OpenDaylight as a Platform for Network Programmability

FOSDEM, 3 February 2018

Charles Eckel, Cisco DevNet

eckelcu@cisco.com

Agenda

- What is SDN
- What is OpenDaylight
- Network programmability
- Installation
- Example use case (VPP)
- Conclusions

What is SDN

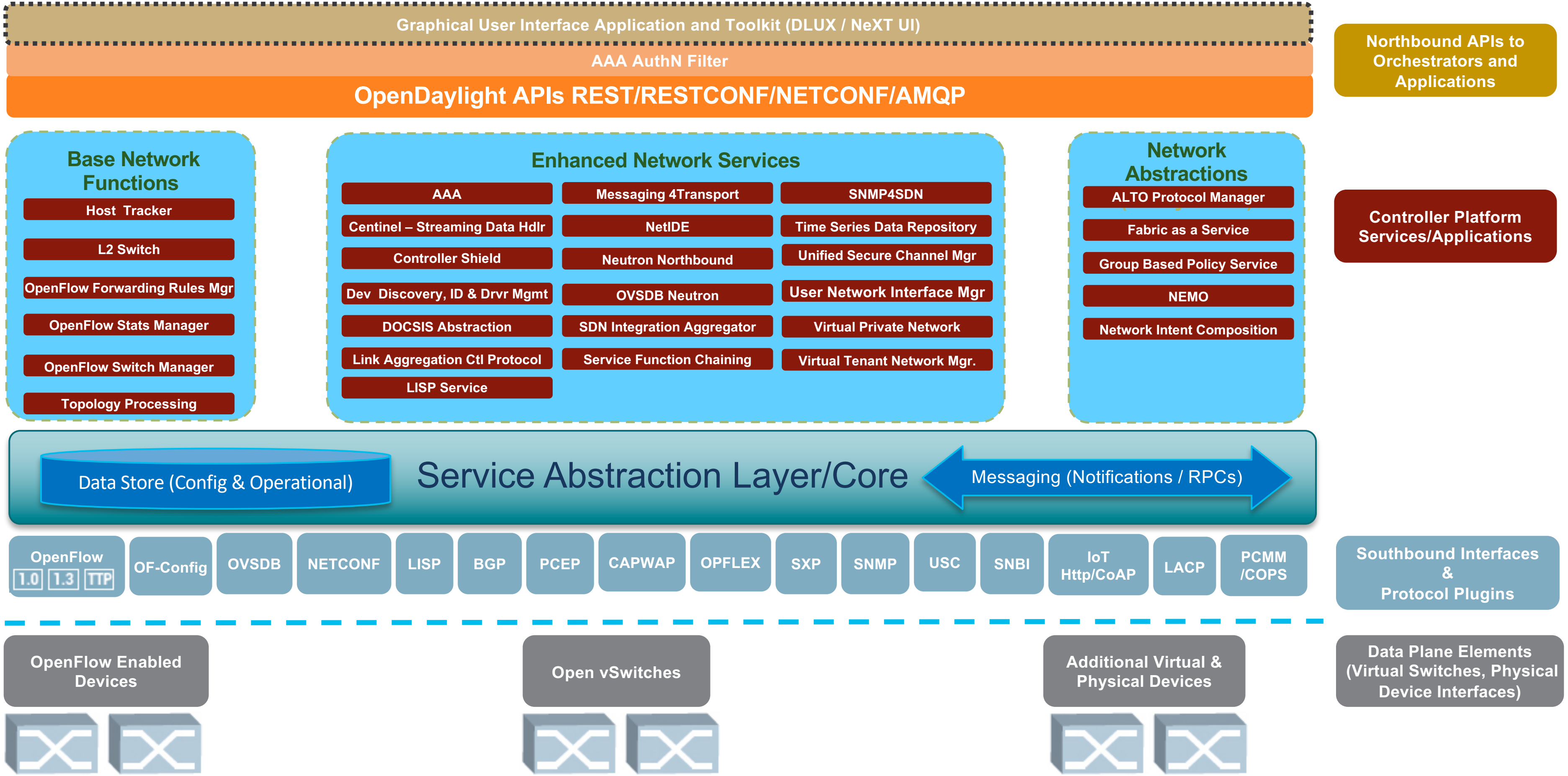
Software Defined Networking (SDN)

- Control & Data Planes separation?
  - OpenFlow?
  - Logically centralized control Plane?
  - White label switches?
- This a valid & useful SDN use case, but...
- SDN can be defined more broadly:
  - Network is a source of vast amount of data...
  - ..that can be utilized by variety of SDN applications
- True power of SDN is network programmability

What Do We Need from an SDN Controller?

- A platform for deploying SDN applications
- Provide an SDN application development environment
  - Developer-friendly APIs to network elements (REST/JSON, pub/sub, etc.)
  - Network-level abstraction through topologies
  - Protocol independence for network-facing applications

What is OpenDaylight



Graphical User Interface Application and Toolkit (DLUX / NeXT UI)

AAA AuthN Filter

OpenDaylight APIs REST/RESTCONF/NETCONF/AMQP

Northbound APIs to Orchestrators and Applications

Base Network Functions

- Host Tracker
- L2 Switch
- OpenFlow Forwarding Rules Mgr
- OpenFlow Stats Manager
- OpenFlow Switch Manager
- Topology Processing

Enhanced Network Services

- AAA
- Centinel – Streaming Data Hdlr
- Controller Shield
- Dev Discovery, ID & Drvr Mgmt
- DOCSIS Abstraction
- Link Aggregation Ctl Protocol
- LISP Service
- Messaging 4Transport
- NetIDE
- Neutron Northbound
- OVSDB Neutron
- SDN Integration Aggregator
- Service Function Chaining
- SNMP4SDN
- Time Series Data Repository
- Unified Secure Channel Mgr
- User Network Interface Mgr
- Virtual Private Network
- Virtual Tenant Network Mgr.

Network Abstractions

- ALTO Protocol Manager
- Fabric as a Service
- Group Based Policy Service
- NEMO
- Network Intent Composition

Controller Platform Services/Applications

Data Store (Config & Operational)

Service Abstraction Layer/Core

Messaging (Notifications / RPCs)

OpenFlow 1.0 1.3 TTP | OF-Config | OVSDB | NETCONF | LISP | BGP | PCEP | CAPWAP | OPFLEX | SXP | SNMP | USC | SNBI | IoT Http/CoAP | LACP | PCMM /COPS

Southbound Interfaces & Protocol Plugins

OpenFlow Enabled Devices

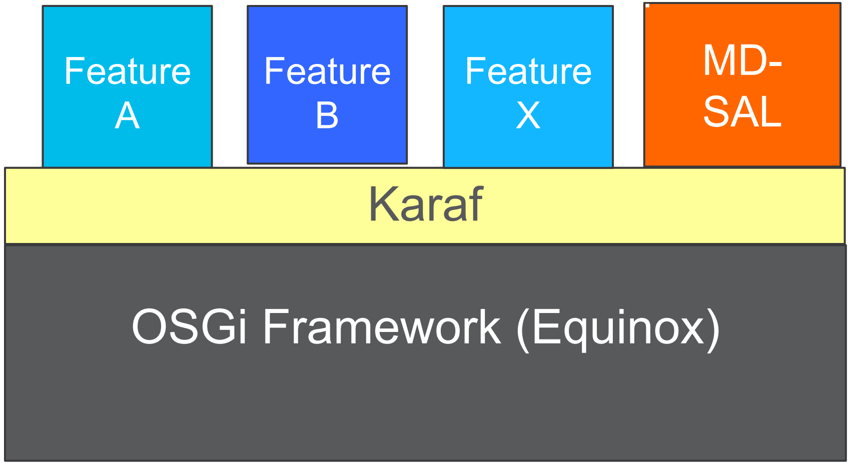
Open vSwitches

Additional Virtual & Physical Devices

Data Plane Elements (Virtual Switches, Physical Device Interfaces)

Software Architecture

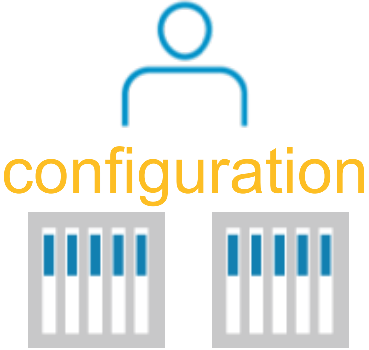
- Java - enterprise-grade, cross-platform compatible language
- Java Interfaces - for event listening, specifications and forming patterns
- Maven – build system
- Karaf – based on OSGi, provides:
  - dynamic loading of bundles
  - registering dependencies and services exported
  - exchanging information across bundles



Network programmability

The Need for Something Better

- **SNMP had failed**
  - For configuration, that is
  - Extensive use in fault handling and monitoring
- CLI scripting
  - “Market share” 70%+

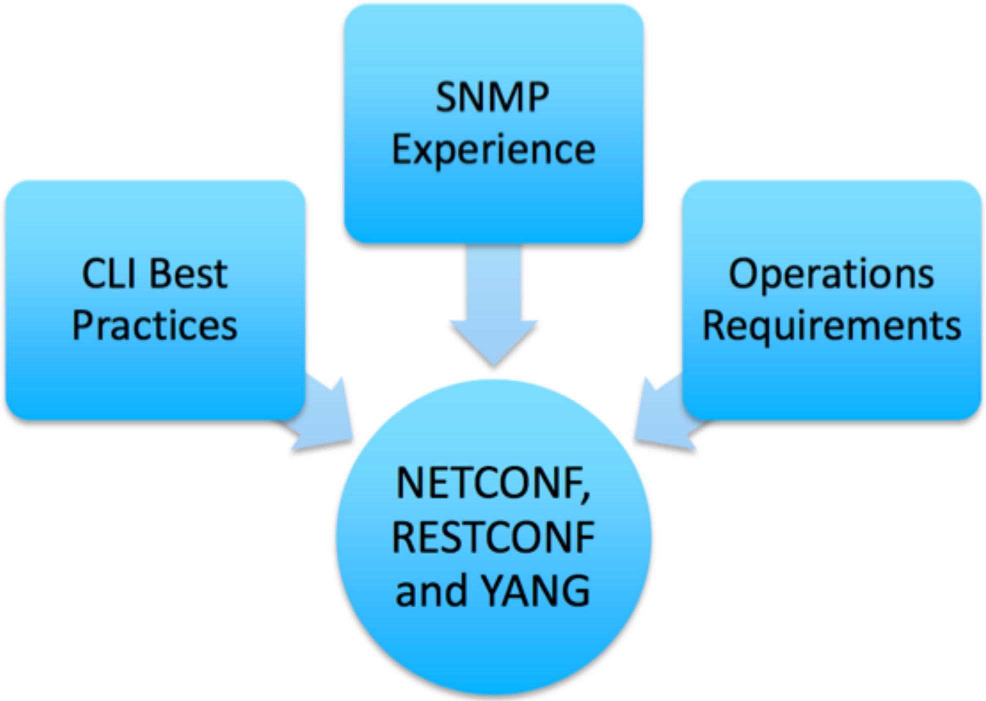


RFC 3535

Abstract

This document provides an overview of a workshop held by the Internet Architecture Board (IAB) on Network Management. The workshop was hosted by CNRI in Reston, VA, USA from June 4 thru June 6, 2002. The goal of the workshop was to continue the important **dialog** started between **network operators** and protocol developers, and to guide the IETFs focus on future work regarding network management.

