Open-source Networking

2020. 11.

(2021년 4월까지 사용 권장)

안 종 석
james@jslab.kr
JS Lab

목차

Opensource Networking
james@jslab.kr

I. 오픈소스 네트워킹 개요

- 개요
- 벤더 솔루션
- Hierarchy of a Network Device
- Legacy Networks
- Opensource at the Market
- Disaggregation
- Modern Networking and SDN

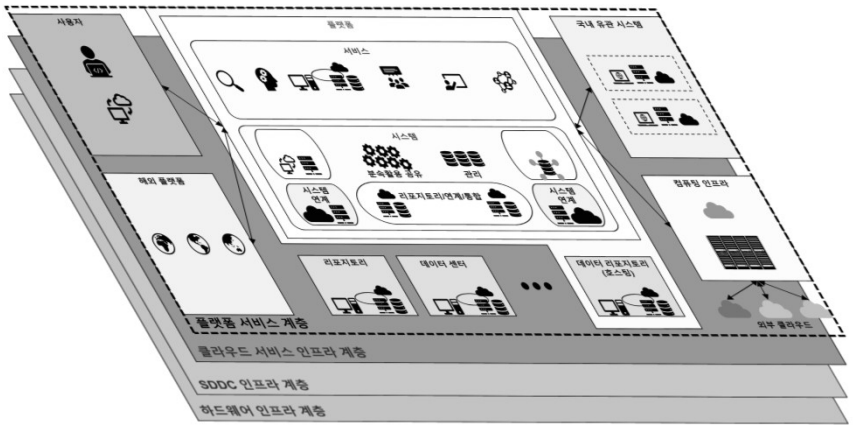
Opensource Networking
james@jslab.kr

https://www.linuxfoundation.org/projects/networking/

I. 오픈소스 네트워킹 개요

❖ 개요

- 계층별 네트워크 연계
- 네트워크 간 연결을 위한 오픈소스 프로젝트 활동 활발



I. 오픈소스 네트워킹 개요

❖ 개요

- 리눅스 재단(LINUX Foundation)의 네트워킹

https://www.linuxfoundation.org/projects/networking/

Opensource Networking
james@jslab.kr

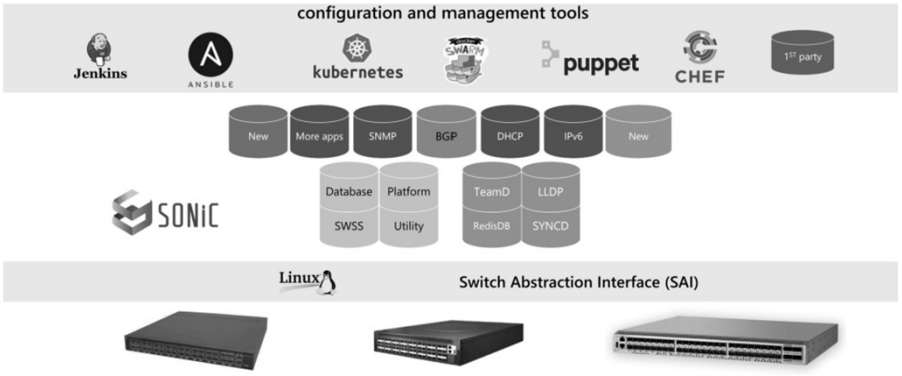
Opensource Networking

james@jslab.kr

I. 오픈소스 네트워킹 개요

❖ 개요

- **SONiC** (Software for Open Networking in the Cloud)



