

# Volcano vgpu device plugin for Kubernetes Example

## Prerequisites

1. GPU driver has been successfully installed.

```
[root@vgpu-on-volcano yum.repos.d]# nvidia-smi
Sat Sep 14 15:41:44 2024

+-----+
| NVIDIA-SMI 550.54.14                  Driver Version: 550.54.14          CUDA Version: 12.4          |
+-----+-----+
| GPU   Name           Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap |      Memory-Usage | GPU-Util  Compute M. |
|                                           MIG M.         |
+-----+-----+
|  0   Tesla T4              Off | 00000000:18:00.0 Off |             0        |
| N/A   73C    P0              33W / 70W |  0MiB / 15360MiB |      0%      Default |
|                                           N/A              |
+-----+-----+
|  1   Tesla T4              Off | 00000000:5E:00.0 Off |             0        |
| N/A   61C    P0              29W / 70W |  0MiB / 15360MiB |      0%      Default |
|                                           N/A              |
+-----+-----+
|  2   Tesla T4              Off | 00000000:AF:00.0 Off |             0        |
| N/A   71C    P0              33W / 70W |  0MiB / 15360MiB |      0%      Default |
|                                           N/A              |
+-----+-----+
|  3   Tesla T4              Off | 00000000:D8:00.0 Off |             0        |
| N/A   68C    P0              32W / 70W |  0MiB / 15360MiB |      8%      Default |
|                                           N/A              |
+-----+-----+

+-----+
| Processes:                               GPU Memory |
|  GPU   GI    CI        PID   Type   Process name                      Usage      |
|-----+-----+
| No running processes found               |
+-----+
```

2. Nvidia-container-toolkit has been installed. Make sure default-runtime is set to nvidia in /etc/docker/daemon.json(Rememeber to restart docker service after the change)

```
{
  "default-runtime": "nvidia",
  "runtimes": {
    "nvidia": {
      "path": "nvidia-container-runtime",
      "runtimeArgs": []
    }
  }
}
```

3. Kubernetes has been properly installed and is functioning normally.

```
[root@vgpu-on-volcano registry]# kubectl get node
NAME                STATUS    ROLES    AGE   VERSION
172.27.231.43      Ready    control-plane 45m   v1.24.14
[root@vgpu-on-volcano registry]# kubectl get pods -A
NAMESPACE          NAME                                                    READY   STATUS    RESTARTS   AGE
kube-system         coredns-c5dfc987b-8wb6x                               1/1     Running   0           43m
kube-system         dns-autoscaler-74ffc79f79-sqh8m                       1/1     Running   0           43m
kube-system         kube-apiserver-172.27.231.43                           1/1     Running   0           45m
kube-system         kube-controller-manager-172.27.231.43                 1/1     Running   1           45m
kube-system         kube-flannel-bpgkw                                       1/1     Running   0           18m
kube-system         kube-proxy-czgvh                                         1/1     Running   0           44m
kube-system         kube-scheduler-172.27.231.43                           1/1     Running   1           45m
kube-system         metrics-server-7bf9d8cb7b-x57wp                       1/1     Running   0           42m
kube-system         registry-k24zm                                           1/1     Running   0           43m
local-path-storage  local-path-provisioner-7695bf6475-649f6               1/1     Running   0           44m
[root@vgpu-on-volcano registry]# helm repo list
Error: no repositories to show
[root@vgpu-on-volcano registry]#
```

## Volcano Installation

1. Make sure volcano version is higher than v1.9.0
2. You can follow the volcano installation documentation: <https://volcano.sh/en/docs/v1-9-0/installation/>
  - helm repo add volcano-sh <https://volcano.sh.github.io/helm-charts>
  - helm repo update
  - helm install volcano volcano-sh/volcano --version 1.9.0 -n volcano-system --create-namespace

```
[root@vgpu-on-volcano registry]# helm repo add volcano-sh https://volcano.sh.github.io/helm-charts
"volcano-sh" has been added to your repositories
[root@vgpu-on-volcano registry]# helm repo update

Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "volcano-sh" chart repository
Update Complete. #Happy Helming!
[root@vgpu-on-volcano registry]#
[root@vgpu-on-volcano registry]# helm install volcano volcano-sh/volcano --version 1.9.0 -n volcano-system --create-namespace
NAME: volcano
LAST DEPLOYED: Sat Sep 14 17:01:21 2024
NAMESPACE: volcano-system
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
Thank you for installing volcano.

Your release is named volcano.

For more information on volcano, visit:
https://volcano.sh/
```

