



Tailor Made Network Operating System

Running Docker Containers on Top of Mellanox's SN2000 Spectrum™-based Ethernet Switches

Modern data-centers have evolved from monolithic architectures running applications on a single host or virtual machine into lightweight Linux-based containers.

CONTAINERS AND DOCKER

A container image is a lightweight, stand-alone, executable package of a piece of software that includes everything needed to run it: code, runtime, system tools, system libraries, and settings. Available for both Linux and Windows based applications, containerized software will always run the same, regardless of the environment. Containers isolate software from its surroundings, for example differences between development and staging environments, and help reduce conflicts between teams running different software on the same infrastructure.

Docker is the world's leading open source software container platform. Docker provides an additional layer of abstraction and automation of operating-system-level virtualization on Windows and Linux. Docker allows independent "containers" to run within a single Linux instance, avoiding the overhead of starting and maintaining virtual machines.

Mellanox's Spectrum™-based 25GbE and 100GbE switches offer end-users the ability to create their own unique differentiation since they are only solution in the market to enable tailored applications by **direct access to Spectrum SDK** on top of Docker containers.

SPECTRUM ADDED VALUE

Container support in Mellanox Onyx™ (Mellanox operating system) or Cumulus Linux offers customers the flexibility to load container-based applications directly onto the Mellanox Spectrum platforms. Running Docker containers over Spectrum simplifies the development and deployment of applications on switches. It also allows deployment of any of the thousands applications available from Docker Hub.

SN2000 platforms are the only solution in the market to enable direct access to system resources via the Mellanox open SDK API:

- Tailor your own application in an isolated container on top of Spectrum SDK
- Fast integration and validations cycles for new features on top of Spectrum SDK

BENEFITS

- Enable direct access to Spectrum SDK API
- Provide DevOps-users the flexibility to build customized unique solutions
- Facilitate time-to-market

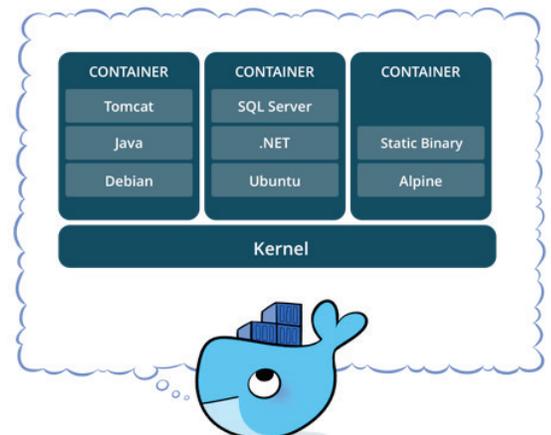


Figure 1. Docker open source software container platform

- Enjoy all of the Spectrum switch capabilities and its features set even if not yet supported by a NOS
- Persistent – up after reload with the same parameters and configuration
- Share selected storage spaces between various containers and the Onyx operating system itself