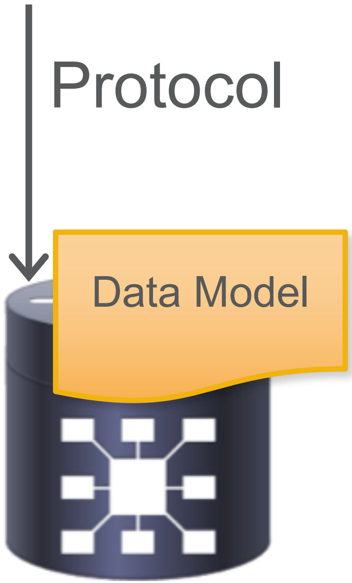
Best Practices Coming Together



YANG

Data Modeling Language for Networking

- Modeling language, defined in RFC 6020
- Represents operational state, configuration, transactions, and notifications
- Defines semantics
  - Constraints (i.e. “MUSTs”)
  - Reusable structures
  - Built-in and derived types

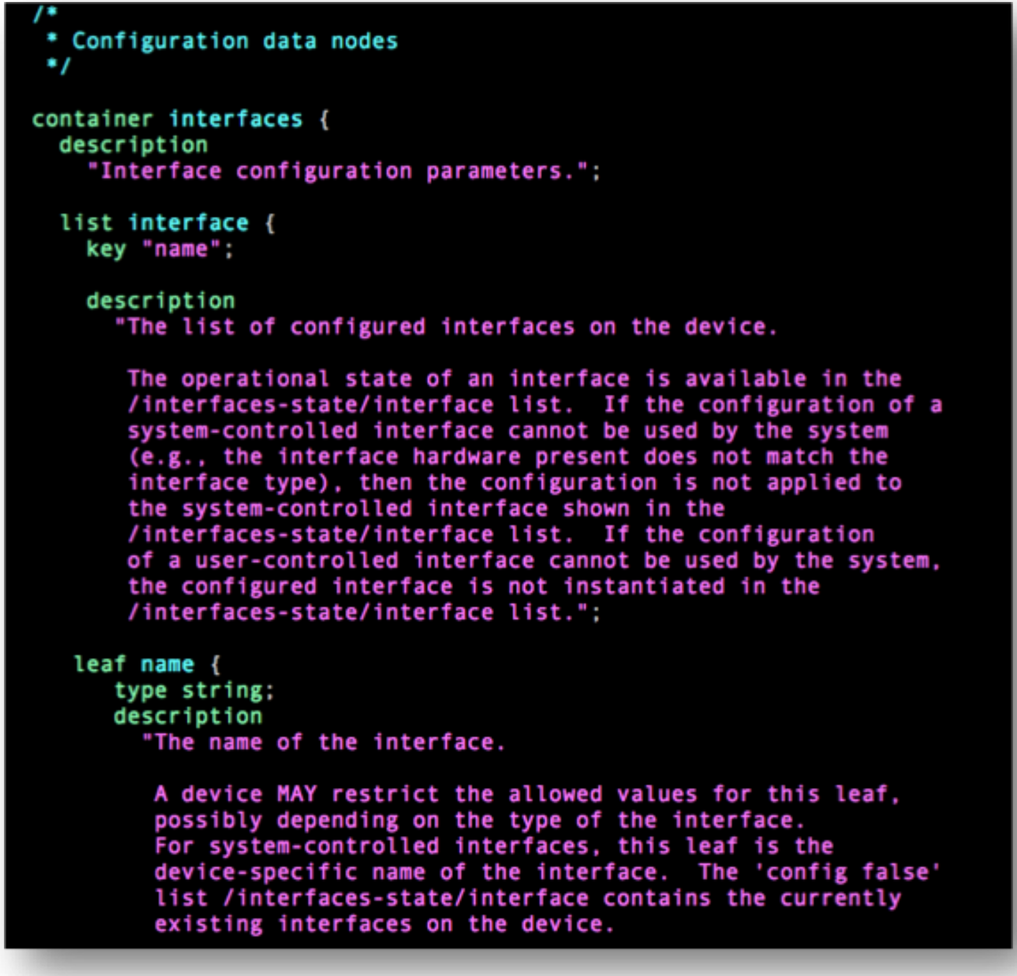


In Summary:

YANG is a full, formal contract language with rich syntax and semantics for network data

YANG Model Example

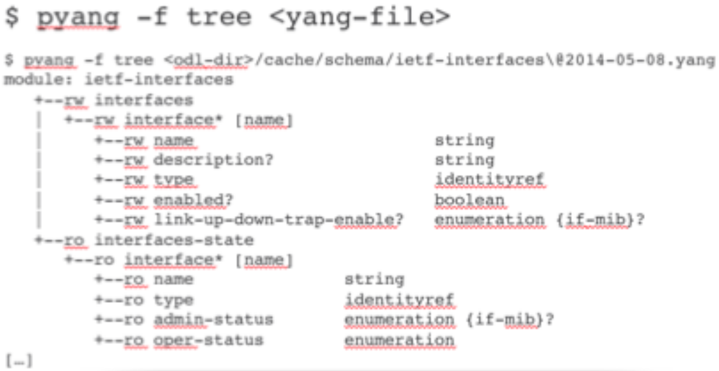
- Screenshot from ietf-interfaces.yang
- Container 'interfaces' with list of 'interface' items
- List items (leafs) have a 'name' which is also the key for the list



```
/*
 * Configuration data nodes
 */

container interfaces {
  description
    "Interface configuration parameters.";

  list interface {
    key "name";

    description
      "The list of configured interfaces on the device.

       The operational state of an interface is available in the
       /interfaces-state/interface list.  If the configuration of a
       system-controlled interface cannot be used by the system
       (e.g., the interface hardware present does not match the
       interface type), then the configuration is not applied to
       the system-controlled interface shown in the
       /interfaces-state/interface list.  If the configuration
       of a user-controlled interface cannot be used by the system,
       the configured interface is not instantiated in the
       /interfaces-state/interface list.";

   leaf name {
      type string;
      description
        "The name of the interface.

         A device MAY restrict the allowed values for this leaf,
         possibly depending on the type of the interface.
         For system-controlled interfaces, this leaf is the
         device-specific name of the interface.  The 'config false'
         list /interfaces-state/interface contains the currently
         existing interfaces on the device.
```

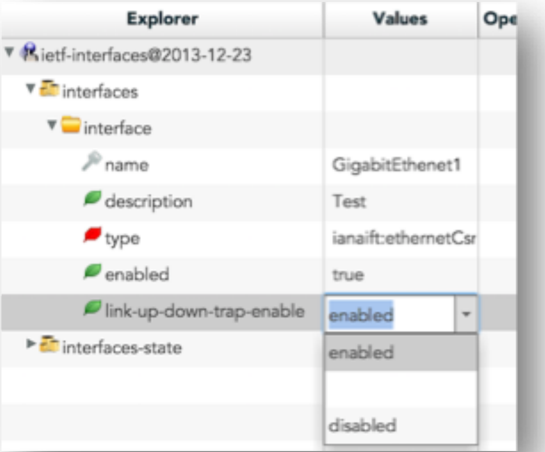
Tools to work with YANG Models

- pyang - An extensible YANG validator and converter in python
  - Source Code - https://github.com/mbj4668/pyang
  - Python Package - https://pypi.python.org/pypi/pyang
- YANG Explorer - YANG Browser / RPC Builder
  - https://github.com/CiscoDevNet/yang-explorer
- OpenDaylight YANG Tools – Tools supporting NETCONF and YANG, code generation from YANG models
  - https://wiki.opendaylight.org/view/YANG_Tools:Main



