

Software Defined Networking with OpenDaylight (LFS465)

Modality: Virtual Classroom

Duration: 2 Days

About this course:

SDN (Software Defined Networking) abstracts networking infrastructure away from the actual physical equipment. This allows network system administrators to maintain the networking environment across multiple vendors and hardware, and operating systems and versions. This course is designed to provide Developers and advanced Network Engineers exposure to modeling in Open Source Software Defined Networking (SDN). We first discuss the context of SDN components then progress to show the code created through modeling in OpenDaylight. This course concludes by writing an application from scratch.

OpenDaylight is an open networking platform that enables SDN and constructs a solid foundation for NFV (Network Functions Virtualization) for all network sizes.

Course Objective:

After completing the course, students will:

- Have a solid understanding of SDN and the protocols, tools and methods used.
- Understand the openDayLight SDN controller with its APIs and how it fits into SDN.
- Write applications on-top of the OpenDaylight APIs.
- Gained understanding of the tools to simulate and debug network topologies.

Audience:

- Developers
- Network Engineers

Prerequisite:

This course is designed to provide Developers and advanced Network Engineers exposure to modeling in Open Source Software Defined Networking (SDN). Students should have a good grasp of typical scalability and system administration issues frequently encountered in enterprise environments. They should also be well versed in Linux command line usage, shell scripting and text file editing.

Course Outline:

Introduction

- Objectives and Goals
- Audience
- The Linux Foundation
- Linux Foundation Training Offerings
- Course Platform: Ubuntu 16.04
- Course Procedures
- Course Registration
- Labs

Software Defined Networking

- Linux Networking and SDN
- Networking Primer
- Data and Control Plane in SDN
- Networking components in SDN
- Knowledge Check

Open vSwitch **

- Introduction
- Open vSwitch Components
- Open vSwitch installation
- Using Open vSwitch
- Knowledge Check
- Labs **

Simulation and Observation

- Objectives
- Mininet
- Mininet Command Line
- Mininet Python API
- Wireshark
- Knowledge Check
- Labs

SDN – History and Evolution

- Early Networking
- Datacenters and Network Operation
- Evolution of Network Programmability
- Knowledge Check

Network Programmability

- TELNET/CLI
- SNMP
- NETCONF
- YANG
- Knowledge Check

OpenFlow

- Introduction
- OpenFlow Basics and Versions
- OpenFlow Protocol
- Knowledge Check
- Labs

Network Virtualization and Multi-tenancy

- Virtualization in the Datacenter and in the Network
- Multi-Tenancy
- OF-Config
- Knowledge Check

Introduction to OpenDaylight

- The OpenDaylight Project
- Components of OpenDaylight
- Project resources
- Knowledge Check
- Labs

YANG

- YANG
- yangtools
- YANG to Java mapping
- YANG in OpenDaylight
- Knowledge Check
- Labs

Apache Karaf – the OSGi container

- OSGi
- Apache Karaf Project
- OpenDaylight and Karaf
- Using Karaf
- Karaf settings for OpenDaylight
- Knowledge Check

- Labs

OpenDaylight Controller and MD-SAL

- How the controller evolved
- MD-SAL
- Brokers and RPC-Calls
- The Datastore
- Clustering MD-SAL
- Plugin Development Workflow
- Development environment setup
- Knowledge Check
- Labs

Eclipse Setup and Importing OpenDaylight

- Eclipse
- Knowledge Check
- Labs

Observing and Logging OpenDaylight

- Observing
- Logging
- Debugging
- Knowledge Check

Writing an Application using OpenDaylight

- Labs

Closing and Evaluation Survey