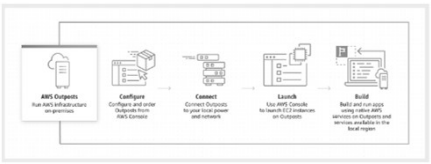
Opensource Networking

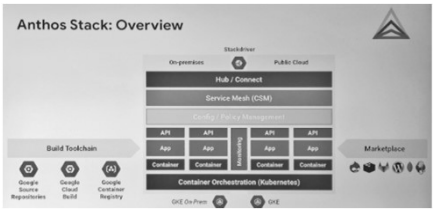
james@jslab.kr

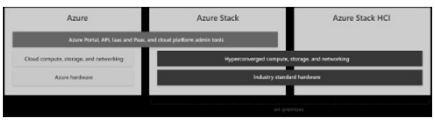
I. 오픈소스 네트워킹 개요

❖ 클라우드 서비스 회사들의 하이브리드/멀티클라우드 솔루션

- **AWS Outpost**
- **MS Azure Stack**
- **Google Anthos**
- **IBM Cloud Pak**







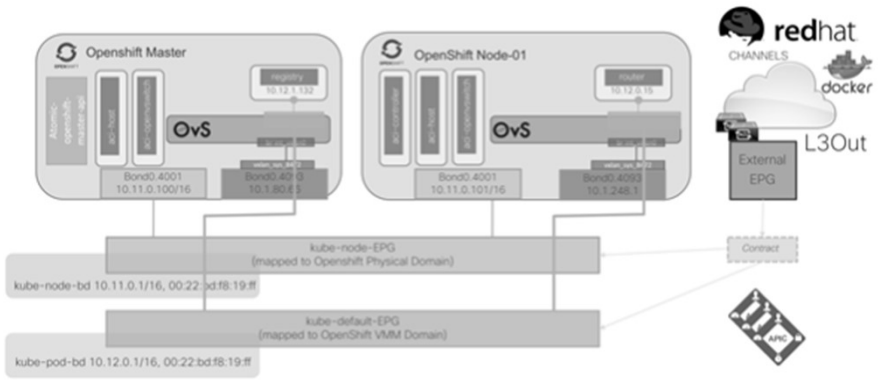
Opensource Networking

james@jslab.kr

I. 오픈소스 네트워킹 개요

❖ 제조사 솔루션 연동

- 제조사들의 멀티클라우드 기반 아키텍처에 오픈소스 네트워킹 기술 채택
- Red Hat, Cisco, VMware 등



JS Lab

https://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/white_papers/Cisco-ACI-CNI-Plugin-for-OpenShift-Architecture-and-Design-Guide.html

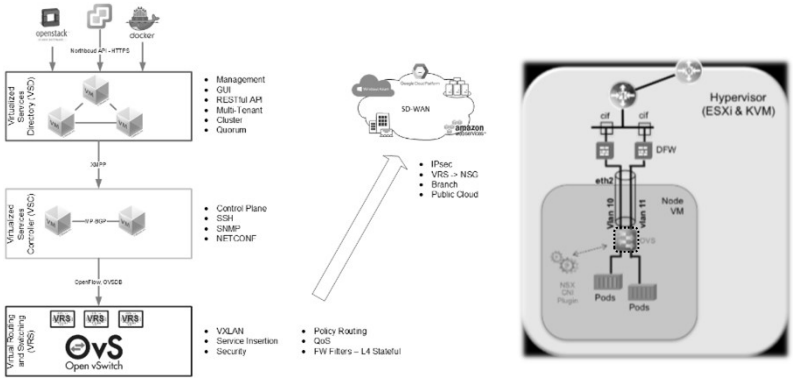
Opensource Networking

james@jslab.kr

I. 오픈소스 네트워킹 개요

❖ 제조사 솔루션 연동

- 제조사들의 멀티클라우드 기반 아키텍처에 오픈소스 네트워킹 기술 채택
- Red Hat, Cisco, VMware 등



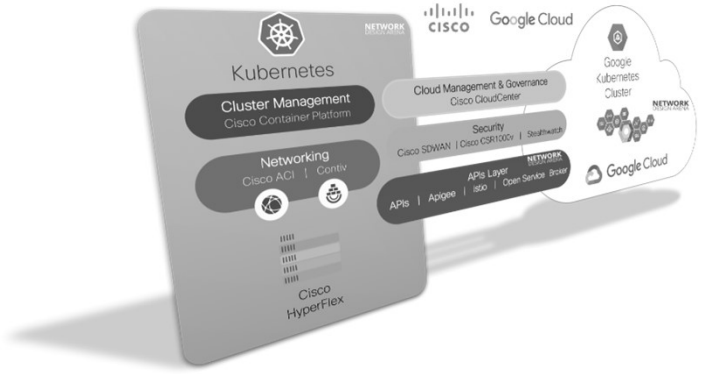
JS Lab

https://thedataplumber.net/nsx-t-vs-nsx-v-and-a-little-bit-of-nuage-vsp/ https://www.routetocloud.com/category/nsx-t/

I. 오픈소스 네트워킹 개요

❖ 클라우드 서비스 / 제조사 협력 솔루션

- 서비스/제조사간 협력 또는 인수를 통한 솔루션 확대
- 네트워킹 오픈소스 활용



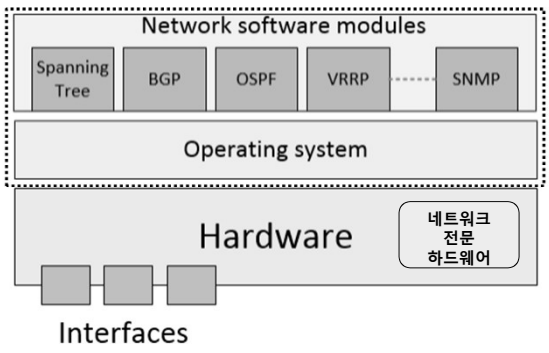
http://www.netdesignarena.com/index.php/2018/05/30/why-cisco-multi-cloud-a-cloud-architect-perspective/

