09_CVD: Deploy NVIDIA GPU Operator on Red Hat OpenShift

The high-level steps for deploying GPUs on Red Hat OpenShift Container Platform (OCP) are:

- Step 1: Deploy Node Feature Discovery (NFD) Operator
- Step 2: Deploy NVIDIA GPU Operator from Red Hat OperatorHub
- Step 3: Create Kubernetes Secret for NVIDIA GPU Cloud (NGC) API Key
- Step 4: Create Kubernetes ConfigMap for NVIDIA License Server (NLS) token
- Step 5: Create the Cluster Policy Instance

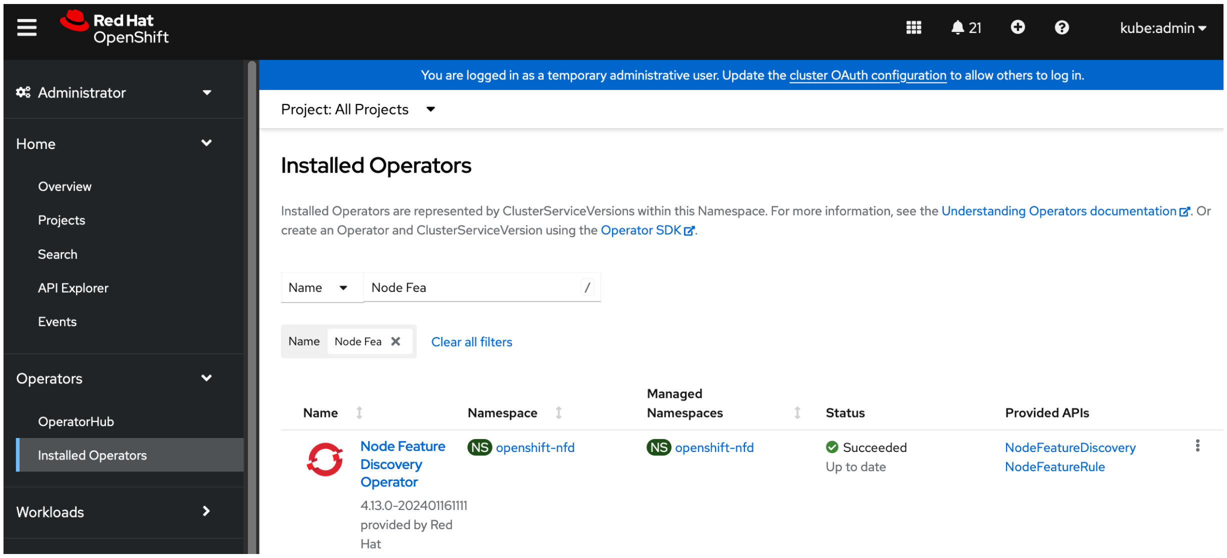
Deploy Node Feature Discovery Operator on Red Hat OpenShift

To deploy NVIDIA's GPU Operator in Red Hat OpenShift, Red Hat's Node Feature Discovery (NFD) Operator must first be deployed. NFD is Kubernetes add-on capability, deployed as an Operator that discover hardware-level features and expose them for use. For nodes with NVIDIA GPUs, NFD will label the worker nodes indicating that a NVIDIA GPU (vGPU in this case) is installed on that node.

For more information, see Red Hat OpenShift documentation for Node Feature Discovery Operator.

Deployment Steps:

1. Use a browser to navigate and login to Red Hat OpenShift's cluster console.
2. From the left navigation menu, navigate to **Operators > Operator Hub** and search for Red Hat's **Node Feature Discovery**.
3. Select **Node Feature Discovery** and click **Install**.
4. Select defaults (**A specific namespace on the cluster)** and click **Install**. This operator will be deployed in the `openshift-nfd` namespace
5. To verify that the NFD Operator installed successfully, navigate to **Operators > Installed Operators** and confirm that it has a **status** of **Succeeded**.



You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.

Project: All Projects

Installed Operators

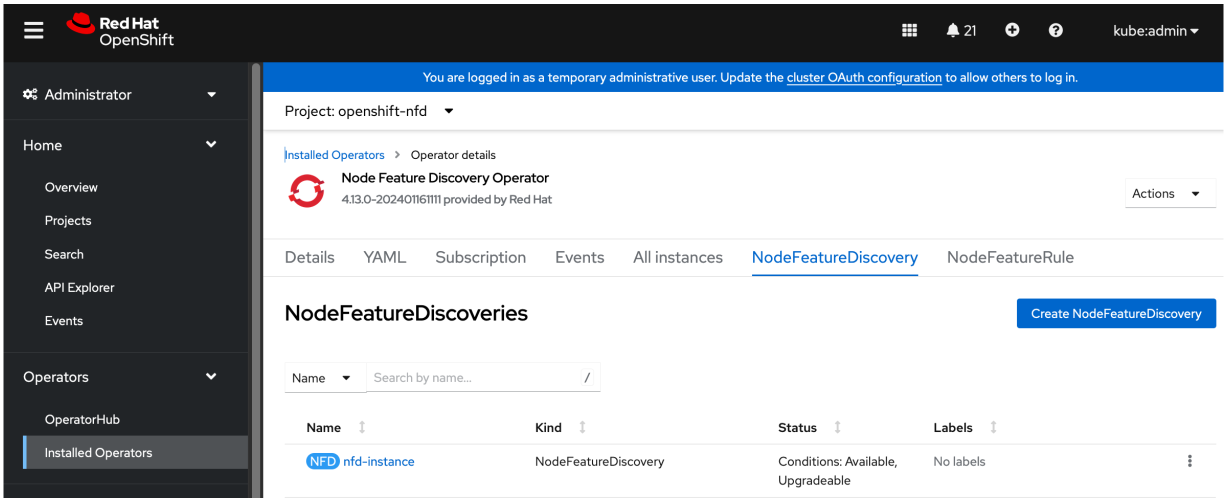
Installed Operators are represented by ClusterServiceVersions within this Namespace. For more information, see the Understanding Operators documentation. Or create an Operator and ClusterServiceVersion using the Operator SDK.

