OpenDaylight SDN Controller Platform (OSCP)

Welcome to OpenDaylight SDN Controller Platform

Introduction

[PROPOSAL](#)

Downloading the Code

First things first: you can get the code from git by running:

```
git clone ssh://<username>@git.opendaylight.org:29418/net-virt-platform.git
```

And **click here** for Installation steps.

Introduction to Software Defined Networking

For over 20 years, network managers have grudgingly accepted the suboptimal nature of closed, proprietary networking architectures, which are inherently non-scalable, inflexible and administratively cumbersome. Traditional networking devices generally have proprietary management systems, implement proprietary protocols, and lack reliable APIs for external programmability or automation. While the enterprise compute and storage sectors have experienced dramatically improved flexibility and scalability over recent years, the networking industry has been slow to evolve and, therefore, is perceived by many enterprise customers as the primary inhibitor to agility and innovation.

The solution is open software-defined networking (Open SDN) — the use of standards-based protocols and open interfaces to abstract the network control plane from the data plane. Open SDN enables unified control and programmability across a network of heterogeneous physical and virtual networking devices. There is a groundswell of support for SDN initiatives among some of the largest technology customers in the world, leading academic institutions, and forward-looking networking vendors. Broad support for the non-profit industry consortium, the Open Networking Foundation (ONF), serves as a strong testament to the disruptive potential of SDN.

The ONF serves as the independent steward of the industry's first standard protocol for a common network control plane abstraction, called OpenFlow, which has now been widely adopted throughout the networking industry. Growing industry momentum and the adoption of these emerging standards have catalyzed the movement toward software-defined networking.

Open SDN Architecture

The Open SDN architecture is built around the OSCP controller, which provides a common data model and policy abstraction for all the network fabric elements. These universal network abstractions and the OVP leverage industry standards and open APIs to provide maximum deployment flexibility. The OVP also enables a broad range of application support, including data center network virtualization.

Open SDN is based on a three-tier architecture of virtual devices. The Open SDN architecture unlocks the latent potential of physical networks and enables network managers to meet the business needs that have emerged recently:

- Northbound open APIs for application developers
- An open-core controller
- Southbound standards-based data plane communication protocols

Project Facts

Project Creation Date:

Primary Contact:

Project Lead:

Committers:

Mailing List: net-virt-platform-dev@lists.opendaylight.org
   Archives: [mailing list archives](#)

Meetings: See [Community Meetings](#)

Repository: git clone https://git.opendaylight.org/gerrit/net-virt-platform

Jenkins: [jenkins silo](#)

Gerrit Patches: [code patches /reviews](#)

Bugs:

- [open bugs](#)

Northbound Open APIs

Open APIs refer to the software interfaces between the software modules of the controller platform and the SDN applications running atop the network platform. The Open SDN architecture employs Northbound APIs, which expose the universal network abstraction data models and functionality within the controller, for use by network applications. These interfaces are published and open to customers, partners, and the open source community for development. The Open APIs enable maximum utility of the Open SDN architecture through the ecosystem of customers and partners developing on the platform.

Click here to go to the Northbound API Reference

Standards-Based Southbound Protocols

The OpenDaylight Virtualization Platform utilizes standards-based connectivity for the Southbound Protocols, which define the control communications between the controller platform and data plane devices, including physical and virtual switches. Furthermore, the Open SDN architecture is flexible and can leverage other protocols in addition to Openflow. Support for a wide range of physical and virtual switches ensure that customers have maximum choice and flexibility for designing and deploying their software-defined network.

The Three-Tier Architecture: OpenDaylight Virtualization Platform, Switches, and Applications

A Cluster of Controller Nodes

The OSCP controller provides the centralized control plane tier in the three-tier Open SDN architecture diagram above. While OSCP is logically centralized, the controller is installed on multiple controller-nodes for redundancy and scale. Each controller-node is simply a separate installed image of the software (or separate hardware appliances with the software installed on it). All the controller-nodes communicate with each other to form a controller cluster, where each node in the cluster shares information with other nodes in the cluster to ensure availability in case any node fails.

blocked URL

OpenFlow-enabled switches, whether physical switches or hypervisor/virtual switches, are configured to connect to the OSCP controller-nodes. Once this is done, the OVP uses OpenFlow to program specific instructions dynamically into the switches' forwarding tables to implement the application-specific forwarding behavior. Note that some switches can connect to multiple controller-nodes simultaneously while some connect to them one at a time. When switches connect to an OpenFlow controller, they identify themselves with a unique "datapath-id" or DPID.

Applications can be enabled to run on top of the OVP and its infrastructure and APIs. In this release, one application is available, the OpenDaylight Network Virtualization (ONV) application. The applications leverage common management infrastructure such as login/security, configuration files, logging and debugging utilities. The applications also use the underlying OpenFlow libraries provided by OVP to control the connected OpenFlow-enabled switches by sending "flow-mods" or "flow-entries" down to the tables inside the switches.

These flow-mods in the switch tables are comprised of three parts:

1. Match conditions that are applied to packets entering the switch - these include matching the ingress port, source/destination MAC addresses, VLAN, and other parts of the packet header.
2. Actions that are taken on the packets - these include dropping the packet, forwarding the packet to a specific set of ports, or asking the controller what action should be taken on the packet
3. Counters that track how many bytes/packets matched a given flow-mod.

Documentation

Programmer Guide
Overview REST Reference Module Loading System Tutorial-Writing a Module

User Guide

Requirements

Release Planning

Release Notes