3. Check if all pods are in running states

```
[root@vgpu-on-volcano registry]# kubectl get pods -n volcano-system
NAME                                READY   STATUS    RESTARTS   AGE
volcano-admission-76645f6857-phzh4  1/1     Running   0           41s
volcano-admission-init-kglm5         0/1     Completed 0           66s
volcano-controllers-7665d47bcd-wsqq4 1/1     Running   0           18s
volcano-scheduler-676c458795-dmtvr   1/1     Running   0           3s
```

## Volcano-vgpu-device-plugin Installation

1. You can follow the volcano-vgpu-device-plugin installation documentation: <https://github.com/Project-HAMi/volcano-vgpu-device-plugin?tab=readme-ov-file#enabling-gpu-support-in-kubernetes>
  - kubectl edit cm -n volcano-system volcano-scheduler-configmap

```

# Please edit the object below. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving this file will be
# reopened with the relevant failures.
#
apiVersion: v1
data:
  volcano-scheduler.conf: |
    actions: "enqueue, allocate, backfill"
    tiers:
    - plugins:
      - name: priority
      - name: gang
        enablePreemptable: false
      - name: conformance
    - plugins:
      - name: overcommit
      - name: drf
        enablePreemptable: false
      - name: deviceshare
        arguments:
          deviceshare.VGPUEnable: true # enable vgpu
      - name: predicates
      - name: proportion
      - name: nodeorder
      - name: binpack
kind: ConfigMap
metadata:
  annotations:
    meta.helm.sh/release-name: volcano
    meta.helm.sh/release-namespace: volcano-system
  creationTimestamp: "2024-09-14T09:01:23Z"
  labels:
    app.kubernetes.io/managed-by: Helm
  name: volcano-scheduler-configmap
  namespace: volcano-system
  resourceVersion: "5550"
  uid: 8303d56e-bb02-497f-96f0-3c4c728d295a
~
~

```

- Save volcano-vgpu-device-plugin.yml to local

```

# Copyright (c) 2019, NVIDIA CORPORATION. All rights reserved.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing,
# software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
# implied.
# See the License for the specific language governing permissions
# and
# limitations under the License.
---
apiVersion: v1
kind: ServiceAccount
metadata:

```

```

    name: volcano-device-plugin
    namespace: kube-system
---
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: volcano-device-plugin
rules:
- apiGroups: ["" ]
  resources: ["nodes"]
  verbs: ["get", "list", "watch", "update", "patch"]
- apiGroups: ["" ]
  resources: ["nodes/status"]
  verbs: ["patch"]
- apiGroups: ["" ]
  resources: ["pods"]
  verbs: ["get", "list", "update", "patch", "watch"]
---
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: volcano-device-plugin
subjects:
- kind: ServiceAccount
  name: volcano-device-plugin
  namespace: kube-system
roleRef:
  kind: ClusterRole
  name: volcano-device-plugin
  apiGroup: rbac.authorization.k8s.io
---
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: volcano-device-plugin
  namespace: kube-system
spec:
  selector:
    matchLabels:
      name: volcano-device-plugin
  updateStrategy:
    type: RollingUpdate
  template:
    metadata:
      # This annotation is deprecated. Kept here for backward
compatibility
      # See https://kubernetes.io/docs/tasks/administer-
cluster/guaranteed-scheduling-critical-addon-pods/
      annotations:

```

```

    scheduler.alpha.kubernetes.io/critical-pod: ""
  labels:
    name: volcano-device-plugin
  spec:
    tolerations:
      # This toleration is deprecated. Kept here for backward
      compatibility
      # See https://kubernetes.io/docs/tasks/administer-
      cluster/guaranteed-scheduling-critical-addon-pods/
      - key: CriticalAddonsOnly
        operator: Exists
      - key: volcano.sh/gpu-memory
        operator: Exists
        effect: NoSchedule
      # Mark this pod as a critical add-on; when enabled, the
      critical add-on
      # scheduler reserves resources for critical add-on pods so
      that they can
      # be rescheduled after a failure.
      # See https://kubernetes.io/docs/tasks/administer-
      cluster/guaranteed-scheduling-critical-addon-pods/
      priorityClassName: "system-node-critical"
      serviceAccount: volcano-device-plugin
      containers:
        - image: docker.io/projecthami/volcano-vgpu-device-
          plugin:v1.9.4
          args: ["--device-split-count=10"]
          lifecycle:
            postStart:
              exec:
                command: ["/bin/sh", "-c", "cp -f /k8s-
          vgpu/lib/nvidia/* /usr/local/vgpu/"]
          name: volcano-device-plugin
          env:
            - name: NODE_NAME
              valueFrom:
                fieldRef:
                  fieldPath: spec.nodeName
            - name: HOOK_PATH
              value: "/usr/local/vgpu"
          securityContext:
            allowPrivilegeEscalation: false
            capabilities:
              drop: ["ALL"]
              add: ["SYS_ADMIN"]
          volumeMounts:
            - name: device-plugin
              mountPath: /var/lib/kubelet/device-plugins
            - name: lib

```

```
    mountPath: /usr/local/vgpu
  - name: hosttmp
    mountPath: /tmp
  - image: docker.io/projecthami/volcano-vgpu-device-
plugin:v1.9.4
    name: monitor
    command:
      - /bin/bash
      - -c
      - volcano-vgpu-monitor
    env:
      - name: NVIDIA_VISIBLE_DEVICES
        value: "all"
      - name: NVIDIA_MIG_MONITOR_DEVICES
        value: "all"
      - name: HOOK_PATH
        value: "/tmp/vgpu"
      - name: NODE_NAME
        valueFrom:
          fieldRef:
            fieldPath: spec.nodeName
    securityContext:
      allowPrivilegeEscalation: false
      capabilities:
        drop: [ "ALL" ]
        add: [ "SYS_ADMIN" ]
    volumeMounts:
      - name: dockers
        mountPath: /run/docker
      - name: containerds
        mountPath: /run/containerd
      - name: sysinfo
        mountPath: /sysinfo
      - name: hostvar
        mountPath: /hostvar
      - name: hosttmp
        mountPath: /tmp
  volumes:
    - hostPath:
        path: /var/lib/kubelet/device-plugins
        type: Directory
        name: device-plugin
    - hostPath:
        path: /usr/local/vgpu
        type: DirectoryOrCreate
        name: lib
    - name: hosttmp
      hostPath:
        path: /tmp
```

```

    type: DirectoryOrCreate
- name: dockers
  hostPath:
    path: /run/docker
    type: DirectoryOrCreate
- name: containerds
  hostPath:
    path: /run/containerd
    type: DirectoryOrCreate
- name: usrbin
  hostPath:
    path: /usr/bin
    type: Directory
- name: sysinfo
  hostPath:
    path: /sys
    type: Directory
- name: hostvar
  hostPath:
    path: /var
    type: Directory

```

- `kubectl create -f volcano-vgpu-device-plugin.yml`

## 2. Check if volcano-device-plugin pod in running states

```

[root@vgpu-on-volcano disk0]# kubectl get pods -n volcano-device-plugin -n kube-system
NAME                                READY   STATUS    RESTARTS   AGE
coredns-c5dfc987b-8wb6x             1/1     Running   0           96m
dns-autoscaler-74ffc79f79-sqh8m     1/1     Running   0           96m
kube-apiserver-172.27.231.43         1/1     Running   0           97m
kube-controller-manager-172.27.231.43 1/1     Running   2 (76s ago) 97m
kube-flannel-bpgkw                  1/1     Running   0           70m
kube-proxy-czgvh                    1/1     Running   0           97m
kube-scheduler-172.27.231.43        1/1     Running   2 (76s ago) 97m
metrics-server-7bf9d8cb7b-x57wp     1/1     Running   0           95m
registry-k24zm                      1/1     Running   0           95m
volcano-device-plugin-nrz9g          2/2     Running   0           29s