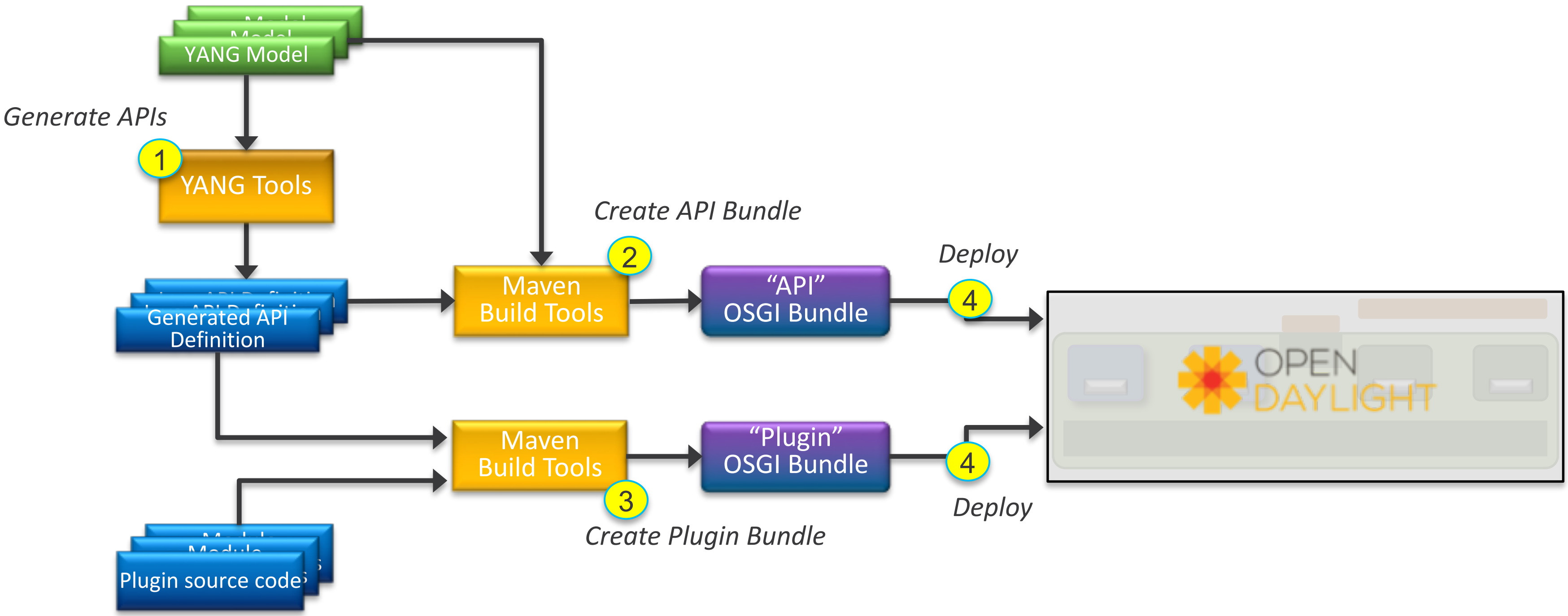
```
$ pyang -f tree <yang-file>
```

```
$ pyang -f tree <odl-dir>/cache/schema/ietf-interfaces\@2014-05-08.yang
module: ietf-interfaces
   +--rw interfaces
   |  +--rw interface* [name]
   |     +--rw name                        string
   |     +--rw description?                string
   |     +--rw type                        identityref
   |     +--rw enabled?                    boolean
   |     +--rw link-up-down-trap-enable?   enumeration {if-mib}?
   +--ro interfaces-state
      +--ro interface* [name]
         +--ro name               string
         +--ro type               identityref
         +--ro admin-status       enumeration {if-mib}?
         +--ro oper-status        enumeration
[…]
```

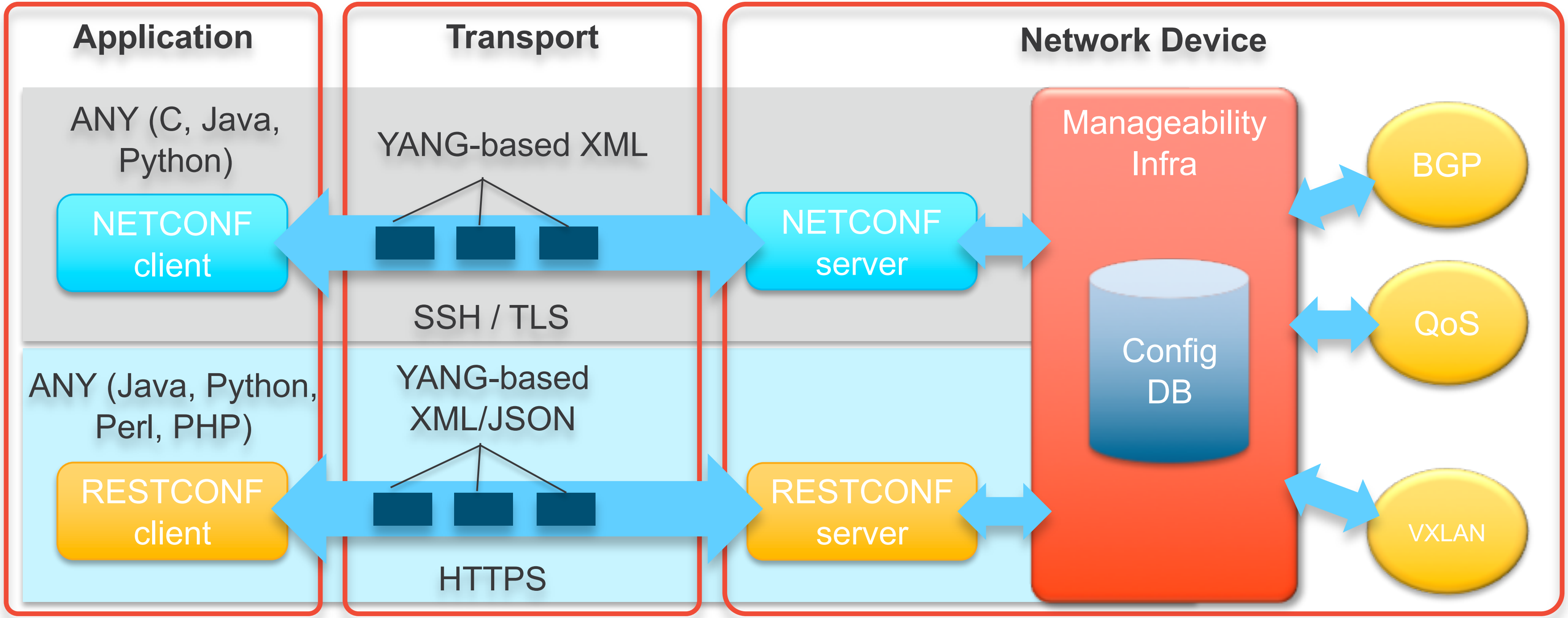


Explorer	Values	Ope
ietf-interfaces@2013-12-23		
interfaces		
interface		
name	GigabitEthenet1	
description	Test	
type	ianaift:ethernetCsr	
enabled	true	
link-up-down-trap-enable	enabled	
interfaces-state	enabled	
	disabled	

Building a Plugin/Application

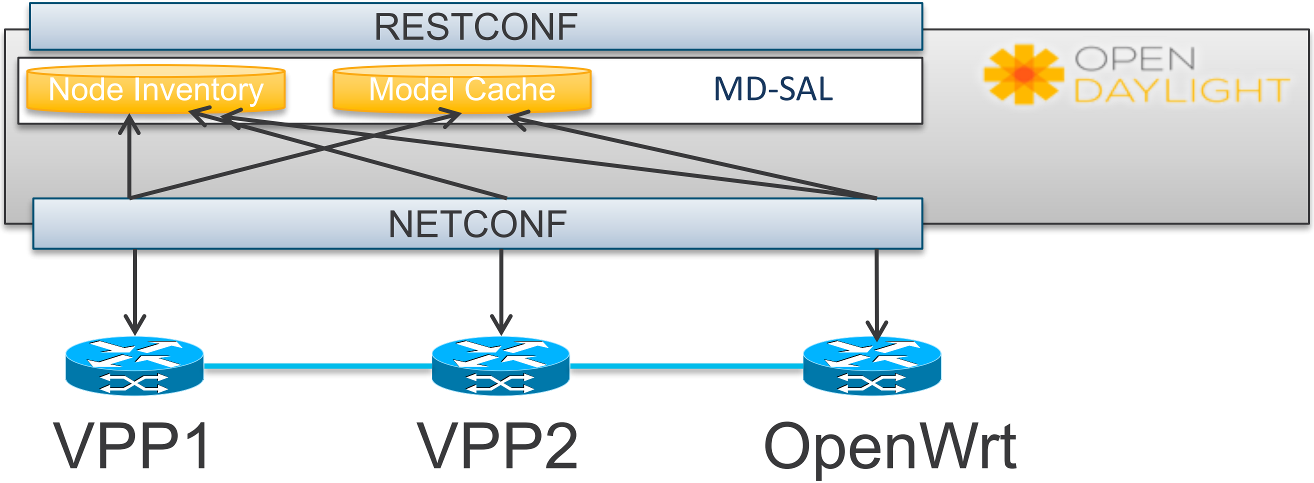


High Level Manageability Architecture



Mounting YANG Datastores
OpenDaylight NETCONF Node “Discovery”

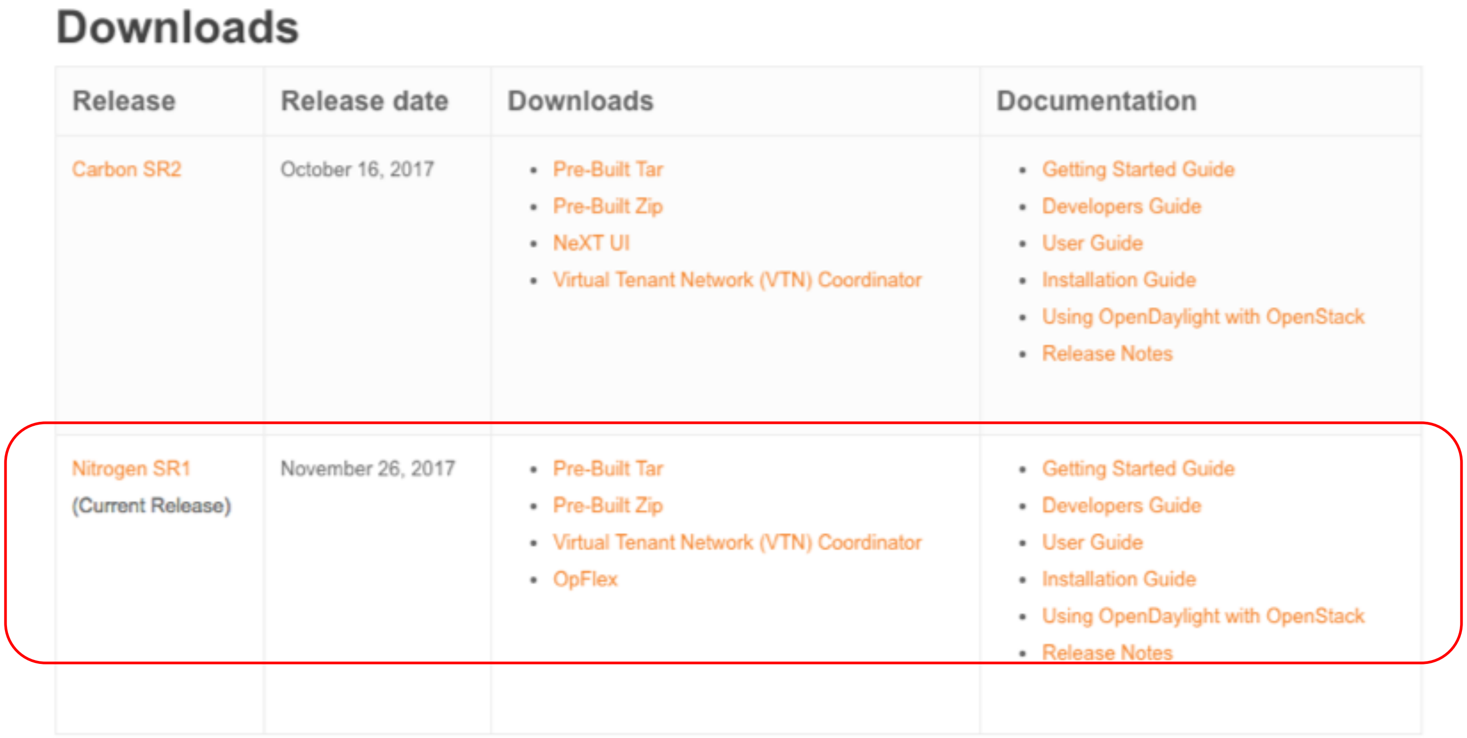
- Nodes added by POSTing to config:modules
- OpenDaylight connects to each node
- OpenDaylight learns capabilities (YANG modules) and stores to cache
    - Cache at ~/cache/schema.   Filenames of form yang-model@2016-07-12.yang.