Opensource Networking
james@jslab.kr

I. 오픈소스 네트워킹 개요

❖ Hierarchy of a Network Device

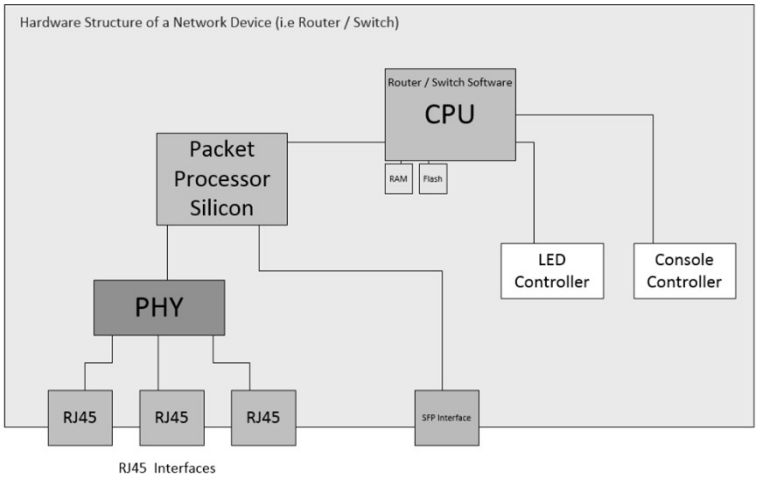
- Packet processor and forwarding
- CPU, along with RAM and flash
- Other controllers, such as a fan controller, console port, LED controller, interface controllers, etc.
- Software.



Opensource Networking
james@jslab.kr

I. 오픈소스 네트워킹 개요

❖ Legacy Networks



I. 오픈소스 네트워킹 개요

❖ 네트워킹 오픈소스

이름	구분	출범일
Edgent	네트워크 분석	2016-12
linkerd	NFVI - 인프라, VNF - L4-7 가속, 캐싱	2016-04
Cilium	NFVI, VNF - L4-7 보안	2017-03
BiRD	NFVI - 스위칭, 라우팅	2013-03
NetBox	NFVI - 스위칭, 라우팅	2016-06
OSM (Open Source MANO)	NFV MANO	2016-05
FBOSS	NFVI - 스위칭, 라우팅, NFVI - NOS	2015-03
Faucet SDN 제어ler	NFVI - 제어	2015-03
GoBGP	NFVI - 스위칭, 라우팅	2017-02
HAProxy	VNF - L4-7 보안, 가속, 캐싱	2001-12
YANFF	NFVI, VNF - L4-7 보안, 가속, 캐싱	2017-03
OpenContrail	NFVI - 스위칭, 라우팅, 제어, NFV MANO	2013-09
OpenDataPlane Project	NFVI - 인프라	2015-02
OpenSwitch	NFVI, 스위칭, 라우팅, NOS	2016
OPNFV	NFVI - 인프라, 하드웨어, 스위칭, 라우팅, NOS, 제어	2017-09
FD.io	NFVI - 인프라, 스위칭, 라우팅, VNF - L4-7 가속, 캐싱	2016-02
OpenDaylight	NFVI - 제어	2013-03

이름	구분	출범일
Open vSwitch	NFVI - 스위칭, 라우팅	2009-07
ONAP	NFVI - 제어, NFV MANO, VNF - L4-7 보안,가속, 캐싱	2017-03
DPDK	NFVI - 인프라, 스위칭, 라우팅	2012-09
FR라우팅 (FRR)	NFVI - 스위칭, 라우팅	2017-10
OpenLSO	NFV MANO	2016-03
NGINX Open Source (OSS)	VNF - L4-7 보안, L4-7 가속, 캐싱	2011-07
Ryu NOS	NFVI - NOS, 제어	2011-12
Open Network Linux	NFVI - NOS	2014-01
ONIE	NFVI - 하드웨어, 설치	2013-06
SONiC	NFVI - 스위칭, 라우팅, NOS	2016-03
OpenConfig Project	NFV MANO	2014-10
CORD	NFVI - 인프라, NOS	미제공
ONOS	NFVI - 제어	2014-12
OpenStack Neutron	NFVI - 인프라	2013-07
OpenStack Tacker	NFV MANO	2015-12
P4	NFVI - 인프라, 스위칭, 라우팅	2015-02
Project Calico	NFVI - 스위칭, 라우팅	2014-07
Open Virtual Network (OVN)	NFVI - 인프라, 스위칭, 라우팅	2015-01

Opensource Networking
james@jslab.kr

II. 오픈소스와 SDN Landscape

- Disaggregated Hardware Layer
- IO 추상화와 Datapath Layer
- Network Operating Systems
- Network Control Layer
- Network Virtualization
- Cloud and Virtual Management Layer
- Orchestration, Management, Policy Layer
- Network Data Analytics

Opensource Networking
james@jslab.kr

II. 오픈소스와 SDN Landscape

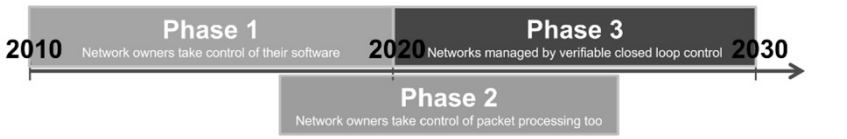
- **Disaggregation**
- **Modern Networking and SDN**
  - **Rip-and-Replace**, **Direct Fabric Programming** (Cloud Networking)
  - **Overlay**
  - **Hybrid**

