



Docker Containers

Table of contents

Docker Using SR-IOV	3
Kubernetes Using SR-IOV	4
Kubernetes with Shared HCA	5

On Linux, Docker uses resource isolation of the Linux kernel, to allow independent "containers" to run within a single Linux kernel instance.

Docker containers are supported on MLNX_OFED using Docker runtime. Virtual RoCE and InfiniBand devices are supported using SR-IOV mode.

Currently, RDMA/RoCE devices are supported in the modes listed in the following table.

Linux Containers Networking Modes

Orchestration and Clustering Tool	Version	Networking Mode	Link Layer	Virtualization Mode
Docker	Docker Engine 17.03 or higher	SR-IOV using sriov-plugin along with docker run wrapper tool	InfiniBand and Ethernet	SR-IOV
Kubernetes	Kubernetes 1.10.3 or higher	SR-IOV using device plugin, and using SR-IOV CNI plugin	InfiniBand and Ethernet	SR-IOV
		VXLAN using IPoIB bridge	InfiniBand	Shared HCA

Docker Using SR-IOV

In this mode, Docker engine is used to run containers along with SR-IOV networking plugin. To isolate the virtual devices, `docker_rdma_sriov` tool should be used. This mode is applicable to both InfiniBand and Ethernet link layers.

To obtain the plugin, visit: hub.docker.com/r/rdma/sriov-plugin

To install the `docker_rdma_sriov` tool, use the container tools installer available via hub.docker.com/r/rdma/container_tools_installer

For instructions on how to use Docker with SR-IOV, refer to [Docker RDMA SRIOV Networking with ConnectX4/ConnectX5/ConnectX6](#) Community post.

Kubernetes Using SR-IOV

In order to use RDMA in Kubernetes environment with SR-IOV networking mode, two main components are required:

1. RDMA device plugin - this plugin allows for exposing RDMA devices in a Pod
2. SR-IOV CNI plugin - this plugin provisions VF net device in a Pod

When used in SR-IOV mode, this plugin enables SR-IOV and performs necessary configuration including setting GUID, MAC, privilege mode, and Trust mode.

The plugin also allocates the VF devices when Pods are scheduled and requested by Kubernetes framework.

Kubernetes with Shared HCA

One RDMA device (HCA) can be shared among multiple Pods running in a Kubernetes worker nodes. User defined networks are created using VXLAN or VETH networking devices. RDMA device (HCA) can be shared among multiple Pods running in a Kubernetes worker nodes.

Copyright 2024. PDF Generated on 08/15/2024