硬件加速器

使用本节选择适合容器工作负载的加速器路径，了解会影响资源请求的 ACP 概念，并继续查看产品文档中的安装、升级、兼容性和故障排查内容。

从最接近您任务的场景开始。厂商集成会为硬件、驱动、运行时以及直接分配做好准备。HAMi 等共享和调度产品建立在这些集成之上，提供共享、虚拟化、调度或资源编排行为。

目录

选择场景

如果您想要...	从这里开始
使用具有厂商标准组件的 NVIDIA GPU	GPU 加速器系列 和 NVIDIA 厂商集成
使用具有厂商标准组件在 Ascend NPU 上运行容器工作负载	NPU 加速器系列 和 Ascend 厂商集成
在多个工作负载之间共享、虚拟化或调度 GPU 或 NPU 设备	HAMi 共享和调度路径

如果您想要...	从这里开始
使用基于 Kubernetes DRA 的资源声明	DRA 机制
在 ACP 配额页面中显示加速器资源，或配置配额字段元数据	加速器资源配额
为虚拟机使用 GPU 直通	使用 虚拟机物理 GPU 配置。这与容器工作负载的 pGPU 或 vGPU 使用路径不同。

按类别浏览

- 概览：Device Plugin、DRA、RuntimeClass、CDI、扩展资源、配额以及监控入口点等概念。
- 加速器系列：面向用户的硬件系列，例如 GPU 和 NPU。
- 产品：厂商集成，例如 NVIDIA 和 Ascend，以及共享或调度产品，例如 HAMi。

相关 ACP 页面

关于配额、插件安装、监控仪表盘、虚拟机 GPU 直通和计费，请继续查看相应的 ACP 功能文档：

- 项目集群配额
- 命名空间资源配额
- 集群插件
- 监控仪表盘
- 虚拟机物理 GPU 配置
- Alauda 成本管理

概览

这些页面说明了会影响加速器使用的 Kubernetes 和 ACP 概念。

当你需要了解某个 workload 为什么请求特定的资源名称、为什么需要 runtime integration，或者为什么不同的 accelerator 产品以不同方式暴露资源时，请使用本节。有关产品安装和故障排查，请继续查看每个主题中链接到的产品页面。

Device Plugin

何时使用

相关产品

相关页面

DRA

它如何影响工作负载

产品文档

相关产品

相关页面

RuntimeCla

它对工作负载的

相关产品

相关页面

加速器配额

配额资源

在 ACP 中显示配额字段

示例

配置配额

配额页面

相关页面

加速器监控

您可以监控的内

指标来源

相关页面

Device Plugin

Device Plugin 是 Kubernetes 中大多数加速器产品用于向调度器声明设备容量，并让选定设备可供 Pods 使用的机制。

在 ACP 中，Device Plugin 很重要，因为 GPU 和 NPU 产品通常通过这一路径暴露扩展资源。安装相应产品后，用户可以在工作负载中请求 NVIDIA GPU、HAMi 虚拟 GPU 或 Ascend NPU 等资源，并通过项目和命名空间设置来管理配额。

目录

何时使用

- 当工作负载通过 Kubernetes 扩展资源名称请求加速器资源时，请使用基于 Device Plugin 的产品。
- 请使用 ACP 的项目和命名空间页面执行配额操作。
- 请参考产品文档了解安装、节点标签、受支持的硬件和故障排查。

相关产品

- NVIDIA GPU Device Plugin 暴露物理 NVIDIA GPU 资源。
- HAMi 使用 device plugin 和调度组件来提供共享加速器资源。
- Ascend NPU Operator 包含将 Ascend 设备暴露给 Kubernetes 工作负载的组件。

相关页面

- GPU 加速器系列
- NPU 加速器系列
- 资源配额

DRA

Dynamic Resource Allocation (DRA) 是一种 Kubernetes 机制，用于通过 resource claim 对象请求和分配设备。它并不只是 GPU 专属能力。

在 ACP 中，DRA 通过产品驱动程序使用。DRA driver 是 DRA 机制的一种实现，而不是该机制本身。

目录

它如何影响工作负载

当满足以下情况时，DRA 相关：

- 工作负载使用 `ResourceClaim` 或 `ResourceClaimTemplate`，而不是仅使用扩展资源名称；
- 加速器产品文档中提到 `DeviceClass`、`ResourceSlice` 或基于 claim 的分配；
- 集群同时存在基于传统 Device Plugin 的产品和基于 DRA 的产品，并且你需要确定工作负载使用的是哪条分配路径。

