Enabling GPU slicing on an Amazon EKS (Elastic Kubernetes Service) cluster allows you to partition a GPU into smaller segments, which can be useful for tasks that don’t need the full power of an entire GPU. This feature is particularly beneficial when using NVIDIA A100 GPUs, as it helps to optimize costs and resource utilization. Here’s a step-by-step guide on how to enable GPU slicing on an EKS cluster:

**Steps to Enable GPU Slicing on EKS**

**Step 1: Update EKS Node Group with NVIDIA GPU Operator**

To manage GPU slicing, use the NVIDIA GPU Operator. The operator automates the provisioning of GPU resources and supports enabling slicing.

1. **Add NVIDIA Helm Repository**:

helm repo add nvidia https://nvidia.github.io/gpu-operator

helm repo update

1. **Install NVIDIA GPU Operator**: Install the GPU operator in your cluster’s nvidia-gpu-operator namespace:

kubectl create namespace nvidia-gpu-operator

helm install --namespace nvidia-gpu-operator gpu-operator nvidia/gpu-operator

**Step 2: Enable GPU Slicing with NVIDIA MIG (Multi-Instance GPU)**

GPU slicing on A100 GPUs is achieved using MIG (Multi-Instance GPU), which allows partitioning a GPU into multiple instances.

1. **Configure MIG Mode**:
   * After the GPU operator is deployed, it will automatically configure MIG support on A100 GPUs if available.
   * Verify that MIG instances are enabled using the following command:

kubectl get nodes -o jsonpath='{.items[\*].status.allocatable}'

* + You should see allocatable resources for nvidia.com/mig-<profile>.

1. **Create a Custom Resource Definition (CRD) for MIG Profiles**:
   * Define the MIG profile configuration by creating a mig-config.yaml file:

apiVersion: nvidia.com/v1

kind: MigConfig

metadata:

name: a100-mig

spec:

devices:

- name: "nvidia.com/gpu"

profiles:

- "1g.5gb"

- "2g.10gb"

- "3g.20gb"

* + Apply the configuration:

kubectl apply -f mig-config.yaml

**Step 3: Test GPU Slicing on EKS**

To test the GPU slicing, you can deploy a sample workload that requests a specific MIG instance. Here’s an example pod.yaml that requests a 1g.5gb MIG slice:

apiVersion: v1

kind: Pod

metadata:

name: gpu-slice-test

spec:

containers:

- name: gpu-slice

image: nvcr.io/nvidia/k8s/cuda-sample

resources:

limits:

nvidia.com/mig-1g.5gb: 1

Apply this configuration with:

kubectl apply -f pod.yaml

**Step 4: Verify GPU Slicing**

Use kubectl describe pod gpu-slice-test to verify that the pod is using the specified MIG instance. You can also use NVIDIA utilities inside the container to confirm the GPU slice.

Following these steps should enable GPU slicing on your EKS cluster, allowing you to efficiently use your GPU resources.