Name: Node Fea

Name Node Fea ✕ Clear all filters

Name	Namespace	Managed Namespaces	Status	Provided APIs
Node Feature Discovery Operator 4.13.0-202401161111 provided by Red Hat	NS openshift-nfd	NS openshift-nfd	Succeeded Up to date	NodeFeatureDiscovery NodeFeatureRule

6. Click on the **Node Feature Discovery Operator** and then click on **Create instance** under **Node Feature Discovery.**
7. Navigate to the **NodeFeatureDiscovery** tab and confirm that the status is: **Available, Upgradeable**



You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.

Project: openshift-nfd

Installed Operators > Operator details

Node Feature Discovery Operator
4.13.0-202401161111 provided by Red Hat

NodeFeatureDiscoveries

Name: Search by name...

Name	Kind	Status	Labels
NFD nfd-instance	NodeFeatureDiscovery	Conditions: Available, Upgradeable	No labels

8. To confirm that NFD labelled the worker nodes with NVIDIA GPUs correctly, navigate to **Compute > Nodes** and a select a worker node with GPU.
9. Go to **Details** and verify that the worker node has the label: `feature.node.kubernetes.io/pci-10de.present`
10. You can also use the following CLI commands to verify (from OCP installer workstation)

```
[administrator@FSV-AI-OCP-Installer etc]$ !1001
oc get nodes -l feature.node.kubernetes.io/pci-10de.present
NAME STATUS ROLES AGE VERSION
ocp3-qg7j6-worker-0-j7ps2 Ready worker 5d4h v1.26.9+52589e6
```

```
ocp3-qg7j6-worker-0-pt9ng Ready worker 125m v1.26.9+52589e6
ocp3-qg7j6-worker-0-w62vj Ready worker 167m v1.26.9+52589e
```

Deploy NVIDIA GPU Operator on Red Hat OpenShift

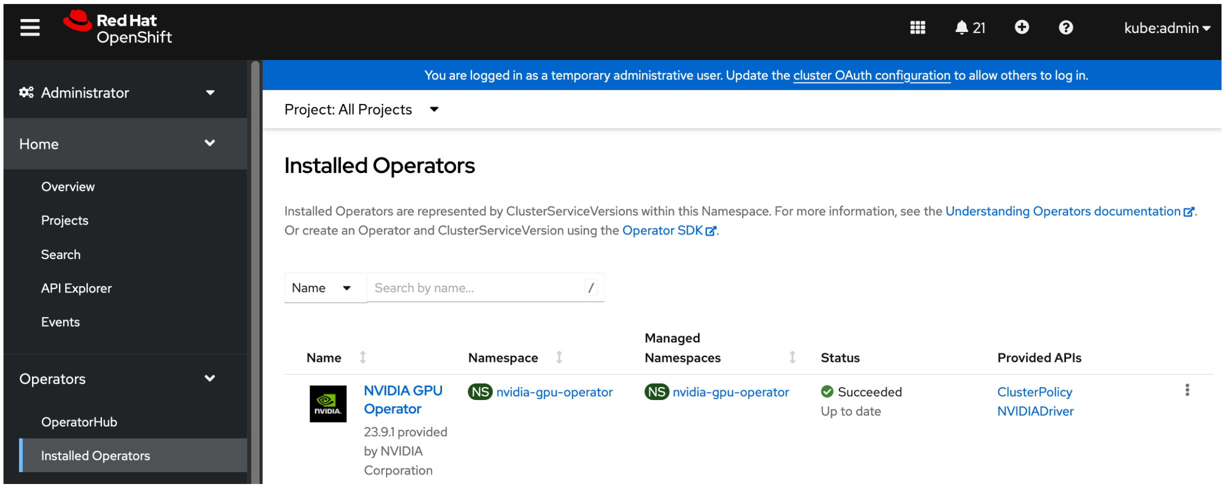
This section covers the following high-level steps for deploying NVIDIA GPUs on a Red Hat OCP cluster.

- Deploy GPU Operator from Red Hat Operator Hub
- Create Kubernetes **ConfigMap** for NVIDIA Licensing System (NLS) token. The licenses were setup using DLS and NVIDIA Licensing Portal in an earlier step. See NVIDIA License System User Guide for instructions.
- Create Kubernetes NVIDIA GPU Cloud (NGC) **Secret** specifying the access details for the driver image in the private or public container registry. This assumes that the driver is already positioned in the private repo (for e.g. quay.io). You can also use the Enterprise registry for vGPU GuestOS drivers on NVIDIA GPU Cloud. This requires special access. When you login into NGC, if you see **CATALOG > Enterprise Catalog** in the left navigation menu, then that you have the right access. You can see the vGPU/guest drivers available and other details of the container image.
- Deploy Cluster Policy specifying the Guest OS driver names, version, private/public repository to use and licensing details (**ConfigMap**) and access information (**Secret**).

For more information, see NVIDIA documentation for deploying GPU Operator.

Deploy GPU Operator from Red Hat OperatorHub

1. Use a browser to navigate and login to Red Hat OpenShift's cluster console.
2. From the left navigation menu, navigate to **Operators > Operator Hub** and search for **NVIDIA GPU Operator**.
3. Select **NVIDIA GPU Operator** and click **Install**.
4. Select defaults for namespace (**nvidia-gpu-operator)** and click **Install**.
5. To verify that the operator installed successfully, navigate to **Operators > Installed Operators** and confirm that it has a **status** of **Succeeded**.