产品文档

使用本页面了解 DRA 对工作负载资源请求的含义，以及如何将 DRA 与传统的基于 Device Plugin 的分配方式区分开来。有关 DRA driver 的安装、Kubernetes 版本要求、feature gate、节点标签、claim 示例、验证和故障排查，请参阅产品文档。

相关产品

仅当相应的 Alauda 产品或 package 明确支持 DRA driver 时，才应在产品站点中记录该 DRA driver。

- 厂商 DRA driver 属于对应的厂商集成。
- HAMi DRA driver 属于 HAMi 的共享和调度路径，因为 HAMi 负责共享、虚拟化、调度以及资源语义。
- 未来的 GPU、NPU 或其他加速器产品也可能使用 DRA。

相关页面

- NVIDIA 厂商集成
- HAMi 共享和调度路径
- GPU 加速器产品族
- NPU 加速器产品族
- RuntimeClass 和 CDI

RuntimeClass 和 CDI

RuntimeClass 和 Container Device Interface (CDI) 是运行时集成机制。它们决定在调度器选择节点和设备之后，如何将选定的设备以及所需的运行时文件暴露给容器。

这些机制与加速器用户相关，因为工作负载可能已成功调度，但如果设备运行时集成缺失或不兼容，仍可能在运行时失败。

目录

它对工作负载的影响

- Device Plugin 和 DRA 决定如何对设备进行广告和分配。
- RuntimeClass 和 CDI 决定容器运行时如何注入设备文件、库和运行时配置。
- 产品文档会说明工作负载是否必须设置 `runtimeClassName`，是否需要 CDI，以及支持哪些容器运行时版本。

相关产品

- NVIDIA 产品可能依赖 NVIDIA Container Toolkit 和 CDI，或依赖 NVIDIA runtime handler，具体取决于所选的产品路径。
- Ascend NPU 产品可能使用 CDI 或厂商运行时组件来注入 Ascend 设备和库。
- HAMi 可能会根据设备类型和产品版本，以不同方式与运行时组件集成。

相关页面

- Device Plugin
- DRA
- GPU accelerator family
- NPU accelerator family

加速器配额

加速器产品通常通过 Kubernetes 扩展资源名称暴露资源。安装产品后，ACP 可以在项目和命名空间配额中使用这些资源名称。

目录

配额资源

在集群中安装加速器产品后，ACP 可以在项目或命名空间作用域下显示相应的加速器配额字段。具体的资源名称和单位取决于产品。

示例包括：

- 物理 NVIDIA GPU 数量；
- HAMi 虚拟 GPU 核心和内存资源；
- Ascend NPU 资源；

- DRA 或厂商特定的资源类型。

在 **ACP** 中显示配额字段

Kubernetes 节点可分配资源与 ACP 配额字段相关，但并不相同。节点可以报告加速器扩展资源，而 ACP 配额页面在将该资源显示为可选配额字段之前，还需要资源元数据。

如果节点已经报告了该加速器资源，但 ACP 配额页面没有显示它，请在 `kube-public` 命名空间中使用 ConfigMap 注册该资源元数据。

> **NOTE**
>
> 要求如下：
>
> - 每个加速器资源名称创建一个 ConfigMap。
> - 在 `kube-public` 命名空间中创建该 ConfigMap。
> - `metadata.name` 使用 `cf-crl-<groupName>-<name>` 格式。
> - 将 `metadata.labels.features.cpaas.io/type` 设置为 `CustomResourceLimitation`。
> - 将 `metadata.labels.features.cpaas.io/group` 和 `data.group` 设置为相同的组名称。
> - 将 `metadata.labels.features.cpaas.io/enabled` 设置为 `"true"`。
> - 将 `data.key` 设置为准确的 Kubernetes 扩展资源名称，例如 `nvidia.com/gpualloc` 或 `huawei.com/Ascend910`。

使用以下字段控制 ACP 在配额页面中如何显示和应用该资源：

- `data.labelEn` 和 `data.labelZh`：显示给用户的名称。
- `data.descriptionEn` 和 `data.descriptionZh`：显示给用户的帮助文本。
- `data.limits`：设置为 `required` 或 `optional`。
- `data.requests`：设置为 `disabled` 或 `fromLimits`。
- `data.resourceUnit`：资源单位，例如 `count`。
- `data.relatedResources`：应一起使用的资源名称。
- `data.excludeResources`：不应与此资源混用的资源名称。

