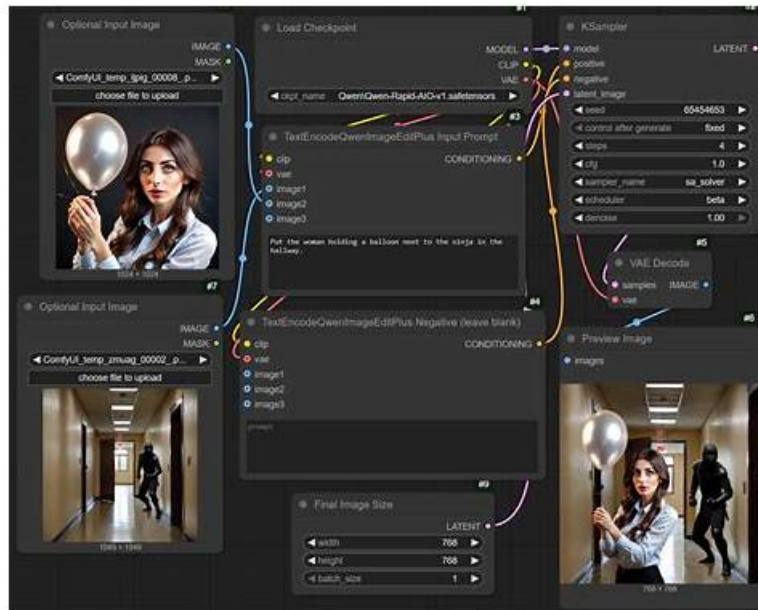


# 信頼的なNCP-AIO教育資料一回合格-高品質なNCP-AIO合格問題



さらに、Fast2test NCP-AIOダンプの一部が現在無料で提供されています：[https://drive.google.com/open?id=1WXVD78g2\\_Oo-kKCES54fa3\\_ulrPJcklJ](https://drive.google.com/open?id=1WXVD78g2_Oo-kKCES54fa3_ulrPJcklJ)

我々の目標はNCP-AIO試験を準備するあなたにヘルプを提供してあなたに試験に合格させることです。この目標を達成するために、我々Fast2testは時間とともに迅速に発展しています。今まで精確的な問題集を開発しています。我々のNCP-AIO問題集を利用しているあなたは一発で試験に合格できると信じています。心配なく我々の資料を利用してください。

## NVIDIA NCP-AIO 認定試験の出題範囲:

トピック	出題範囲
トピック 1	<ul style="list-style-type: none"> <li>インストールと展開: このセクションでは、システム管理者のスキルを測定し、インフラストラクチャのインストールと展開に関するコアプラクティスを扱います。受験者は、Base Command Manager のインストールと設定、NVIDIA ホストでの Kubernetes の初期化、NVIDIA NGC およびクラウド VMI コンテナからのコンテナの展開についてテストされます。また、AI データセンターにおけるストレージ要件の理解、DPU Armプロセッサへの DOCA サービスの展開、AI ドリブン環境の堅牢な構築についても学習します。</li> </ul>
トピック 2	<ul style="list-style-type: none"> <li>ワークロード管理: このセクションでは、AIインフラストラクチャエンジニアのスキルを評価し、AI環境におけるワークロードの効率的な管理に焦点を当てます。Kubernetes クラスターの管理、ワークロード効率の維持、システム管理ツールを用いた運用上の問題のトラブルシューティング能力を評価します。NVIDIAテクノロジーと連携し、異なる環境間でワークロードがスムーズに実行されることを重視します。</li> </ul>
トピック 3	<ul style="list-style-type: none"> <li>トラブルシューティングと最適化: NVIこの試験セクションでは、AIインフラストラクチャエンジニアのスキルを評価し、高度なAIシステムで発生する技術的な問題の診断と解決に焦点を当てます。出題範囲には、Docker、NVIDIA NVlinkおよびNVSwitchシステムの Fabric Managerサービス、Base Command Manager、Magnum IOコンポーネントのトラブルシューティングが含まれます。受験者は、ストレージパフォーマンスの問題を特定して解決し、AIワークロード全体で最適なパフォーマンスを確保する能力も実証する必要があります。</li> </ul>

#### トピック 4

- 管理: このセクションでは、システム管理者のスキルを評価し、データセンターにおけるAIワークロードの管理に不可欠なタスクを網羅します。受験者は、フリートコマンド、Slurmクラスタ管理、そしてAI環境に特有のデータセンターアーキテクチャ全体を理解していなければなりません。また、Base Command Manager (BCM)、クラスタプロビジョニング、Run.ai管理、そしてAIとハイパフォーマンスコンピューティングアプリケーションの両方に対応したマルチインスタンスGPU (MIG) の構成に関する知識も求められます。

>> NCP-AIO教育資料 <<

## 最高のNCP-AIO教育資料 & 合格スムーズNCP-AIO合格問題 | 効果的なNCP-AIO認定資格試験

NCP-AIO資格は重要な認証科目です。人数は少なくても需要は大きいため、この認証を持っている人は給料が一番高い人になっています。NCP-AIO試験に合格したら、あなたの知識と能力を証明することができます。あなたはそれらの専門家の一員になれば、あなたはいい仕事を探せます。我々のNCP-AIO問題集を利用して、試験に参加しましょう。

### NVIDIA AI Operations 認定 NCP-AIO 試験問題 (Q25-Q30):

#### 質問 # 25

A Docker container that runs a PyTorch model is experiencing CUDA out-of-memory errors during training, even though 'nvidia-smi' reports that the GPU has sufficient free memory. You suspect memory fragmentation is the cause. How do you diagnose and mitigate this issue within the Docker environment?