You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.

Project: All Projects

Installed Operators

Installed Operators are represented by ClusterServiceVersions within this Namespace. For more information, see the Understanding Operators documentation. Or create an Operator and ClusterServiceVersion using the Operator SDK.

Name	Namespace	Managed Namespaces	Status	Provided APIs
NVIDIA GPU Operator 23.9.1 provided by NVIDIA Corporation	NS nvidia-gpu-operator	NS nvidia-gpu-operator	Succeeded Up to date	ClusterPolicy NVIDIADriver

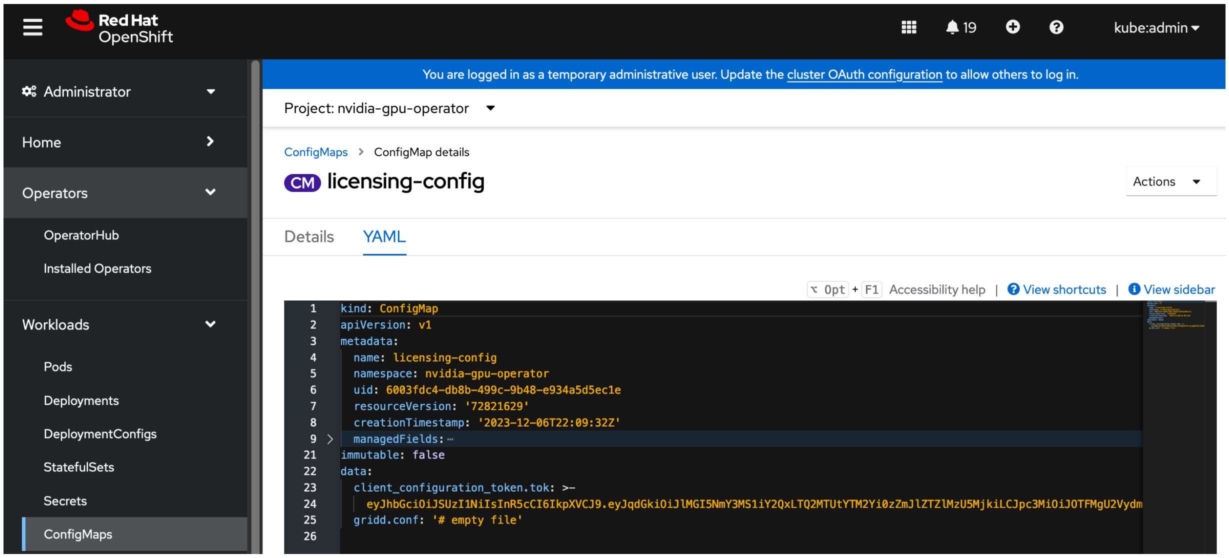
Create Kubernetes `ConfigMap` for NVIDIA Licensing System (NLS) Token

To generate and download a NLS client license token, complete the following steps.

1. Navigate to **Home** > **Projects** and ensure the `nvidia-gpu-operator` is selected.
2. Select the **Workloads** Drop Down menu.
3. Select **ConfigMaps**.
4. Click **Create ConfigMap**.
5. Enter the details for your **ConfigMap**.
   - The `name` must be `licensing-config`.
   - Copy and paste the information for your NLS client token into the `client_configuration_token.tok` parameter.

```
kind: ConfigMap
apiVersion: v1
metadata:
name: licensing-config
namespace: nvidia-gpu-operator
data:
client_configuration_token.tok: >-
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.ey <REMOVED>
gridd.conf: '# empty file'
```

6. Click **Create**. The YAML file created is shown below.



Red Hat OpenShift
19
kube:admin
Administrator
Home
Operators
OperatorHub
Installed Operators
Workloads
Pods
Deployments
DeploymentConfigs
StatefulSets
Secrets
ConfigMaps
You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.
Project: nvidia-gpu-operator
ConfigMaps > ConfigMap details
CM licensing-config
Actions
Details
YAML
⌥ Opt + F1 Accessibility help | View shortcuts | View sidebar

