Getting Started with OpenDaylight

Charles Eckel, Open Source Developer Evangelist
Giles Heron, Principal Engineer, CTAO

FOSDEM 2017 – SDN / NFV Dev Room

Agenda

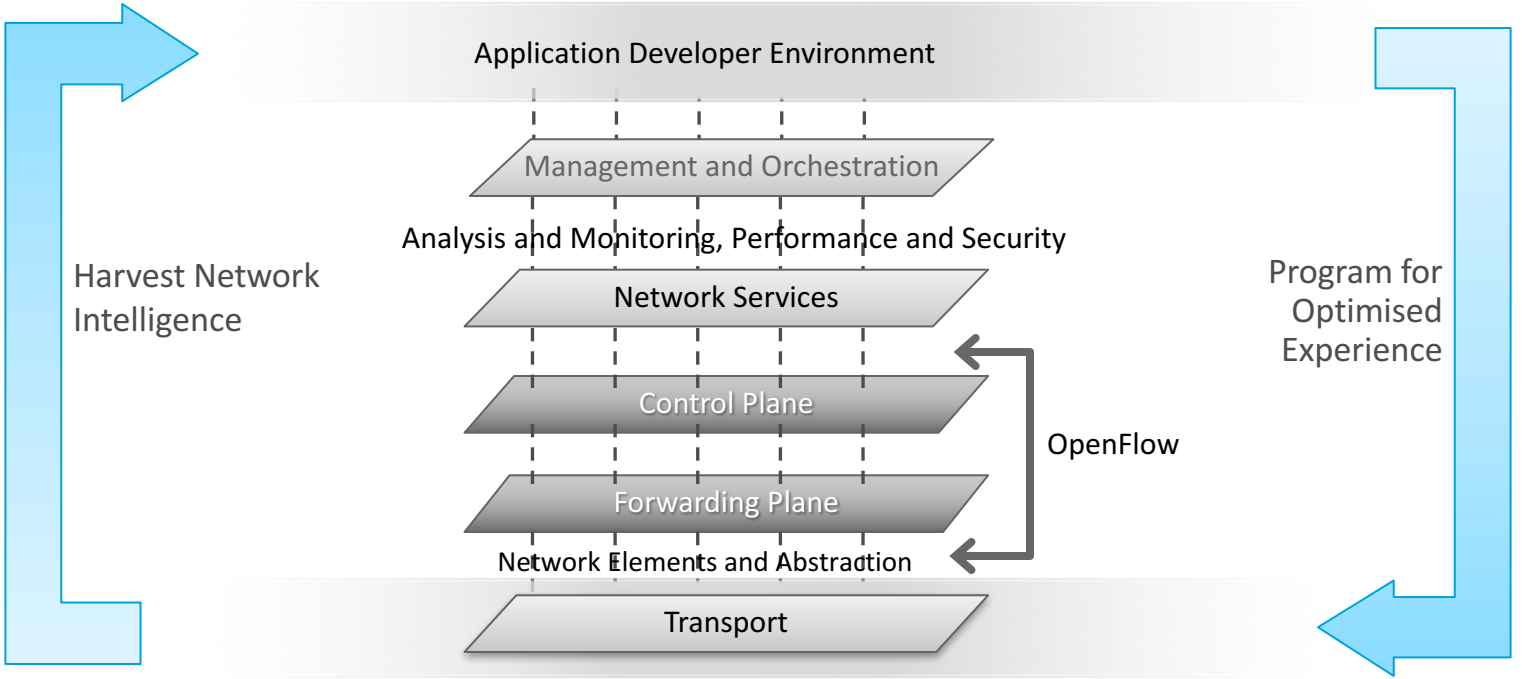
- What is SDN?
- What is OpenDaylight?
- Installation
- Example Use Cases
- Additional resources

What is SDN?

Software Defined Networking (SDN)

- Control & Data Planes separation?
  - OpenFlow?
  - Logically centralised 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 other useful data...
  - ..that can be utilised by variety of SDN applications
- True power of SDN is network programmability

SDN - A Broader Definition



Generic feedback/control/policy loop between apps and the network

What Do We Expect from an SDN Controller?

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

NOTE – we didn't use the word "OpenFlow"

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

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)

Model Driven Service Abstraction Layer/Core (MD-SAL)

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)

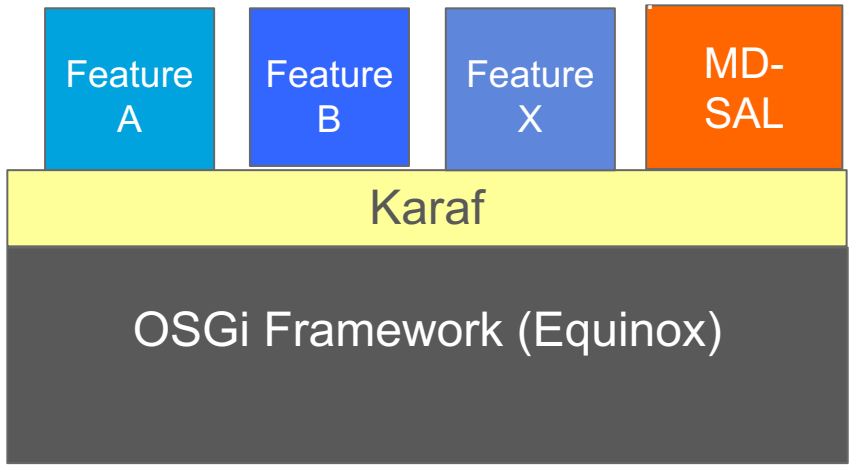
The OpenDaylight Community

- Founded in February 2013
- Run by the Linux Foundation
- Eclipse Public License
- 15 founding companies provided software and developers
- 600+ contributors
- 2.5M+ lines of code
- Mostly Java