- `data.ignoreNodeCheck`：当配额字段派生自其他资源且不需要直接检查节点可分配资源时，设置为 `"true"`。

如果产品暴露多个资源名称，请为每个资源名称创建一个 ConfigMap。例如，共享 GPU 产品可能会分别暴露 GPU 核心和 GPU 内存的资源名称。

示例

HAMi NVIDIA

以下示例注册了 HAMi NVIDIA 资源字段，用于 GPU 数量、vGPU 核心和 vGPU 内存：

```
apiVersion: v1
data:
  dataType: integer
  defaultValue: "1"
  descriptionEn: Declare how many physical GPUs needs
  descriptionZh: 申请的物理 gpu 个数
  group: hami-nvidia
  groupI18n: '{"zh": "HAMi NVIDIA", "en": "HAMi NVIDIA"}'
  key: nvidia.com/gpualloc
  labelEn: gpu number
  labelZh: gpu 个数
  limits: optional
  requests: disabled
  resourceUnit: "count"
  relatedResources: "nvidia.com/gpucores,nvidia.com/gpumem"
  excludeResources: "nvidia.com/mps-core,nvidia.com/mps-memory,tencen
t.com/vcuda-core,tencent.com/vcuda-memory"
  runtimeClassName: ""
kind: ConfigMap
metadata:
  labels:
    features.cpaas.io/enabled: "true"
    features.cpaas.io/group: hami-nvidia
    features.cpaas.io/type: CustomResourceLimitation
  name: cf-crl-hami-nvidia-gpualloc
  namespace: kube-public
---
apiVersion: v1
data:
  dataType: integer
  defaultValue: "20"
  descriptionEn: Declare how many cores needs per physical GPU (%)
  descriptionZh: 每个物理 GPU 申请的算力（%）
  group: hami-nvidia
  groupI18n: '{"zh": "HAMi NVIDIA", "en": "HAMi NVIDIA"}'
  key: nvidia.com/gpucores
  labelEn: vgpu cores
  labelZh: vgpu 算力
  limits: optional
  requests: disabled
  resourceUnit: "%"
  relatedResources: "nvidia.com/gpualloc,nvidia.com/gpumem"
  excludeResources: "nvidia.com/mps-core,nvidia.com/mps-memory,tencen
```

```
t.com/vcuda-core,tencent.com/vcuda-memory"
  runtimeClassName: ""
  ignoreNodeCheck: "true"
kind: ConfigMap
metadata:
  labels:
    features.cpaas.io/enabled: "true"
    features.cpaas.io/group: hami-nvidia
    features.cpaas.io/type: CustomResourceLimitation
  name: cf-crl-hami-nvidia-gpucores
  namespace: kube-public
---
apiVersion: v1
data:
  dataType: integer
  defaultValue: "4000"
  descriptionEn: Declare how many memory needs per physical GPU (Mi)
  descriptionZh: 每个物理 GPU 申请的显存（Mi）
  group: hami-nvidia
  groupI18n: '{"zh": "HAMi NVIDIA", "en": "HAMi NVIDIA"}'
  key: nvidia.com/gpumem
  labelEn: vgpu memory
  labelZh: vgpu 显存
  limits: optional
  requests: disabled
  resourceUnit: "Mi"
  relatedResources: "nvidia.com/gpualloc,nvidia.com/gpucores"
  excludeResources: "nvidia.com/mps-core,nvidia.com/mps-memory,tencen
t.com/vcuda-core,tencent.com/vcuda-memory"
  runtimeClassName: ""
  ignoreNodeCheck: "true"
kind: ConfigMap
metadata:
  labels:
    features.cpaas.io/enabled: "true"
    features.cpaas.io/group: hami-nvidia
    features.cpaas.io/type: CustomResourceLimitation
  name: cf-crl-hami-nvidia-gpumem
  namespace: kube-public
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: cf-crl-hami-config
```

```
  namespace: kube-public
  labels:
    device-plugin.cpaas.io/config: "true"
data:
  deviceName: "HAMi"
  nodeLabelKey: "gpu"
  nodeLabelValue: "on"
```

