



Red Hat Blog

Menu ▾

Entitlement-Free Deployment of the NVIDIA GPU Operator on OpenShift

December 14, 2021 | Kevin Pouget | 8-minute read

[Hybrid cloud](#)[Partners](#)[Containers](#)[< Back to all posts](#)

The Performance and Latency Sensitive Applications (PSAP) team at Red Hat works on the enablement and optimization of Red Hat OpenShift to run compute-intensive enterprise workloads HPC and AI/ML applications effectively and efficiently. As a team of Linux and performance enthusiasts who are always pushing the limits of what's possible with the latest and greatest upstream technologies, we've been collaborating with NVIDIA on the GPU Operator on various enhancements to the operator for enterprise use cases.

Version 1.9.0 of the GPU Operator has just landed in OpenShift OperatorHub, with many different updates. We're proud to announce that this version comes with the support of the entitlement-free deployment of NVIDIA GPU Driver.

Thanks to the OpenShift Driver Toolkit, a container image built with all the packages required to build kernel drivers, we've been able to remove the need for accessing the Red Hat RHEL repository servers to deploy the NVIDIA GPU Operator. The dependency on a valid RHEL

subscription certificate, plus the need to access packages from Red Hat servers, were major burdens for the NVIDIA GPU Operator users.

In this new release, the operator now relies on an OpenShift core image to build the GPU driver. The removal of the access to the package servers also simplifies the accelerator-enablement in disconnected environments (proxy, disconnected, and air-gapped), as customers will not have to configure the access to the package server mirrors anymore.

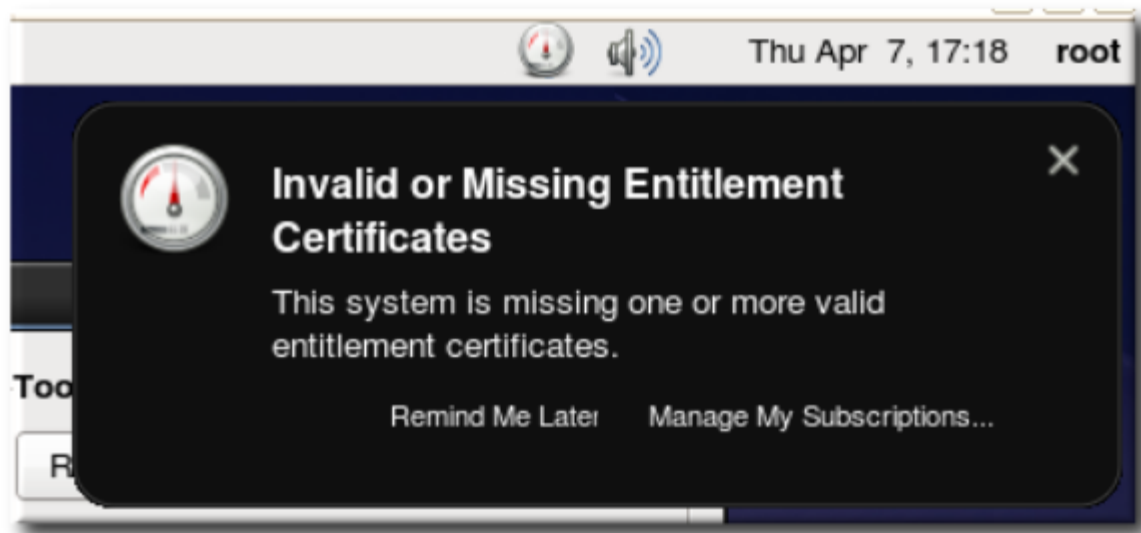
The past: RHEL entitlement required to build drivers

Building and loading the NVIDIA GPU Driver is the cornerstone of the GPU Operator work: Without this base layer, all the other elements of the GPU computing stack are moot. So the GPU Operator first deploys a DaemonSet targeting all the nodes with GPU PCI cards and, in turn, the DaemonSet deploys a privileged Pod running with NVIDIA's driver container image.

