TechyEdz Solutions

Training | Consulting | Developement | Outsourcing

Cisco SDN NFV

9032803832

9032803832

contact@techyedz.com

www.techyedz.com

Cisco Software Defined Networking and Network Function Virtualization (SDN & NFV)

Course Overview

In this course, students learn Software Defined Network architecture and the important protocols related to SDN implementations. This course thoroughly explains what SDN is, how it works, and then does a deep dive into the SDN protocols themselves. SDN can both manage and control physical network elements as well as Network Function Virtualization, allowing network professionals to deploy and maintain a clean integration between cloud environments and the physical network itself.

Often we are asked by network personnel to teach them what the network looks like when it enters the cloud. This is why the study of Network Function Virtualization is a natural progression in this type of study, so we have included both SDN and NFV in one course. This course will clarify what happens at the cloud boundary and then look into the virtual network within the cloud itself. If you are already a networking professional and you take a look at what is going on inside the cloud, you will learn that there is no reason not to take all those good ideas and implement them outside the cloud. The networking control layer as you may currently understand it, will change radically with SDN. We will show you that the change is both amazing and powerful.

In this course, you will build, configure, and deploy the most popular network functions, routing, bridging, and OpenFlow switches along with requisite protocols. You will integrate these components with an emulated physical environment and perform verification testing. The cloud environment will be represented with a *very* deep dive into OpenStack Neutron and Neutron-compute.

Course Outline:

1. SDN Introduction

- Southbound Interface and Northbound Interface
    - Controller Southbound Interface (SBI) & Northbound Interface (NBI)
- Data Plane
    - Classic Forwarding Device

- Data Plane

- Control Plane
    - Distributed Control Plane
- Problems with the current distributed Control Plane design
    - Interfacing with the Distributed Control Plane
    - Problems with Distributed Control Plane
- Problems solved by the Centralized Control Plane
    - Clean Interface for new Applications
    - Clean Interface for new Applications
    - Declarative vs Imperative Control
    - What about the Southbound Interface?
- Data Plane
    - Service Chaining
- Management Plane Functions
    - RFC 7426 SDN Layers and Architecture Terminology
- Northbound API Abstractions
    - Northbound API Abstractions
    - Recognizing Cloud Types

2. NFV Practical Application

- Universal Data Center Options
    - Data Center Layout - Basic Cloud Components
    - Data Center Layout – Network Fabric
    - Data Center Layout – NFV Network
    - Data Center Layout – Controller Node
    - Data Center Layout – Network Node
    - Data Center Layout – Compute Nodes
    - Data Center Layout – Storage Nodes
    - A Data Center Rack - Generic!
    - Compute Node Functions
- Cisco Data Center Options
    - A Data Center Rack according to Cisco ACI
    - Data Center Layout - Cisco ACI
- NSX VMware Data Center Options
    - A Data Center Rack according to NSX (VMware)
    - Data Center Layout - NSX Vmware + Cisco-driven Fabric

- OpenStack Data Center Options
  - A Data Center Rack - Openstack
  - Data Center Layout - Openstack

3. NFV

- NFV Terminology
  - NFV Terminology
- NFV Architecture
  - ETSI NFV ISG Interfaces and Architecture IFA WG
  - Network Functions Virtualization: VNF, Network Service and E2E Network Service
  - Network Functions Virtualization: Management of NFV Components
  - Management and Orchestration: Architecture
  - Virtualized Infrastructure Manager (VIM)
  - VNF Manager (VNFM)
  - NFV Orchestrator (NFVO)
  - VNF Forwarding Graph and Network Forwarding Path on top of a Network Service
  - Base Information Elements
- NFV Reference Points
  - MANO Architectural Framework- Reference Points and Interfaces
- Service Function Chaining Architecture (RFC 7665)
  - Service Chaining

4. NFV Commands

- net-tools vs iproute2
  - net-tools (Legacy) vs iproute2 (NFV friendly)
- iproute2
  - iproute2 Package Commands
  - Linux Container Building Blocks
- Linux Network Devices
  - Linux Network Devices Used in this Course
  - Linux Network Devices Basics – Linux Bridge
  - OVSwitch
  - TAP (1 of 2)

        - Other search expression
        - tcpdump Essentials
        - BPF Berkley Packet Filter Primer
- Troubleshooting
    - a3diff
    - ip address vs. ip link