HAMi Ascend vNPU

以下示例注册了 HAMi Ascend vNPU 资源字段，用于 Ascend 310P vNPU 数量和 vNPU 内存：

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: cf-crl-hami-ascend310p
  namespace: kube-public
  labels:
    features.cpaas.io/enabled: "true"
    features.cpaas.io/group: hami-ascend-vnpu
    features.cpaas.io/type: CustomResourceLimitation
data:
  dataType: integer
  defaultValue: "1"
  descriptionEn: Number of Ascend 310P vNPU devices requested
  descriptionZh: 申请的 Ascend 310P vNPU 数量
  group: hami-ascend-vnpu
  groupI18n: '{"zh":"HAMi Ascend vNPU","en":"HAMi Ascend vNPU"}'
  key: huawei.com/Ascend310P
  labelEn: Ascend 310P vNPU
  labelZh: Ascend 310P vNPU
  limits: optional
  requests: disabled
  resourceUnit: "count"
  relatedResources: "huawei.com/Ascend310P-memory"
  runtimeClassName: ""
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: cf-crl-hami-ascend310p-memory
  namespace: kube-public
  labels:
    features.cpaas.io/enabled: "true"
    features.cpaas.io/group: hami-ascend-vnpu
    features.cpaas.io/type: CustomResourceLimitation
data:
  dataType: integer
  defaultValue: "3072"
  descriptionEn: Ascend 310P vNPU memory requested per device
  descriptionZh: 每个 Ascend 310P vNPU 申请的显存
  group: hami-ascend-vnpu
  groupI18n: '{"zh":"HAMi Ascend vNPU","en":"HAMi Ascend vNPU"}'
  key: huawei.com/Ascend310P-memory
  labelEn: Ascend 310P vNPU memory
```

```
    labelZh: Ascend 310P vNPU 显存
    limits: optional
    requests: disabled
    resourceUnit: "Mi"
    relatedResources: "huawei.com/Ascend310P"
    runtimeClassName: ""
    ignoreNodeCheck: "true"
  ---
  apiVersion: v1
  kind: ConfigMap
  metadata:
    name: cf-crl-hami-ascend-vnpu-config
    namespace: kube-public
    labels:
      device-plugin.cpaas.io/config: "true"
  data:
    deviceName: "HAMi Ascend vNPU"
    nodeLabelKey: "ascend"
    nodeLabelValue: "on"
```

内存配额字段使用 `ignoreNodeCheck: "true"`，因为它与 vNPU 设备数量字段配合使用，而不是作为独立的节点可分配设备数量进行检查。

Ascend 910

以下示例注册了报告为 `huawei.com/Ascend910` 的 Ascend 910 资源，以及 Huawei NPU 节点的设备显示元数据：

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: cf-crl-huawei-ascend910
  namespace: kube-public
  labels:
    features.cpaas.io/enabled: "true"
    features.cpaas.io/group: huawei-npu
    features.cpaas.io/type: CustomResourceLimitation
data:
  dataType: integer
  defaultValue: "1"
  descriptionEn: Number of Ascend 910 NPUs requested
  descriptionZh: 申请的 Ascend 910 NPU 数量
  group: huawei-npu
  groupI18n: '{"zh":"Huawei NPU","en":"Huawei NPU"}'
  key: huawei.com/Ascend910
  labelEn: Ascend 910
  labelZh: Ascend 910
  limits: optional
  requests: disabled
  resourceUnit: "count"
  runtimeClassName: ""
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: cf-crl-huawei-npu-config
  namespace: kube-public
  labels:
    device-plugin.cpaas.io/config: "true"
data:
  deviceName: "Huawei NPU"
  nodeLabelKey: "ascend"
  nodeLabelValue: "on"
```

例如，即使 NPU Operator 在节点可分配资源中报告了 `huawei.com/Ascend910`，ACP 也只有在相应的 `CustomResourceLimitation` 元数据可用后，才能在配额页面中显示 `Ascend 910`。

```
{
  "huawei.com/Ascend910": "8"
}
```

