Contents

NVIDIA GPU Operator Setup with Grafana and Prometheus

Overview

Currently, two Kubernetes clusters are deployed: `p1` running Prometheus and `c2` running NVIDIA GPU Operator. Long-term, these may be consolidated into a single cluster for simplicity.

Cluster Configuration

Two Kubernetes clusters are running:

- **P1 cluster:** Runs Grafana and Prometheus in the `monitoring` namespace
- **P2 cluster:** Has Grafana-Prometheus in the `monitoring` namespace

The P1 cluster runs Grafana/Prometheus with the main prometheus-grafana service set up as NodePort on port 31600. Launch a web browser to `p1-worker-vm:31600`; the default login is `admin`/`prom-operator`.

P1 Cluster Details

```
wsl=> k config current-context
p1-admin@p1.grafana

wsl=> k get node
NAME                                STATUS   ROLES           AGE   VERSION
p1-master-vm.hst.enablement.local   Ready    control-plane   79d   v1.33.1
p1-worker-vm.hst.enablement.local   Ready    <none>          79d   v1.33.1
hjma@HSTHJMA02:~


wsl=> k get svc -n monitoring
NAME                                      TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)
                          AGE
alertmanager-operated                     ClusterIP   None             <none>        9093/
    TCP,9094/TCP,9094/UDP   54d
prometheus-operated                       ClusterIP   None             <none>        9090/
    TCP                     54d
test-prometheus-grafana                   NodePort    10.97.92.135     <none>
    80:31600/TCP                 54d
test-prometheus-kube-prome-alertmanager   ClusterIP   10.100.1.114     <none>        9093/
    TCP,8080/TCP            54d
test-prometheus-kube-prome-operator       ClusterIP   10.100.184.136   <none>        443/TCP
                          54d
test-prometheus-kube-prome-prometheus     ClusterIP   10.107.215.146   <none>        9090/
    TCP,8080/TCP            54d
test-prometheus-kube-state-metrics        ClusterIP   10.101.113.18    <none>        8080/
    TCP                     54d
test-prometheus-prometheus-node-exporter  ClusterIP   10.100.132.103   <none>        9100/
    TCP                     54d

wsl=> curl 10.16.160.42:31600
```

```
<a href="/login">Found</a>.
```

Prometheus Helm Configuration

In the P1 cluster, Grafana/Prometheus was set up using Helm:

```
wsl=> helm list -A --filter 'prometheus'
NAME            NAMESPACE       REVISION        UPDATED
    STATUS    CHART                           APP VERSION
test-prometheus monitoring      3               2025-07-22 17:11:52.125453 -0700 MST
    deployed  kube-prometheus-stack-75.4.0     v0.83.0
hjma@HSTHJMA02:~

wsl=> helm -n monitoring get values test-prometheus --revision 3
USER-SUPPLIED VALUES:
prometheus:
  prometheusSpec:
    additionalScrapeConfigs:
    - job_name: c2-dcgm-exporter
      static_configs:
      - targets:
        - 10.16.160.54:30639
        - 10.16.160.54:30639
hjma@HSTHJMA02:~
wsl=> helm -n monitoring get values test-prometheus --revision 2
USER-SUPPLIED VALUES:
prometheus:
  prometheusSpec:
    additionalScrapeConfigs:
    - job_name: c2-dcgm-exporter
      static_configs:
      - targets:
        - 10.16.160.54:31065
        - 10.16.160.55:31065
hjma@HSTHJMA02:~
wsl=> helm -n monitoring get values test-prometheus --revision 1
USER-SUPPLIED VALUES:
null
hjma@HSTHJMA02:~
```

C2 Cluster Configuration

The C2 cluster has the GPU Operator in the `gpu-operator` namespace.

DCGM Exporter Overview

The NVIDIA GPU Operator Helm chart deploys a DCGM (Data Center GPU Manager) exporter by default, but there are important nuances:

- The DCGM exporter Pod will be created automatically when the operator detects a node with an NVIDIA GPU and the dcgm-exporter component is enabled in its values.

- In the stock gpu-operator Helm chart from NVIDIA's repo, the DCGM exporter is enabled by default (`dcgmExporter.enabled: true`).