Rip-and-Replace

Opensource Networking
james@jslab.kr

II. 오픈소스와 SDN Landscape

- **SDN의 발전**
- **Nick McKeown's 3 Phases of SDN** (ONF Connect, September 2019)
  1. **Phase 1:** 네트워크 소유자가 자신의 소프트웨어로 제어
  2. **Phase 2:** 네트워크 소유자가 패킷처리도 제어
  3. **Phase 3:** 네트워크는 확인 가능한 폐쇄 루프 제어에 의해 관리



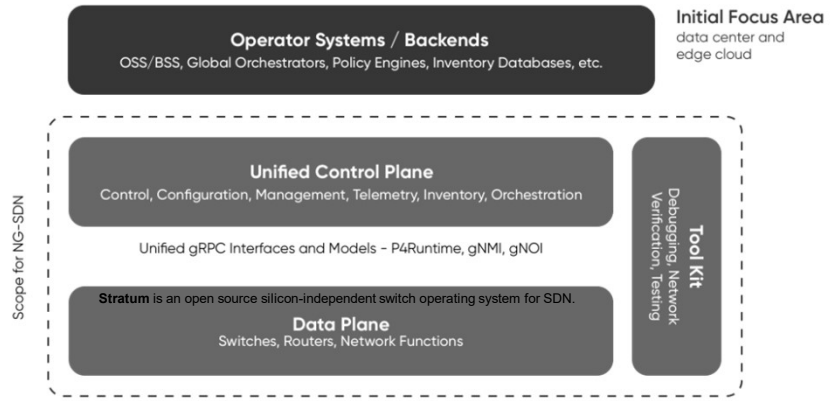
Opensource Networking
james@jslab.kr

Open Networking Foundation **(ONF)**

I. 개요

❖ **차세대 SDN (NG-SDN)**

❖ **요소 프로젝트:** Stratum, P4, μONOS, ONOS SDN Apps, OpenConfig, gRPC Network Management Interface(gNMI), gRPC Network Operations Interface (gNOI), Open Network Linux (ONL), Open Compute Project(OCP)

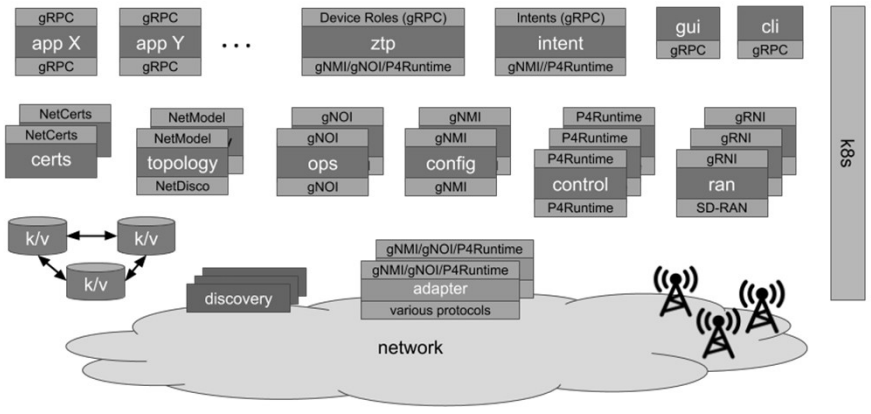


https://www.opennetworking.org/reference-designs/ng-sdn/

JS Lab

II. 오픈소스와 SDN Landscape

❖ **μONOS Deployment Architecture**



gRPC Network Operations Interface (gNOI) gRPC Network Management Interface(gNMI)

https://docs.onosproject.org/

JS Lab

II. 오픈소스와 SDN Landscape

❖ 제조사 네트워킹 솔루션의 오케스트레이션 연동 (엔터프라이즈/Telco)

- SDN 기반 클라우드 네이티브(Cloud Native)화 데이터센터 네트워킹 분야 집중
- 컨테이너, 서비스 메쉬, 마이크로서비스, 변경 불가능 인프라(Immutable Infrastructure) 및 선언적 API를 사용하는 접근 방식