NVIDIA GPU

以下示例注册了报告为 `nvidia.com/gpu` 的物理 NVIDIA GPU 资源：

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: cf-crl-nvidia-gpu
  namespace: kube-public
  labels:
    features.cpaas.io/enabled: "true"
    features.cpaas.io/group: nvidia-gpu
    features.cpaas.io/type: CustomResourceLimitation
data:
  dataType: integer
  defaultValue: "1"
  descriptionEn: Number of NVIDIA GPUs requested
  descriptionZh: 申请的 NVIDIA GPU 数量
  group: nvidia-gpu
  groupI18n: '{"zh":"NVIDIA GPU","en":"NVIDIA GPU"}'
  key: nvidia.com/gpu
  labelEn: NVIDIA GPU
  labelZh: NVIDIA GPU
  limits: optional
  requests: disabled
  resourceUnit: "count"
  runtimeClassName: ""
```

对于其他加速器产品，请保持相同的 ACP 元数据模型，并将资源 `key`、标签、显示名称、单位和设备信息替换为该产品暴露的值。

配置配额

使用 ACP 配额页面执行以下操作：将集群添加到项目、创建命名空间或更改命名空间配额。请参阅加速器产品文档，了解某个产品暴露了哪些资源名称以及每个单位的含义。

配额页面

- 项目集群资源分配
- 命名空间资源配额

相关页面

- GPU 加速器系列
- NPU 加速器系列

加速器监控

ACP 中的加速器监控将产品提供的指标与 ACP 可观测性功能连接起来。

目录

您可以监控的内容

准确的指标名称和监控面板取决于集群中安装的加速器产品。ACP 可观测性页面说明了如何查看监控面板、查询指标以及配置告警。

指标来源

- NVIDIA GPU 指标可能来自 DCGM Exporter 或产品特定的收集器。
- HAMi 可能提供虚拟 GPU 分配和使用指标。
- Ascend NPU 产品可能提供 NPU exporter 指标。

相关页面

- 使用 ACP 可观测性文档了解 监控概念 和 监控面板管理。
- 使用 Alauda 成本管理 文档了解 GPU 或 NPU 计费模型。
- 使用各加速器产品站点了解产品特定的指标名称和监控面板。

加速器系列

当你已经知道要使用的加速器硬件类型（例如 GPU 或 NPU），并需要选择产品路径时，从这里开始。

GPU

选择 GPU 路径

容器和 VM

相关页面

NPU

选择 NPU 路径

使用 NPU 资源

产品文档

相关页面

GPU

当你希望在 GPU 上运行容器工作负载，或在 ACP 中选择正确的 GPU 产品路径时，请使用此页面。

对于硬件启用、直接 GPU 分配、DRA、GPU Operator 或 NVIDIA 指标，请使用 NVIDIA vendor integration。对于用户任务从基于 GPU 基础的共享、虚拟化或调度行为开始的场景，请使用 HAMi。

目录

选择 GPU 路径

需求	典型路径
使用供应商标准组件的 NVIDIA GPU	NVIDIA vendor integration
在多个工作负载之间共享、虚拟化或调度 GPU 资源	HAMi sharing and scheduling path

需求	典型路径
使用基于 claim 的 GPU 分配	DRA，当目标产品交付明确支持 DRA driver 时
对虚拟机使用 GPU 直通	ACP Virtualization 虚拟机创建文档

容器和 VM

容器工作负载的 GPU 使用与虚拟机 GPU 直通是不同的路径。

- 容器工作负载的 GPU 使用由此加速器部分及相关产品站点涵盖。
- 如果 GPU 资源可在节点上分配，但在 ACP 配额页面中不可见，请在 Accelerator resource quota 中注册 accelerator quota 字段元数据。
- 虚拟机物理 GPU 直通在 ACP Virtualization documentation 中配置。

相关页面

- NVIDIA vendor integration
- HAMi sharing and scheduling path
- Device Plugin
- DRA

NPU

当您希望在 Ascend NPU 上运行容器工作负载时，请使用此页面。

目录

选择 NPU 路径

当需要进行硬件启用、driver 生命周期管理、runtime 集成、CDI、直接 NPU 分配以及厂商标准操作时，请使用 Ascend 厂商集成。当用户任务从共享、虚拟化或基于 Ascend 基础构建的调度行为开始时，请使用 HAMi。