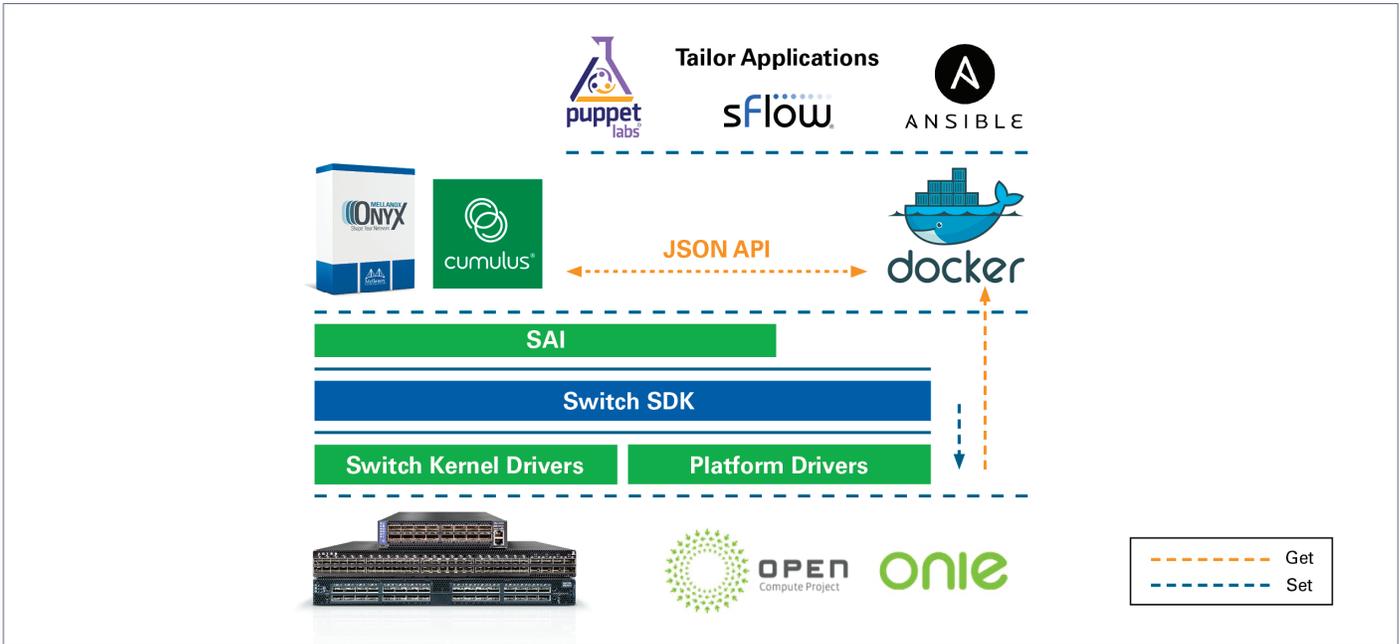


Figure 2. Running Docker on top of Spectrum

DOCKER USE CASES USING SPECTRUM

Use Case 1 - Telemetry

Network change cycles are growing increasingly shorter and more frequent. As a result, operators need as much visibility into the network as possible to help them identify, isolate and resolve potential issues as quickly as possible to keep their networks running smoothly. A network with increased and real-time visibility enables better reliability and real-time control.

Spectrum provides real-time hardware buffer monitoring. The architecture enables the switch to detect delays that occur in traffic flows when packets are buffered. Detecting the presence of any congestion at the hardware level provides useful visibility into the switch. Awareness of the location, depth and duration of the congestion enables for additional optimizations. Indications like buffer level, latency and bandwidth supporting watermarks, histograms and thresholds while streaming data to a local CPU. Docker on top of Spectrum switches gives the user a unique ability to collect, alert and analyze in real time, any event or alert supported by Spectrum and streamed out to visualization tools, such as Grafana, to gain better insight into metrics and enable fabric optimizations (and thus increasing revenue).

