## TRAINING GOALS:

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 and Tungsten Fabric. 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.

Tungsten Fabric is an open source network virtualization solution for providing connectivity and security for virtual, containerized or bare-metal workloads.

We discuss both of these platforms in detail:

- **SDN**
  - **OpenDaylight**
  - Tools such as mininet and wireshark
  - Applications using OpenDaylight and Tungsten Fabric APIs

After completing the course, students will:
CONSPECT:

- **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

- 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 as well as the role of Tungsten Fabric.
- Write applications on-top of the OpenDaylight and Tungsten Fabric APIs.
- Gained understanding of the tools to simulate and debug network topologies.
- 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
- Data Acceleration Technologies
  - DPDK and Other Intel EPA features
  - SR-IOV/PCI Passthrough
  - FD.io/VPP
  - IOVisor
  - Labs
- SDN Use Cases
  - Network Slicing
  - Transport Network Optimization
  - Cloud/Edge
  - SD-WAN/vCPE
  - Intra DC Connection
  - Labs
○ Introduction to OpenDaylight
  ○ The OpenDaylight Project
  ○ Components of OpenDaylight
  ○ Project resources
  ○ Knowledge Check
  ○ Labs

○ Introduction to Tungsten Fabric
  ○ The TF Project
  ○ Community; Benefits of TF
  ○ Describe TF Architecture and Overlay Network Principles
  ○ Explain Components of TF
  ○ Service Chaining
  ○ Monitoring
  ○ Multi Tenancy
  ○ Labs

○ Introduction to VM networking
  ○ OpenStack Neutron
  ○ ML-2/L3 Plugins
  ○ Labs

○ Introduction to Container Networking
  ○ CNI
  ○ Calico
  ○ Flannel
  ○ Labs

○ Virtual Networking Service Assurance
  ○ Introduction to OPNFV Calipso
  ○ 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
- TF Architecture Deep Dive
  - TF Architecture Overview
  - TF Control Plane
  - TF Data Plane
  - TF Management
  - TF Basic Troubleshooting
  - TF Security Policy Framework
  - Labs
- TF Configuration
Training: The Linux Foundation
LFS465 Software Defined Networking Essentials

- Configuration Techniques
- vRouter Configuration
- Virtual Networks Configuration
- Network Policy/Security Group Configuration
- TF API
- Remote Edge
- Labs
- TF And External Networks
  - Connecting Virtual and Physical Networks
  - Floating IPs
  - Simple Virtual Gateway Configuration
  - EVPN
  - Labs
- TF Network Services
  - TF and Docker containers
  - Baremetal Workloads
  - BGP-as-a-service
  - LBaaS
  - vRouter Deployment Models (Kernel, DPDK, SRIOV, SmartNic)
  - DNS Server
  - Broadcast/Multicast
  - Device Manager
  - Labs
- Observing and Logging TF
  - Monitoring
  - Logging
  - Analytics
  - Labs
- Closing and Evaluation Survey

REQUIREMENTS:

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.
Difficulty level

CERTIFICATE:

The participants will obtain certificates signed by The Linux Foundation.

TRAINER:

Certified The Linux Foundation Trainer.