Installation

Distributions

https://www.opendaylight.org/technical-community/getting-started-for-developers/downloads-and-documentation



Downloads

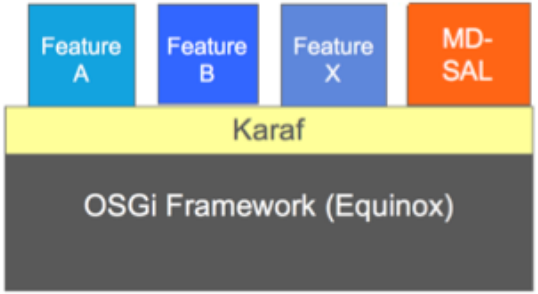
Release	Release date	Downloads	Documentation
Carbon SR2	October 16, 2017	- Pre-Built Tar - Pre-Built Zip - NeXT UI - Virtual Tenant Network (VTN) Coordinator	- Getting Started Guide - Developers Guide - User Guide - Installation Guide - Using OpenDaylight with OpenStack - Release Notes
Nitrogen SR1 (Current Release)	November 26, 2017	- Pre-Built Tar - Pre-Built Zip - Virtual Tenant Network (VTN) Coordinator - OpFlex	- Getting Started Guide - Developers Guide - User Guide - Installation Guide - Using OpenDaylight with OpenStack - Release Notes

```
$ unzip karaf-0.7.1.zip
Archive:  karaf-0.7.1.zip
   creating: karaf-0.7.1/system/ …
$ cd karaf-0.7.1
$ ./bin/karaf
karaf: Enabling Java debug options: -Xdebug -Xnoagent -Djava.compiler=NONE
   -Xrunjdwp:transport=dt_socket,server=y,suspend=n,address=5005
Listening for transport dt_socket at address: 5005
Apache Karaf starting up. Press Enter to open the shell now...
100% [========================================================================]

Karaf started in 0s. Bundle stats: 10 active, 10 total
________                       ________                .__  .__       .__     __
\_____  \ ______   ____   ____ \______ \ _____  ___.__.|  | |__| ____ |  |___/  |_
 /   |   \\____ \_/ __ \ /    \ |    |  \\__  \<   |  ||  | |  |/ ___\|  |  \   __\
/    |    \  |_> >  ___/|   |  \|    `   \/ __ \\___  ||  |_|  / /_/  >   Y  \  |
\_______  /   __/ \___  >___|  /_______  (____  / ____||____/__\___  /|___|  /__|
        \/|__|        \/     \/        \/     \/\/            /_____/      \/

Hit '<tab>' for a list of available commands
and '[cmd] --help' for help on a specific command.
Hit '<ctrl-d>' or type 'system:shutdown' or 'logout' to shutdown OpenDaylight.

opendaylight-user@root>
```

Install Features using Karaf

- OpenDaylight distro comes without any features enabled by default
- All features are available for you to install

• *feature:list*	list all features available
• *feature:list -i*	list all features installed
• *feature:list –r*	list all features required
• *feature:install <feature>*	install the <feature> feature
• *feature:install <feature-1> <feature-2> … <feature-n>*	install list of features
• *feature:uninstall <feature>*	uninstalls the <feature> feature

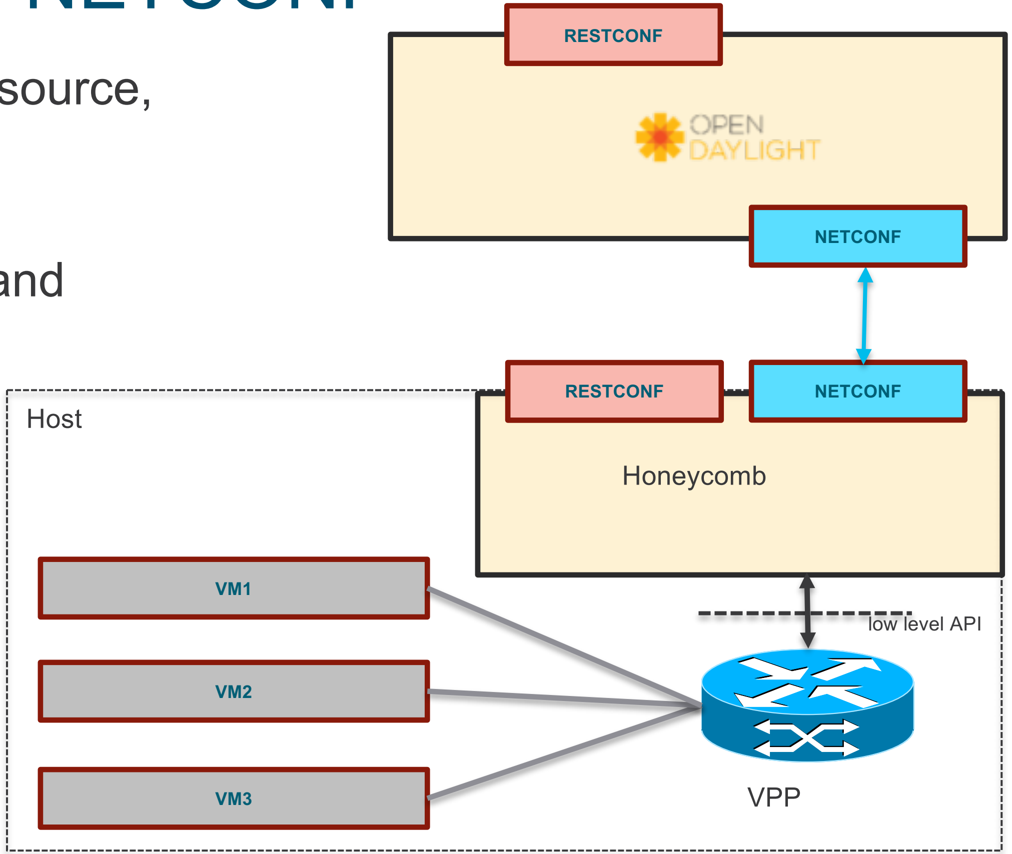


Install DLUX, NETCONF, and RESTCONF

```
opendaylight_user@root> feature:install odl-dlux-core
opendaylight_user@root> feature:install odl-dluxapps-yangui
opendaylight_user@root> feature:install odl-restconf-all
opendaylight_user@root> feature:install odl-netconf-all
opendaylight_user@root> feature:install odl-netconf-topology
Opendaylight_user@root> feature:install odl-netconf-connector-ssh
opendaylight_user@root> feature:list -r

Name                     | Version Required State
------------------------------------------------------
odl-netconf-topology      | 1.3.1 | x       | Started
odl-restconf-all          | 1.6.1 | x       | Started
odl-netconf-connector-ssh | 1.3.1 | x       | Started
odl-dluxapps-yangui       | 0.6.1 | x       | Started
odl-netconf-all           | 1.3.1 | x       | Started
odl-dlux-core             | 0.6.1 | x       | Started
wrap                      | 0.0.0 | x       | Started
standard                  | 4.0.10| x       | Started
```

Honeycomb/VPP using NETCONF

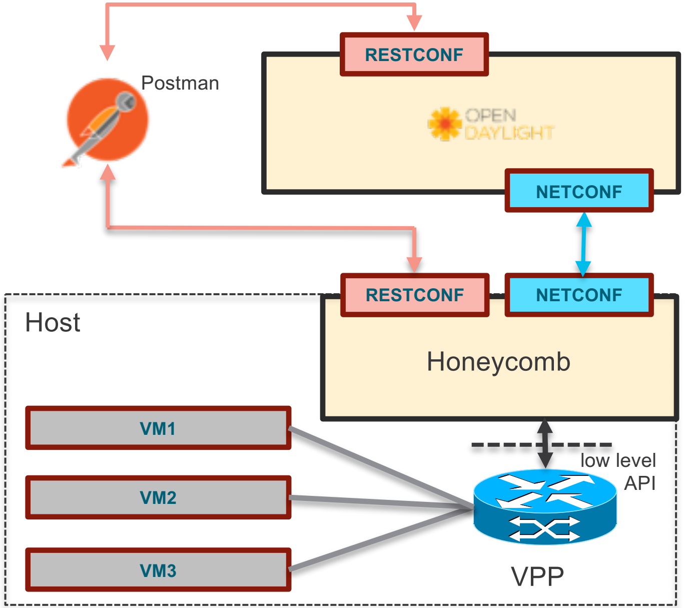
- VPP is a high-performance, open source, software forwarder
  - http://www.fd.io
- Honeycomb provides NETCONF and RESTCONF interfaces to VPP



Honeycomb/VPP Using NETCONF

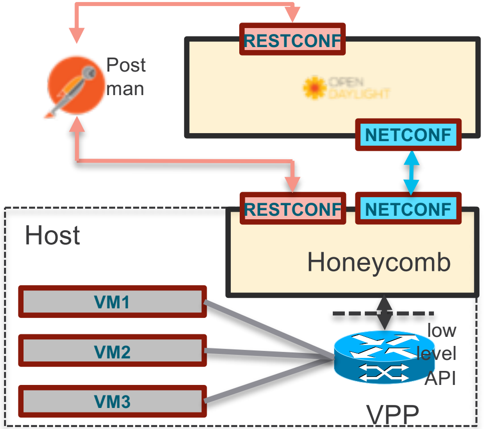
Step by Step

1. Create VM for Honeycomb and VPP
2. Install VPP and Honeycomb on VM
3. Start VPP and Honeycomb
4. Connect to VPP using CLI
5. Add interface(s) to VPP
6. Connect to VPP using Honeycomb/NETCONF
7. Connect to VPP using OpenDaylight