```

## 3. Check node status

```
Addresses:
  InternalIP: 172.27.231.43
  Hostname: 172.27.231.43
Capacity:
  cpu: 96
  ephemeral-storage: 515928320Ki
  hugepages-1Gi: 0
  hugepages-2Mi: 0
  memory: 527794028Ki
  pods: 220
  volcano.sh/vgpu-cores: 400
  volcano.sh/vgpu-memory: 61440
  volcano.sh/vgpu-number: 40
Allocatable:
  cpu: 96
  ephemeral-storage: 475479538925
  hugepages-1Gi: 0
  hugepages-2Mi: 0
  memory: 527691628Ki
  pods: 220
  volcano.sh/vgpu-cores: 400
  volcano.sh/vgpu-memory: 61440
  volcano.sh/vgpu-number: 40
```

## Running VGPU Jobs

---

### 1. Running a demo vgpu job

```
cat <<EOF | kubectl apply -f -
apiVersion: v1
kind: Pod
metadata:
  name: gpu-pod1
spec:
  schedulerName: volcano
  containers:
  - name: cuda-container
    image: nvidia/cuda:12.4.1-cudnn-devel-ubuntu22.04
    imagePullPolicy: IfNotPresent
    command: ["sleep"]
    args: ["100000"]
    resources:
      limits:
        volcano.sh/vgpu-number: 2 # requesting 2 gpu cards
        volcano.sh/vgpu-memory: 3000 # (optional)each vGPU uses 3G
        device memory
        volcano.sh/vgpu-cores: 50 # (optional)each vGPU uses 50% core
```



EOF

## 2. Check pod status

```
[root@vgpu-on-volcano workspace]# kubectl get pod -A
NAMESPACE          NAME                                                    READY   STATUS    RESTARTS   AGE
default             gpu-pod1                                               1/1     Running   0          4s
kube-system         coredns-c5dfc987b-8wb6x                              1/1     Running   0          3d23h
kube-system         dns-autoscaler-74ffc79f79-sqh8m                     1/1     Running   0          3d23h
kube-system         kube-apiserver-172.27.231.43                          1/1     Running   0          3d23h
kube-system         kube-controller-manager-172.27.231.43                1/1     Running   2 (3d21h ago) 3d23h
kube-system         kube-flannel-bpgkw                                    1/1     Running   0          3d22h
kube-system         kube-proxy-czgvh                                       1/1     Running   0          3d23h
kube-system         kube-scheduler-172.27.231.43                          1/1     Running   2 (3d21h ago) 3d23h
kube-system         metrics-server-7bf9d8cb7b-x57wp                     1/1     Running   0          3d23h
kube-system         registry-k24zm                                         1/1     Running   0          3d23h
kube-system         volcano-device-plugin-nrz9g                           2/2     Running   0          3d21h
local-path-storage  local-path-provisioner-7695bf6475-649f6              1/1     Running   1 (3d21h ago) 3d23h
volcano-system      volcano-admission-76645f6857-phzh4                   1/1     Running   0          3d22h
volcano-system      volcano-admission-init-kglm5                          0/1     Completed 0          3d22h
volcano-system      volcano-controllers-7665d47bcd-wsqq4                  1/1     Running   0          3d22h
volcano-system      volcano-scheduler-676c458795-dmtvr                   1/1     Running   0          3d22h
```

