in AI data centers and deploying DOCA services on DPU Arm processors, ensuring robust setup of AI-driven environments.
Topic 2	• Workload Management: This section of the exam measures the skills of AI infrastructure engineers and focuses on managing workloads effectively in AI environments. It evaluates the ability to administer Kubernetes clusters, maintain workload efficiency, and apply system management tools to troubleshoot operational issues. Emphasis is placed on ensuring that workloads run smoothly across different environments in alignment with NVIDIA technologies.
Topic 3	• Administration: This section of the exam measures the skills of system administrators and covers essential tasks in managing AI workloads within data centers. Candidates are expected to understand fleet command, Slurm cluster management, and overall data center architecture specific to AI environments. It also includes knowledge of Base Command Manager (BCM), cluster provisioning, Run.ai administration, and configuration of Multi-Instance GPU (MIG) for both AI and high-performance computing applications.

Topic 4	• Troubleshooting and Optimization: NVIThis section of the exam measures the skills of AI infrastructure engineers and focuses on diagnosing and resolving technical issues that arise in advanced AI systems. Topics include troubleshooting Docker, the Fabric Manager service for NVIDIA NVlink and NVSwitch systems, Base Command Manager, and Magnum IO components. Candidates must also demonstrate the ability to identify and solve storage performance issues, ensuring optimized performance across AI workloads.

NCP-AIO Practice Exam & Exam NCP-AIO Simulations

We offer free demos as your experimental tryout before downloading our real NCP-AIO exam questions. For more textual content about practicing exam questions, you can download our products with reasonable prices and get your practice begin within 5 minutes. After getting to know our NCP-AIO Test Guide by free demos, many exam candidates had their volitional purchase. So our NCP-AIO latest dumps are highly effective to make use of.

NVIDIA AI Operations Sample Questions (Q23-Q28):

NEW QUESTION # 23
Consider this YAML snippet for deploying the NVIDIA device plugin. Which statement is true about the highlighted segment?

- A. It will make sure that the device plugin gets deployed on nodes with 'accelerator: nvidia-tesla-t4' label
- B. This will ensure that the pod is only deployed on the node with certain taints, but with no other scheduling requirements
- C. It will make sure that only tolerations are set, but there's no affinity.
- D. It's a deprecated way of defining affinities. Consider using nodeAffinity instead.
- E. It will make sure that the node affinity is ignored during scheduling.

Answer: A

Explanation:
The 'nodeselector' is used to target the deployment to nodes with label 'accelerator: nvidia-tesla-t4'. "nodeAffinity' is a more advanced way of doing this and is recommended, but in the absence of explicit nodeAffinity, nodeSelector is sufficient.

```yaml
piVersion: apps/v1
ind: DaemonSet
etadata:
 name: nvidia-device-plugin-daemonset
 namespace: kube-system
pec:
 selector:
   matchLabels:
     name: nvidia-device-plugin-pod
 template:
   metadata:
     labels:
       name: nvidia-device-plugin-pod
   spec:
      nodeSelector:
        accelerator: nvidia-tesla-t4
     containers:
     - image: nvcr.io/nvidia/k8s/device-plugin:v0.14.
       name: nvidia-device-plugin
       securityContext:
         allowPrivilegeEscalation: false
         capabilities: {add: ["CAP_MKNOD"]}
       volumeMounts:
       - name: device-plugin
         mountPath: /var/lib/kubelet/device-plugins
     volumes:
     - name: device-plugin
       hostPath:
         path: /var/lib/kubelet/device-plugins
```

NEW QUESTION # 24
You need to monitor the GPU utilization of pods in your Kubernetes cluster running AI workloads. You want to use Prometheus to collect these metrics. Which of the following approaches is most suitable and efficient for collecting GPU metrics within the Kubernetes environment?

- A. Install the 'gpu-exporter' as a Deployment. Configure Prometheus to scrape these endpoints.
- B. Install the 'nvidia-dcgm-exporter' as a DaemonSet to expose GPU metrics from each node. Configure Prometheus to scrape these endpoints.
- C. Manually SSH into each node and run 'nvidia-smi' to record GPU utilization, then input the data into Prometheus.
- D. Use the Kubernetes metrics server to collect GPU utilization metrics directly from the kubelet on each node.
- E. Run 'nvidia-smi' inside each pod and expose the output as a Prometheus metric using a custom script and the Prometheus client library.