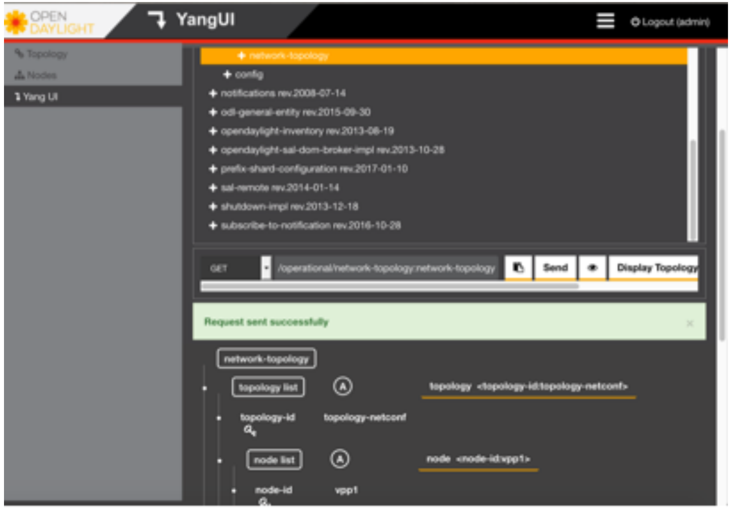


Honeycomb/VPP Using NETCONF

7. Connect to VPP Using OpenDaylight



- Enable NETCONF interface on OpenDaylight
  - feature:install odl-restconf-all odl-netconf-all odl-netconf-topology odl-netconf-connector-ssh
- Add VPP to OpenDaylight using Postman
  - PUT http://{{odl_address}}:8181/restconf/config/network-topology:network-topology/topology/topology-netconf/node/vpp1
  - Postman collection
    - https://github.com/CiscoDevNet/opendaylight-sample-apps/blob/master/postman-collections/ODL-VPP.json
- Interact with VPP using OpenDaylight DLUX

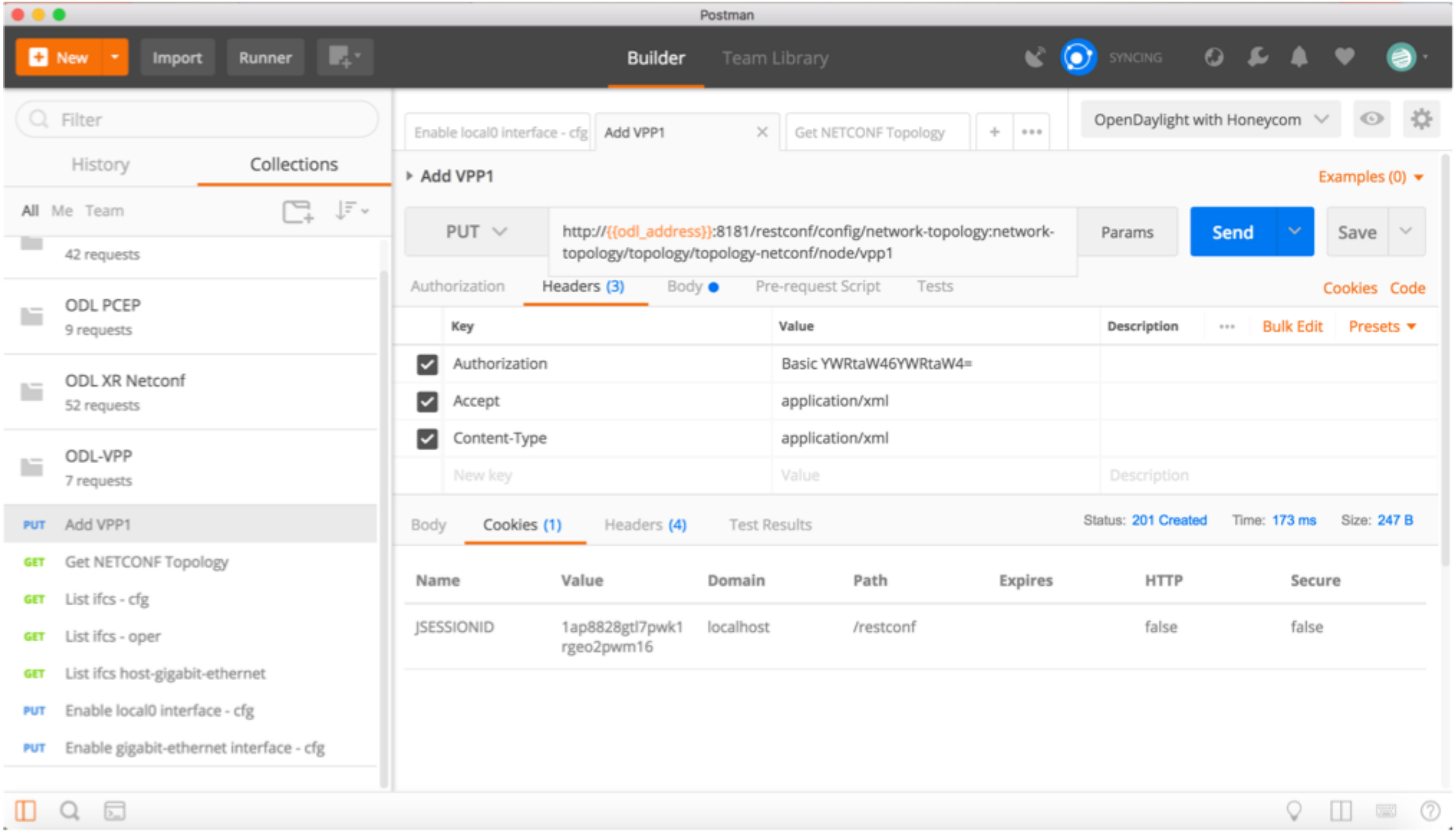


+ network-topology
+ config
+ notifications rev.2008-07-14
+ odl-general-entity rev.2015-09-30
+ opendaylight-inventory rev.2013-08-19
+ opendaylight-sal-dom-broker-impl rev.2013-10-28
+ prefix-shard-configuration rev.2017-01-10
+ sal-remote rev.2014-01-14
+ shutdown-impl rev.2013-12-18
+ subscribe-to-notification rev.2016-10-28

GET /operational/network-topology:network-topology

Request sent successfully

network-topology
topology list topology <topology-id:topology-netconf>
topology-id topology-netconf
node list node <node-id:vpp1>
node-id vpp1



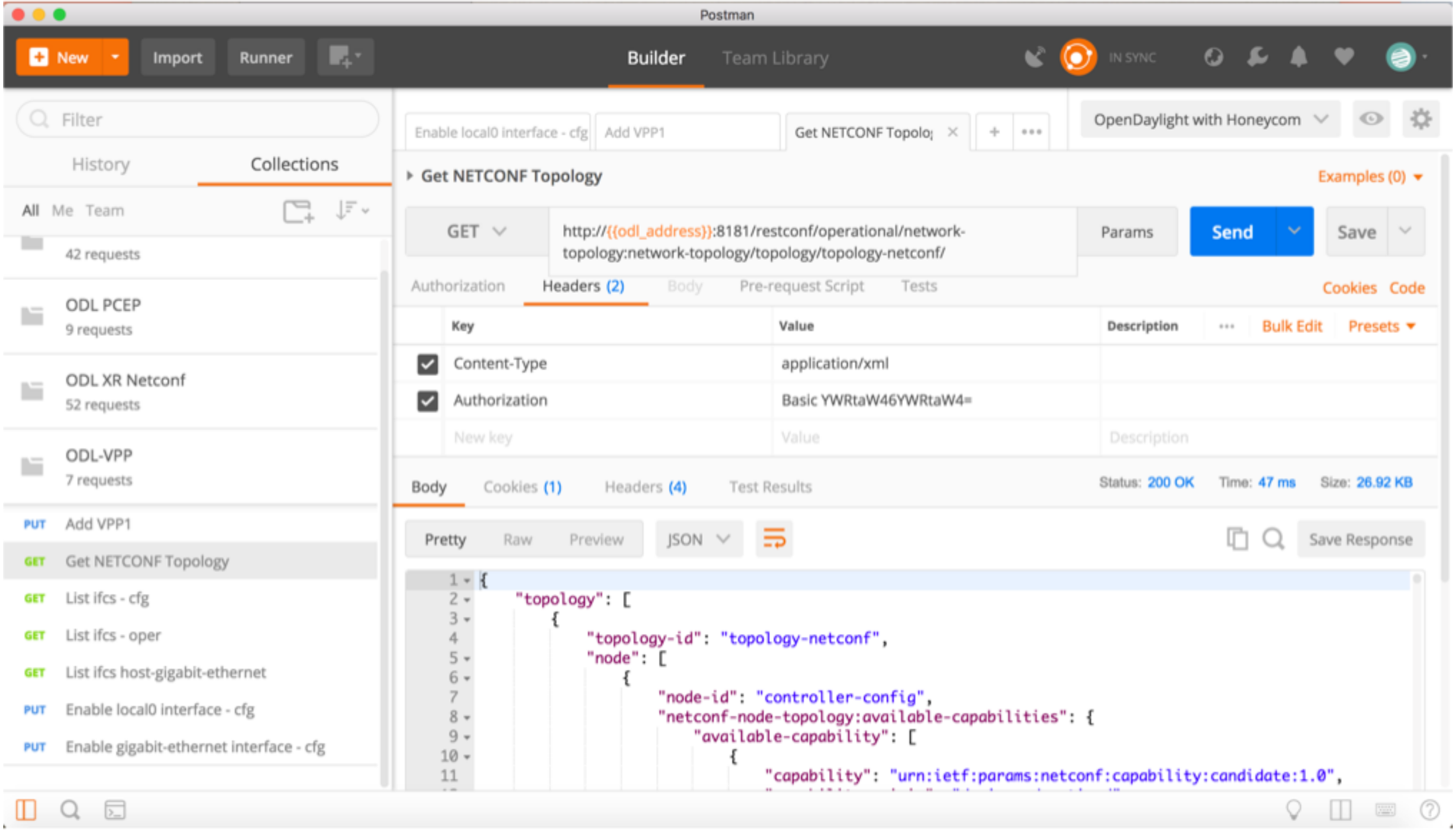
Add VPP1

PUT http://{{odl_address}}:8181/restconf/config/network-topology:network-topology/topology/topology-netconf/node/vpp1