This driver container image was designed to be independent of the OS version running in the node. Indeed, each minor version of OpenShift runs a slightly different version of RHCOS and Linux kernel, so it's not possible to ship the image with fully built, kernel-specific, driver binary files. To work around this, the driver container image only contains the generic installer binary, plus a set of scripts required to setup the container, build the driver, and load it in the kernel memory.

The complications come within this first container setup phase: To be able to build the driver, the setup script must install the RHCOS kernel packages (kernel-header, kernel-core and kernel-devel, plus the version of GCC used to compile the driver, and the elfutils-libelf-devel package). The issue is that all these packages are behind Red Hat RHEL entitlement, meaning that a customer subscription certificate key is required to download them from the Red Hat RPM Repositories.

This step is well-known to create issues, as the entitlement key simply might not be deployed in the cluster, or a MachineConfigOperator failure occurred during the deployment, or the certificate may be invalid or expired ... to list only a few issues.



So, even if efforts have been made to simplify the handling of entitlement keys in an OpenShift cluster, it was becoming crucial to find a way to lift this requirement for an entitlement key.

The future: Entitlement-free driver builds with OpenShift Driver Toolkit image

The OpenShift Driver Toolkit container image project, started earlier this year, aims at easing hardware enablement in OpenShift clusters. This container image, now a built-in part of the latest OpenShift releases, contains all the packages required to build and load out-of-tree Linux kernel modules. It's already being used as part of the Special Resource Operator.

The Driver Toolkit container image provides all the necessary bits for lifting the NVIDIA GPU Operator dependency on Red Hat RHEL package repositories, including the need for the subscription certificate.

By design, the image is specific to a given RHCOS OS (and kernel) version. The following command shows how to look up the image address for a specific OpenShift version:

```
$ oc adm release info 4.8.0 --image-for=driver-toolkit  
quay.io/openshift-release-dev/ocp-v4.0-art-dev@sha256:e0f9b9154538af082596f60a  
f99290b5a751e61fd61100912defb71b6cac15c6
```

However, this command is not convenient to run inside the GPU Operator controller, as we cannot always know the exact OpenShift version running in a given node (in particular, during a cluster upgrade). The only reliable information we have about the system running on a node is its RHCOS version, so we had to design a new way to match the nodes' RHCOS version and the driver toolkit image.

OpenShift Driver Toolkit imagestream

The first part of the solution involves giving access to the Driver Toolkit image matching the version of the operating system running on the nodes. The easiest way for the operators is to generate the image address from a property of the cluster, such as the node's RHCOS version.

So the Driver Toolkit project has been extended to expose an imagestream (that is, a cluster-internal image repository), with tags dynamically generated when OpenShift is built. The tags specify the RHCOS version against which the image was built, and the latest tag references the Driver Toolkit image of the current version of the cluster. During cluster upgrades, the imagestream is extended with new RHCOS-version tags, and the latest tag is overwritten:

OpenShift Node Feature Discovery

The second part of the solution involves being able to define a DaemonSet targeting all the nodes running the same RHCOS version. The DaemonSet controller relies on the node labels to determine which nodes it should run Pods on. Until now, the GPU Operator was deploying the compute stack "on all the nodes with NVIDIA GPU cards."

But for the Driver Toolkit, this is too broad, as different GPU nodes may, under some circumstances, run different RHCOS version (for example, in the middle of an upgrade). This means we can no longer use a single DaemonSet to cover all GPU nodes, as each of them will require the DaemonSet to specify a different Driver Toolkit image.

Labeling nodes with hardware and system properties is the task of OpenShift Node Feature Discovery (NFD) Operator, and the GPU Operator already relies on it to discover nodes with

NVIDIA PCI cards. The NFD exposes various labels related to the system (OpenShift, RHEL, and Linux kernel versions), but nothing was available to discover the exact RHCOS version, so we extended it.