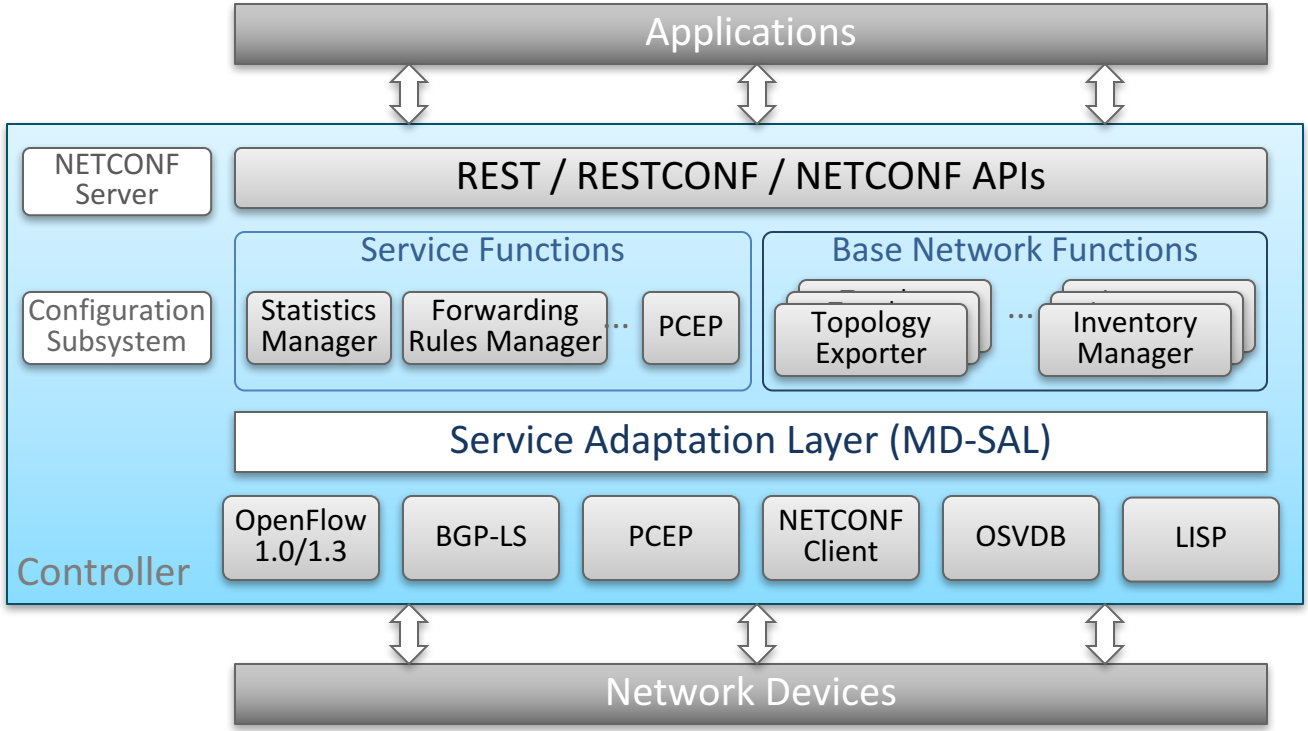
- First release “Hydrogen”
  - February 2014
- Releases roughly every 8 months
- Current stable release - “Boron”
  - “Boron-SR2” released 2 Dec, 2016
- Next release is Carbon
  - Target is 5 May, 2017