	Key	Value	Description
☑	Authorization	Basic YWRtaW46YWRtaW4=	
☑	Accept	application/xml	
☑	Content-Type	application/xml	
	New key	Value	Description

Status: 201 Created Time: 173 ms Size: 247 B

Name	Value	Domain	Path	Expires	HTTP	Secure
JSESSIONID	1ap8828gtl7pwk1rgeo2pwm16	localhost	/restconf		false	false



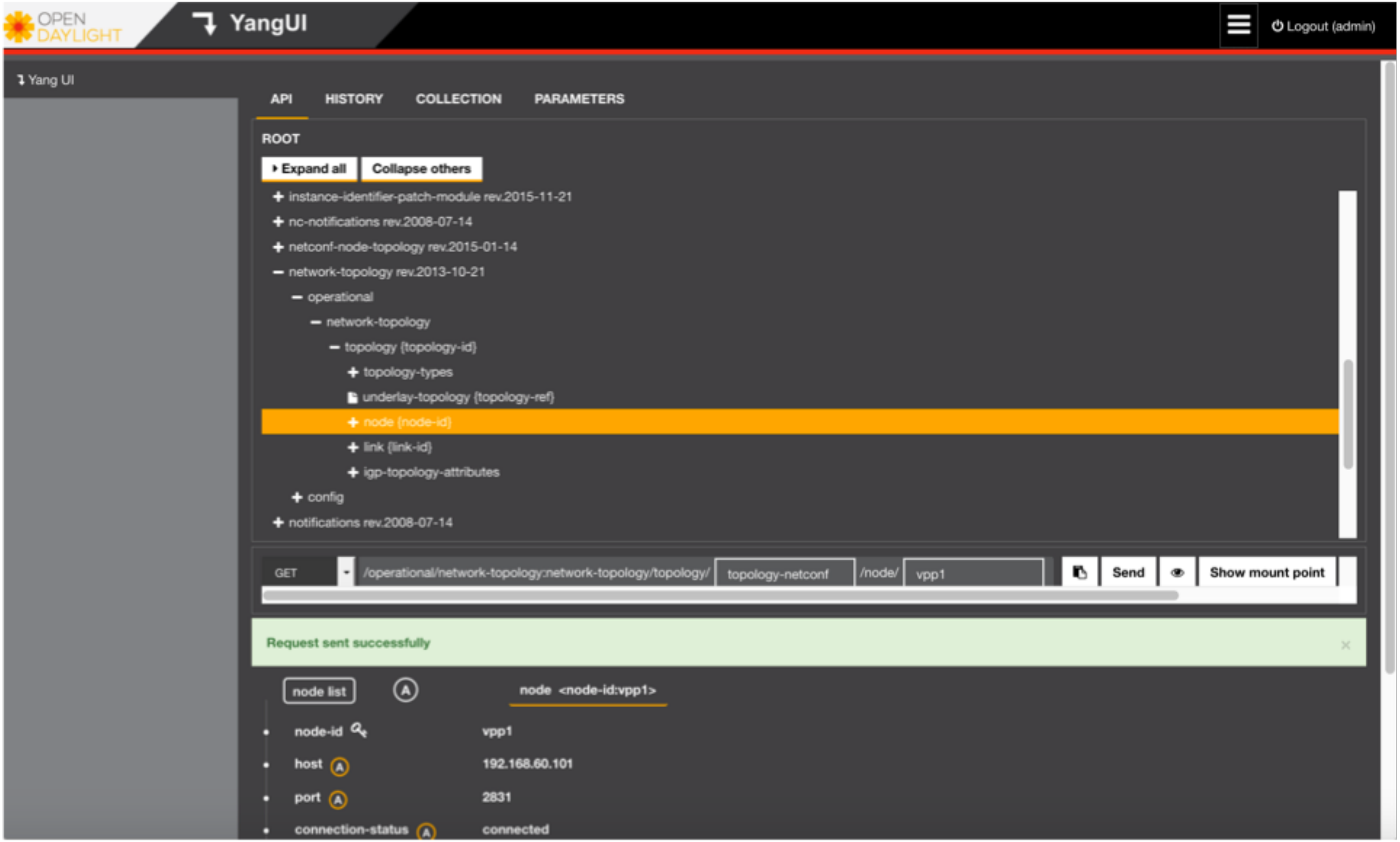
Get NETCONF Topology

GET http://{{odl_address}}:8181/restconf/operational/network-topology:network-topology/topology/topology-netconf/

	Key	Value	Description
☑	Content-Type	application/xml	
☑	Authorization	Basic YWRtaW46YWRtaW4=	
	New key	Value	Description

Status: 200 OK Time: 47 ms Size: 26.92 KB

```
{
    "topology": [
        {
            "topology-id": "topology-netconf",
            "node": [
                {
                    "node-id": "controller-config",
                    "netconf-node-topology:available-capabilities": {
                        "available-capability": [
                            {
                                "capability": "urn:ietf:params:netconf:capability:candidate:1.0",
```



- instance-identifier-patch-module rev.2015-11-21
- nc-notifications rev.2008-07-14
- netconf-node-topology rev.2015-01-14
- network-topology rev.2013-10-21
  - operational
    - network-topology
      - topology {topology-id}
        - topology-types
        - underlay-topology {topology-ref}
        - node {node-id}
        - link {link-id}
        - igp-topology-attributes
  - config
- notifications rev.2008-07-14

```
GET  /operational/network-topology:network-topology/topology/ topology-netconf /node/ vpp1
```

Request sent successfully

node-id	vpp1
host	192.168.60.101
port	2831
connection-status	connected

Conclusions

Key Takeaways

- SDN is more than just OpenFlow
- Network programmability is key benefit of SDN
- OpenDaylight provides a platform for network applications and programmable network infrastructure via YANG, NETCONF, RESTCONF

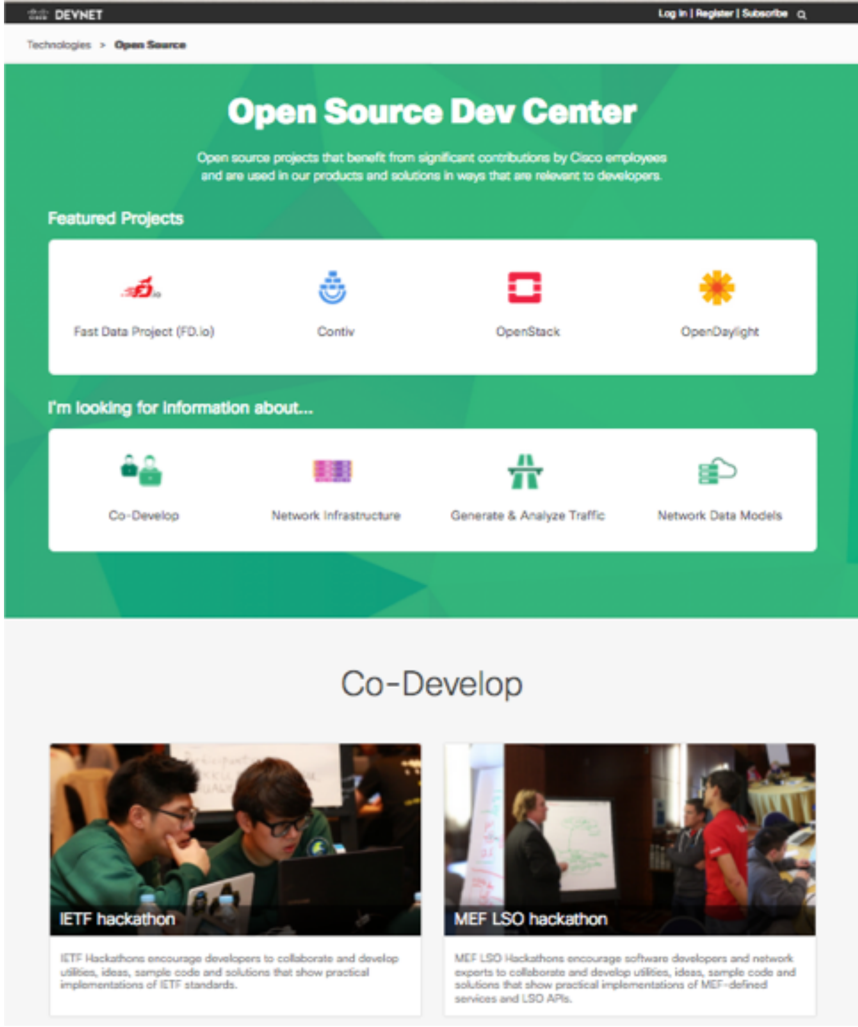
Additional resources

Open Source Dev Center

Your Source for Open Source at Cisco

https://developer.cisco.com/opensource

- Contributions to open source
- Use in products/solutions
- Community forums, blogs
- Developer Events
  - IETF Hackathons and MEF LSO Hackathons featuring open source implementations of open standards


DEVNET
Log in | Register | Subscribe
Technologies > Open Source

Open Source Dev Center

Open source projects that benefit from significant contributions by Cisco employees and are used in our products and solutions in ways that are relevant to developers.

Featured Projects

Fast Data Project (FD.io)
Contiv
OpenStack
OpenDaylight

I'm looking for information about...