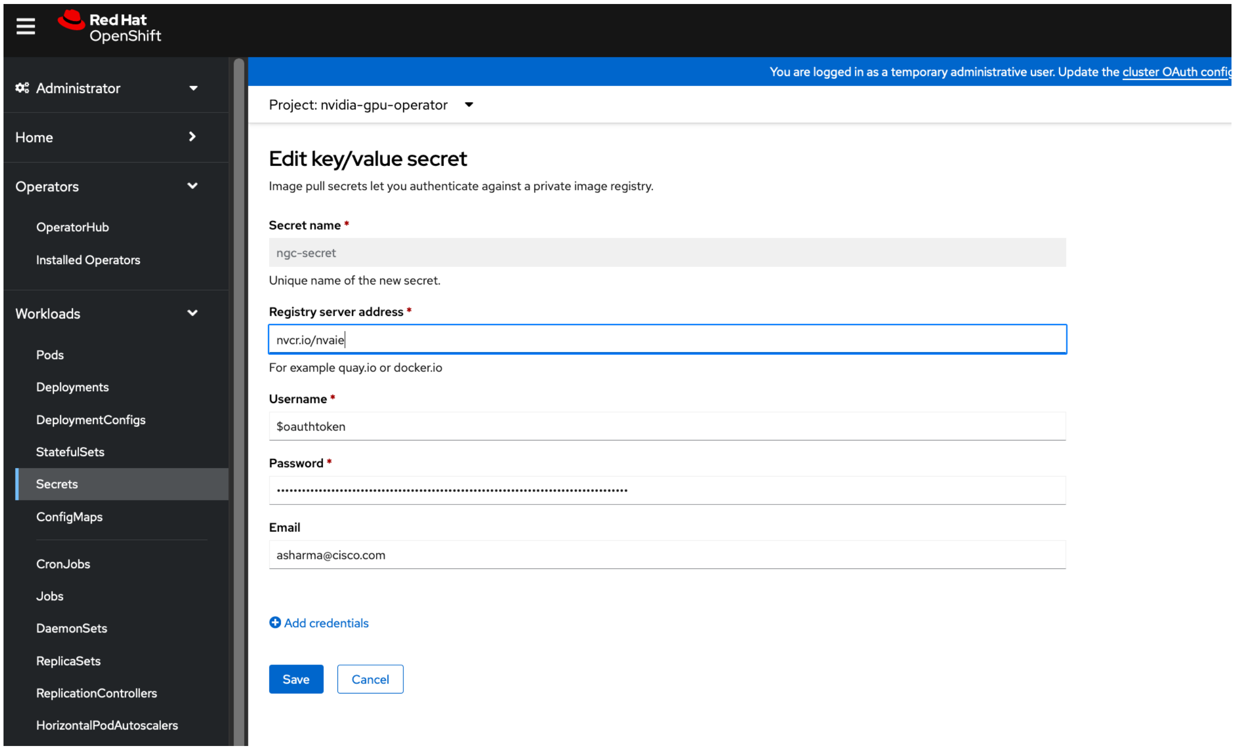
```yaml
kind: ConfigMap
apiVersion: v1
metadata:
  name: licensing-config
  namespace: nvidia-gpu-operator
  uid: 6003fdc4-db8b-499c-9b48-e934a5d5ec1e
  resourceVersion: '72821629'
  creationTimestamp: '2023-12-06T22:09:32Z'
  managedFields: …
immutable: false
data:
  client_configuration_token.tok: >-
    eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiJlMGI5NmY3MS1iY2QxLTQ2MTUtYTM2Yi0zZmJlZTZlMzU5MjkiLCJpc3MiOiJOTFMgU2Vydm
  gridd.conf: '# empty file'
```

Create NGC Secret in namespace `nvidia-gpu-operator`

OpenShift has a secret object type which provides a mechanism for holding sensitive information such as passwords and private source repository credentials. Next you will create a secret object for storing our NGC API key (the mechanism used to authenticate your access to the NGC container registry).

Note: This is only required when using NVIDIA GPU Cloud registry for images and not if it is a private image

1. Click **Secrets** from the **Workloads** drop down.
2. From the top right side of the window, click on **Create** to see the options in the drop-down list.
3. Select **Image Pull Secret**.
4. Enter the following into each field. See instructions for creating API Key here.
    - Secret name: **ngc-secret**
    - Authentication type: Image registry credentials
    - Registry server address: nvcr.io/nvaie
    - Username: $oauthtoken
    - Password: <API-KEY>
    - Email: <YOUR-EMAIL>



You are logged in as a temporary administrative user. Update the cluster OAuth confi

Project: nvidia-gpu-operator

Edit key/value secret

Image pull secrets let you authenticate against a private image registry.

Secret name *

ngc-secret

Unique name of the new secret.