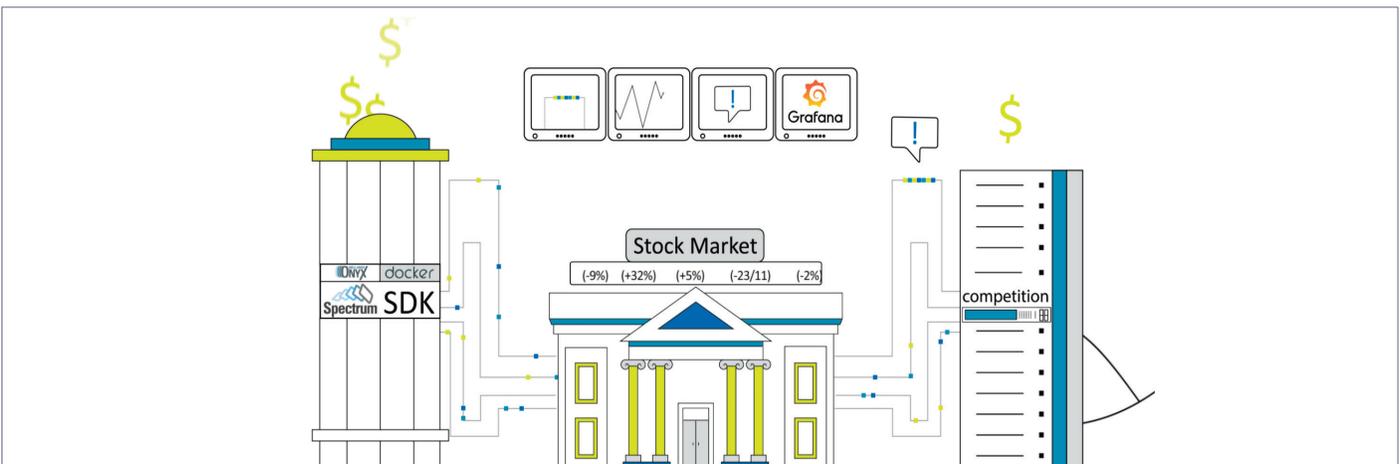


Figure 3. High frequency trading utilizing telemetry

Use Case 2 - Automotive

Modern cars and soon to come autonomous vehicles are small data centers with dozens of end customers (i.e. sensors, cameras) with need for fast and synced interconnect in order to ensure a safe ride.

The automotive industry mandates the unique but rarely implemented 802.1AS networking protocol which ensures not only precise timing and synchronization between network components, but also high bandwidth performance. Spectrum’s extensive accessibility allows end-users to achieve a competitive edge even faster than the switch vendors themselves by implementing 802.1AS on top of Spectrum.

Running tailored applications allows users to achieve the needed clock accuracy which is cardinal for the application to work properly and accurately. The container can configure the SDK to get the unique PTP packets from the network to enable a fast and safe integration of new code.



Figure 4. Autonomous cars will require high bandwidth accurate telemetry

Use Case 3 – Software Defined Storage (SDS)

Software-defined storage is a marketing term for computer data storage software for policy-based provisioning and management of data storage independent of the underlying hardware. Software-defined storage typically includes a form of storage virtualization to separate the storage hardware from the software that manages it. The software enabling a software-defined storage environment may also provide policy management for features such as data deduplication, replication, thin provisioning, snapshots and backup.

Here, a customer is building his own tailored SDS application using containers running on top of each of the switches in the fabric using a client-server/master-client architecture. Each of the clients monitor the switch status by polling all of the Spectrum counters exposed through the SDK open APIs. Moreover, the master can manage and configure the fabric according to its pre-defined policies.