Software Architecture

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



OpenDaylight as an SDN Controller

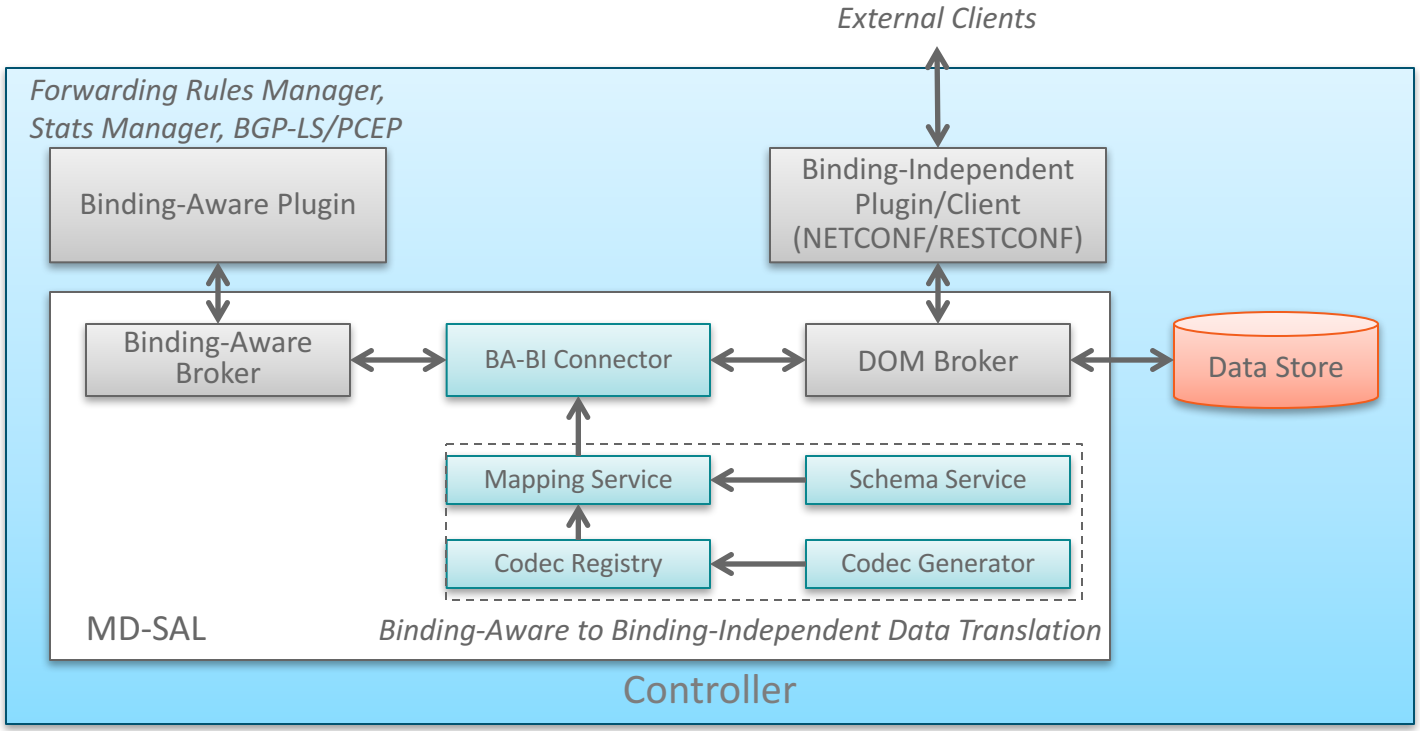


Network Applications Orchestration & Services

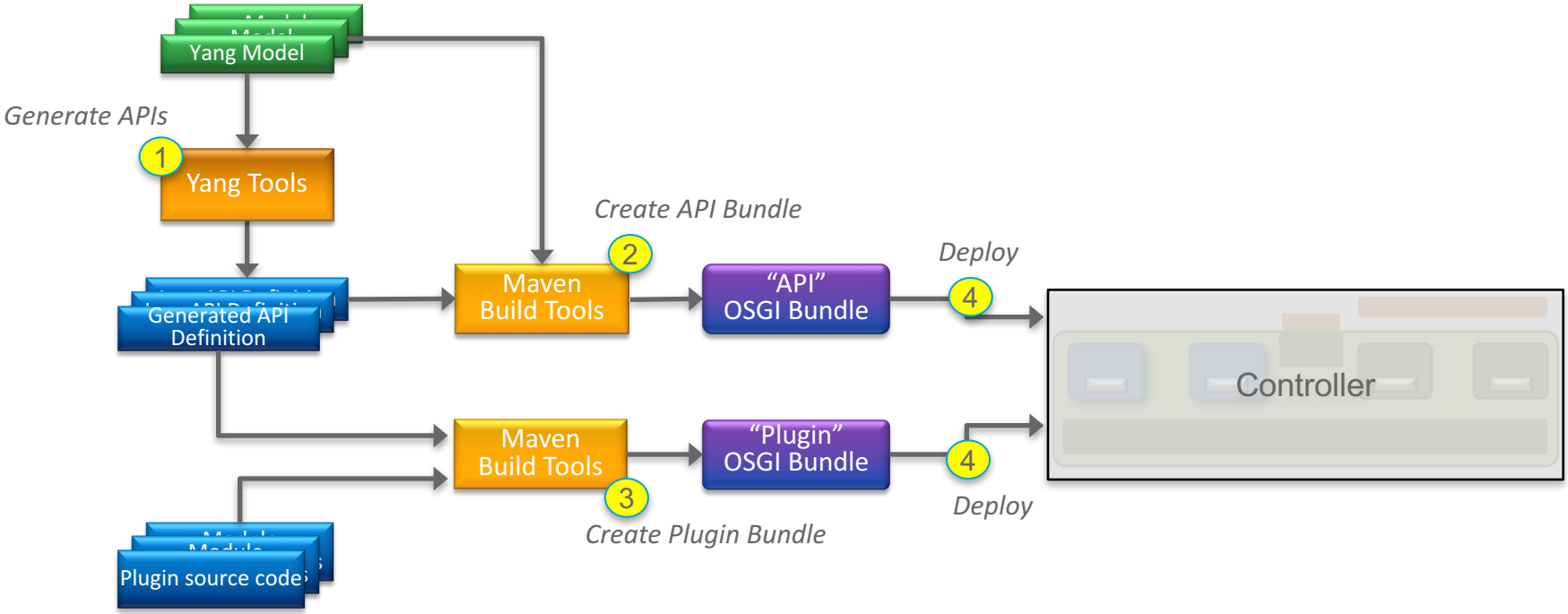
Controller Platform

Southbound Interfaces & Protocol Plugins

MD-SAL Details



Building a Plugin/Application



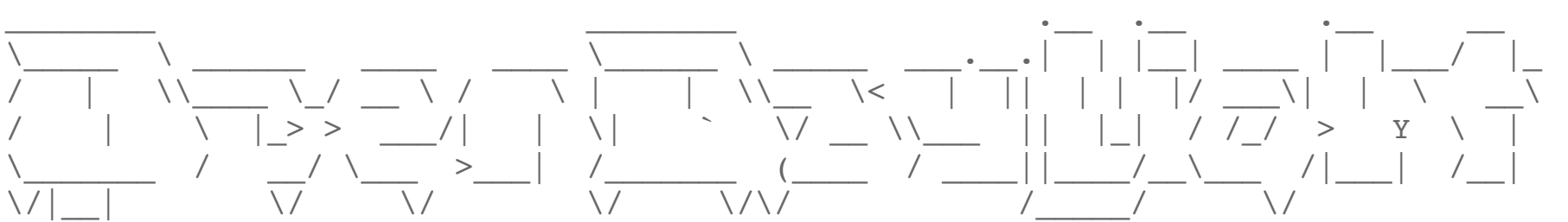
Installation

Distributions
https://www.opendaylight.org/downloads

Release	Edition	Version	Release date	Downloads	Virtual Machines	Documentation	Additional Downloads
Boron-SR2	n/a	n/a	December 20, 2016	Pre-Built Tar Pre-Built Zip Virtual Tenant Network (VTN) Coordinator NeXt UI Toolkit OpFlex		Getting Started Guide Developers Guide User Guide Installation Guide Using OpenDaylight with OpenStack Release Notes	Additional Downloads
Beryllium-SR4	N/A	N/A	October 21, 2016	Pre-built zip file Pre-built tar file NeXt UI Toolkit OpFlex Virtual Tenant Network (VTN) Coordinator		Getting Started Guide Developers Guide User Guide Installation Guide Using OpenDaylight with OpenStack TSC Approval Release Notes	Additional Downloads

Release Archives

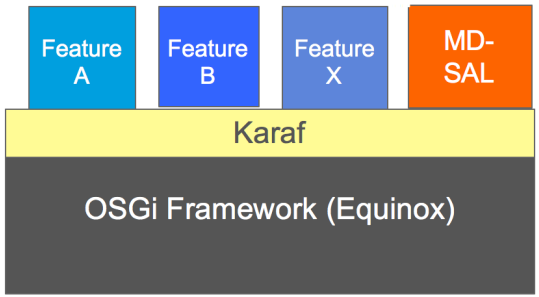
```
$ unzip distribution-karaf-0.5.2-Boron-SR2.zip
Archive:  distribution-karaf-0.5.2-Boron-SR2.zip
   creating: distribution-karaf-0.5.2-Boron-SR2 …
$ cd distribution-karaf-0.5.2-Boron-SR2/
$ ./bin/karaf
karaf: Enabling Java debug options: -Xdebug -Xnoagent -Djava.compiler=NONE …
Listening for transport dt_socket at address: 5005
Apache Karaf starting up. Press Enter to open the shell now...
100% [========================================================================]

Karaf started in 3s. Bundle stats: 64 active, 64 total
```



```
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: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
    - HOWEVER, it will remain installed until you shutdown and restart with *./bin/karaf clean*



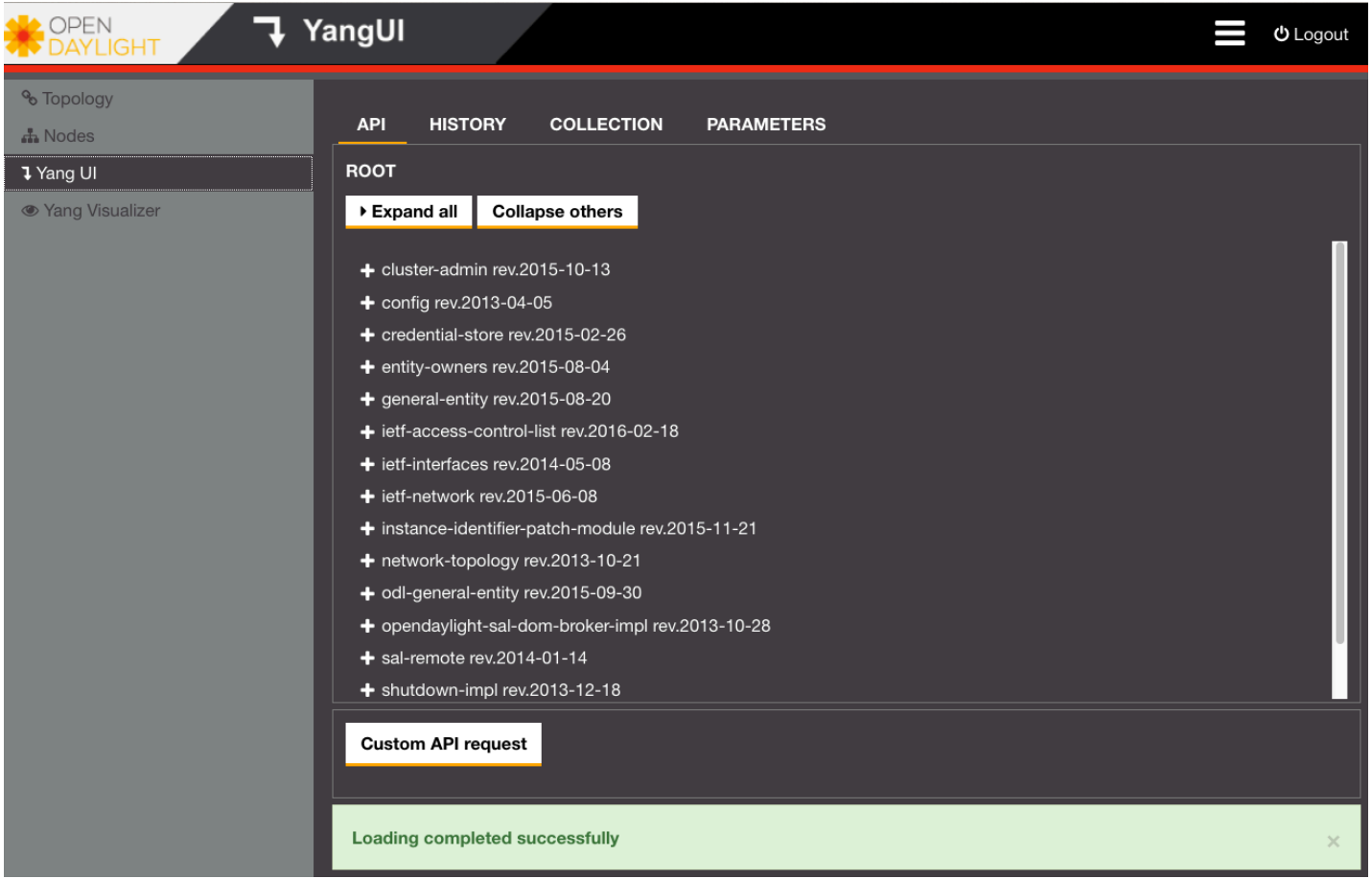
OpenDaylight User Interface - DLUX