Registry server address *

nvcr.io/nvaie

For example quay.io or docker.io

Username *

$oauthtoken

Password *

••

Email

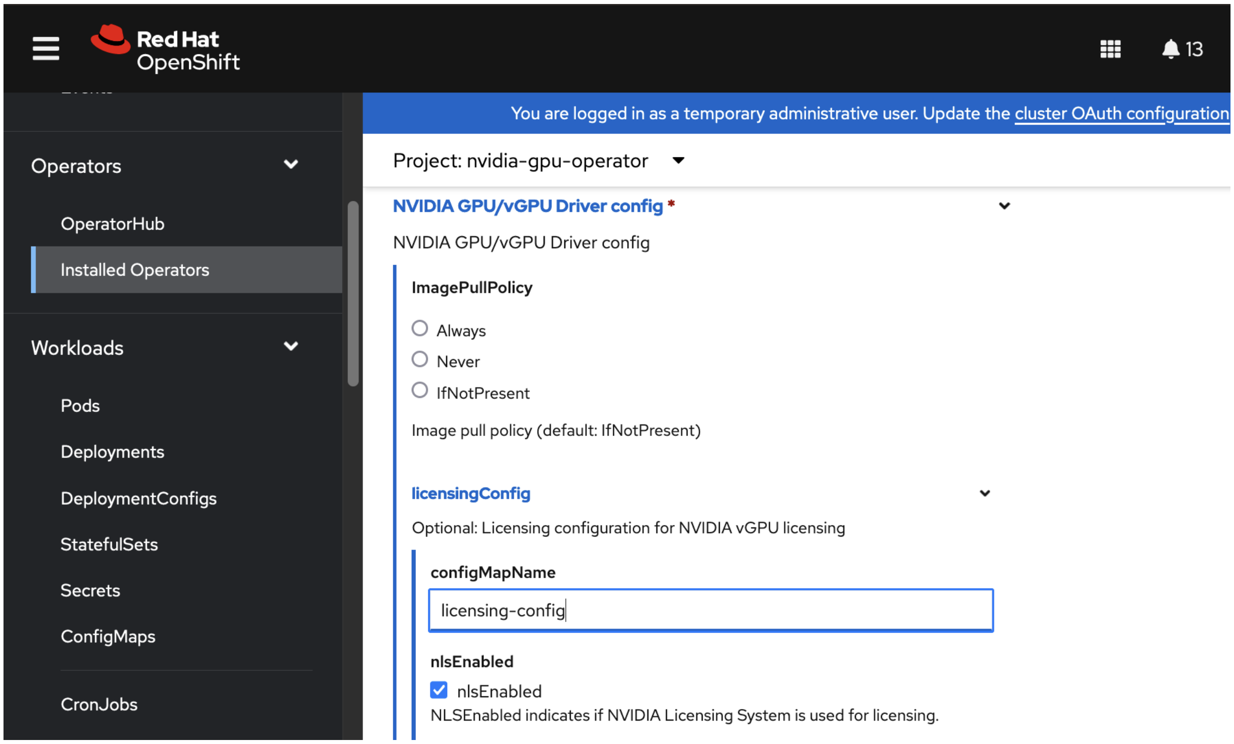
asharma@cisco.com

5. Click **Create.**

Create Cluster Policy for GPU Operator

1. To create a **Cluster Policy** instance, navigate to **Operators > Installed Operators** and click on **NVIDIA GPU Operator**.
2. Select the **ClusterPolicy** tab, then click on **Create ClusterPolicy**. The platform assigns the default name of **gpu-cluster-policy**.
3. Customize the cluster policy to reflect your deployment. In this solution, the following parameters were used as a custom guestOS driver image was created in Red Hat quay.iofor this effort. Alternatively, you can use NVIDIA's NGC repository for the driver directly as well. Provide the following information in **Driver** section.
    - Licensing `ConfigMap` that was setup earlier (**Driver > licensingConfig: configMapname: <**ConfigMap>)
    - Guest OS/vGPU Driver Image Repository Path (**Driver> repository:**)
    - Guest OS/vGPU Driver Image name (**Driver> image: <**image_name**>**)
    - Guest OS/vGPU Driver Image version (**Driver> version:: <**image_version**>**)
    - Guest OS/vGPU Driver Image pull secret - use `Secret` that was setup earlier (expand **Advanced)**

Parameter	**Name**	**Description**
ConfigMap	`configMapName: licensing-config`	NVAIE Licensing using DLS
Driver Repository	`repository: quay.io/aiml/nvaie`	
Image Version	`version: 1.0.0`	**1.0.0-rhcos4.13** (name in quay.io)
Image Name	`image: driver`	
Secret	`ngc-secret`	Image Pull Secret (not necessary for private build)



You are logged in as a temporary administrative user. Update the cluster OAuth configuration

Project: nvidia-gpu-operator

NVIDIA GPU/vGPU Driver config *

NVIDIA GPU/vGPU Driver config

ImagePullPolicy

○ Always
○ Never
○ IfNotPresent

Image pull policy (default: IfNotPresent)

licensingConfig

Optional: Licensing configuration for NVIDIA vGPU licensing

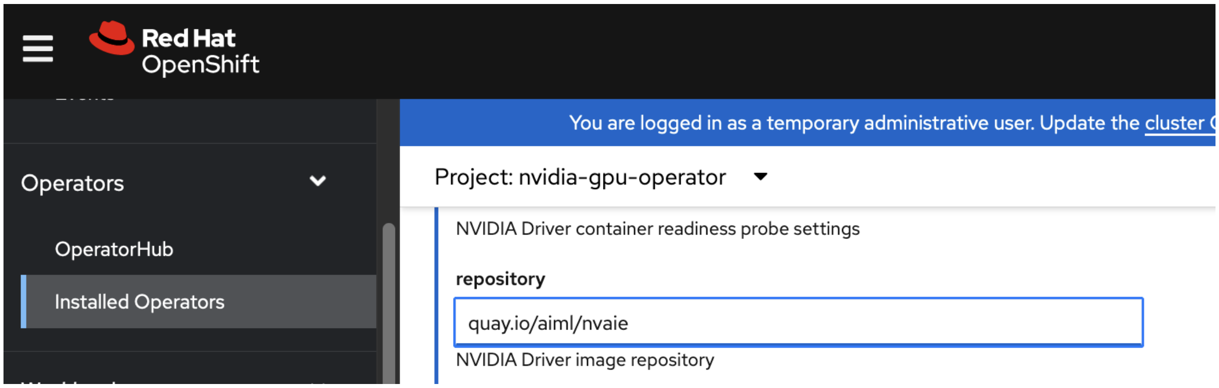
configMapName

licensing-config

nlsEnabled

☑ nlsEnabled

NLSEnabled indicates if NVIDIA Licensing System is used for licensing.