Co-Develop
Network Infrastructure
Generate & Analyze Traffic
Network Data Models

Co-Develop

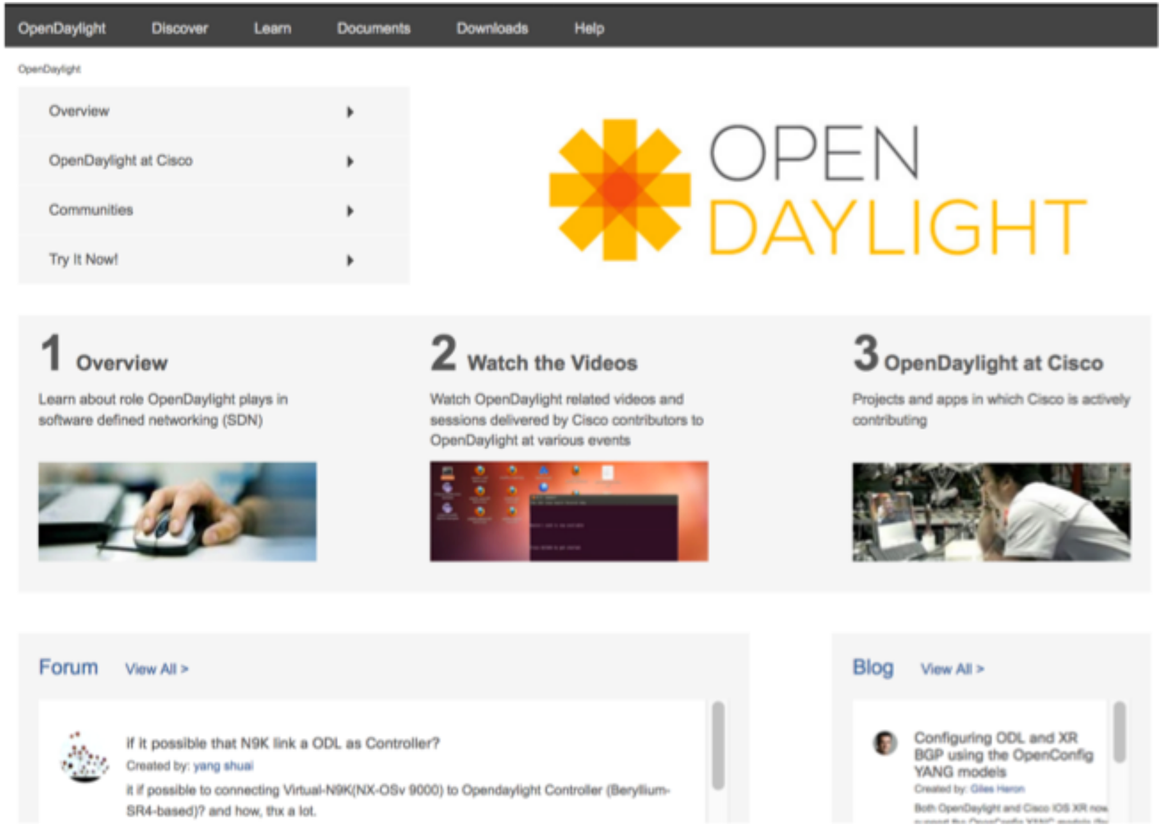
IETF hackathon

IETF Hackathons encourage developers to collaborate and develop utilities, ideas, sample code and solutions that show practical implementations of IETF standards.

MEF LSO hackathon

MEF LSO Hackathons encourage software developers and network experts to collaborate and develop utilities, ideas, sample code and solutions that show practical implementations of MEF-defined services and LSO APIs.

OpenDaylight Microsite
https://developer.cisco.com/opendaylight



1 Overview

Learn about role OpenDaylight plays in software defined networking (SDN)

2 Watch the Videos

Watch OpenDaylight related videos and sessions delivered by Cisco contributors to OpenDaylight at various events

3 OpenDaylight at Cisco

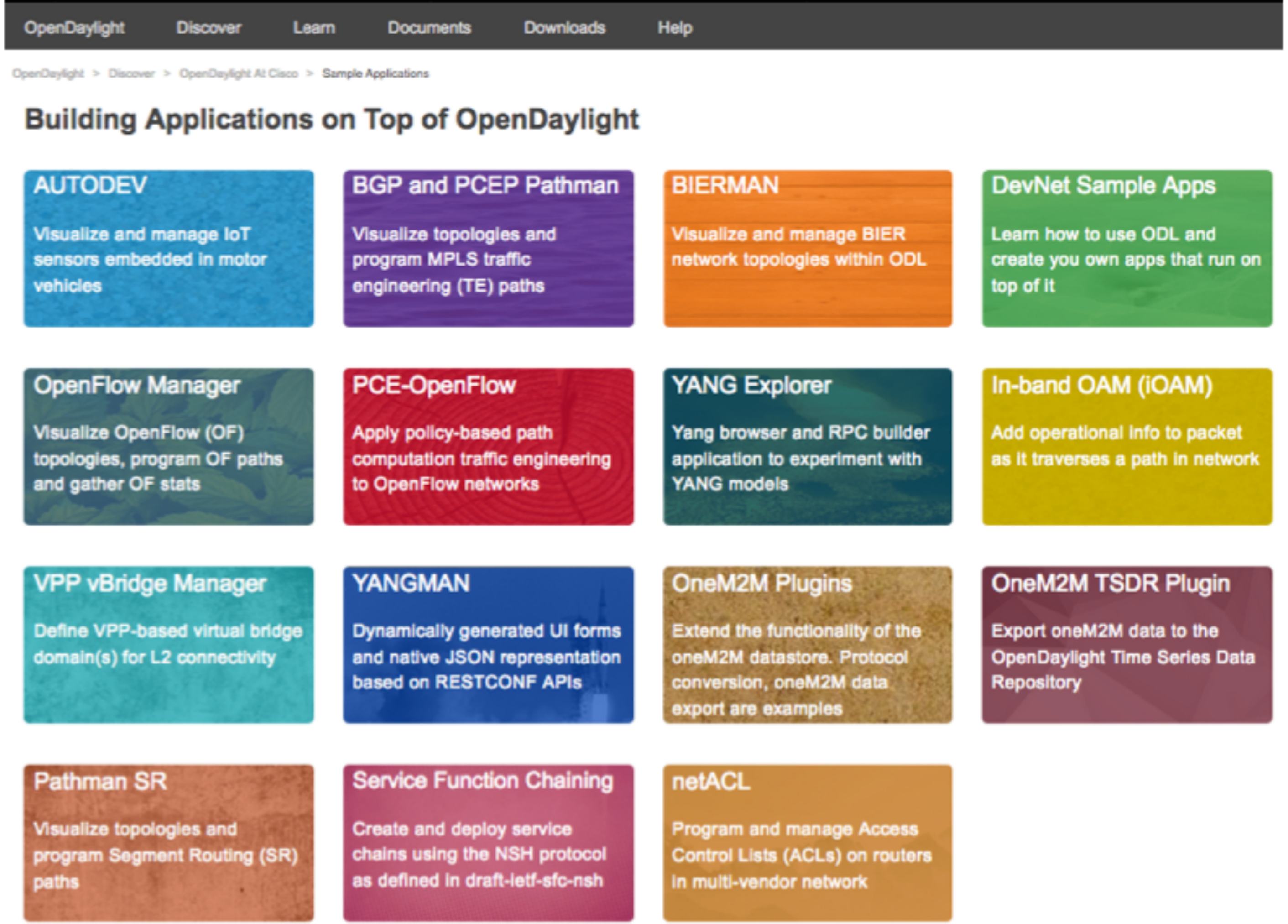
Projects and apps in which Cisco is actively contributing

Forum View All >

If it possible that N9K link a ODL as Controller?
Created by: yang shuai
it if possible to connecting Virtual-N9K(NX-OSv 9000) to Opendaylight Controller (Beryllium-SR4-based)? and how, thx a lot.

Blog View All >

Configuring ODL and XR BGP using the OpenConfig YANG models
Created by: Giles Heron
Both OpenDaylight and Cisco IOS XR now


OpenDaylight
Discover
Learn
Documents
Downloads
Help
OpenDaylight > Discover > OpenDaylight At Cisco > Sample Applications

Building Applications on Top of OpenDaylight

AUTODEV
Visualize and manage IoT sensors embedded in motor vehicles

BGP and PCEP Pathman
Visualize topologies and program MPLS traffic engineering (TE) paths

BIERMAN
Visualize and manage BIER network topologies within ODL

DevNet Sample Apps
Learn how to use ODL and create you own apps that run on top of it

OpenFlow Manager
Visualize OpenFlow (OF) topologies, program OF paths and gather OF stats

PCE-OpenFlow
Apply policy-based path computation traffic engineering to OpenFlow networks

YANG Explorer
Yang browser and RPC builder application to experiment with YANG models

In-band OAM (iOAM)
Add operational info to packet as it traverses a path in network

VPP vBridge Manager
Define VPP-based virtual bridge domain(s) for L2 connectivity

YANGMAN
Dynamically generated UI forms and native JSON representation based on RESTCONF APIs

OneM2M Plugins
Extend the functionality of the oneM2M datastore. Protocol conversion, oneM2M data export are examples