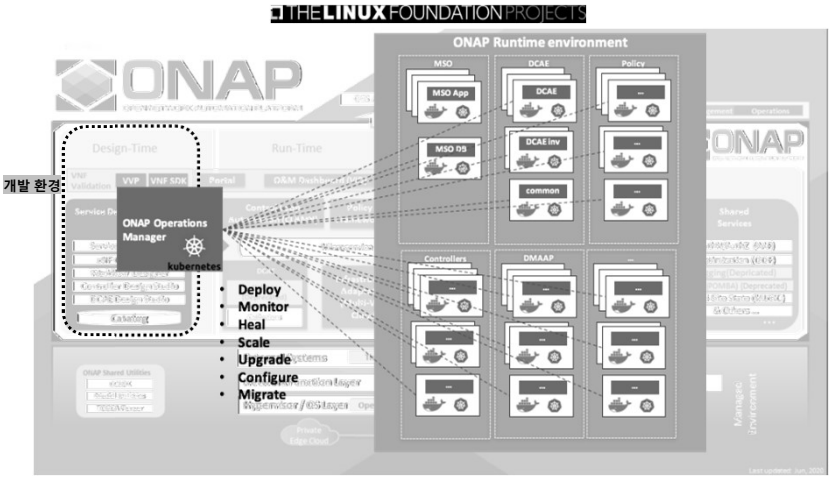
제조사	솔루션 이름	오케스트레이션 연동
Big Switch Networks	Big Cloud Fabric	쿠버네티스, 오픈스택, VMware, OpenShift
Huawei	CloudFabric	오픈스택, FusionSphere, ManageOne, Red Hat, Mirantis
Lenovo	RackSwitch	오픈스택, VMware vCloud Suite, Microsoft Azure Stack, Tungsten Fabric
Netronome	Agilio SmartNIC	오픈스택
Plexxi	Plexxi HCN	쿠버네티스, 오픈스택, vCloud, Nutanix
ZTE	ZENIC	쿠버네티스, 오픈스택
Dell EMC	Z9100-ON Switch	쿠버네티스, 오픈스택, VMware vCloud Suite
Altoline	99xx/69xx	쿠버네티스, 오픈스택, VMware vCloud Suite
Mellanox	Open Composable Networks	오픈스택, VMware vCloud Suite, NEO
Cisco	Application Policy Infrastructure Controller (APIC)	VMware vCloud Suite
Ericsson	Cloud SDN	쿠버네티스, 오픈스택

제조사	솔루션 이름	오케스트레이션 연동
Juniper Networks	Contrail	오픈스택
Nuage Networks	Virtualized Services Platform(VSP)	쿠버네티스, 오픈스택, VMware vCloud Suite, 클라우드스택
Pluribus	Netvisor OS, Adaptive Cloud Fabric	VMware vCloud Suite, Ansible, Puppet, Chef
FlowEngine	FlowEngine TDE-2000	오픈스택, VMware vCloud Suite
Red Hat	NFV solution	쿠버네티스, 오픈스택
VMware	NSX	쿠버네티스, 오픈스택, VMware vCloud Suite
Wind River	Titanium Cloud	오픈스택
A10	Thunder ADC	오픈스택, VMware vCloud Suite
Cumulus	Cumulus Linux	오픈스택
ipinfusion	OcNOS	오픈스택
Pulse Secure	Pulse Access Suite	쿠버네티스, 오픈스택, VMware vCloud Suite

II. 오픈소스와 SDN Landscape

❖ ONAP: 물리/가상 네트워크 기능을 위한 오픈소스 소프트웨어 플랫폼

(real-time, policy-driven orchestration, automation)



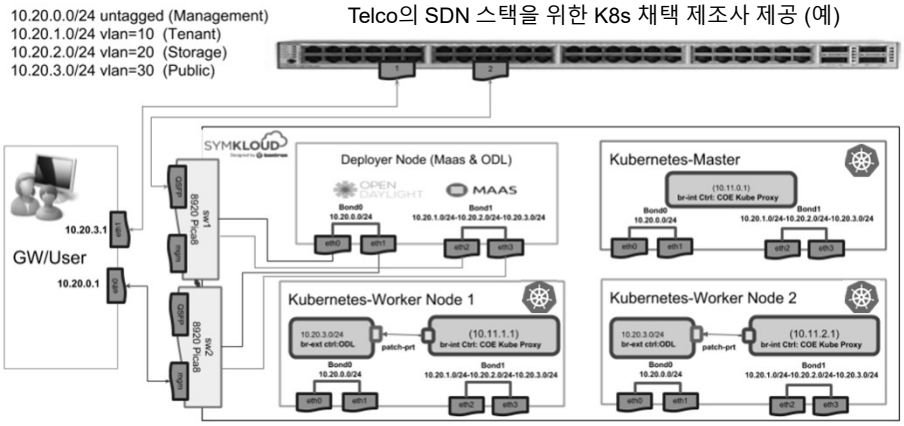
https://www.onap.org/architecture

II. 오픈소스와 SDN Landscape

❖ **제조사의 Telco를 위한 K8s의 제어플레인 기능 배포**

- Telco의 SDN 스택을 위한 K8s 채택 (예)
- K8s는 제어기능의 배포 위치 변경 요구 수용 필요 (DC or UP)

Telco의 SDN 스택을 위한 K8s 채택 제조사 제공 (예)