5. OpenFlow

- OpenDaylight Soutbound APIs
    - OpenFlow Interface
- Active Networking
    - Active Networking
- ForCES Architecture
    - ForCES Architecture
    - ForCES Architecture- FE Model
- Clean Slate
    - OpenFlow
- Layers - API vs Control vsInfrastructure
    - OpenFlow in a SDN Architecture
- Switch Specification
    - OpenFlow Switch Specification
- Linux Installation and Deployment
    - Installed on a Linux Machine using x86 Hardware
- Components
    - What is OpenFlow?
- Main Components the Switch and Controller
    - Main Components of an OpenFlow Switch
    - Open Source Controllers
- Traditional L2
    - The MAC Address
    - An Ethernet Access
    - The Ethernet Link
    - Ethernet Broadcast Domain
    - The Source and Destination IP Addresses
    - Referencing the Host Routing Table
    - Ethernet Broadcast Domain
    - Ethernet Switch MAC Address Learning

  - Ethernet Broadcast Domain
  - ARP Response
  - ARP Response
  - Analyzing the ARP Table
  - Switch Forwarding to “Known” MAC Address
  - MAC table aging on no activity
- Basic Operations and Messages
  - OpenFlow Learning Bridge example
  - OpenFlow Ethernet BroadcastDomain
  - OpenFlow Learning SwitchApplication
  - OpenFlow Learning Switch - ARP
  - OpenFlow Learning Switch - Flow match
  - OpenFlow Learning Switch - Packet IN
  - OpenFlow Learning Switch - Learning
  - OpenFlow Learning Switch - Packet OUT
  - OpenFlow Learning Switch - Flooding
  - OpenFlow Learning Switch - ARP Response
  - OpenFlow Learning Switch - Packet IN
  - OpenFlow Learning Switch - Packet OUT
  - OpenFlow Learning Switch - L2Forwarding
  - OpenFlow Learning Switch - L2Forwarding
  - Flow Timers
  - Populating Flow Entries Reactively
  - Populating Flow Entries Reactively
  - HW vs SW Switches
- Flow Table
  - Flow Table
  - Flow Entries
  - Table ID
  - Priority
  - Packets
  - Match
  - Actions/Instructions
- Review of OpenFlow Specification (current or 1.1.0, Wire Protocol 0x02)
  - Normal Port
- Flow Tables, Pipeline Processing
  - Pipeline Processing

    - Multiple Match Tables (MMT)
- Group Table, Matching, Instructors
    - Instructions
    - Action Set
    - Instructions that modify action set
    - Actions
    - Flow Table Entry
    - Flow Switching/Routing
    - Group Tables (OF 1.1)
    - OpenFlow 1.2
    - OpenFlow 1.3
    - OpenFlow 1.4
    - OF 1.5
- Segment Routing

6. Open vSwitch

- Architecture and Components
    - What is Open vSwitch?
    - What is Virtual Switch?
    - Open vSwitch Design
    - Open Virtual Network Architecture
    - Open Virtual Network Architecture
    - Open Virtual Network Architecture
    - Open Virtual Network Architecture
    - Open Virtual Network Architecture
    - Open Virtual Network Architecture
    - Open Virtual Network Architecture
    - Open Virtual Network Architecture
    - Open Virtual Network Architecture
    - Open Virtual Network Architecture
- OpenvSwitch Daemon
    - ovs-vswitchd
- ovsdb-server
    - Lifecycle of a VIF
- Core Tables
    - Open vSwitch

- Linux Bridge vs. OpenvSwitch Design
    - Virtual Network Topology in OpenStack Example
- Ovs-ofctl, ovs-dpctl
    - Management
- Traditional VM Ethernet Processing
    - Traditional VM Ethernet Processing
- Intel DPDK intro
    - Intel DPDK
- Intel SR-IOV
    - Intel SR-IOV (Single Root IO Virtualization)
- OVS Kernel Module
    - OVS Kernel Module: openvswitch_mod.ko
- Intel DPDK Effect
    - Why is OVS-DPDK faster than OVS?
    - OVS vs OVS-DPDK
    - Cross Socket Tests
- ovs-vswitchd.conf.db(5)
    - ovs-vswitchd.conf.db - Open_vSwitch database schema
    - ovs-vswitchd.conf.db - Open_vSwitch TABLE SUMMARY
    - OpenFlow Switch Specification

