OpenShift Container Platform 4.16

ハードウェアアクセラレーター

ハードウェアアクセラレーター

Last Updated: 2025-11-24

OpenShift Container Platform 4.16 ハードウェアアクセラレーター

ハードウェアアクセラレーター

Legal Notice

Abstract

このドキュメントでは、人工知能および機械学習 (AI/ML) アプリケーションを作成するのに提供されるハードウェアアクセラレーション機能用に、Red Hat OpenShift AI でサポートされている GPU Operator をインストールおよび設定する手順を説明します。

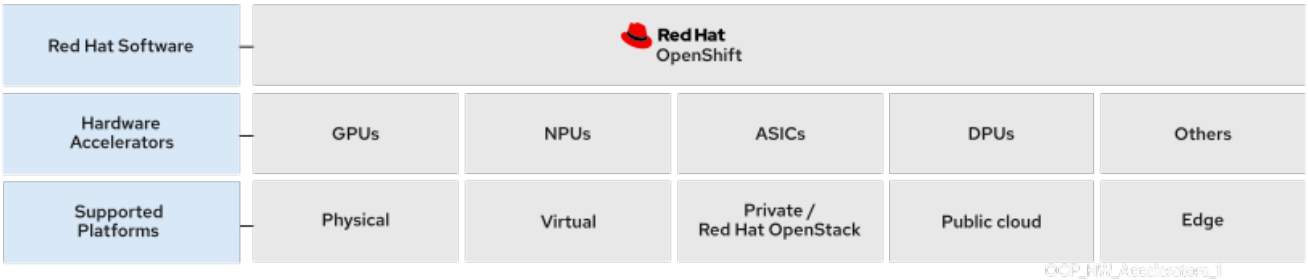
Table of Contents

第1章 ハードウェアアクセラレーターについて

専用ハードウェアアクセラレーターは、新しい生成人工知能および機械学習 (AI/ML) 業界で重要な役割を果たします。具体的には、ハードウェアアクセラレーターは、この新しいテクノロジーを支える大規模言語モデルやその他の基礎モデルのトレーニングと提供に不可欠です。データサイエンティスト、データエンジニア、ML エンジニア、開発者は、データ量の多い変換やモデルの開発と提供に特化したハードウェアアクセラレーションを活用できます。そのエコシステムの多くはオープンソースであり、複数の貢献パートナーとオープンソース財団が存在します。

Red Hat OpenShift Container Platform は、ハードウェアアクセラレーターの構成要素である次の処理ユニットを追加するカードと周辺ハードウェアをサポートしています。

- グラフィックスプロセッシングユニット (GPU)
- ニューラルプロセッシングユニット (NPU)
- 特定用途向け集積回路 (ASIC)
- データプロセッシングユニット (DPU)



専用ハードウェアアクセラレーターは、AI/ML 開発にさまざまな利点をもたらします。

1つのプラットフォームであらゆる用途に対応

開発者、データエンジニア、データサイエンティスト、DevOps のためのコラボレーション環境

Operator による機能拡張

Operator により OpenShift Container Platform に AI/ML 機能を導入可能

ハイブリッドクラウドのサポート

モデルの開発、提供、デプロイのためのオンプレミスサポート

AI/ML ワークロードのサポート

モデルのテスト、イテレーション、統合、プロモートを行い、サービスとして運用環境に提供

Red Hat は、Red Hat Enterprise Linux (RHEL) および OpenShift Container Platform プラットフォームの Linux (カーネルとユーザー空間) および Kubernetes レイヤーで、このような専用ハードウェアアクセラレーターを有効にするために最適化されたプラットフォームを提供しています。これを実現するために、Red Hat は、Red Hat OpenShift AI と Red Hat OpenShift Container Platform の実証済みの機能を、単一のエンタープライズ対応 AI アプリケーションプラットフォームに統合しました。

ハードウェア Operator は、Kubernetes クラスターのオペレーティングフレームワークを使用して、必要なアクセラレーターリソースを有効にします。提供されているデバイスプラグインを手動で、またはデーモンセットとしてデプロイすることもできます。このプラグインにより、クラスターに GPU が登録されます。