You are logged in as a temporary administrative user. Update the cluster (

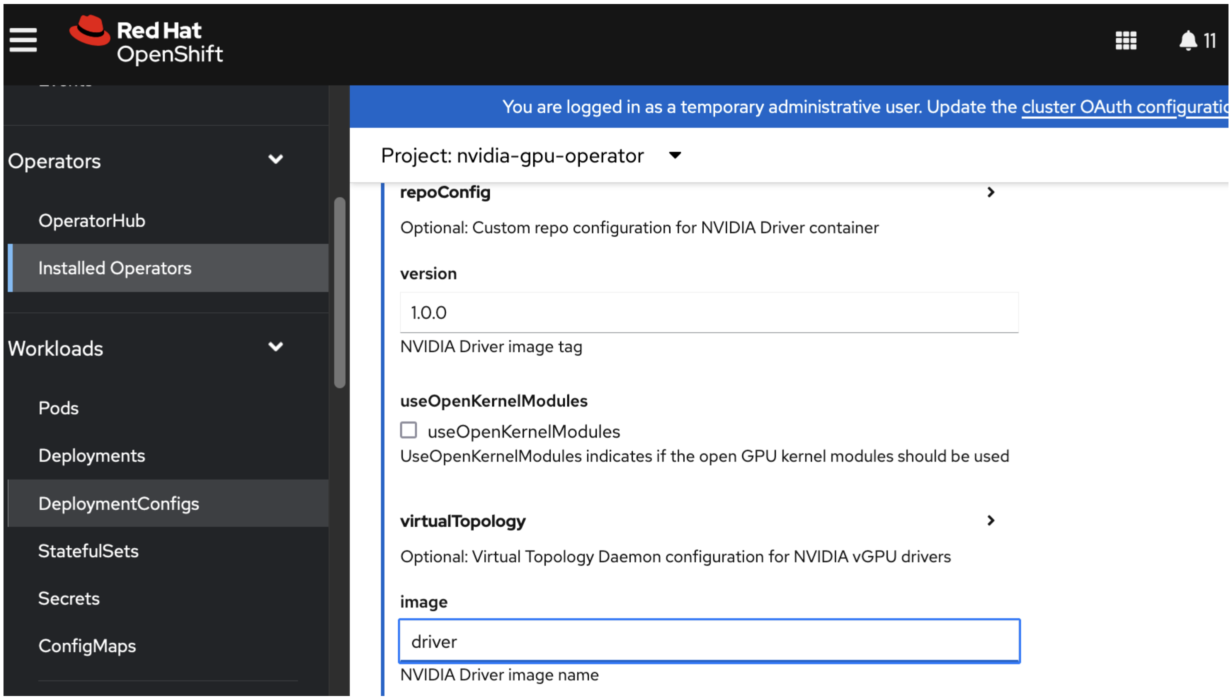
Project: nvidia-gpu-operator

NVIDIA Driver container readiness probe settings

repository

quay.io/aiml/nvaie

NVIDIA Driver image repository



You are logged in as a temporary administrative user. Update the cluster OAuth configuratio

Project: nvidia-gpu-operator

repoConfig

Optional: Custom repo configuration for NVIDIA Driver container

version

1.0.0

NVIDIA Driver image tag

useOpenKernelModules

☐ useOpenKernelModules

UseOpenKernelModules indicates if the open GPU kernel modules should be used

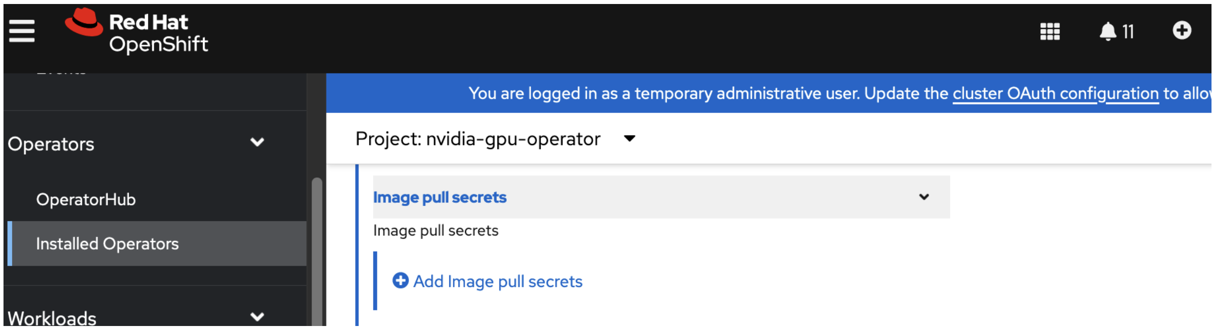
virtualTopology

Optional: Virtual Topology Daemon configuration for NVIDIA vGPU drivers

image

driver

NVIDIA Driver image name



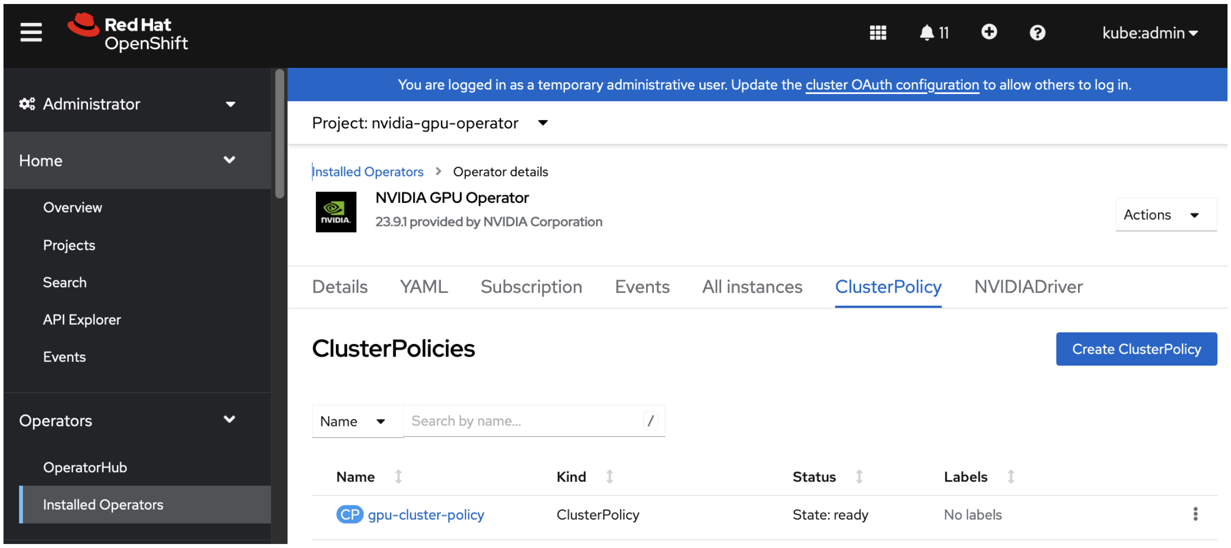
You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allo

Project: nvidia-gpu-operator

Image pull secrets

Image pull secrets

Add Image pull secrets



You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.