7. OpenFlow Controller

- Northbound vs. Southbound Interfaces
    - Northbound API Abstractions
- RYU SDN Framework
    - What is Ryu?
    - What's Ryu?
    - Supported features/protocols
    - OF/firewall/router REST API
    - IDS Support
    - Ryu Implementation
    - Ryu Architecture
    - Event Dispatcher
    - Event Source/Sink
    - Event Request/Reply
    - Connection to OpenFlow Switch
    - Overview of Ryu Plugin

- OpenStack L2 Isolation: Physical View
- Flow Table Usage
- GRE Tunneling with OpenStack
- Python
- AIO Libraries
- Threading
- Hello Packets and Discovery
- Default Match
- PacketIN and PacketOut
- Source MAC learning at the controller
- Simple Switch via FlowMod

8. NETCONF and YANG

- Overview of Network Configuration
  - What is NETCONF and YANG?
  - Why NETCONF and YANG?
  - YANG: Data Schema for Networking
- Introduction to SDN with NETCONF
  - NETCONF Configuration Data Stores
  - NETCONF Layers
  - NETCONF Transactions, Network-wideTransactions
  - NETCONF Transactions, Network-wideTransactions
  - NETCONF Base Operations
  - NETCONF Example Configuration Sequence
- Introduction to SDN with YANG
  - YANG Data Modeling Nodes
  - YANG Data Modeling Nodes
  - YANG Example
  - YANG - Toaster
  - YANG - Toaster
- SDN Programming with YANG
  - YANG - Structure
  - YANG - Header
  - YANG - Identities
  - YANG - Container
  - YANG - RPC Example

- YANG - RPC example 2
- YANG - Notifications

- SDN Programming with NETCONF
  - NETCONF RFC 6241 Optional Capabilities
  - Non-base NETCONF Capabilities
  - NETCONF <hello> Operation
  - NETCONF <get-config>Operation
  - NETCONF <get-config>Operation
  - NETCONF <edit-config> Operation
  - NETCONF <lock>, <unlock> Operation
  - NETCONF <get> Operation
  - NETCONF <close-session> Operation
  - Additional NETCONF operations by capabilities
- VPN Scenario
  - VPN Scenario
- RFC 7149
  - RFC 7149 A Perspective from within a Service Provider Environment
  - RFC 7426 SDN Layers and Architecture Terminology

9. OpFlex

- Introduction
  - OpFlex Background
  - draft-smith-opflex-03 Status
- Group Policy
  - Group Based Policy (1 of 3) "As desired"
  - Group Based Policy (2 of 3) "in Reality"
  - Group Based Policy (3 of 3) "How Group based Policy is Deployed"
  - Group Based Policy Constructs
  - The Translation from Group based Policy to Reality
  - The OpenFlow Conversion from Policy to Reality
  - The OpFlex Conversion from Policy to Reality
  - OpFlex Elements
  - Logical constructs required for OpFlex operation
  - OpFlex Transaction Types

10. Introduction to OpenDaylight

- Fundamentals for OpenDaylight Programming
    - Open Daylight- 4th Release “Beryllium” Production-Ready Open SDNPlatform
- Setup
    - OPEN DAYLIGHT Versions
    - OPEN DAYLIGHT Versions
- OSGI
    - OSGI: As the Architect Designed it
    - OSGI: As the Boss Changed it
    - OSGI: Business Requirements Changed it
    - OSGI: As the programmer developed it
    - OSGI: Design vs Deployment without OSGI
    - OSGI: As the programmer Maintains the Code
    - OSGI: Unknown Dependencies!?
    - OSGI: How OSGI Helps
    - Open Service Gateway Initiative (OSGI)
- Fundamentals – Maven and Project Building
    - Maven Package Manager for Java
- Apache Karaf
    - Karaf: OSGI Management
- Fundamentals – Mininet
    - Apache Karaf Overview

11. SAL

- Controller Functionality
    - Critical Northbound Applications
- Standardization
    - Diagram of Standardization

12. OpFlex

- Big Picture Diagramming
    - The Big Picture

13. SAL

- Standardization
    - MD-SAL Communication Model
- Restful Interface YANG
    - MD-SAL's Restful Interface
- Model Driven Service Abstraction Layer
    - MD-SAL's Interaction with the Controller
- Network Abstraction
    - Network Abstractions (Policy/Intent)
- Alto Protocol
    - Alto Protocol Manger
- Fabric as a Service
    - New, Fabric as a Service (FaaS)
- Network Modeling Language NEMO
    - New, Nemo – A NEtwork MOdeling Language
- Group Based Policy Service Example
    - Network Intent Composition