The exact RHCOS version is exposed to the system in the `/etc/os-release` file, with the `OSTREE_VERSION` key. This value identifies a specific build of RHCOS, including the full set of userland packages installed, plus the kernel. This is a perfect fit for matching with the Driver Toolkit imagestream tag. After extending NFD, the RHCOS version is now exposed as part of the node labels:

Integrating the Driver Toolkit as part of the GPU Operator

The last part of the entitlement-free GPU Operator rework was to actually integrate it in the GPU Operator. This was done in two main steps: (1) Integrate the Driver Toolkit as a side-car container of the main driver container and (2) support multiple RHCOS versions running side by side.

Integrate the Driver Toolkit as a side-car container

The NVIDIA driver container image, shipped by NVIDIA, contains the driver installer binary; some scripts to set up the system, trigger the build, and load the driver; and a set of pre-installed NVIDIA packages providing core services (such as `nvidia-persistenced` or `nv-fabricmanager`, required for NVIDIA DGX systems). Because of these pre-installed services, it was not possible to simply replace the NVIDIA driver container image by the OpenShift Driver Toolkit image. Instead, the Driver Toolkit image gets configured as a side-car container: Both containers run side by side inside the Pod, and they exchange the driver installer and binary files via a shared directory.

Internally, the NVIDIA driver container copies the files required to build the driver to the shared

directory, and the OpenShift Driver Toolkit container picks them up from there, builds the driver binary files, and copies them back to the shared directory. And eventually, the NVIDIA driver container loads the driver binaries in the kernel memory and starts NVIDIA core services. Once all of these steps are completed, both containers sleep forever, as their job is completed.

Multi-DaemonSet upgrade support

During cluster upgrade, the worker nodes may have different RHCOS versions for a while. That means that for a seamless cluster upgrade support, the operator driver DaemonSet must either work well with any RHCOS version, or spawn multiple DaemonSets, one per RHCOS version. The GPU Operator previously followed the former option, as the NVIDIA driver container image is independent of the underlying system version. However, this is no longer the case with the OpenShift Driver Toolkit container image.

Thus, we had to extend the GPU Operator controller and make the driver DaemonSet RHCOS version-specific. This is done by enumerating the RHCOS version used by GPU nodes (from their NFD tags), creating one DaemonSet per RHCOS version, and specifying the corresponding Driver Toolkit image tag in the Pod template.

At each iteration of the GPU Operator reconciliation loop, the controller checks if new RHCOS versions appeared in the cluster (at the beginning of an upgrade), or if existing RHCOS-specific DaemonSets are no longer in use (at the end of an upgrade). If a new version appears, a new DaemonSet is created, and if a version is no longer in use, the DaemonSet is deleted. Between these two steps, the cluster nodes will be rebooted and upgraded by the Machine Config Operator, and they will slip from one DaemonSet to the other.

Summary and future work

In this blog post, we presented the new design of the GPU Operator driver DaemonSet on OpenShift, which now supports entitlement-free deployment of the NVIDIA GPU Driver, including seamless cluster upgrade. This support is enabled by default when deploying the GPU Operator v1.9 from OperatorHub. See the release notes for further details about the OpenShift versions supported.

We also presented how we extended OpenShift Driver Toolkit and the Node Feature Discovery to implement this feature, so that other hardware-enablement operators can reuse this design to build and deploy their drivers on OpenShift nodes.

Finally, a follow-up step of this work would be to use images with pre-compiled driver binaries (built specifically for each version of OpenShift/RHCOS). This would reuse the seamless upgrade and RHCOS-specific DaemonSet deployment code but include a pre-compiled driver

instead of building them in the OpenShift Driver Toolkit container. That would result in a shorter deployment time, but at the cost of maintaining one driver imager per RHCOS version. In this context, the OpenShift Driver Toolkit container could be used as a fallback, if the pre-compiled driver image is not available.

We believe that Linux containers and container orchestration engines, most notably Kubernetes, are well positioned to power future software applications spanning multiple industries. Red Hat has embarked on a mission to enable some of the most critical workloads like machine learning, deep learning, artificial intelligence, big data analytics, high-performance computing, and telecommunications, on Red Hat OpenShift. By developing the necessary performance and latency-sensitive application platform features of OpenShift, the PSAP team is supporting this mission across multiple footprints (public, private, and hybrid cloud), industries and application types.

About the author



More like this