如果某个部署使用 NPU Operator 准备 Ascend 节点，但通过 HAMi Ascend Device Plugin 暴露资源，请将 NPU 产品文档用于基础前提条件，将 HAMi 产品文档用于设备暴露、资源请求、共享语义和故障排除。

使用 NPU 资源

- 工作负载请求由所选分配产品暴露的 NPU 资源。
- 使用项目和命名空间页面进行配额操作。
- 如果某个 NPU 资源在节点上可分配，但在 ACP 配额页面中不可见，请在 Accelerator resource quota 中注册加速器配额字段元数据。
- 使用可观测性页面进行 dashboard 和告警配置。
- 使用 NPU 产品文档了解 Ascend 基础资源名称、driver 生命周期和验证工作负载。使用 HAMi 文档了解 HAMi 特有的 Ascend 资源名称或共享行为。

产品文档

使用 NPU 产品站点获取安装、driver 生命周期、兼容性、release notes 和故障排除信息。

> **Note**
>
> 因为 Alauda Ascend NPU 的发版周期与灵雀云容器平台不同，所以 Alauda Ascend NPU 的文档现在作为独立的文档站点托管在 Alauda Ascend NPU ↗。

相关页面

- Ascend 厂商集成
- HAMi 共享与调度路径
- Device Plugin
- RuntimeClass 和 CDI
- Accelerator resource quota

加速器产品

当您知道所需的加速器家族后，产品可帮助您选择要使用的 vendor integration 或 sharing and scheduling product。

Vendor integrations 会准备硬件、驱动程序、运行时、直接分配以及特定于 vendor 的监控。Sharing and scheduling products 基于一个或多个 vendor integrations 构建，以提供共享、虚拟化、调度、队列管理或资源编排。

供应商集成

NVIDIA

选择 NVIDIA 路径

相关 ACP 页面

相关概念

产品文档

Ascend

选择 Ascend 路径

相关 ACP 页面

产品文档

相关概念

共享与调度

HAMi

在以下场景使用 HAMi

HAMi 后端

相关概念

产品文档

供应商集成

供应商集成会为加速器硬件、驱动程序、运行时、直接分配以及供应商特定的监控路径做好准备。

当您希望为容器加速器工作负载选择供应商基础时，请使用本部分。有关安装、升级、兼容性、release notes 和故障排查，请继续查看产品文档。

NVIDIA

选择 NVIDIA 路径

相关 ACP 页面

相关概念

产品文档

Ascend

选择 Ascend 路径

相关 ACP 页面

产品文档

相关概念

NVIDIA

NVIDIA 厂商集成涵盖 ACP 集群中的 NVIDIA GPU 启用，包括直接 GPU 分配、基于 DRA 的 GPU 分配、GPU Operator，以及 NVIDIA GPU 指标组件（例如 DCGM-Exporter），前提是这些插件产品可通过 Application Marketplace 或已批准的交付包获取。

目录

选择 NVIDIA 路径

选项	适用场景
NVIDIA GPU Device Plugin	你需要将物理 NVIDIA GPU 资源作为 Kubernetes 扩展资源公开。
NVIDIA DRA Driver for GPUs	你需要用于 NVIDIA GPU 分配的 Kubernetes DRA resource claims。

选项	适用场景
GPU Operator	当该产品可在你的交付中获得时，你需要统一的 NVIDIA stack 部署路径，例如 driver、toolkit、device plugin 或监控组件。
DCGM-Exporter	你需要用于 ACP 监控、计费或产品仪表板的 NVIDIA GPU 指标。
HAMi on NVIDIA GPU	你需要基于 NVIDIA GPU 前置条件实现共享、虚拟化或调度行为。请为 HAMi 工作流中的 HAMi 部分使用 HAMi 共享和调度路径。

WARNING

为每个物理 NVIDIA GPU 池选择一种 GPU 分配路径。

除非该拓扑已被明确记录，并且已针对你的产品版本完成验证，否则不要让多个 GPU 分配组件管理同一组物理 GPU。

相关 **ACP** 页面

使用此页面来选择 NVIDIA 集成路径。关于安装、节点标签、运行时前置条件、兼容性和故障排除，请使用相应的产品文档。如果 HAMi 暴露了 GPU 资源，则仅将此路径用于 NVIDIA 前置条件，并继续参考 HAMi 文档了解资源请求和共享语义。

有关通用插件安装，请参见 Cluster Plugin。

相关概念

- Device Plugin
- DRA
- RuntimeClass and CDI
- Accelerator resource quota

产品文档

NVIDIA GPU 文档

> **Note**
>