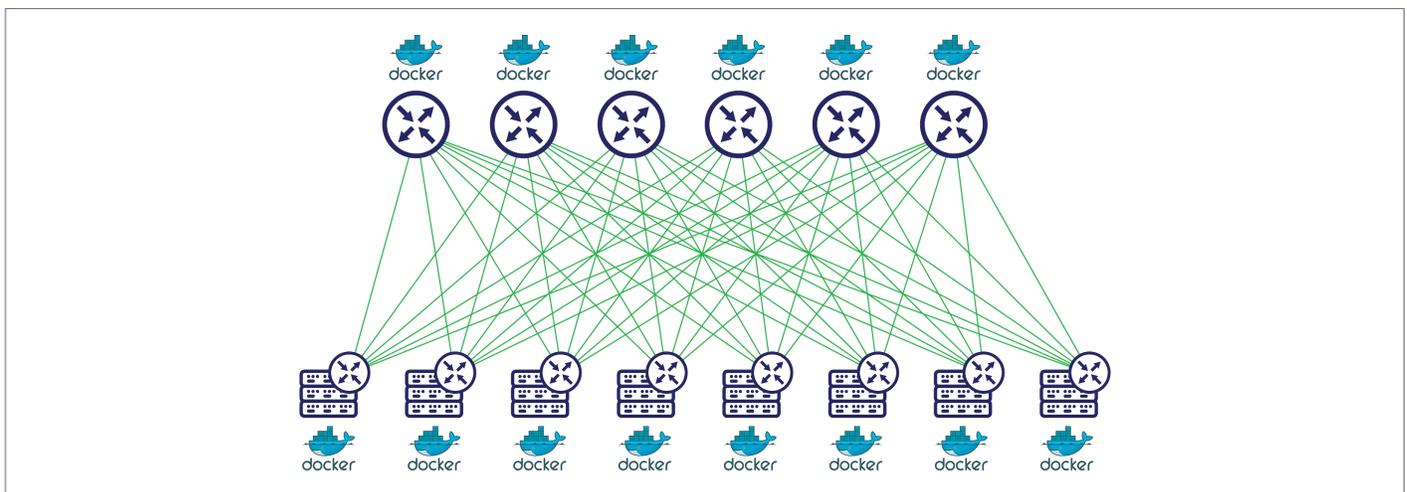


Figure 5. Software Defined Storage (SDS) Docker example

HOW TO RUN DOCKER OVER SPECTRUM

Docker can be installed directly on a Spectrum switch running either Onyx or Cumulus Linux, and Docker containers can be run natively on that switch. To run Docker in Onyx:

1. Start Docker service on the switch:

```
panther-144 [standalone: master] (config) # no docker shutdown
```

2. Pull a new image:

```
panther-144 [standalone: master] (config) # docker pull ubuntu
```

3. Start a new container:

```
panther-144 [standalone: master] (config) # docker start ubuntu latest my-ubuntu now
```

When should I start my Docker?

<Starting point>	<The starting point of this container>
Init	Persistent, Start the container after boot, when system configuration is done
data-path-ready	Persistent, Start the container after boot, when data-path is ready to be configured
now	Start the container now, this is not persistent

The user can also enable network and SDK support via:

```
panther-144 [standalone: master] (config) # docker start ubuntu latest my-ubuntu now privileged sdk network
```

The privileged, sdk, network options can go with any start point.

4. To get the SDK object and start developing an SDK application run:

```
panther-144 [standalone: master] (config) # docker copy-sdk my-ubuntu to /
```

To run Docker on top of Cumulus Linux, please refer to: <https://docs.cumulusnetworks.com/display/DOCS/Docker+on+Cumulus+Linux>.

Note: MLNX-OS 3.6.4006 and Cumulus Linux 3.3 or higher support the Docker engine. Mellanox Onyx was previously named MLNX-OS for Ethernet.

PAVE YOUR OWN PATH

Mellanox Spectrum switches are the only solution in the market to offer a full package to support speeds from 10GbE up to 200GbE with a full set of Layer 2 and Layer 3 features and unique capabilities, with no compromise on performance. With the latest addition of Docker containers access to Spectrum SDK API, Spectrum is the only solution to enable direct access to switches and provide DevOps-users with flexibility, faster time-to-market, and a unique edge when developing tailor made next-generation applications.

EXPLORE FURTHER

More information can be found under Mellanox/community.

- [Getting Started with Docker Container over MLNX-OS](#)
- [How To Enable Wireshark Over Docker Container \(MLNX-OS\)](#)
- [How To Enable DHCP Service Over Docker Container \(MLNX-OS\)](#)
- [Example of Python script running over Docker reading Spectrum counters](#)

About Mellanox

Mellanox Technologies (NASDAQ: MLNX) is a leading supplier of end-to-end Ethernet and InfiniBand intelligent interconnect solutions and services for servers, storage, and hyper-converged infrastructure. Mellanox intelligent interconnect solutions increase data center efficiency by providing the highest throughput and lowest latency, delivering data faster to applications and unlocking system performance. Mellanox offers a choice of high performance solutions: network and multi-core processors, network adapters, switches, cables, software and silicon, that accelerate application runtime and maximize business results for a wide range of markets including high performance computing, enterprise data centers, Web 2.0, cloud, storage, network security, telecom and financial services. More information is available at www.mellanox.com.

About Cumulus Networks

Cumulus Linux embodies native Linux networking. Supercharged versions of the kernel and other networking-related packages encompass the latest industry thinking in networking while retaining compatibility with the full range of software available in Debian. The SN200 series running Cumulus Linux provides standard networking functions such as bridging, routing, VLANs, MLAGs, IPv4/IPv6, OSPF/BGP, access control, VRF and VXLAN overlays.

Learn more about Cumulus Network operating systems at www.cumulusnetworks.com.