専用ハードウェアアクセラレーターの中には、開発とテストのためのセキュリティーを確保する必要がある非接続環境内で動作するように設計されているものもあります。

1.1. ハードウェアアクセラレーター

Red Hat OpenShift Container Platform では、次のハードウェアアクセラレーターが有効になります。

- NVIDIA GPU
- AMD Instinct® GPU
- Intel® Gaudi®

関連情報

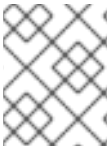
- Red Hat OpenShift AI の概要
- Red Hat OpenShift Container Platform 上の NVIDIA GPU Operator
- AMD Instinct Accelerators
- Intel Gaudi AI Accelerators

第2章 NVIDIA GPU アーキテクチャー

NVIDIA は、OpenShift Container Platform でのグラフィックスプロセッシングユニット (GPU) リソースの使用をサポートしています。OpenShift Container Platform は、大規模な Kubernetes クラスターのデプロイと管理用に Red Hat が開発およびサポートする、セキュリティーを重視して強化された Kubernetes プラットフォームです。OpenShift Container Platform には Kubernetes の拡張機能が含まれているため、ユーザーはが簡単に NVIDIA GPU リソースを設定し、それを使用してワークロードを高速化できます。

NVIDIA GPU Operator は、OpenShift Container Platform 内の Operator フレームワークを使用して、GPU で高速化されたワークロードの実行に必要な NVIDIA ソフトウェアコンポーネントのライフサイクル全体を管理します。

これらのコンポーネントには、NVIDIA ドライバー (CUDA を有効にするため)、GPU 用の Kubernetes デバイスプラグイン、NVIDIA Container Toolkit、GPU Feature Discovery (GFD) を使用した自動ノードタグ付け、DCGM ベースのモニタリングなどが含まれます。



注記

NVIDIA GPU Operator をサポートしているのは NVIDIA だけです。NVIDIA からサポートを受ける方法は、NVIDIA サポートの利用方法 を参照してください。