```
opendaylight-user@root>feature:list -i
Name       | Version | Installed | Repository     | Description
-------------------------------------------------------------------------------
standard   | 3.0.7   | x         | standard-3.0.7 | Karaf standard feature
config     | 3.0.7   | x         | standard-3.0.7 | Provide OSGi ConfigAdmin
region     | 3.0.7   | x         | standard-3.0.7 | Provide Region Support
package    | 3.0.7   | x         | standard-3.0.7 | Package commands and mbeans
kar        | 3.0.7   | x         | standard-3.0.7 | Provide KAR (KARaf archive)
ssh        | 3.0.7   | x         | standard-3.0.7 | Provide a SSHd server on Karaf
management | 3.0.7   | x         | standard-3.0.7 | Provide a JMX MBeanServer
```

Install DLUX Feature

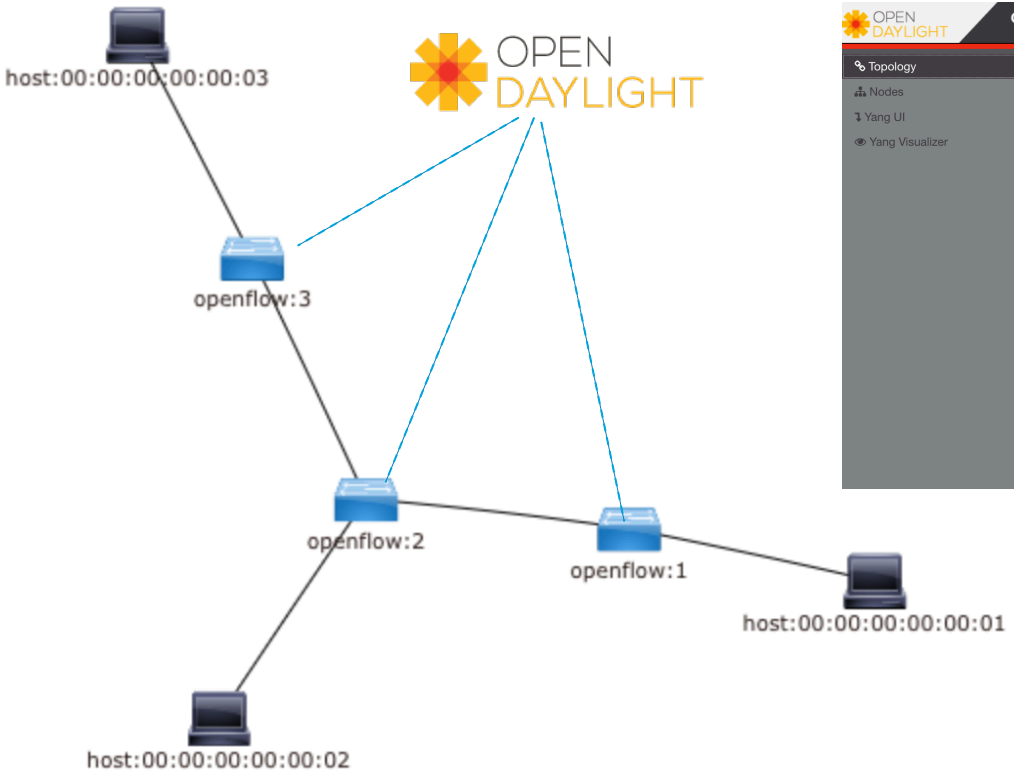
```
opendaylight-user@root>feature:list | grep dlux
odl-snbi-dlux            | 1.3.2-Boron-SR2  |           | odl-snbi-1.3.2-Boron-SR2
odl-dlux-all             | 0.4.2-Boron-SR2  |           | odl-dlux-0.4.2-Boron-SR2
odl-dlux-core            | 0.4.2-Boron-SR2  |           | odl-dlux-0.4.2-Boron-SR2
odl-dlux-node            | 0.4.2-Boron-SR2  |           | odl-dlux-0.4.2-Boron-SR2
odl-dlux-yangui          | 0.4.2-Boron-SR2  |           | odl-dlux-0.4.2-Boron-SR2
odl-dlux-yangman         | 0.4.2-Boron-SR2  |           | odl-dlux-0.4.2-Boron-SR2
odl-dlux-yangvisualizer | 0.4.2-Boron-SR2  |           | odl-dlux-0.4.2-Boron-SR2