14. Overlays and Underlays

- Architecture for Overlay Networks (draft-ietf-nvo3-arch-04)
    - An Architecture for Data Center Network Virtualization Overlays
    - An Architecture for Data Center Network Virtualization Overlays (Continued)
- Security Requirements of NVO3 (draft-ietf-nvo3-security-requirements-07)
    - Security Requirements of NVO3 (draft-ietf-nvo3-security-requirements-07)
    - Introduction to Cloud Overlay Networks
    - L3 Based Fabric Advantages
    - L3: A Better Design
    - Tunnels in the Physical World
    - VXLAN: Virtual eXtensible LAN
    - VXLAN: Virtual eXtensible LAN
    - VXLAN: Virtual eXtensible LAN
    - How do VTEPs handle BUM (Broadcast, Unknown Unicast, Multicast)?
    - VXLAN: Virtual eXtensible LAN
    - VXLAN Service Node
    - How many L2 networks in this picture? Two!
    - How many L2 networks in this picture? Still only two!
    - VLAN

- How L2 VLAN tagging works
    - How many L2 networks in this picture? Four!
    - Again, how many L2 networks in this picture? Four!
    - VXLAN Packet Headers
    - GRE Packet Headers
    - How L2 VLAN tagging works with L3 subnets
    - VTEP allows L2 connectivity despite L3 boundaries
    - VTEP allows L2 connectivity despite L3 boundaries (1 of 2)
    - VTEP allows L2 connectivity despite L3 boundaries (2 of 2)
    - Examining VXLAN tagging in Wireshark
    - Decode as VXLAN
    - Now Wireshark shows vxlan-encapsulated internal packets!

15. OpenStack Neutron Networking

- Bare Metal Interfaces
    - Neutron Networks
    - Same Tenant, Same VM
    - Neutron Networking same compute, same subnet
    - Neutron Networking same compute, different subnet, no DVR
    - Neutron Networking VXLAN Option without DVR
    - Neutron Networking same compute, different subnet, with DVR
    - Neutron Networking different compute, different subnet
    - Neutron Networking same compute, different subnet, no DVR
    - Neutron Networks
    - Neutron Networks
    - Neutron Networks
    - Neutron Networks
- OpenvSwitch
    - Neutron Architecture (OVS)
- Type Drivers – VLAN
    - Neutron ML2 Type Drivers
    - Neutron ML2 Mechanism Drivers Drivers
- Neutron Network Types
    - Neutron Network Types
- Type Drivers – VXLAN
    - OpenStack VXLAN

- Restful Interface
    - Restful Interface
    - Writing an Application using RESTful API

17. ONOS Controller

- East-West ONOS Cluster
    - ONOS without Open_vSwitch table summarization
- Shared Aggregate Network Topology
    - ONOS Open_vSwitch with summarization (east)
    - ONOS Open_vSwitch view after summarization
- Provider Clustering Diagram
    - Open_vSwitch TABLE SUMMARY
- ONOS0
    - ONOS0
- ONOS1
    - ONOS1
- Ecosystem
    - ONOS Ecosystem
- Control and Data Planes
    - ONOS Control & Data Planes
    - Redundant ONOS Controllers
- Network Slicing
    - Network topology determined by slice aggregation
    - ONOS controllers topology map sharing.
    - An ONOS primary controller is elected per slice
    - ONON network when a primary controller fails
    - Rapid Raft Consensus Protocol
    - ONOS after controller recovery
- Versions
    - ONOS after controller recovery

18. Securing SDN

- Securing the Controller
    - Security Challenges
- Security Challenges

- SDN-Specific Security Challenges
- SDN-Specific Security Challenges

➢ Security Principles
  - Security Principles
  - Security Principles
  - Security Principles
  - Security Principles

➢ Attack Model
  - Attack Model

SDN Labs

- **Lab 1:** Linux iproute2 - Linux is standardly equipped with most of the NFV components, let's learn the basics first.
  - legacy vs iproute2 commands
  - create a virtual interface
  - assign an ip address to virtual interfaces
  - assign static ip routes
- **Lab 2:** NFV debugging tools
  - Using live tcpdump to view SDN protocol analysis
  - tcpdump pcap files for Wireshark
- **Lab 3:** Virtual

