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

Application Developer Environment

Management and Orchestration

Analysis and Monitoring, Performance and Security

Network Services

Control Plane

Forwarding Plane

Network Elements and Abstraction

Transport

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

Harvest Network Intelligence

Program for Optimised Experience

OpenFlow

FOSDEM - Getting Started with OpenDaylight
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?
**OpenDaylight APIs**

<table>
<thead>
<tr>
<th>REST/RESTCONF/NETCONF/AMQP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Model Driven Service Abstraction</strong></td>
</tr>
<tr>
<td><strong>Layer/Core (MD-SAL)</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>OpenFlow Enabled Devices</strong></td>
</tr>
<tr>
<td><strong>Open vSwitches</strong></td>
</tr>
<tr>
<td><strong>Additional Virtual &amp; Physical Devices</strong></td>
</tr>
<tr>
<td><strong>Controller Platform Services/Applications</strong></td>
</tr>
<tr>
<td><strong>Data Plane Elements</strong></td>
</tr>
</tbody>
</table>

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

**AAA AuthN Filter**

**Data Plane Elements (Virtual Switches, Physical Device Interfaces)**

**Southbound Interfaces & Protocol Plugins**

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

**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 Hdr
- Controller Shield
- Dev Discovery, ID & Drvr Mgmt
- DOCSIS Abstraction
- Link Aggregation Ctrl 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.

**Data Store (Config & Operational)**

**Messaging (Notifications / RPCs)**

**OpenFlow Stats Manager**

**OpenFlow Forwarding Rules Mgr**

**OpenFlow Switch Manager**

**Topology Processing**

**FOSDEM - Getting Started with OpenDaylight**

All rights reserved. Cisco Public
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

Applications

Network Applications Orchestration & Services

Controller Platform

Southbound Interfaces & Protocol Plugins

Controller

Service Functions

Service Adaptation Layer (MD-SAL)

Base Network Functions

Network Devices

NETCONF Server

Configuration Subsystem

REST / RESTCONF / NETCONF APIs

NETCONF

Client

Network Applications Orchestration & Services

Controller Platform

Southbound Interfaces & Protocol Plugins

Service Functions

Statistics Manager

Forwarding Rules Manager

PCEP

Base Network Functions

Topology Exporter

Inventory Manager

NETCONF

Client

OSVDB

LISP

Service Adaptation Layer (MD-SAL)

Controller

OpenFlow 1.0/1.3

BGP-LS

PCEP

NETCONF

Client

OSVDB

LISP

Network Devices

Applications
MD-SAL Details

Forwarding Rules Manager, Stats Manager, BGP-LS/PCEP

Binding-Aware Plugin

Binding-Aware Broker

BA-BI Connector

DOM Broker

Mapping Service

Schema Service

Codec Registry

Codec Generator

Binding-Independent Plugin/Client (NETCONF/RESTCONF)

Data Store

External Clients

Controller

Binding-Aware to Binding-Independent Data Translation

MD-SAL

12

FOSDEM - Getting Started with OpenDaylight

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Public
Building a Plugin/Application

1. **Yang Tools** → Generate APIs
   - Yang Model
   - Generated API Definition
   - Plugin source code

2. **Maven Build Tools** → Create API Bundle
   - “API” OSGI Bundle
   - Deploy

3. **Maven Build Tools** → Create Plugin Bundle
   - “Plugin” OSGI Bundle
   - Deploy

4. Deploy Controller

---

FOSDEM - Getting Started with OpenDaylight

© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Public
Installation
### Distributions

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

<table>
<thead>
<tr>
<th>Release</th>
<th>Edition</th>
<th>Version</th>
<th>Release date</th>
<th>Downloads</th>
<th>Virtual Machines</th>
<th>Documentation</th>
<th>Additional Downloads</th>
</tr>
</thead>
</table>

**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

```bash
opendaylight-user@root>feature:list -i
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Version</th>
<th>Installed</th>
<th>Repository</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>standard</td>
<td>3.0.7</td>
<td>x</td>
<td>standard-3.0.7</td>
<td>Karaf standard feature</td>
</tr>
<tr>
<td>config</td>
<td>3.0.7</td>
<td>x</td>
<td>standard-3.0.7</td>
<td>Provide OSGi ConfigAdmin</td>
</tr>
<tr>
<td>region</td>
<td>3.0.7</td>
<td>x</td>
<td>standard-3.0.7</td>
<td>Provide Region Support</td>
</tr>
<tr>
<td>package</td>
<td>3.0.7</td>
<td>x</td>
<td>standard-3.0.7</td>
<td>Package commands and mbeans</td>
</tr>
<tr>
<td>kar</td>
<td>3.0.7</td>
<td>x</td>
<td>standard-3.0.7</td>
<td>Provide KAR (KARaf archive)</td>
</tr>
<tr>
<td>ssh</td>
<td>3.0.7</td>
<td>x</td>
<td>standard-3.0.7</td>
<td>Provide a SSHd server on Karaf</td>
</tr>
<tr>
<td>manage</td>
<td>3.0.7</td>
<td>x</td>
<td>standard-3.0.7</td>
<td>Provide a JMX MBeanServer</td>
</tr>
</tbody>
</table>
Install DLUX Feature

```bash
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
```

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

```bash
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](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)

- Use Pathman-SR application to create Segment Routed LSPs
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

* 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

```bash
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

• 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

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

Blog

Configuring ODL and XR BGP using the OpenConfig YANG models

© 2017 Cisco and/or its affiliates. All rights reserved.
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 your 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 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

Engage with OpenDaylight Content

- how to modify XRV interface configuration by odl
  1 day ago
  by aajay9321

- Configuring ODL and XR BGP using the OpenConfig YANG models
  1 week ago
  by gheron

- Mounting XRv (or ASR9K) from ODL or Cisco OSC using NETCONF/YANG
  12 months ago
  by gheron

- Error Adding Node to ODL via RESTCONF
  3 months ago
  by brfoster

- Error Connecting to ODL Karaf
  3 months ago
  by brfoster

- Using BGP-LS/PCE-P with XR and ODL
  6 months ago
  by gheron

- OpenDaylight - It Just Got Real
  3 months ago
  by eckelcu

- OpenDaylight Summit 2016 - See You There!
  4 months ago
  by eckelcu

- Running XRv and CSR1Kv under QEMU/KVM
  by gheron

Open Source

- Dec 20, 2016
  FOSDEM 2017 - Join Us in the SDN and NFV Dev Room

- Nov 17, 2016
  IETF 87 Hackathon – Improving Open Standards through Open Source

- Apr 3, 2015
  IETF Hackathon brings running code back to IETF

- Aug 22, 2016
  Record Breaking Hackathon at IETF 96 in Berlin

- Jul 6, 2016
  Open Source Continues Climb from Red Hat Summit to CiscoLive!

OpenStack

- Jan 13, 2017
  Jumbo Mumbo in OpenStack using

FOSDEM - Getting Started with OpenDaylight
Questions?
Thank You