Project: nvidia-gpu-operator

Installed Operators > Operator details

NVIDIA GPU Operator

23.9.1 provided by NVIDIA Corporation

Actions

Details | YAML | Subscription | Events | All instances | ClusterPolicy | NVIDIADriver

ClusterPolicies

Create ClusterPolicy

Name | Search by name... /

Name	Kind	Status	Labels
CP gpu-cluster-policy	ClusterPolicy	State: ready	No labels

4. Once the installation completes successfully, the status of the cluster policy will change to `State:ready`

Verify that the GPU (vGPU in this case) has the correct software

Verify that the GPU (vGPU in this case) is licensed

- Execute the following command from the OCP installer/client workstation to verify Licensing:

```
[administrator@FSV-AI-OCP-Installer OCP3]$ oc exec -it nvidia-driver-daemonset-
413.92.202309261804-0-bp8fg -- nvidia-smi -q

==============NVSMI LOG==============

Timestamp : Fri Feb 2 06:41:53 2024
Driver Version : 525.60.13
CUDA Version : 12.0

Attached GPUs : 1
GPU 00000000:02:00.0
Product Name : GRID A100D-40C
Product Brand : NVIDIA Virtual Compute Server
Product Architecture : Ampere
Display Mode : Enabled
Display Active : Disabled
Persistence Mode : Enabled
MIG Mode
Current : Disabled
Pending : Disabled
Accounting Mode : Disabled
Accounting Mode Buffer Size : 4000
Driver Model
Current : N/A
Pending : N/A
Serial Number : N/A
GPU UUID : GPU-0eb17efc-1dd3-11b2-8411-dd8b953c6ec4
Minor Number : 0
VBIOS Version : 00.00.00.00.00
MultiGPU Board : No
Board ID : 0x200
Board Part Number : N/A
GPU Part Number : 20B5-893-A1
Module ID : N/A
Inforom Version
Image Version : N/A
OEM Object : N/A
ECC Object : N/A
Power Management Object : N/A
GPU Operation Mode
```

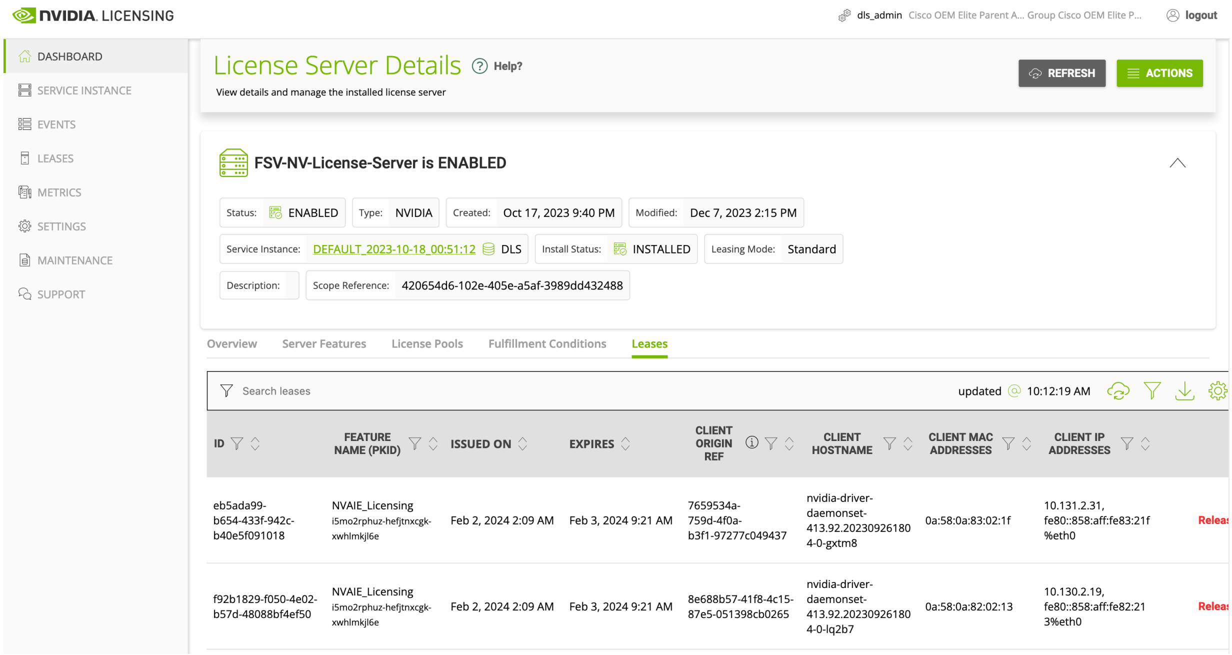
```
Current : N/A
Pending : N/A
GSP Firmware Version : N/A
GPU Virtualization Mode
Virtualization Mode : VGPU
Host VGPU Mode : N/A
vGPU Software Licensed Product
Product Name : NVIDIA Virtual Compute Server
License Status : Licensed (Expiry: 2024-2-3 5:23:19 GMT)
IBMNPU
Relaxed Ordering Mode : N/A
PCI
Bus : 0x02
Device : 0x00
Domain : 0x0000
Device Id : 0x20B510DE
Bus Id : 00000000:02:00.0
Sub System Id : 0x159B10DE
GPU Link Info
PCIe Generation
Max : N/A
Current : N/A
Device Current : N/A
Device Max : N/A
Host Max : N/A
Link Width
Max : N/A
Current : N/A
Bridge Chip
Type : N/A
Firmware : N/A
Replays Since Reset : N/A
Replay Number Rollovers : N/A
Tx Throughput : N/A
Rx Throughput : N/A
Atomic Caps Inbound : N/A
Atomic Caps Outbound : N/A
Fan Speed : N/A
Performance State : P0
Clocks Throttle Reasons : N/A
FB Memory Usage
Total : 40960 MiB
Reserved : 3092 MiB
Used : 0 MiB
Free : 37867 MiB
BAR1 Memory Usage
Total : 4096 MiB
```

```
Used : 0 MiB
Free : 4096 MiB
Compute Mode : Default
Utilization
Gpu : 0 %
Memory : 0 %
Encoder : 0 %
Decoder : 0 %
Encoder Stats
Active Sessions : 0
Average FPS : 0
Average Latency : 0
FBC Stats
Active Sessions : 0
Average FPS : 0
Average Latency : 0
Ecc Mode
Current : Enabled
Pending : Enabled
ECC Errors
Volatile
SRAM Correctable : 0
SRAM Uncorrectable : 0
DRAM Correctable : 0
DRAM Uncorrectable : 0
Aggregate
SRAM Correctable : 0
SRAM Uncorrectable : 0
DRAM Correctable : 0
DRAM Uncorrectable : 0
Retired Pages
Single Bit ECC : N/A
Double Bit ECC : N/A
Pending Page Blacklist : N/A
Remapped Rows : N/A
Temperature
GPU Current Temp : N/A
GPU Shutdown Temp : N/A
GPU Slowdown Temp : N/A
GPU Max Operating Temp : N/A
GPU Target Temperature : N/A
Memory Current Temp : N/A
Memory Max Operating Temp : N/A
Power Readings
Power Management : N/A
Power Draw : N/A
Power Limit : N/A
```

```
Default Power Limit : N/A
Enforced Power Limit : N/A
Min Power Limit : N/A
Max Power Limit : N/A
Clocks
Graphics : 210 MHz
SM : 210 MHz
Memory : 1512 MHz
Video : 795 MHz
Applications Clocks
Graphics : N/A
Memory : N/A
Default Applications Clocks
Graphics : N/A
Memory : N/A
Deferred Clocks
Memory : N/A
Max Clocks
Graphics : N/A
SM : N/A
Memory : N/A
Video : N/A
Max Customer Boost Clocks
Graphics : N/A
Clock Policy
Auto Boost : N/A
Auto Boost Default : N/A
Voltage
Graphics : N/A
Fabric
State : N/A
Status : N/A
Processes : None

[administrator@FSV-AI-OCP-Installer OCP3]$
```