> 因为 Alauda NVIDIA GPU 的发版周期与灵雀云容器平台不同，所以 Alauda NVIDIA GPU 的文档现在作为独立的文档站点托管在 Alauda NVIDIA GPU ↗。

GPU Operator

当该产品可在你的交付中获得时，请使用 GPU Operator 产品文档。

Ascend

Ascend vendor 集成涵盖 ACP 集群中的 Huawei Ascend NPU 启用能力，包括 NPU Operator、driver 生命周期、runtime 集成、CDI、直接 NPU 分配，以及在目标 release 中随之交付的 vendor-specific 监控组件。

目录

选择 Ascend 路径

NPU Operator 管理 Ascend NPU software stack，并可通过 vendor-standard path 将 NPU 资源暴露给 Kubernetes 工作负载。有关 installation、driver 生命周期、upgrade、compatibility、release notes 和 troubleshooting，请使用 NPU 产品文档。

如果部署仅使用 NPU Operator 来准备 Ascend base，并使用 HAMi Ascend Device Plugin 来暴露资源，请使用 HAMi 产品文档了解 device exposure、resource requests、sharing semantics 和 troubleshooting。

相关 **ACP** 页面

- 使用 ACP accelerator 页面了解 Ascend 在 GPU/NPU capability 导航中的位置。
- 使用 ACP quota 页面进行 project 和 namespace quota 操作。
- 使用 ACP observability 页面进行 dashboard 和告警操作。
- 使用 NPU 产品文档了解特定版本的 Ascend base 行为。
- 使用 HAMi 文档了解 Ascend NPU 上的 HAMi 或 Ascend vNPU 上的 HAMi 行为。

产品文档

有关 installation、driver 生命周期、compatibility、release notes 和 troubleshooting，请使用 NPU 产品站点。

> **Note**
>
> 因为 Alauda Ascend NPU 的发版周期与灵雀云容器平台不同，所以 Alauda Ascend NPU 的文档现在作为独立的文档站点托管在 Alauda Ascend NPU ↗。

相关概念

- Device Plugin
- RuntimeClass 和 CDI
- Accelerator resource quota
- NPU accelerator family
- HAMi sharing and scheduling path

共享与调度

共享与调度产品基于一个或多个供应商集成，提供加速器共享、虚拟化、调度、排队或资源编排行为。

当用户任务从共享或调度行为开始，而不是从基本设备启用开始时，请使用本节。

HAMi

在以下场景使用 HAMi

HAMi 后端

相关概念

产品文档

HAMi

HAMi 是一个用于加速器共享、虚拟化和调度行为的共享与调度产品路径。它建立在供应商集成之上，例如 NVIDIA GPU 或 Ascend NPU，具体取决于产品版本和后端支持。

目录

在以下场景使用 HAMi

- 多个工作负载需要共享加速器设备；
- 当目标后端支持时，用户需要虚拟 GPU、虚拟 NPU、硬分区或软分区等加速器共享模式；
- 产品文档确认支持目标硬件、运行时和 ACP 版本。

HAMi 后端

HAMi 文档涵盖组合部署中的 HAMi 部分：HAMi 安装、调度器行为、HAMi 设备插件、资源键、共享语义、监控集成和故障排查。

HAMi 后端页面应指向其所依赖的供应商集成前提条件：

- NVIDIA GPU 上的 HAMi 依赖 NVIDIA GPU 基础路径中的驱动程序、运行时、toolkit 和 NVIDIA 特定前提条件。
- Ascend NPU 或 Ascend vNPU 上的 HAMi 依赖 Ascend 基础路径中的 NPU 节点、驱动程序生命周期、运行时集成和 CDI 前提条件。

对于不使用 HAMi 共享语义的加速器分配，请使用相应的供应商集成：

- 对于不使用 HAMi 共享的 NVIDIA GPU 分配，请使用 NVIDIA 供应商集成。
- 对于不使用 HAMi 共享的 Ascend NPU 分配，请使用 Ascend 供应商集成。

相关概念

- Device Plugin
- DRA
- RuntimeClass 和 CDI
- 加速器资源配额
- GPU 加速器家族
- NPU 加速器家族

产品文档

使用 HAMi 产品站点获取安装、兼容性、release notes、故障排查以及后端特定行为相关信息。

> **Note**
>
> 因为 Alauda Build of HAMi 的发版周期与灵雀云容器平台不同，所以 Alauda Build of HAMi 的文档现在作为独立的文档站点托管在 Alauda Build of HAMi ↗。