- A. Reduce the batch size and gradient accumulation steps to lower the overall memory footprint of the training process.
- B. Set the environment variable to force PyTorch's memory allocator to be more aggressive in garbage collecting and splitting large memory blocks.
- C. Use CUDA memory profiling tools like 'NVIDIA Nsight Systems' to identify specific memory allocations and deallocations causing fragmentation.
- D. Use the function periodically during training to release unused GPU memory and defragment the memory pool.
- E. Restart the Docker container frequently during training to clear the memory and start with a fresh allocation state.

正解: B、C、D

解説:

Memory fragmentation can lead to out-of-memory errors even with sufficient free memory. 'PYTORCH CUDA ALLOC CONF' (A) helps manage PyTorch's memory allocation. (B) defragments the memory. Profiling tools (D) pinpoint fragmentation sources. Reducing batch size (C) avoids the problem. Frequent restarts (E) are a workaround, not a solution.

#### 質問 # 26

You are deploying a multi-GPU training job using a container from NGC on a Slurm cluster. The container expects the number of GPUs to be available in the 'CUDA\_VISIBLE\_DEVICES' environment variable. How do you ensure this variable is correctly set within the Slurm job script?

- A. Configure the NVIDIA Container Toolkit to automatically detect and set 'CUDA\_VISIBLE\_DEVICES'.
- B. Set the environment variable manually in the Slurm job script to a fixed value (e.g.,
- C. Utilize the Slurm environment variable 'SLURM\_JOB\_GPUS' to dynamically set 'CUDA\_VISIBLE\_DEVICES' in the job script (e.g., 'export
- D. Use the Slurm command 'srun' with the '-gpus' option to allocate GPUs and automatically set
- E. Define the 'CUDA\_VISIBLE\_DEVICES' environment variable in the containers Docket-file.

正解: C、D

解説:

B and D are correct. 'srun -gpus' handles GPU allocation and sets the environment variable. 'SLURM\_JOB\_GPUS' provides a dynamic way to access allocated GPUs within the script. A is incorrect as it doesn't adapt to the actual allocation. C is incorrect

because it's not a Slurm configuration. E depends on the specific toolkit version and might not be reliable without explicit configuration in the job script.

#### 質問 # 27

You have successfully pulled a TensorFlow container from NGC and now need to run it on your stand-alone GPU-enabled server. Which command should you use to ensure that the container has access to all available GPUs?

- A. `docker run --gpus all nvcr.io/nvidia/tensorflow:<tag>`
- B. `kubectl create pod --gpu=all nvcr.io/nvidia/tensorflow:<tag>`
- C. `docker start nvcr.io/nvidia/tensorflow:<tag>`
- D. `docker run nvcr.io/nvidia/tensorflow:<tag>`

正解: A

解説:

When running a GPU-enabled container directly on a server with Docker, the flag `--gpus all` is required to allow the container access to all GPUs on the host system. This ensures that the TensorFlow container can utilize GPU resources fully. The other options either do not specify GPU access correctly or are Kubernetes-specific commands.

#### 質問 # 28

A data science team is using Fleet Command to deploy AI models to edge devices in a smart city project. They've noticed that some devices are consistently failing to update due to insufficient disk space. Which of the following is the MOST effective strategy to mitigate this issue?

- A. Optimize the deployed models to reduce their size, and update the affected devices only.
- B. Ignore the failing devices and focus on the ones that are updating successfully.
- C. Roll back the updates to the previous version for all devices.
- D. Implement a process to automatically clean up unused files and data on the edge devices before each update.
- E. Increase the disk space on all edge devices remotely via Fleet Command.

正解: D

解説:

Optimizing models (B) is helpful, but a cleanup process (E) addresses the root cause. Increasing disk space (A) might not be feasible or cost-effective. Ignoring devices (C) is unacceptable. Rolling back updates (D) is a temporary solution. Thus, automatically cleaning up unused files is the most proactive and sustainable approach.

#### 質問 # 29

You are using an all-flash array (AFA) for your AI training data.

a. You observe that the storage utilization is very low, but you are still experiencing performance bottlenecks. What could be the potential reasons for this and how can you troubleshoot them?

- A. The AFA's warranty has expired. Renewing the warranty will magically fix the performance issues.
- B. The AFA is not configured correctly to handle the specific I/O patterns of your AI workload (e.g., random reads, large sequential writes). Check the AFA's configuration settings for block size, caching policies, and prefetching.
- C. The AFA is over-provisioned, and the internal garbage collection processes are interfering with I/O operations. Reduce the amount of provisioned space.
- D. The network connection between the compute nodes and the AFA is the bottleneck. Upgrade the network infrastructure or optimize the data transfer protocols.
- E. The AFA's internal controllers are overloaded, even though the overall storage utilization is low. Monitor the controller utilization and consider upgrading the AFA or distributing the workload across multiple AFAs.

正解: B、D、E

解説:

Incorrect AFA configuration can lead to performance issues even with low utilization. Network bottlenecks can limit data transfer rates. Overloaded controllers within the AFA can become a bottleneck.