## 3. Running a single command to check if its working

```
root@gpu-pod1:/# nvidia-smi
[HAMI-core Msg(16:140627257530176:libvgpu.c:836)]: Initializing.....
Wed Sep 18 07:34:11 2024

+-----+
| NVIDIA-SMI 550.54.14                Driver Version: 550.54.14    CUDA Version: 12.4     |
+-----+
| GPU   Name                               Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf              Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                           MIG M. |
+-----+
|  0   Tesla T4                               Off | 00000000:AF:00:00 Off |             0        |
| N/A   38C    P8              9W / 70W     |  0MiB / 3000MiB |      0%    Default |
|                                           |
+-----+
|  1   Tesla T4                               Off | 00000000:D8:00:00 Off |             0        |
| N/A   41C    P8             11W / 70W     |  0MiB / 3000MiB |      0%    Default |
|                                           |
+-----+

+-----+
| Processes:                               |
|  GPU   GI    CI        PID   Type   Process name                        GPU Memory |
|          ID    ID                                   Usage   |
+-----+
| No running processes found              |
+-----+
[HAMI-core Msg(16:140627257530176:multiprocess_memory_limit.c:468)]: Calling exit handler 16
```

# Monitor

1. You can access the metrics interface of the volcano scheduler in cluster. For example: `curl -vvv volcano-scheduler-service.volcano-system:8080/metrics`

```

# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 0.00015242
go_gc_duration_seconds{quantile="0.25"} 0.000221242
go_gc_duration_seconds{quantile="0.5"} 0.000250834
go_gc_duration_seconds{quantile="0.75"} 0.000282792
go_gc_duration_seconds{quantile="1"} 0.000522629
go_gc_duration_seconds_sum 3.024513783
go_gc_duration_seconds_count 11622
# HELP go_gc_gogc_percent Heap size target percentage configured by the user, otherwise 100. This value is set by the GOGC environmen
t variable, and the runtime/debug.SetGCPercent function.
# TYPE go_gc_gogc_percent gauge
go_gc_gogc_percent 100
# HELP go_gc_gomemlimit_bytes Go runtime memory limit configured by the user, otherwise math.MaxInt64. This value is set by the GOMEM
LIMIT environment variable, and the runtime/debug.SetMemoryLimit function.
# TYPE go_gc_gomemlimit_bytes gauge
go_gc_gomemlimit_bytes 9.223372036854776e+18
# HELP go_gc_heap_allocs_by_size_bytes Distribution of heap allocations by approximate size. Bucket counts increase monotonically. No
te that this does not include tiny objects as defined by /gc/heap/tiny/allocs:objects, only tiny blocks.
# TYPE go_gc_heap_allocs_by_size_bytes histogram
go_gc_heap_allocs_by_size_bytes_bucket{le="8.999999999999998"} 7.914473e+06
go_gc_heap_allocs_by_size_bytes_bucket{le="24.999999999999996"} 2.94996679e+08
go_gc_heap_allocs_by_size_bytes_bucket{le="64.99999999999999"} 5.30667287e+08
go_gc_heap_allocs_by_size_bytes_bucket{le="144.99999999999997"} 5.79497177e+08
go_gc_heap_allocs_by_size_bytes_bucket{le="320.99999999999994"} 6.30376345e+08
go_gc_heap_allocs_by_size_bytes_bucket{le="704.9999999999999"} 6.44929911e+08
go_gc_heap_allocs_by_size_bytes_bucket{le="1536.9999999999998"} 6.48467187e+08
go_gc_heap_allocs_by_size_bytes_bucket{le="3200.9999999999995"} 6.49427558e+08
go_gc_heap_allocs_by_size_bytes_bucket{le="6528.999999999999"} 6.50365899e+08
go_gc_heap_allocs_by_size_bytes_bucket{le="13568.999999999998"} 6.51421992e+08
go_gc_heap_allocs_by_size_bytes_bucket{le="27264.999999999996"} 6.51570469e+08
go_gc_heap_allocs_by_size_bytes_bucket{le="+Inf"} 6.51604793e+08
go_gc_heap_allocs_by_size_bytes_sum 6.2970815968e+10
go_gc_heap_allocs_by_size_bytes_count 6.51604793e+08
# HELP go_gc_heap_allocs_bytes_total Cumulative sum of memory allocated to the heap by the application.
# TYPE go_gc_heap_allocs_bytes_total counter
go_gc_heap_allocs_bytes_total 6.2970815968e+10