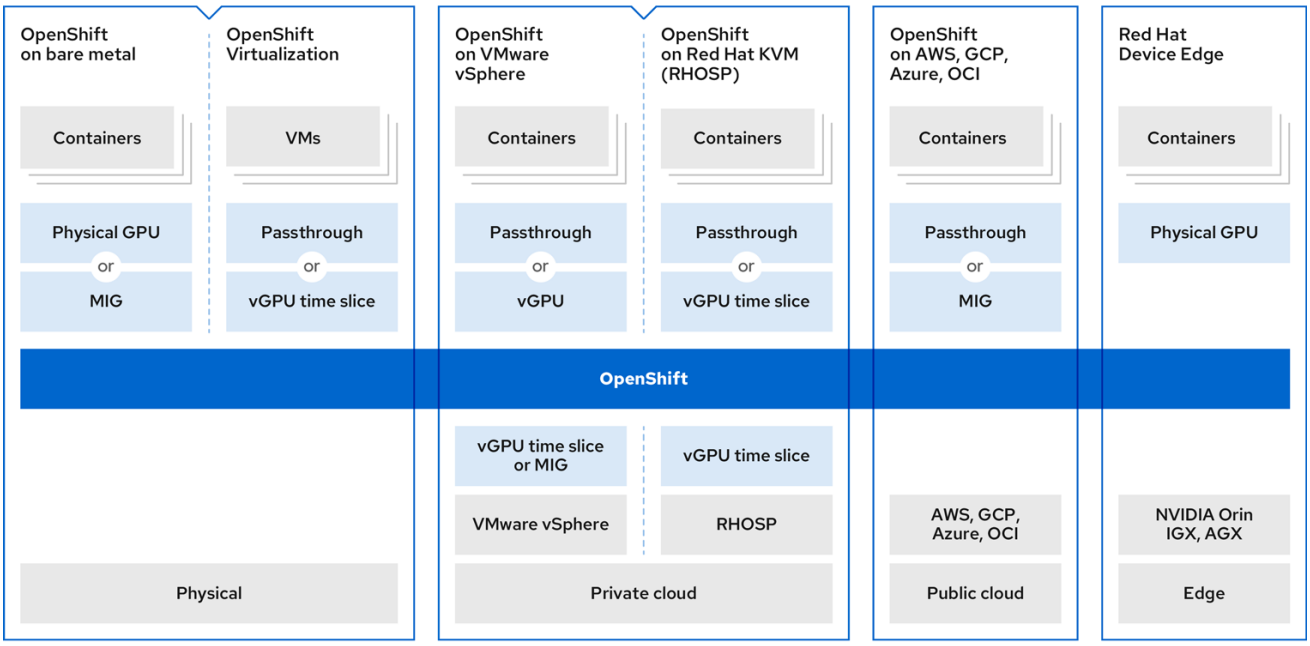
2.1. NVIDIA GPU の前提条件

- 1 つ以上の GPU ワーカーノードを備えた OpenShift クラスターが稼働している。
- 必要な手順を実行するために **cluster-admin** として OpenShift クラスターにアクセスできる。
- OpenShift CLI (**oc**) がインストールされている。
- Node Feature Discovery (NFD) Operator をインストールし、**nodefeaturediscovery** インスタンスを作成している。

2.2. NVIDIA GPU の有効化

以下の図は、OpenShift で GPU アーキテクチャーがどのように有効になっているかを示しています。

図2.1 NVIDIA GPU の有効化



注記

MIG は、A30、A100、A100X、A800、AX800、H100、H800 でのみサポートされます。

2.2.1. GPU とベアメタル

NVIDIA 認定のベアメタルサーバーに OpenShift Container Platform をデプロイできますが、いくつかの制限があります。

- コントロールプレーンノードは CPU ノードにできます。
- AI/ML ワークロードがワーカーノードで実行される場合、そのワーカーノードは GPU ノードである必要があります。
  さらに、ワーカーノードは 1 つ以上の GPU をホストできますが、すべて同じタイプである必要があります。たとえば、ノードには 2 つの NVIDIA A100 GPU が存在することは可能ですが、A100 GPU と T4 GPU を 1 つずつ備えたノードはサポートされません。Kubernetes の NVIDIA デバイスプラグインは、同じノード上で異なる GPU モデルの組み合わせをサポートしません。
- OpenShift を使用する場合は、1 台または 3 台以上のサーバーが必要な点に注意してください。2 台のサーバーを含むクラスターはサポートされません。単一サーバーのデプロイメントはシングルノード openShift (SNO) と呼ばれ、この設定を使用すると、高可用性 OpenShift 環境が得られません。

以下のいずれかの方法で、コンテナー化された GPU にアクセスできます。

- GPU パススルー
- マルチインスタンス GPU (MIG)

関連情報

- Red Hat OpenShift on Bare Metal Stack

2.2.2. GPU と仮想化

多くの開発者や企業がコンテナー化されたアプリケーションやサーバーレスインフラストラクチャーに移行していますが、仮想マシン上で実行されるアプリケーションの開発と保守は引き続き注目されています。Red Hat OpenShift Virtualization はこの機能を提供し、企業はこの機能を使用して仮想マシンをクラスター内のコンテナー化されたワークフロー組み込むことができます。

ワーカーノードを GPU に接続する場合は、次のいずれかの方法を選択できます。

- 仮想マシン内の GPU ハードウェアにアクセスして使用するための GPU パススルー。
- GPU コンピュート容量がワークロードでいっぱいになっていない場合の GPU (vGPU) のタイムスライス。

関連情報

- NVIDIA GPU Operator with OpenShift Virtualization

2.2.3. GPU と vSphere

OpenShift Container Platform は、さまざまな GPU タイプをホストできる NVIDIA 認定の VMware vSphere サーバーにデプロイできます。

仮想マシンで vGPU インスタンスが使用されている場合は、NVIDIA GPU ドライバーをハイパーバイザーにインストールする必要があります。VMware vSphere の場合、このホストドライバーは VIB ファイルの形式で提供されます。

ワーカーノード仮想マシンに割り当てることができる vGPUS の最大数は、vSphere のバージョンによって異なります。

- vSphere 7.0: 仮想マシンごとに最大 4 つの仮想 GPU
- vSphere 8.0: 仮想マシンごとに最大 8 つの仮想 GPU

注記

vSphere 8.0 では、仮想マシンに関連付けられた複数の完全または部分的な異種プロファイルのサポートが導入されました。

次のいずれかの方法を選択して、ワーカーノードを GPU に割り当てることができます。

- 仮想マシン内の GPU ハードウェアにアクセスして使用するための GPU パススルー
- すべての GPU が必要でない場合の GPU (vGPU) タイムスライス