  Interfaces o
  veth
  - network namespace
  - interconnecting namespaces
- **Lab 4:** Linux Bridge
  - link show
  - create a linux bridge
  - installing new ports
  - interconnecting namespaces with veth and linux bridge
- **Lab 5:** ADVANCED – Augmenting bash for working with Network

  Namespaces o Modify bash prompt to indicate shell
  environment's current namespace
- **Lab 6:** Open vSwitch (OVS) o

  create an OvS
  bridge o Install ports

  - Install bridge internal ports
  - Interconnect namespaces using OvS
  - Show advantages of OvS over Linux bridge
- **Lab 7:** Wireshark set up for SDN Protocol analysis
  - How to set up wireshark to perform OpenFlow Protocol analysis
- **Lab 8:** Introduction to Mininet o

  Set up
  mininet

- mininet commands
- Standard mininet topologies
- **Lab 9:** Using MiniEdit to create custom MiniNet
  Topologies o How to setup a custom mininet
  topology
- **Lab 10:** Mininet Namespaces – Learning About Linux Network
  Namespaces o Learn how mininet manages network
  namespaces
  - Basic python example to examine mininet namespaces
- **Lab 11:** ADVANCED – SDN Topology Analysis Using Python
  - An optional lab, teaching how to use python to build complextopologies
- **Lab 12:** Detailed Wireshark analysis of live OpenFlow
  Traffic o HELLO
  - OFPT_FEATURES (request/reply)
  - OFPT_MULTIPART
  - OFPMP_PORT_DESC
  - reserved ports
  - OFPT_PACKET_IN
  - OFPT_PACKET_OUT1
  - OFPT_FLOW_MOD
  - OFPT_ECHO
- **Lab 13:** OPTIONAL – Using vim
  - Optional lab for people not familiar with linux text editing
- **Lab 14:** Introducing the Controllers (Ryu)
  - We will analyze and run already-written python scripts to perform
    - broadcast domain management
- **Lab 15:** Writing a FlowMod to Handle a Table-Miss – Controller Application
  (ryu-app) o We add the northbound logic to perform table-missprocessing
- **Lab 16:** PacketIn Hub Logic with an SDN (Ryu)

    - We add even more logic to make the OvS switch act like a hub
- **Lab 17:** Deploying Simple Switch Logic with an SDN (Ryu)
    - We yet more logic to make the OvS switch learn mac addresses
- **Lab 18:** Deploying Simple Switch Logic with an SDN (Ryu) Part 2
    - We add logic to populate switch mac tables, greatly reducing OpenFlow chatter and speeding up the OvS.
- **Lab 19:** Neutron Networking. All labs leading up to this point are mandatory to understand this lab.
    - A deep dive into the following OpenStack NFV stack (NOTHING isomitted).
        - veths
        - namespaces
        - OvS
        - OvS patches
        - OvS Bridge internal
        - Linux bridge security group (firewall)
        - Bonding
        - Vlan Management
        - Integration bridge
        - Tunnel bridge
        - Vlan bridge
        - Provide bridge
    - On a newly installed OpenStack datacenter, Install a Neutron networking from scratch. While performing each of the following steps, analyze EXACTLY what NFV components have been changed by neutron.
        - Install the provider net
        - define provider gateway router
        - define provider subnet
        - create a tenant network
        - Install a tenant router
        - install a tenant subnet
        - configure firewall rules
        - Launch the virtual machine

Prerequisites:

- Delegates should be from an engineering background with a solid appreciation of carrier class or enterprise networks.
- Understanding of a programming language such as Python or Java is helpful but not essential.

Who can Attend:

- Strategic planners, network architects, network managers, systems engineers, service planners and carrier operation staff who are responsible for planning, implementing and deploying networks which may require SDN and/or NFV techniques in the future.

Number of Hours: 50hrs

Certification: AWS Certified Solutions Architect – Associate (SAA-C02)

Key Features:

- One to One Training
- Online Training
- Fastrack & Normal Track
- Resume Modification
- Mock Interviews
- Video Tutorials
- Materials
- Real Time Projects
- Virtual Live Experience
- Preparing for Certification