However:

1. ServiceMonitor is not enabled by default.

- This means Prometheus won't automatically scrape the DCGM exporter unless you either:
  - Enable the ServiceMonitor (`dcgmExporter.serviceMonitor.enabled: true`), or
  - Manually define a scrape config in Prometheus.

2. No GPUs → No exporter pods

- If the GPU Operator is installed into a cluster without GPU-capable nodes, the DaemonSet for dcgm-exporter may not schedule any pods.

ServiceMonitor Overview

A ServiceMonitor is not a built-in Kubernetes object like a Pod, Deployment, or Service. It's a Custom Resource Definition (CRD) that comes from the Prometheus Operator (or kube-prometheus-stack Helm chart).

How it works:

- You first deploy Prometheus Operator (usually via kube-prometheus-stack Helm chart).
- The Prometheus Operator introduces new CRDs, including:
  - ServiceMonitor
  - PodMonitor
  - PrometheusRule
- Prometheus Operator watches for ServiceMonitor objects and dynamically updates Prometheus' scrape configuration to match them.

For NVIDIA GPU Operator:

- If set to true, it will create a ServiceMonitor for the DCGM exporter so Prometheus can scrape GPU metrics automatically.
- If you don't have Prometheus Operator installed, creating a ServiceMonitor will do nothing—because only the Prometheus Operator knows how to use it.

C2 Cluster Details

The DCGM exporter typically uses port `9400`.

```
wsl=> k config use-context c2-admin@c2.gpu
Switched to context "c2-admin@c2.gpu".
hjma@HSTHJMA02:~


wsl=> k get svc -n gpu-operator
NAME                   TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)    AGE
gpu-operator           ClusterIP   10.233.44.80   <none>        8080/TCP   23d
nvidia-dcgm-exporter   ClusterIP   10.233.15.59   <none>        9400/TCP   23d
hjma@HSTHJMA02:


wsl=> helm list -A
NAME                     NAMESPACE       REVISION        UPDATED
                                       STATUS          CHART                   APP VERSION
gpu-operator-1753140595 gpu-operator    1               2025-07-21 16:30:01.929686717 -0700
    MST deployed        gpu-operator-v25.3.0    v25.3.0
wsl=> helm show values nvidia/gpu-operator | grep -A20 dcgmExporter:
dcgmExporter:
  enabled: true
  repository: nvcr.io/nvidia/k8s
  image: dcgm-exporter
  version: 4.2.3-4.1.3-ubuntu22.04
  imagePullPolicy: IfNotPresent
  env:
    - name: DCGM_EXPORTER_LISTEN
      value: ":9400"
    - name: DCGM_EXPORTER_KUBERNETES
      value: "true"
    - name: DCGM_EXPORTER_COLLECTORS
      value: "/etc/dcgm-exporter/dcp-metrics-included.csv"
  resources: {}
  service:
    internalTrafficPolicy: Cluster
  serviceMonitor:
    enabled: false
    interval: 15s
    honorLabels: false
    additionalLabels: {}
```

Making C2 DCGM Exporter Service Available to P1 Prometheus

We can change the `dcgm-exporter` service from the default `ClusterIP` to `NodePort`. If we patch Kubernetes using `kubectl`, the change will not be persistent, as Helm values remain at their defaults. The next time you upgrade the gpu-operator Helm chart, the service will revert to ClusterIP. We need to modify Helm values to make the change persistent.

> [!NOTE] After hours of investigation, Helm does not have a value to lock the NodePort port number. It can only define the type as NodePort and the `internalTrafficPolicy` field. Attempting to define NodePort 39400 to lock the port (preventing changes after restart or chart

> upgrade) resulted in “field not supported” errors. This limitation led to the decision to install Prometheus on the C2 cluster to avoid this complexity.

> [!TIP] This is a useful tip or suggestion.

> [!IMPORTANT] This is important information that users should not miss.

> [!WARNING] This is a warning about a potential issue or a caution.

!!! note “Custom Title” Content here

??? note “Collapsible Note” Click to expand

!!! warning “Watch Out!” Important warning

> Note:
> this is a note to see how github render compare with MKDocs. this line has two trailing spaces.
> test second line
> this is 3rd line
> disable hl2 try disable code hl by using yaml listing-options