ベアメタルデプロイメントと同様に、1 台または 3 台以上のサーバーが必要です。2 台のサーバーを含むクラスターはサポートされません。

関連情報

- OpenShift Container Platform on VMware vSphere with NVIDIA vGPUs

2.2.4. GPU および Red Hat KVM

OpenShift Container Platform は、NVIDIA 認定のカーネルベースの仮想マシン (KVM) サーバー上で使用できます。

ベアメタルデプロイメントと同様に、1 台または 3 台以上のサーバーが必要です。2 台のサーバーを含むクラスターはサポートされません。

ただし、ベアメタルデプロイメントとは異なり、サーバーで異なるタイプの GPU を使用できます。これは、GPU を Kubernetes ノードとして機能する別の仮想マシンに割り当てることができるためです。唯一の制限として、Kubernetes ノードがノードと同レベルで GPU タイプのセットを持つ必要があります。

以下のいずれかの方法で、コンテナー化された GPU にアクセスできます。

- 仮想マシン内の GPU ハードウェアにアクセスして使用するための GPU パススルー
- すべての GPU が必要でない場合の GPU (vGPU) タイムスライス

vGPU 機能を有効にするには、特別なドライバーをホストレベルでインストールする必要があります。このドライバーは RPM パッケージとして提供されます。このホストドライバーは、GPU パススルーの割り当てにはまったく必要ありません。

関連情報

- How To Deploy OpenShift Container Platform 4.13 on KVM

2.2.5. GPU と CSP

OpenShift Container Platform は、主要なクラウドサービスプロバイダー (CSP) である Amazon Web Services (AWS)、Google Cloud、Microsoft Azure のいずれかにデプロイできます。

フルマネージドデプロイメントとセルフマネージドデプロイメントの 2 つのオペレーションモードを使用できます。

- フルマネージドデプロイメントでは、Red Hat が CSP と連携してすべてを自動化します。お客様は CSP の Web コンソールを使用して OpenShift インスタンスを要求できます。クラスターは自動的に作成され、Red Hat によって完全に管理されます。この環境内では、ノードの障害やエラーを心配する必要はありません。クラスターの稼働時間を維持する責任は Red Hat がすべて負います。フルマネージドサービスは、AWS、Azure、Google Cloud で利用できます。AWS の場合、OpenShift サービスは ROSA (Red Hat OpenShift Service on AWS) と呼ばれます。Azure の場合、このサービスは Azure Red Hat OpenShift と呼ばれます。Google Cloud の場合、このサービスは OpenShift Dedicated on Google Cloud と呼ばれます。
- セルフマネージドデプロイメントでは、お客様が OpenShift クラスターのインスタンス化と維持を行う必要があります。この場合、Red Hat は OpenShift クラスターのデプロイを支援するために、OpenShift-install ユーティリティーを提供します。セルフマネージドサービスは、世界中のすべての CSP で利用できます。

このコンピュートインスタンスが GPU により高速化されたコンピュートインスタンスであること、および GPU タイプが NVIDIA AI Enterprise でサポートされている GPU のリストと一致することが重要です。たとえば、T4、V100、A100 はこのリストに含まれます。

以下のいずれかの方法で、コンテナー化された GPU にアクセスできます。

- 仮想マシン内の GPU ハードウェアにアクセスして使用するための GPU パススルー。
- GPU 全体を必要としない場合 GPU (vGPU) タイムスライス。

関連情報

- Red Hat Openshift in the Cloud

2.2.6. GPU と Red Hat Device Edge

Red Hat Device Edge は MicroShift へのアクセスを提供します。MicroShift は、シングルノードデプロイメントのシンプルさと、リソースに制約のある (エッジ) コンピューティング求められる機能とサービスを備えています。Red Hat Device Edge は、リソースに制約のある環境にデプロイされるベアメタル、仮想、コンテナー化された、または Kubernetes のワークロードのニーズを満たします。