# HELP go_gc_heap_allocs_objects_total Cumulative count of heap allocations triggered by the application. Note that this does not inc
lude tiny objects as defined by /gc/heap/tiny/allocs:objects, only tiny blocks.
# TYPE go_gc_heap_allocs_objects_total counter
go_gc_heap_allocs_objects_total 6.51604793e+08
# HELP go_gc_heap_frees_by_size_bytes Distribution of freed heap allocations by approximate size. Bucket counts increase monotonically.
Note that this does not include tiny objects as defined by /gc/heap/tiny/allocs:objects, only tiny blocks.
# TYPE go_gc_heap_frees_by_size_bytes histogram
go_gc_heap_frees_by_size_bytes_bucket{le="8.999999999999998"} 7.91245e+06
go_gc_heap_frees_by_size_bytes_bucket{le="24.999999999999996"} 2.94980586e+08
go_gc_heap_frees_by_size_bytes_bucket{le="64.99999999999999"} 5.30642353e+08
go_gc_heap_frees_by_size_bytes_bucket{le="144.99999999999997"} 5.79466542e+08
go_gc_heap_frees_by_size_bytes_bucket{le="320.99999999999994"} 6.30342973e+08
go_gc_heap_frees_by_size_bytes_bucket{le="704.9999999999999"} 6.4489438e+08
go_gc_heap_frees_by_size_bytes_bucket{le="1536.9999999999998"} 6.48431285e+08
go_gc_heap_frees_by_size_bytes_bucket{le="3200.9999999999995"} 6.4939152e+08
go_gc_heap_frees_by_size_bytes_bucket{le="6528.999999999999"} 6.50329791e+08
go_gc_heap_frees_by_size_bytes_bucket{le="13568.999999999998"} 6.5138572e+08
go_gc_heap_frees_by_size_bytes_bucket{le="27264.999999999996"} 6.5153416e+08
go_gc_heap_frees_by_size_bytes_bucket{le="+Inf"} 6.51568464e+08
go_gc_heap_frees_by_size_bytes_sum 6.2963620496e+10
go_gc_heap_frees_by_size_bytes_count 6.51568464e+08
# HELP go_gc_heap_frees_bytes_total Cumulative sum of heap memory freed by the garbage collector.
# TYPE go_gc_heap_frees_bytes_total counter
go_gc_heap_frees_bytes_total 6.2963620496e+10
# HELP go_gc_heap_frees_objects_total Cumulative count of heap allocations whose storage was freed by the garbage collector. Note tha
t this does not include tiny objects as defined by /gc/heap/tiny/allocs:objects, only tiny blocks.
# TYPE go_gc_heap_frees_objects_total counter
go_gc_heap_frees_objects_total 6.51568464e+08
# HELP go_gc_heap_goal_bytes Heap size target for the end of the GC cycle.
# TYPE go_gc_heap_goal_bytes gauge
go_gc_heap_goal_bytes 1.4019072e+07
# HELP go_gc_heap_live_bytes Heap memory occupied by live objects that were marked by the previous GC.
# TYPE go_gc_heap_live_bytes gauge
go_gc_heap_live_bytes 6.677888e+06
# HELP go_gc_heap_objects_objects Number of objects, live or unswept, occupying heap memory.
# TYPE go_gc_heap_objects_objects gauge
go_gc_heap_objects_objects 36329
# HELP go_gc_heap_tiny_allocs_objects_total Count of small allocations that are packed together into blocks. These allocations are co
unted separately from other allocations because each individual allocation is not tracked by the runtime, only their block. Each bloc
k is already accounted for in allocs-by-size and frees-by-size.
# TYPE go_gc_heap_tiny_allocs_objects_total counter
go_gc_heap_tiny_allocs_objects_total 2.2216151e+07
# HELP go_gc_limiter_last_enabled_gc_cycle GC cycle the last time the GC CPU limiter was enabled. This metric is useful for diagnosin
g the root cause of an out-of-memory error, because the limiter trades memory for CPU time when the GC's CPU time gets too high. This
is most likely to occur with use of SetMemoryLimit. The first GC cycle is cycle 1, so a value of 0 indicates that it was never enabl
ed.
# TYPE go_gc_limiter_last_enabled_gc_cycle gauge
go_gc_limiter_last_enabled_gc_cycle 0
# HELP go_gc_pauses_seconds Distribution of individual GC-related stop-the-world pause latencies. Bucket counts increase monotonically.
# TYPE go_gc_pauses_seconds histogram
go_gc_pauses_seconds_bucket{le="6.399999999999999e-08"} 0
go_gc_pauses_seconds_bucket{le="6.399999999999999e-07"} 0
go_gc_pauses_seconds_bucket{le="7.167999999999999e-06"} 0
go_gc_pauses_seconds_bucket{le="8.191999999999999e-05"} 2256
go_gc_pauses_seconds_bucket{le="0.0009175039999999999"} 23233
go_gc_pauses_seconds_bucket{le="0.010485759999999998"} 23244