Answer: B

Explanation:
The correct answer is A. The 'nvidia-dcgm-exporter' is designed to efficiently expose GPU metrics from each node as Prometheus endpoints. Running it as a DaemonSet ensures that it runs on every node with GPUs. Option B is inefficient and requires significant overhead. Option C, the Kubernetes metrics server, does not collect detailed GPU utilization metrics. Option D is a manual and impractical approach. Option E is not standard and relies on an unspecified 'gpu-exporter' , whereas 'nvidia-dcgm-exporter' is an official solution.

NEW QUESTION # 25
You want to limit the GPU memory usage of a specific container within a Kubernetes pod running an AI inference service. How can you achieve this using NVIDIA tools and Kubernetes resources?

- A. Set resource limits for 'nvidia.com/gpu' in the pod's resource requests and limits.
- B. Set the 'CUDA VISIBLE_DEVICES' environment variable to an empty string for that container.
- C. Utilize the NVIDIA MPS (Multi-Process Service) and configure memory limits for each process using MPS control commands.
- D. Use the 'nvidia-smi' command within the container to limit the GPU memory usage of the process.
- E. Configure the Kubernetes scheduler to only schedule pods with GPU memory limits on nodes with sufficient free GPU memory.

Answer: C

Explanation:
The correct answer is C. NVIDIA MPS (Multi-Process Service) allows multiple processes to share a single GPU, and it provides mechanisms to control the memory usage of each process. By configuring MPS, you can limit the GPU memory available to a specific container. Option A disables GPU access entirely. Option B is not a reliable way to enforce memory limits. Option D only controls the number of GPUs, not the memory usage per container. Option E describes scheduling based on available memory, but doesn't enforce limits.

NEW QUESTION # 26
You need to deploy a containerized AI application from NGC using a CI/CD pipeline. The pipeline should automatically build, test, and deploy the container image to a Kubernetes cluster whenever changes are pushed to the code repository. Which of the following CI/CD tools and practices are most suitable for this scenario?

- A. Employ CircleCl with a custom script to build the image and directly push it to the container registry.
- B. Use Jenkins with the Docker plugin to build the image and 'kubectl apply' to deploy it.
- C. Use AWS CodePipeline with AWS CodeBuild to build the image and AWS Elastic Kubernetes Service (EKS) to deploy the application.
- D. Utilize GitLab CI/CD with kaniko to build the image and Helm to manage deployments.
- E. Integrate NVIDIA GPU Cloud (NGC) CLI into the CIICD pipeline to fetch and deploy pre-built containers.

Answer: C,D,E

Explanation:
B, D and E are the correct. GitLab CI/CD with kaniko and Helm provides a robust and scalable solution for building and deploying container images. NGC CLI allows fetching and deploying pre-built containers, simplifying the process. AWS CodePipeline, CodeBuild, and EKS offer a complete CI/CD solution within the AWS ecosystem. Option A is a valid but less modern approach. Option C lacks a structured deployment process.

NEW QUESTION # 27
A user complains that their AI training job is running very slowly. Upon investigation, you discover that the pod is scheduled onto a node with a slow network connection, causing significant delays in data transfer. How would you ensure that future similar jobs are scheduled onto nodes with faster network connections?

- A. Manually reschedule the pod onto a node with a faster network.
- B. Configure the kubelet to prioritize pods based on their network usage.
- C. Increase the resource requests for the pod to trigger rescheduling.
- D. Implement node affinity rules based on network bandwidth labels, and label the nodes appropriately.

- E. Use inter-pod affinity to force the job onto nodes already running network-intensive workloads.

Answer: D

Explanation:
The correct answer is B. By labeling nodes with their network bandwidth capabilities (e.g., 'network-bandwidth: 100GbpS), you can then use node affinity rules in your pod specifications to ensure that jobs requiring high bandwidth are scheduled onto suitable nodes. Option A is a temporary fix. Options C and D do not address the core issue of network bandwidth. Option E would exacerbate the problem by concentrating network-intensive workloads on the same nodes.

NEW QUESTION # 28
......