Training Contents

SDN, OpenFlow and NFV Deep Dive with SDNit and Ivan Pepelnjak 21-22nd September in Stockholm

The Need for Software Defined Networking

While the whole IT industry has been moving toward highly automated solutions in the last decade, networking has remained stuck – most networking engineers are still manually configuring individual devices.

There's high time we change the deployment and operational processes and reduce the amount of time spent doing repetitive manual tasks. this part of the workshop will give you some high-level guidelines.

The first part of the workshop focuses on technologies underlying SDN and NFV – OpenFlow, NETCONF, APIs, virtualization and virtual appliances

Software Defined Networking Explained

Software defined networking is not a new technology – we've been using the concepts of programmable networks for decades.

This section describes the motivations behind the SDN movement, its principles and perfect use cases, and numerous technologies you can use to program the network devices. It will also try to answer the fundamental questions: When, Why and How should you program your network.

Introduction to OpenFlow

This section describes the concepts of OpenFlow, a new protocol used to decouple control plane (topology discovery, path calculation…) from data plane (packet forwarding). It covers the following topics:

- Traditional forwarding with distributed routing protocols
- Controller-based forwarding
- Basics of OpenFlow protocol
- Benefits and drawbacks of OpenFlow

OpenFlow Deep Dive

After the introduction to OpenFlow concepts, the training includes a deep dive into the details of OpenFlow protocol, including:

- OpenFlow forwarding model.
- OpenFlow ports, classifiers, and actions.
- OpenFlow groups and multi-table support.
- QoS in OpenFlow networks.
- OpenFlow protocol details.

- Simple OpenFlow use cases, from controller-based topology discovery and learning bridges to distributed routing and control-plane protocols.
- OpenFlow deployment models and real-life implementations.

OpenFlow Scalability Challenges

OpenFlow concepts are not new and share scalability challenges with similar technologies and architectures including Frame Relay, ATM, ForCES and MPLS-TP. This section discusses the major OpenFlow scalability challenges:

- Hardware limitations
- Proactive and reactive forwarding table setup
- Hop-by-hop and path-based forwarding
- Control-plane scalability and lack of shared fate

Benefits of Network Function Virtualization

If you open a firewall, load balancer, WAN accelerator or almost any other network services appliance, you'll find one or more x86 processors, standard GE/10GE NICs and some custom packet handling logic. Is there any reason we have to be tied to physical hardware? Wouldn't it be better to deploy the same services in virtual machine format and make them flexible? That's the fundamental concept of Network Function Virtualization.

Does it really make sense to replace physical network services appliances with virtual machines? What are the benefits and drawbacks of NFV approach? This section will give you the answers you need to start evaluating applicability of NFV in your environment.

BGP-based SDN

Numerous SDN solutions use BGP as the controller-to-device communication protocol. This section explains the basics of BGP-based SDN, documents several typical use cases and gives practical deployment guidelines, including sample open-source-based controller implementation.

Network Programmability with NETCONF and YANG

NETCONF is a protocol widely used to configure networking devices (it's supported by Brocade, Cisco, Juniper and other vendors). This section describes NETCONF and YANG (the data model description language used by NETCONF), their benefits and shortcomings, and the vendor-specific implementation details. It includes the following topics:

- What is NETCONF and YANG
- Why are SNMP, CLI and REST not good enough?
- Where did NETCONF and YANG come from?
- How does NETCONF work over XML?
- How does YANG work?
- Tools you can use to test your NETCONF code
- Differences in NETCONF implementations
- Deployment examples

Network Automation with Chef, Puppet and Ansible

Chef, Puppet, and Ansible are the most popular server configuration management tools, and all of them get used in network automation solutions.

This section describes the fundamentals of all three tools, their typical implementation on network devices, and the potential benefits and drawbacks of using them. It then focuses on Ansible is one of, which is commonly the tool-of-choice due to its agentless design.

SDN and Controller-Based Networking Deployment Considerations

Networking solutions with centralized network intelligence or control plane have existed for almost half a century (IBM SNA, ATM, Frame Relay, Ipsilon Flow Management Protocols).
Not surprisingly, novel SDN architectures using centralized controller clusters exhibit similar challenges:

- Single points of failure.
- Impact of network partitions.
- Balance between tightly- and loosely-coupled elements.
- Control plane and controller security.
- Impact of data plane activity on control-plane performance (punting to control plane).
- Control plane denial of service (DoS) attacks.

This section describes typical SDN deployment considerations, ranging from architectural and design challenges to security and operational considerations.

Real-Life SDN Use Cases

Service providers and enterprises are already deploying SDN, using NETCONF, BGP or OpenFlow as the implementation technology. This section describes numerous use cases based on real-life deployments:

- Data center fabrics (Arista XMPP, Juniper QFabric, NEC ProgrammableFlow, Plexxi controller)
- Forwarding optimizations and exception routing with BGP (Microsoft)
- Optimized WAN edge forwarding (Spotify/Arista)
- Centralized traffic engineering with OpenFlow (Google)
- Programmable network taps and tap aggregation networks (Arista, NEC, Big Switch, Cisco)
- Network monitoring (Plexxi Control, HP SDN VAN controller)
- Network services insertion (NEC ProgrammableFlow, segment routing, virtualization solutions)
- Software-defined WAN
- Scale-out load balancing (NEC/Riverbed) and firewalling (Arista/Palo Alto)
- Scale-out intrusion detection system (University of Indiana)
- DoS mitigation tools (Remote-triggered black holes, BGP Flowspec, NEC/Radware)
- Edge policy enforcement