opendaylight-user@root>feature:install odl-dlux-core

opendaylight-user@root>feature:list -i | grep dlux
odl-dlux-core            | 0.4.2-Boron-SR2  | x         | odl-dlux-0.4.2-Boron-SR2
odl-dlux-node            | 0.4.2-Boron-SR2  | x         | odl-dlux-0.4.2-Boron-SR2
odl-dlux-yangui          | 0.4.2-Boron-SR2  | x         | odl-dlux-0.4.2-Boron-SR2
odl-dlux-yangvisualizer | 0.4.2-Boron-SR2  | x         | odl-dlux-0.4.2-Boron-SR2
```

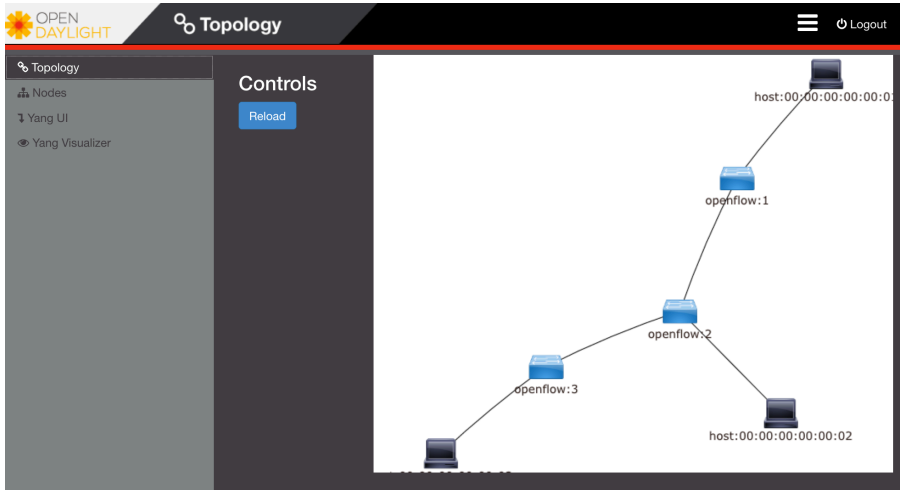
http://localhost:8181/index.html#/yangui/index



Example Use Cases

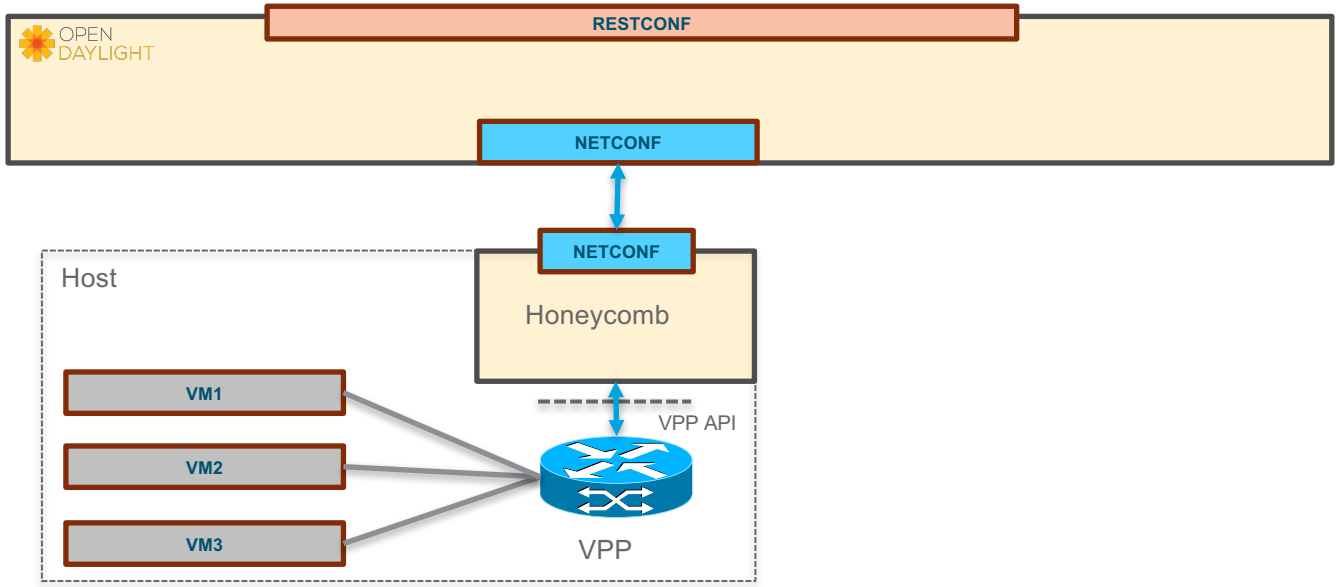
OpenDaylight with Mininet, OVSDB and OpenFlow





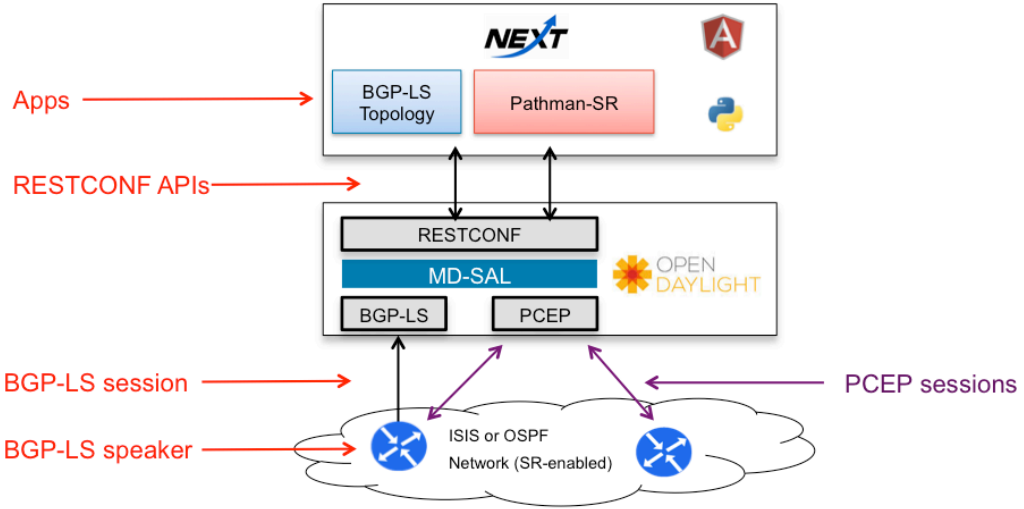
Honeycomb/VPP using NETCONF

- VPP is a high-performance software forwarder (see http://www.fd.io)
- Honeycomb provides NETCONF management for VPP

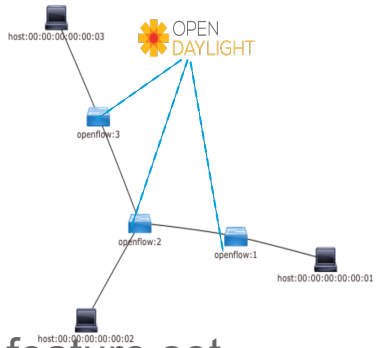


Cisco IOS XR using BGP-LS and PCE-P

- Cisco XRv topology in dCloud
  - dCloud is http://dcloud.cisco.com (requires CCO login)