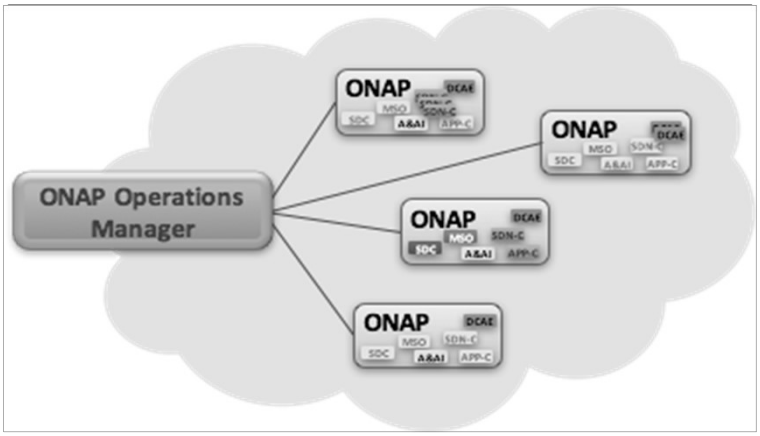


https://blog.symkloud.com/adapting-kubernetes-to-the-sdn-stack-for-telco

II. 오픈소스와 SDN Landscape

❖ **Cloud and Virtual Management Layer**

❖ **Cloud and Virtual Management Layer Communicates with Network Controllers**



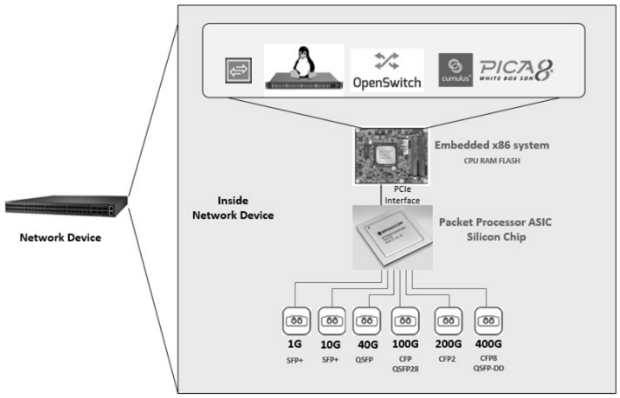
II. 오픈소스와 SDN Landscape

- ❖ **Open Source and Software Defined Networking**
- ❖ **Open Networking - 계층구조** (리눅스재단의 오픈소스네트워킹)
  - **Disaggregated Hardware**
  - **IO Abstraction and Datapath**
  - **Network Operating Systems**
  - **Network Control**
  - **Network Virtualization**
  - **Cloud and Virtual Management**
  - **Orchestration, Management, Policy**
  - **Network Data Analytics**
  - **Application Layer.**

Opensource Networking
james@jslab.kr

II. 오픈소스와 SDN Landscape

- ❖ **Disaggregated Hardware Layer**
- ❖ **Disaggregated Network Device inside an Open Hardware**



Opensource Networking
james@jslab.kr

II. 오픈소스와 SDN Landscape

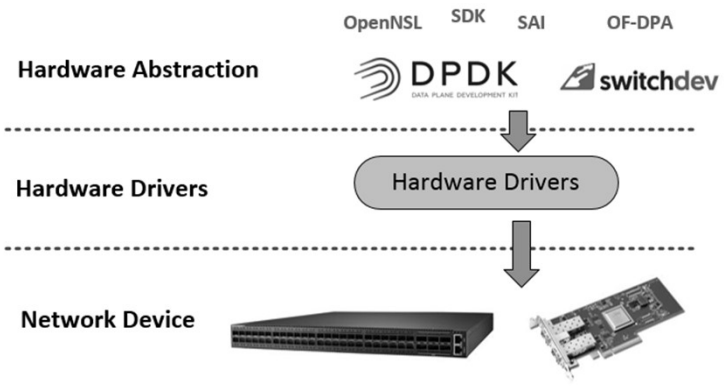
❖ **Disaggregated Hardware Layer**

기능 (Function)	리눅스 탑재 x86 서버 (An x86 server running Linux)	이더넷 스위치/라우터 (Dedicated Ethernet switch or router)
패킷처리 (Packet Processing)	Packets are sent to CPU and OS to make forwarding, routing or firewalling decisions	Packets are processed in packet processor ASIC (Application-Specific Integrated Circuit) silicon, not the CPU
처리량 (Throughput)	Limited to server hardware, such as CPU, I/O bus, kernel. Normally, limited to Gbps	Depends on the ASIC model. Varies from Gbps to tens of Tbps
부팅 (Boot process)	System boots as a normal PC, loads the Linux kernel, OS and networking software (for example, iptables, etc.)	A tiny OS runs on CPU and starts driving the ASIC
포트 수 (Port density)	Limited to the number of ports on the server, or additional ports via a PCIe (Peripheral Component Interconnect Express) card	Ethernet switches can support 48 or 52 ports in 1U form factor

II. 오픈소스와 SDN Landscape

❖ **IO Abstraction and Datapath Layer**

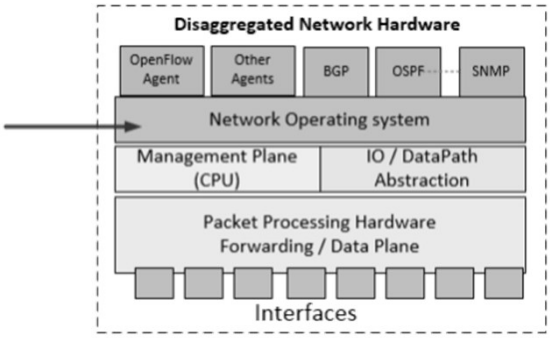
❖ **Hardware Abstraction in an Open Network Device**