Red Hat Device Edge 環境のコンテナー上で NVIDIA GPU を有効にできます。

コンテナー化された GPU へのアクセスには、GPU パススルーを使用します。

関連情報

- Red Hat Device Edge 上の NVIDIA GPU を使用してワークロードを高速化する方法

2.3. GPU の共有方法

Red Hat と NVIDIA は、エンタープライズレベルの OpenShift Container Platform クラスター上で、GPU 加速コンピューティングを簡略化するための GPU 同時実行性と共有メカニズムを開発しました。

通常、アプリケーションにはさまざまなコンピューティング要件があり、GPU が十分に活用されていない可能性があります。デプロイメントコストを削減し、GPU 使用率を最大化するには、ワークロードごとに適切な量のコンピュートリソースを提供することが重要です。

GPU 使用率を改善するための同時実行メカニズムは、プログラミングモデル API からシステムソフトウェアやハードウェアパーティショニングまで、仮想化を含めて幅広く存在します。次のリストは、GPU 同時実行メカニズムを示しています。

- Compute Unified Device Architecture (CUDA) ストリーム
- タイムスライス
- CUDA マルチプロセスサービス (MPS)
- マルチインスタンス GPU (MIG)
- vGPU による仮想化

さまざまな OpenShift Container Platform シナリオで GPU 同時実行メカニズムを使用する場合は、次の GPU 共有に関する推奨事項を考慮してください。

ベアメタル
: vGPU は使用できません。MIG 対応カードの使用を検討してください。

仮想マシン
: vGPU が最良の選択です。

ベアメタル上の MIG を持たない古い NVIDIA カード
: タイムスライスの使用を検討してください。

複数の GPU を搭載し、パススルーと vGPU が必要な仮想マシン
: 個別の仮想マシンの使用を検討してください。

OpenShift Virtualization と複数の GPU を備えたベアメタル

ホストされた仮想マシンにはパススルー、コンテナーにはタイムスライスの使用を検討してください。

関連情報

- GPU 使用率の向上

2.3.1. CUDA ストリーム

Compute Unified Device Architecture (CUDA) は、GPU での計算全般のために NVIDIA が開発した並列コンピューティングプラットフォームおよびプログラミングモデルです。

ストリームは、GPU 上で発行順に実行される一連の操作です。CUDA コマンドは通常、デフォルトストリームで順次実行され、前のタスクが完了するまでタスクは開始されません。

ストリームをまたいだ操作の非同期処理により、タスクの並列実行が可能になります。あるストリームで発行されたタスクは、別のタスクが別のストリームで発行される前、実行中、または発行された後に実行されます。これにより、GPU は指定された順序に関係なく複数のタスクを同時に実行できるようになり、パフォーマンスの向上につながります。

関連情報

- Asynchronous Concurrent Execution

2.3.2. タイムスライス

GPU タイムスライスは、複数の CUDA アプリケーションを実行しているときに、過負荷になった GPU でスケジュールされたワークロードをインターリーブします。

Kubernetes で GPU のタイムスライスを有効にするには、GPU のレプリカセットを定義し、それを個別に Pod に配分してワークロードを実行できるようにしっます。マルチインスタンス GPU (MIG) とは異なり、メモリーや障害はレプリカ間で分離されませんが、一部のワークロードでは一切共有しないより、こちらの方が適切です。内部的には、GPU タイムスライスを使用して、基礎である同じ GPU のレプリカからのワークロードを多重化します。

クラスター全体のデフォルト設定をタイムスライスに適用できます。ノード固有の設定を適用することもできます。たとえば、タイムスライス設定を Tesla T4 GPU を備えたノードにのみ適用し、他の GPU モデルを備えたノードは変更しないようにできます。

クラスター全体のデフォルト設定を適用し、ノードにラベルを付けて、それらのノードにノード固有の設定が適用されるようにすることで、2 つのアプローチを組み合わせることができます。

2.3.3. CUDA マルチプロセスサービス

CUDA マルチプロセスサービス (MPS) を使用すると、単一の GPU で複数の CUDA プロセスを使用できます。プロセスは GPU 上で並行して実行されるため、GPU コンピュートリソースの飽和が発生しなくなります。MPS を使用すると、カーネル操作や、別のプロセスからのメモリーコピーの同時実行または重複も可能になり、使用率が向上します。