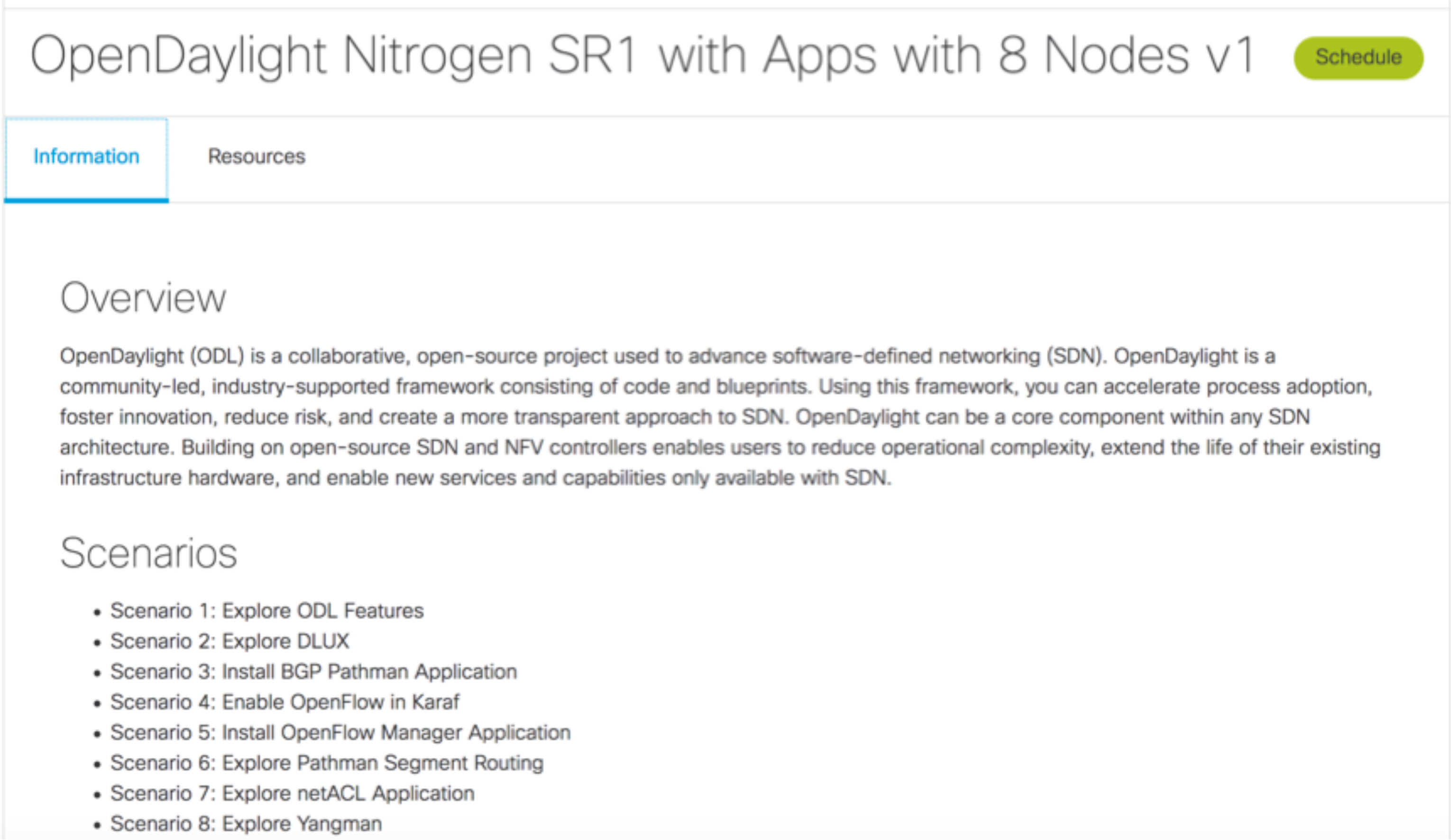
OneM2M TSDR Plugin
Export oneM2M data to the OpenDaylight Time Series Data Repository

Pathman SR
Visualize topologies and program Segment Routing (SR) paths

Service Function Chaining
Create and deploy service chains using the NSH protocol as defined in draft-ietf-sfc-nsh

netACL
Program and manage Access Control Lists (ACLs) on routers in multi-vendor network

Tutorials and Sandboxes



OpenDaylight Nitrogen SR1 with Apps with 8 Nodes v1

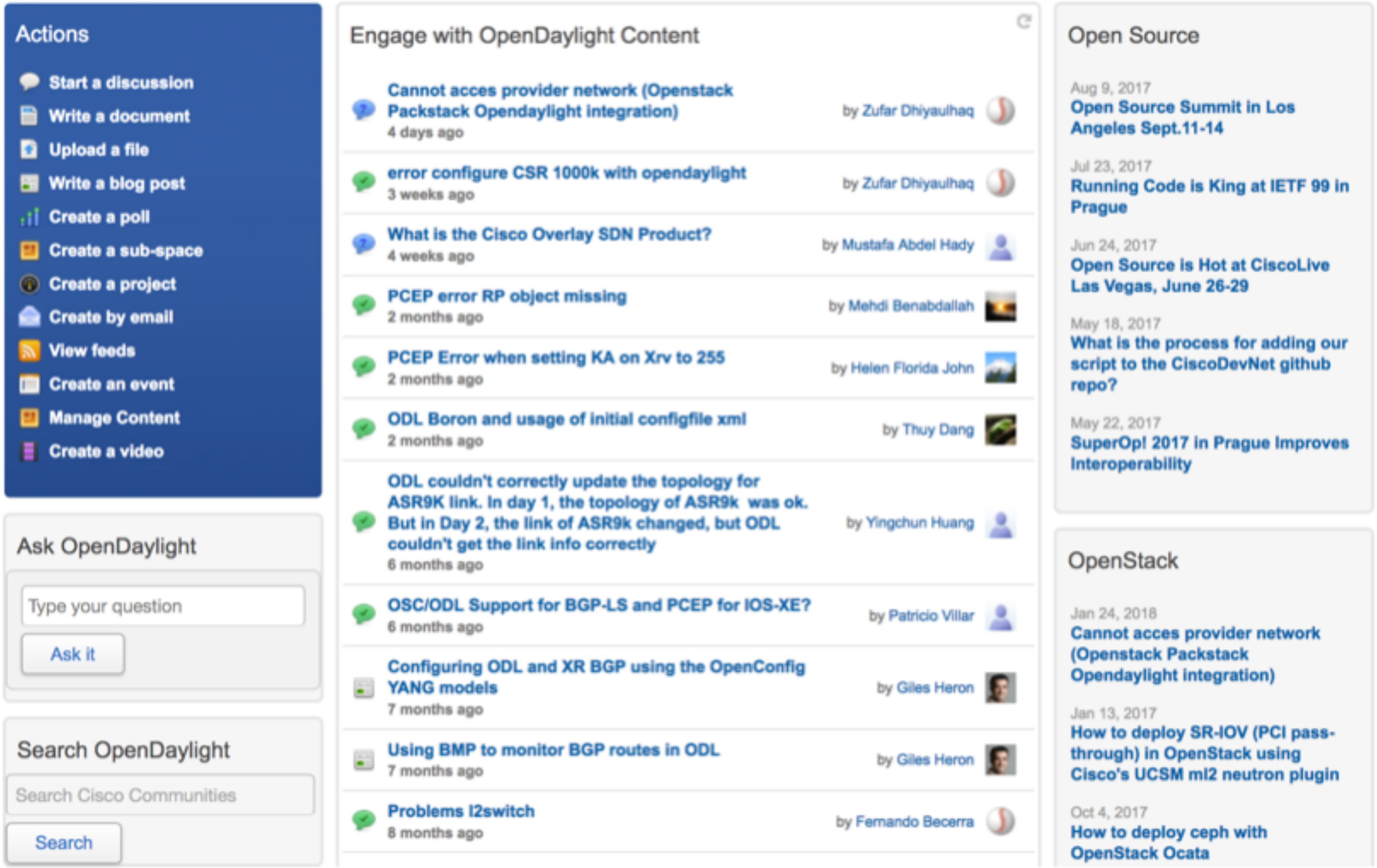
Overview

OpenDaylight (ODL) is a collaborative, open-source project used to advance software-defined networking (SDN). OpenDaylight is a community-led, industry-supported framework consisting of code and blueprints. Using this framework, you can accelerate process adoption, foster innovation, reduce risk, and create a more transparent approach to SDN. OpenDaylight can be a core component within any SDN architecture. Building on open-source SDN and NFV controllers enables users to reduce operational complexity, extend the life of their existing infrastructure hardware, and enable new services and capabilities only available with SDN.

Scenarios

- Scenario 1: Explore ODL Features
- Scenario 2: Explore DLUX
- Scenario 3: Install BGP Pathman Application
- Scenario 4: Enable OpenFlow in Karaf
- Scenario 5: Install OpenFlow Manager Application
- Scenario 6: Explore Pathman Segment Routing
- Scenario 7: Explore netACL Application
- Scenario 8: Explore Yangman

https://communities.cisco.com/community/developer/opendaylight



Actions

- Start a discussion
- Write a document
- Upload a file
- Write a blog post
- Create a poll
- Create a sub-space
- Create a project
- Create by email
- View feeds
- Create an event
- Manage Content
- Create a video

Ask OpenDaylight

Type your question

Ask it

Search OpenDaylight

Search Cisco Communities

Search

Engage with OpenDaylight Content

Cannot acces provider network (Openstack Packstack Opendaylight integration)
4 days ago — by Zufar Dhiyaulhaq

error configure CSR 1000k with opendaylight
3 weeks ago — by Zufar Dhiyaulhaq

What is the Cisco Overlay SDN Product?
4 weeks ago — by Mustafa Abdel Hady

PCEP error RP object missing
2 months ago — by Mehdi Benabdallah

PCEP Error when setting KA on Xrv to 255
2 months ago — by Helen Florida John

ODL Boron and usage of initial configfile xml
2 months ago — by Thuy Dang

ODL couldn't correctly update the topology for ASR9K link. In day 1, the topology of ASR9k was ok. But in Day 2, the link of ASR9k changed, but ODL couldn't get the link info correctly
6 months ago — by Yingchun Huang

OSC/ODL Support for BGP-LS and PCEP for IOS-XE?
6 months ago — by Patricio Villar

Configuring ODL and XR BGP using the OpenConfig YANG models
7 months ago — by Giles Heron

Using BMP to monitor BGP routes in ODL
7 months ago — by Giles Heron

Problems l2switch
8 months ago — by Fernando Becerra

Open Source

Aug 9, 2017
Open Source Summit in Los Angeles Sept.11-14

Jul 23, 2017
Running Code is King at IETF 99 in Prague

Jun 24, 2017
Open Source is Hot at CiscoLive Las Vegas, June 26-29

May 18, 2017
What is the process for adding our script to the CiscoDevNet github repo?

May 22, 2017
SuperOp! 2017 in Prague Improves Interoperability

OpenStack

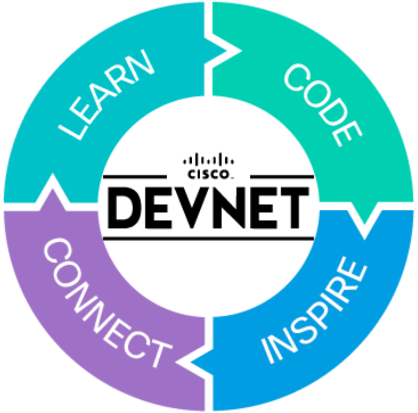
Jan 24, 2018
Cannot acces provider network (Openstack Packstack Opendaylight integration)

Jan 13, 2017
How to deploy SR-IOV (PCI pass-through) in OpenStack using Cisco's UCSM ml2 neutron plugin

Oct 4, 2017
How to deploy ceph with OpenStack Ocata

Continue Your Education

- Become a DevNet Member:
  - https://developer.cisco.com/join
- Access OpenDaylight resources
  - https://developer.cisco.com/site/opendaylight/
- Visit our Open Source Dev Center: https://developer.cisco.com/site/opensource/



Thank you!