II. 오픈소스와 SDN Landscape

❖ **Network Operating Systems**

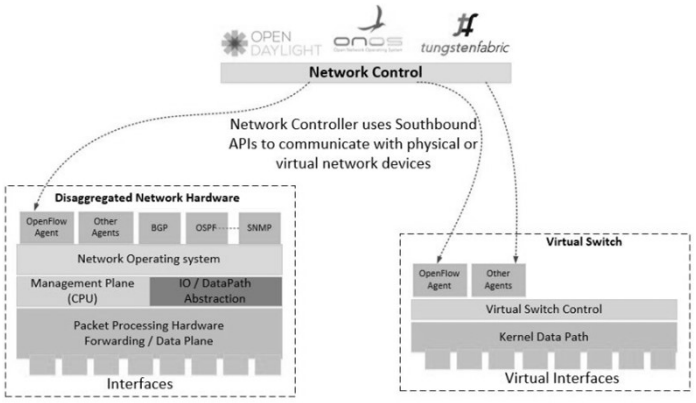
- **A network operating system is an operating system that runs on the management plane of a network device. This operating system is designed to drive the packet processor hardware chipset, such as a switch silicon, and perform the tasks required for forwarding, routing, and switching.**



II. 오픈소스와 SDN Landscape

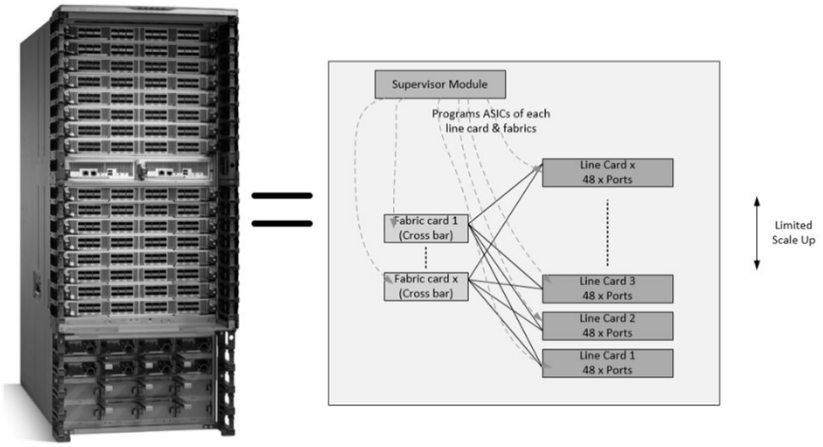
❖ **Network Control Layer**

- **The Network Control layer is about SDN controllers that can manage multiple network operating systems via agents and protocols (aka Southbound APIs).**