関連情報

- CUDA MPS

2.3.4. マルチインスタンス GPU

マルチインスタンス GPU (MIG) を使用すると、GPU コンピュートユニットとメモリーを複数の MIG インスタンスに分割できます。これらの各インスタンスは、システムの観点からはスタンドアロン GPU デバイスであり、ノード上で実行されている任意のアプリケーション、コンテナー、または仮想マシンに接続できます。GPU を使用するソフトウェアは、これらの各 MIG インスタンスを個別の GPU として扱います。

MIG は、GPU 全体のフルパワーを必要としないアプリケーションがある場合に役立ちます。新しい NVIDIA Ampere アーキテクチャーの MIG 機能を使用すると、ハードウェアリソースを複数の GPU インスタンスに分割できます。各インスタンスは、オペレーティングシステムで独立した CUDA 対応 GPU として利用できます。

NVIDIA GPU Operator バージョン 1.7.0 以降では、A100 および A30 Ampere カードの MIG サポートを提供しています。これらの GPU インスタンスは、最大 7 つの独立した CUDA アプリケーションをサポートするように設計されており、専用のハードウェアリソースをしようしてそれぞれ完全に分離された状態で稼働します。

関連情報

- NVIDIA Multi-Instance GPU User Guide

2.3.5. vGPU による仮想化

仮想マシンは、NVIDIA vGPU を使用して単一の物理 GPU に直接アクセスできます。企業全体の仮想マシンで共有され、他のデバイスからアクセスできる仮想 GPU を作成できます。

この機能は、GPU パフォーマンスのパワーと、vGPU がもたらす管理およびセキュリティーの利点を組み合わせたものです。vGPU には他にも、仮想環境のプロアクティブな管理と監視、混合 VDI とコンピュートワークロードのワークロードバランシング、複数の仮想マシン間でのリソース共有などの利点があります。

関連情報

- Virtual GPUs

2.4. OPENSHIFT CONTAINER PLATFORM の NVIDIA GPU 機能

NVIDIA Container Toolkit

NVIDIA Container Toolkit を使用すると、GPU で高速化されたコンテナーを作成して実行できます。ツールキットには、コンテナーが NVIDIA GPU を使用するように自動的に設定するためのコンテナーランタイムライブラリーとユーティリティーが含まれています。

NVIDIA AI Enterprise

NVIDIA AI Enterprise は、NVIDIA 認定システムで最適化、認定、サポートされている AI およびデータ分析ソフトウェアのエンドツーエンドのクラウドネイティブスイートです。
NVIDIA AI Enterprise には、Red Hat OpenShift Container Platform のサポートが含まれています。サポートされているインストール方法は以下のとおりです。

- GPU パススルーを使用するベアメタルまたは VMware vSphere 上の OpenShift Container Platform。
- NVIDIA vGPU を使用する VMware vSphere 上の OpenShift Container Platform。

GPU Feature Discovery

NVIDIA GPU Feature Discovery for Kubernetes は、ノード上で使用可能な GPU のラベルを自動的に生成できるソフトウェアコンポーネントです。GPU Feature Discovery は、Node Feature Discovery (NFD) を使用してこのラベル付けを実行します。
Node Feature Discovery (NFD) Operator は、ハードウェア固有の情報でノードにラベル付けを行うことで、OpenShift Container Platform クラスターのハードウェア機能と設定の検出を管理します。NFD は、PCI カード、カーネル、OS バージョンなどのノード固有の属性で、ホストにラベル付けを行います。

Operator Hub で NFD Operator を見つけるには、"Node Feature Discovery" で検索してください。

NVIDIA GPU Operator with OpenShift Virtualization

これまで、GPU Operator は、GPU で高速化されたコンテナーを実行するためにワーカーノードのみをプロビジョニングしていました。現在は、GPU Operator を使用して、GPU で高速化された仮想マシンを実行するためのワーカーノードもプロビジョニングできます。
GPU Operator を、どの GPU ワークロードがそのワーカーノード上で実行するように設定されたかに応じて、異なるソフトウェアコンポーネントをワーカーノードにデプロイするように設定できます。