- Navigate to the DLS instance deployed on-prem for licensing and verify that you have have active leases as shown below.



License Server Details

View details and manage the installed license server

FSV-NV-License-Server is ENABLED

Status: ENABLED | Type: NVIDIA | Created: Oct 17, 2023 9:40 PM | Modified: Dec 7, 2023 2:15 PM

Service Instance: DEFAULT_2023-10-18_00:51:12 DLS | Install Status: INSTALLED | Leasing Mode: Standard

Description: | Scope Reference: 420654d6-102e-405e-a5af-3989dd432488

updated @ 10:12:19 AM

ID	FEATURE NAME (PKID)	ISSUED ON	EXPIRES	CLIENT ORIGIN REF	CLIENT HOSTNAME	CLIENT MAC ADDRESSES	CLIENT IP ADDRESSES	
eb5ada99-b654-433f-942c-b40e5f091018	NVAIE_Licensing i5mo2rphuz-hefjtnxcgk-xwhlmkjl6e	Feb 2, 2024 2:09 AM	Feb 3, 2024 9:21 AM	7659534a-759d-4f0a-b3f1-97277c049437	nvidia-driver-daemonset-413.92.202309261804-0-gxtm8	0a:58:0a:83:02:1f	10.131.2.31, fe80::858:aff:fe83:21f%eth0	Releas
f92b1829-f050-4e02-b57d-48088bf4ef50	NVAIE_Licensing i5mo2rphuz-hefjtnxcgk-xwhlmkjl6e	Feb 2, 2024 2:09 AM	Feb 3, 2024 9:21 AM	8e688b57-41f8-4c15-87e5-051398cb0265	nvidia-driver-daemonset-413.92.202309261804-0-lq2b7	0a:58:0a:82:02:13	10.130.2.19, fe80::858:aff:fe82:213%eth0	Releas

Enable NVIDIA GPU (vGPU) DCGM Monitoring on Red Hat OpenShift

Using https://docs.nvidia.com/datacenter/cloud-native/openshift/latest/enable-gpu-monitoring-dashboard.html, enable to GPU Monitoring Dashboard to monitor vGPUs in the Openshift Web-Console. Note that GPU hardware parameters such as Temperature and Power are not available with vGPUs.

Enable NVIDIA GPU (physical GPU) DCGM Monitoring on VMware vCenter

Deployment Steps:

1. In VMware vCenter, select an ESXi host that has GPU(s) installed and select the Monitor tab in the center pane. Select **Performance > Advanced**.
2. Click **Chart Options**. In the list on the left under Chart Metrics, select **GPU**. Select up to two counters (Temperature and Utilization recommended) and make sure all Target Objects are selected.
3. Click **SAVE OPTIONS AS**. Enter GPU for the Chart options name and click **OK**.
4. Click **OK** to view the chart. This chart is now available on all ESXi hosts with GPUs.