  - “OpenDaylight Boron SR2 with Apps with 8 Nodes v1”
  - ODL runs in dCloud (or use anyconnect/openconnect VPN to use local ODL instance)
  - http://github.com/CiscoDevNet/opendaylight-setup
- Use Pathman-SR application to create Segment Routed LSPs
  - http://github.com/CiscoDevNet/pathman-sr



OpenDaylight with Mininet – Step by Step



- Install, setup, and start Mininet VM using VirtualBox
  - Great instructions at http://www.brianlinkletter.com/set-up-mininet/
- In one xterm, start OpenDaylight (151.216.133.238*), enable required feature set
  - `opendaylight-user@root> feature:install odl-restconf odl-l2switch-switch odl-mdsal-apidocs odl-dlux-all`
- In second xterm, connect to Mininet VM (192.168.58.102*)
  - `ssh -X mininet@192.168.58.102` (password mininet)
- Start 3 switches, each with one host and controlled by OpenDaylight
  - `mininet@mininet-vm:~$ sudo mn --topo linear,3 --mac --controller=remote,ip=151.216.133.238,port=6633 --switch ovs,protocols=OpenFlow13`
- From Browser, log into OpenDaylight DLUX
  - http://151.216.133.238:8181/index.html (credentials: admin/admin)

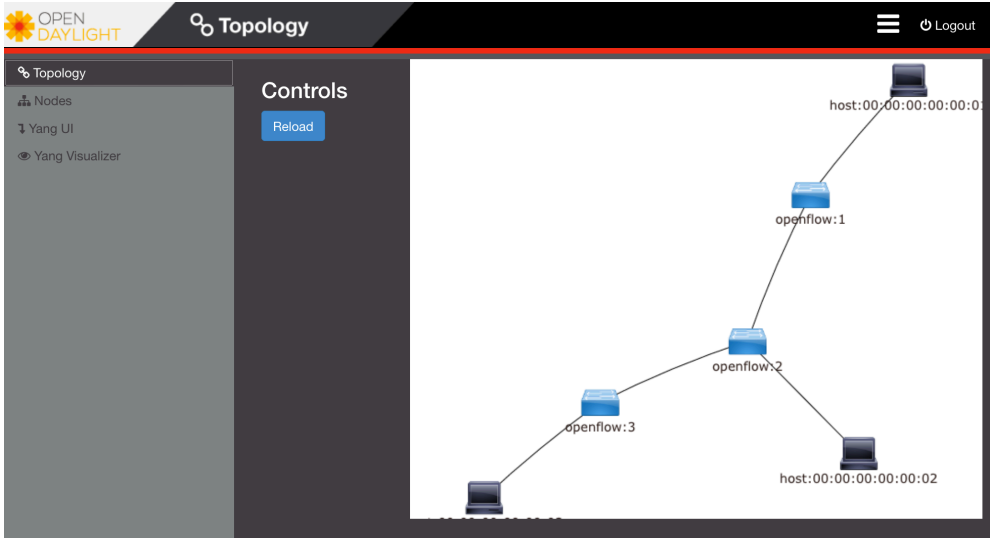
* The IP addresses in your setup will likely be different. You can find the IP address using “ifconfig” or “ip addr show”

Mininet Network Start

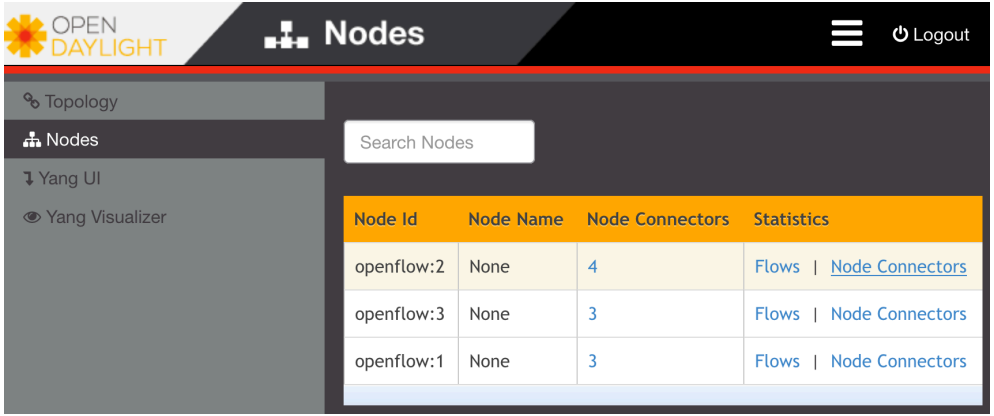
```
mininet@mininet-vm:~$ sudo mn --topo linear,3 --mac --controller=remote,ip=192.168.40.18,
port=6633 --switch ovs,protocols=OpenFlow13
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1 s2 s3
*** Adding links:
(h1, s1) (h2, s2) (h3, s3) (s2, s1) (s3, s2)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 3 switches
s1 s2 s3 ...
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3
h2 -> h1 h3
h3 -> h1 h2
*** Results: 0% dropped (6/6 received)
mininet>
```

Using DLUX

- From Browser, log into OpenDaylight DLUX
  - http://151.216.133.238:8181/index.html (credentials: admin/admin)

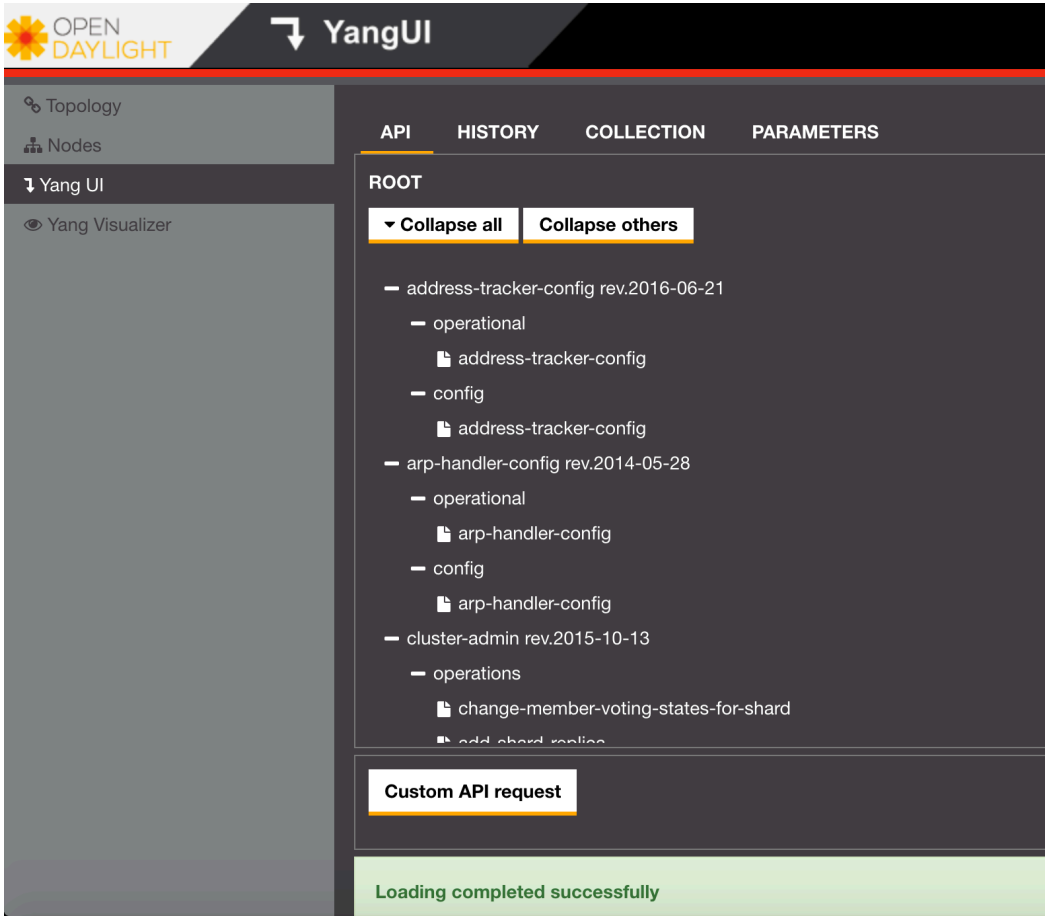


- Check out the network and switches by clicking on *Nodes*, *Node Connectors*