GPU モニタリングダッシュボード

モニタリングダッシュボードをインストールして、OpenShift Container Platform Web コンソールのクラスターの **Observe** ページに、GPU の使用状況に関する情報を表示できます。GPU 使用状況に関する情報には、使用可能な GPU の数、消費電力 (ワット単位)、温度 (摂氏)、使用率 (パーセント)、および各 GPU のその他のメトリクスが含まれます。

関連情報

- NVIDIA-Certified Systems
- NVIDIA AI Enterprise
- NVIDIA Container Toolkit
- GPU モニタリングダッシュボードの有効化
- MIG Support in OpenShift Container Platform
- OpenShift での NVIDIA GPU のタイムスライス
- Deploy GPU Operators in a disconnected or airgapped environment
- Node Feature Discovery Operator

第3章 AMD GPU OPERATOR

OpenShift Container Platform クラスター内で AMD Instinct GPU アクセラレーターと AMD GPU Operator を併用することで、機械学習、生成 AI、および GPU アクセラレーションアプリケーション向けのコンピューティング能力をシームレスに活用できます。

このドキュメントでは、AMD GPU Operator を有効化、設定、テストするために必要な情報を提供します。詳細は、AMD Instinct™ Accelerators を参照してください。

3.1. AMD GPU OPERATOR について

AMD GPU Operator のハードウェアアクセラレーション機能は、Red Hat OpenShift AI を使用して人工知能および機械学習 (AI/ML) アプリケーションを作成するデータサイエンティストや開発者に、高いパフォーマンスとコスト効率を提供します。GPU 機能の特定の領域を高速化すると、CPU 処理とメモリー使用量を最小限に抑え、全体的なアプリケーション速度、メモリー消費、帯域幅の制約を改善できます。

3.2. AMD GPU OPERATOR のインストール

クラスター管理者は、OpenShift CLI と Web コンソールを使用して AMD GPU Operator をインストールできます。これは複数のステップから成る手順であり、Node Feature Discovery Operator、Kernel Module Management Operator、AMD GPU Operator のインストールが必要です。Operator の AMD コミュニティー版リリースをインストールするには、次の手順を順に実行します。

次のステップ

1. Node Feature Discovery Operator をインストールします。
2. Kernel Module Management Operator をインストールします。
3. AMD GPU Operator をインストールして設定します。

3.3. AMD GPU OPERATOR のテスト

ROCmInfo のインストールをテストし、AMD MI210 GPU のログを表示するには、次の手順を使用します。

手順

1. ROCmInfo をテストする YAML ファイルを作成します。

```
$ cat << EOF > rocminfo.yaml

apiVersion: v1
kind: Pod
metadata:
 name: rocminfo
spec:
 containers:
 - image: docker.io/rocm/pytorch:latest
   name: rocminfo
   command: ["/bin/sh","-c"]
   args: ["rocminfo"]
   resources:
```

```
    limits:
      amd.com/gpu: 1
    requests:
      amd.com/gpu: 1
 restartPolicy: Never
EOF
```

2. **rocminfo** Pod を作成します。

```
$ oc create -f rocminfo.yaml
```

出力例

```
apiVersion: v1
pod/rocminfo created
```

3. 1 つの MI210 GPU を含む **rocmnfo** ログを確認します。

```
$ oc logs rocminfo | grep -A5 "Agent"
```

出力例

```
HSA Agents
==========
*******
Agent 1
*******
  Name:                    Intel(R) Xeon(R) Gold 6330 CPU @ 2.00GHz
  Uuid:                    CPU-XX
  Marketing Name:          Intel(R) Xeon(R) Gold 6330 CPU @ 2.00GHz
  Vendor Name:             CPU
--
Agent 2
*******
  Name:                    Intel(R) Xeon(R) Gold 6330 CPU @ 2.00GHz
  Uuid:                    CPU-XX
  Marketing Name:          Intel(R) Xeon(R) Gold 6330 CPU @ 2.00GHz
  Vendor Name:             CPU
--
Agent 3
*******
  Name:                    gfx90a
  Uuid:                    GPU-024b776f768a638b
  Marketing Name:          AMD Instinct MI210
  Vendor Name:             AMD
```

4. Pod を削除します。

```
$ oc delete -f rocminfo.yaml
```

出力例

```
pod "rocminfo" deleted
```