```

2. You can also change the Volcano service from ClusterIP mode to NodePort mode, which will allow external access to the metrics interface.

```
apiVersion: v1
kind: Service
metadata:
  annotations:
    meta.helm.sh/release-name: volcano
    meta.helm.sh/release-namespace: volcano-system
    prometheus.io/path: /metrics
    prometheus.io/port: "8080"
    prometheus.io/scrape: "true"
  creationTimestamp: "2024-09-14T09:01:23Z"
  labels:
    app: volcano-scheduler
    app.kubernetes.io/managed-by: Helm
  name: volcano-scheduler-service
  namespace: volcano-system
  resourceVersion: "5580"
  uid: 402ee6a3-5d0f-4e9e-8f7f-879920e54c49
spec:
  clusterIP: 10.42.187.209
  clusterIPs:
    - 10.42.187.209
  internalTrafficPolicy: Cluster
  ipFamilies:
    - IPv4
  ipFamilyPolicy: SingleStack
  ports:
    - name: metrics
      port: 8080
      protocol: TCP
      targetPort: 8080
  selector:
    app: volcano-scheduler
  sessionAffinity: None
  type: NodePort
status:
  loadBalancer: {}
```

```
[root@vgpu-on-volcano worspace]# kubectl get svc -A
```

NAMESPACE	NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
default	kubernetes	ClusterIP	10.42.0.1	<none>	443/TCP	3d23h
kube-system	coredns	ClusterIP	10.42.0.3	<none>	53/UDP,53/TCP,9153/TCP	3d23h
kube-system	metrics-server	ClusterIP	10.42.45.187	<none>	443/TCP	3d23h
kube-system	registry	NodePort	10.42.61.224	<none>	5000:35000/TCP	3d23h
volcano-system	volcano-admission-service	ClusterIP	10.42.238.92	<none>	443/TCP	3d22h
volcano-system	volcano-scheduler-service	NodePort	10.42.187.209	<none>	8080:33410 TCP	3d22h