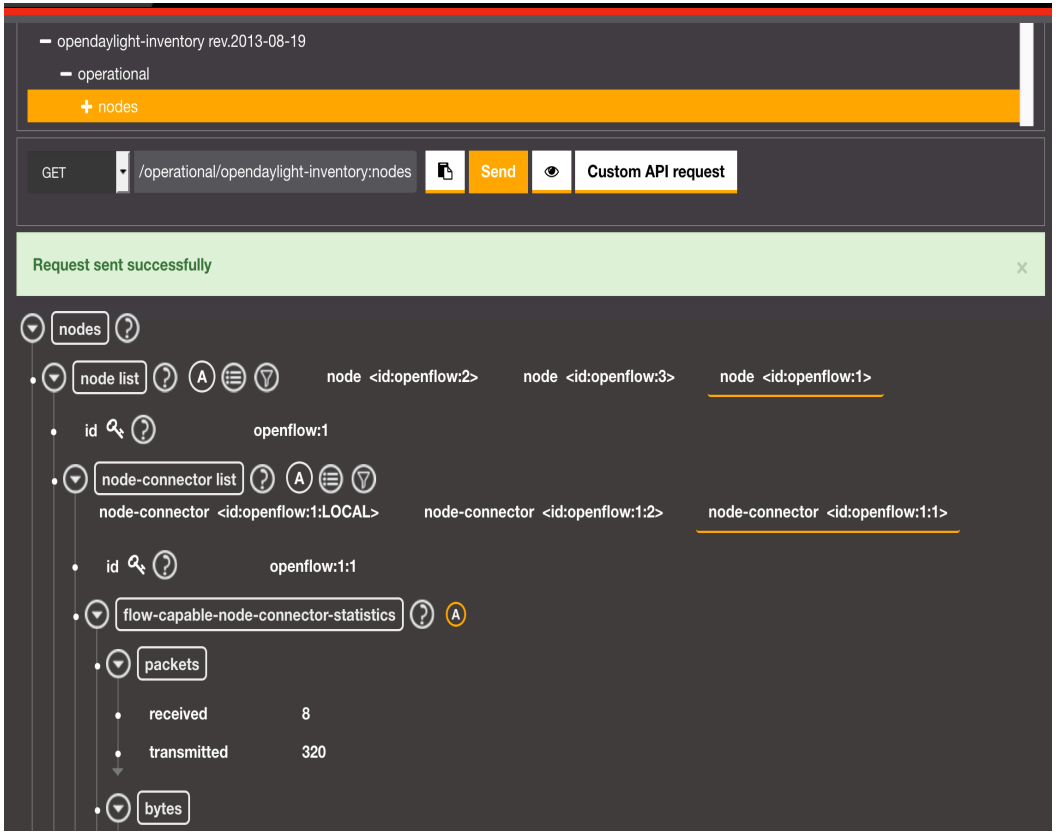
REST APIs

- Click on *Yang UI* and *Expand All* to see all the REST APIs available



Inventory of Network Nodes

- GET opendaylight-inventory -> operational -> nodes

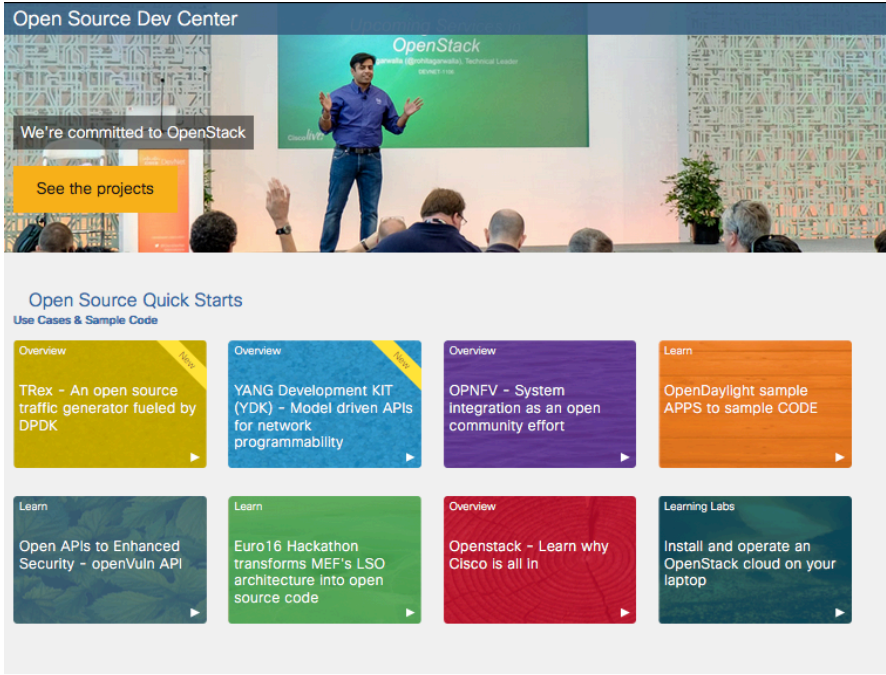


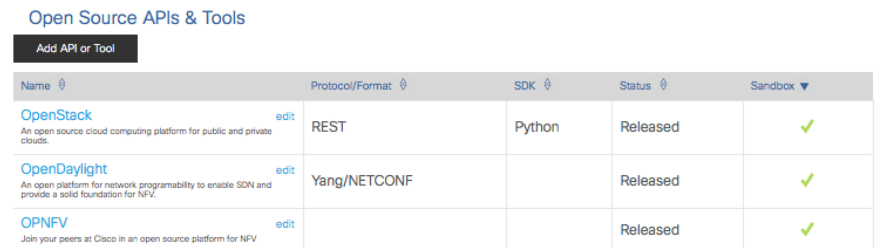
Additional resources

Open Source Dev Center

Your Source for Open Source at Cisco
https://developer.cisco.com/opensource

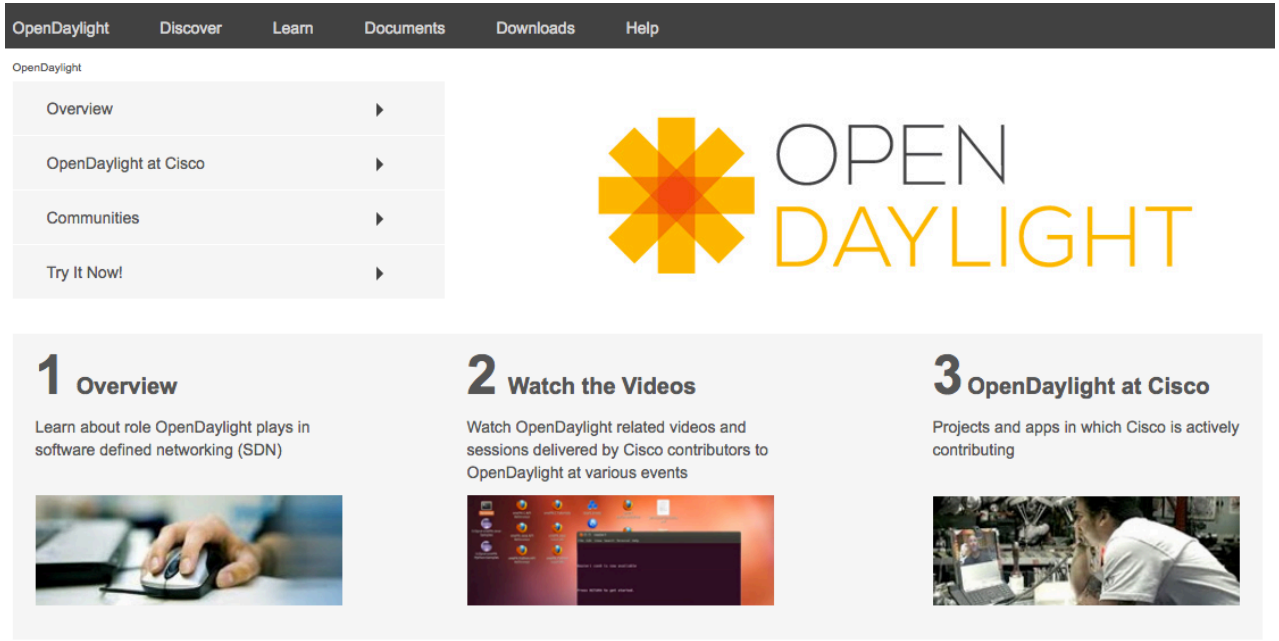
- Contributions to open source
- Use in products/solutions
- Community forums, blogs
- Developer Events
  - IETF Hackathons and MEF LSO Hackathons featuring open source implementations of open standards

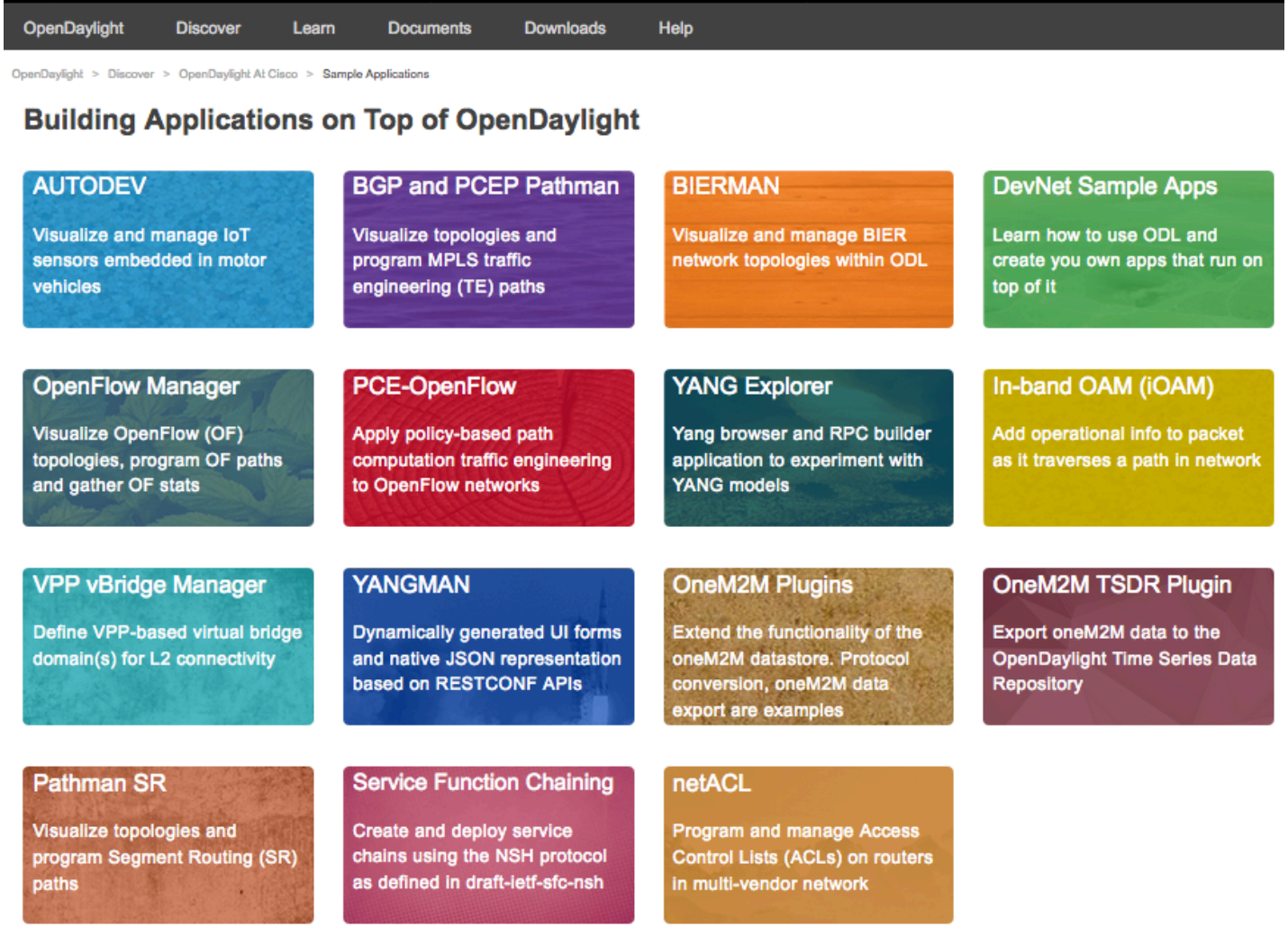




OpenDaylight Microsite

https://developer.cisco.com/opendaylight





Tutorials and Sandboxes

OpenDaylight Boron SR2 with Apps with 16 Nodes v1

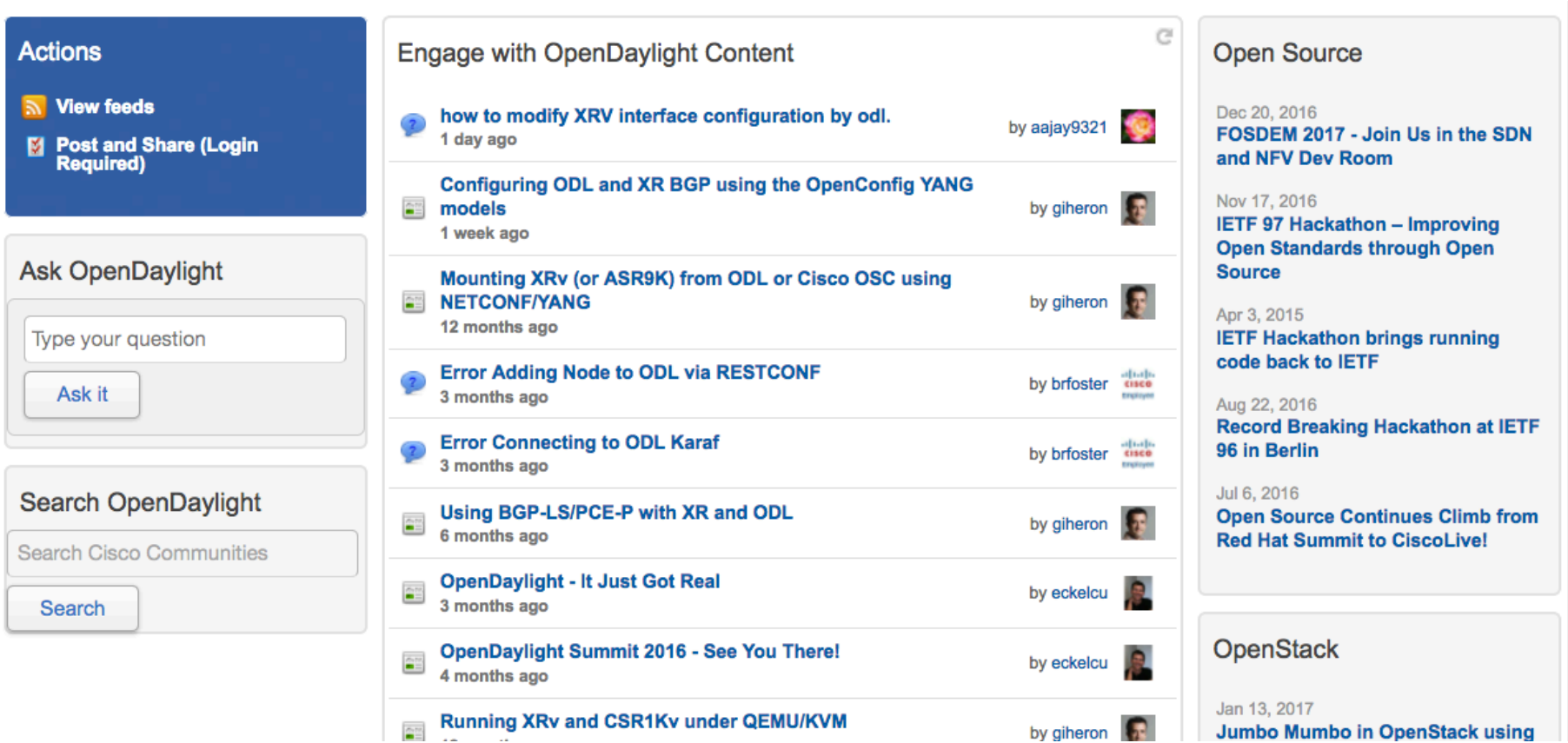
Overview

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



Questions?

Thank You