← → ↻ 🔒 不安全 172.27.231.43:33410/metrics ☆ 无痕模式 (已打开 2 个窗口) 有新版 Chrome 可用

```
# HELP apiserver_audit_event_total [ALPHA] Counter of audit events generated and sent to the audit backend.
# TYPE apiserver_audit_event_total counter
apiserver_audit_event_total 0
# HELP apiserver_audit_requests_rejected_total [ALPHA] Counter of apiserver requests rejected due to an error in audit logging backend.
# TYPE apiserver_audit_requests_rejected_total counter
apiserver_audit_requests_rejected_total 0
# HELP cardinality_enforcement_unexpected_categorizations_total [ALPHA] The count of unexpected categorizations during cardinality enforcement.
# TYPE cardinality_enforcement_unexpected_categorizations_total counter
cardinality_enforcement_unexpected_categorizations_total 0
# HELP disabled_metrics_total [BETA] The count of disabled metrics.
# TYPE disabled_metrics_total counter
disabled_metrics_total 0
# HELP go_cgo_go_to_c_calls_calls_total Count of calls made from Go to C by the current process.
# TYPE go_cgo_go_to_c_calls_calls_total counter
go_cgo_go_to_c_calls_calls_total 0
# HELP go_cpu_classes_gc_mark_assist_cpu_seconds_total Estimated total CPU time goroutines spent performing GC tasks to assist the GC and prevent it from falling behind the application. This metric is an overestimate, and not directly comparable to system CPU time measurements. Compare only with other /cpu/classes metrics.
# TYPE go_cpu_classes_gc_mark_assist_cpu_seconds_total counter
go_cpu_classes_gc_mark_assist_cpu_seconds_total 4.184921835
# HELP go_cpu_classes_gc_mark_dedicated_cpu_seconds_total Estimated total CPU time spent performing GC tasks on processors (as defined by GOMAXPROCS) dedicated to those tasks. This metric is an overestimate, and not directly comparable to system CPU time measurements. Compare only with other /cpu/classes metrics.
# TYPE go_cpu_classes_gc_mark_dedicated_cpu_seconds_total counter
go_cpu_classes_gc_mark_dedicated_cpu_seconds_total 414.062459679
# HELP go_cpu_classes_gc_mark_idle_cpu_seconds_total Estimated total CPU time spent performing GC tasks on spare CPU resources that the Go scheduler could not otherwise find a use for. This should be subtracted from the total GC CPU time to obtain a measure of compulsory GC CPU time. This metric is an overestimate, and not directly comparable to system CPU time measurements. Compare only with other /cpu/classes metrics.
# TYPE go_cpu_classes_gc_mark_idle_cpu_seconds_total counter
go_cpu_classes_gc_mark_idle_cpu_seconds_total 27.086009616
# HELP go_cpu_classes_gc_pause_cpu_seconds_total Estimated total CPU time spent with the application paused by the GC. Even if only one thread is running during the pause, this is computed as GOMAXPROCS times the pause latency because nothing else can be executing. This is the exact sum of samples in /gc/pause:seconds if each sample is multiplied by GOMAXPROCS at the time it is taken. This metric is an overestimate, and not directly comparable to system CPU time measurements. Compare only with other /cpu/classes metrics.
# TYPE go_cpu_classes_gc_pause_cpu_seconds_total counter
go_cpu_classes_gc_pause_cpu_seconds_total 290.641986912
# HELP go_cpu_classes_gc_total_cpu_seconds_total Estimated total CPU time spent performing GC tasks. This metric is an overestimate, and not directly comparable to system CPU time measurements. Compare only with other /cpu/classes metrics. Sum of all metrics in /cpu/classes/gc.
# TYPE go_cpu_classes_gc_total_cpu_seconds_total counter
go_cpu_classes_gc_total_cpu_seconds_total 735.975377242
# HELP go_cpu_classes_idle_cpu_seconds_total Estimated total available CPU time not spent executing any Go or Go runtime code. In other words, the part of /cpu/classes/total:cpu-seconds that was unused. This metric is an overestimate, and not directly comparable to system CPU time measurements. Compare only with other /cpu/classes metrics.
# TYPE go_cpu_classes_idle_cpu_seconds_total counter
go_cpu_classes_idle_cpu_seconds_total 3.269560798900441e+07
# HELP go_cpu_classes_scavenge_assist_cpu_seconds_total Estimated total CPU time spent returning unused memory to the underlying platform in response eagerly in response to memory pressure. This metric is an overestimate, and not directly comparable to system CPU time measurements. Compare only with other /cpu/classes metrics.
# TYPE go_cpu_classes_scavenge_assist_cpu_seconds_total counter
go_cpu_classes_scavenge_assist_cpu_seconds_total 8.35e-07
# HELP go_cpu_classes_scavenge_background_cpu_seconds_total Estimated total CPU time spent performing background tasks to return unused memory to the underlying platform. This metric is an overestimate, and not directly comparable to system CPU time measurements. Compare only with other /cpu/classes metrics.
# TYPE go_cpu_classes_scavenge_background_cpu_seconds_total counter
```