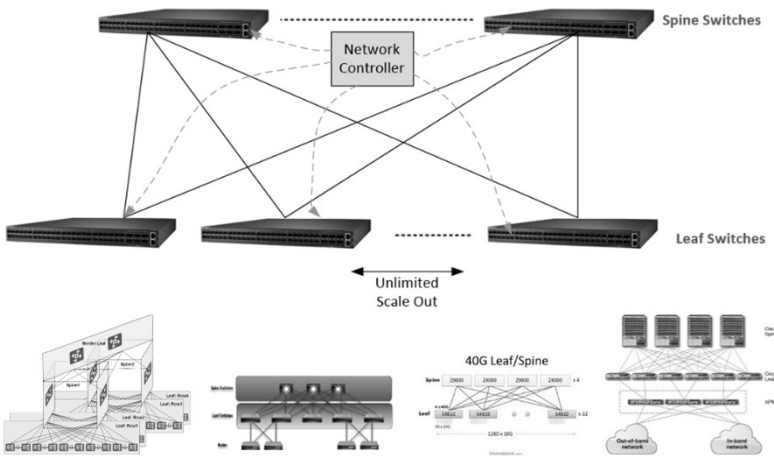
II. 오픈소스와 SDN Landscape

- ❖ Network Control Layer
- ❖ A Chassis-Based Switch



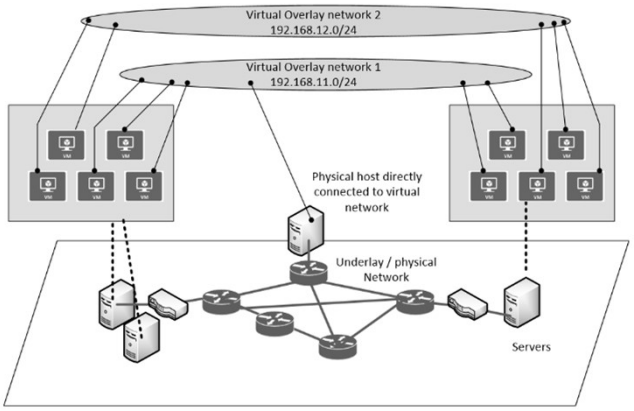
II. 오픈소스와 SDN Landscape

- ❖ Network Control Layer
- ❖ An SDN Network Can Scale Out



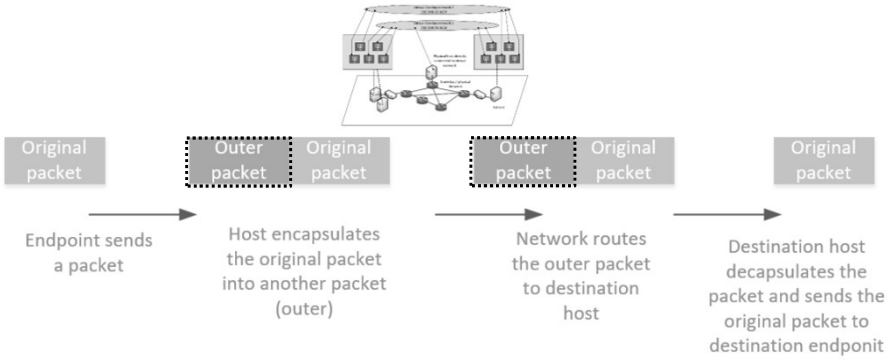
II. 오픈소스와 SDN Landscape

- ❖ **Network Virtualization**
- ❖ **Overlay Networks Are Virtual Networks on Top of Physical Networks, Built Using Encapsulation and Tunnels**



II. 오픈소스와 SDN Landscape

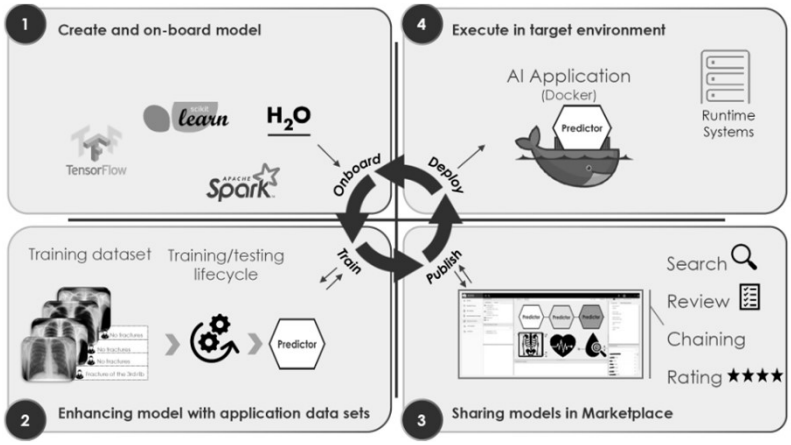
- ❖ **Packet Transfer Steps in an Overlay Network**



II. 오픈소스와 SDN Landscape

❖ **Network Data Analytics**

- **Acumos AI**



https://lfai.foundation/

JS Lab

JS Lab

Opensource Networking
james@jslab.kr

Opensource Networking

james@jslab.kr

III. 소프드웨어/하드웨어 분리

- Proprietary Network Products vs. Disaggregated Open Hardware
- Open Compute Project
- OCP Projects
- Telecom Infra Project
- How Ethernet Switches Are Built
- Types of Switches
- Bare Metal Ethernet Switches
- White Box Ethernet Switch Hardware
- Bare Metal Wireless Access Points

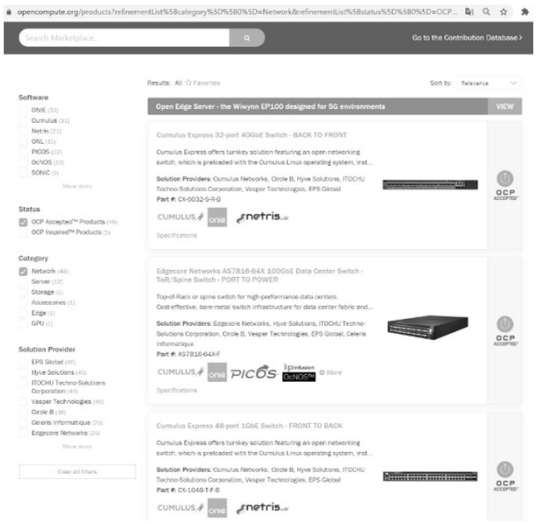
Opensource Networking

james@jslab.kr

III. 소프드웨어/하드웨어 분리

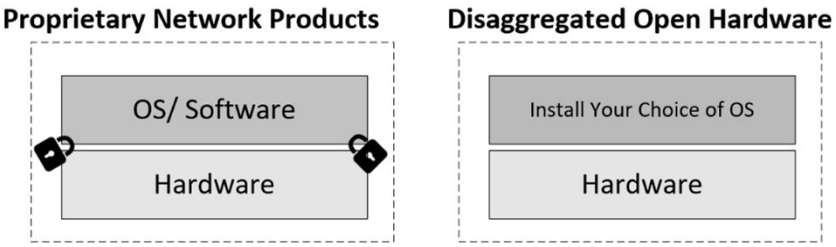
❖ **OCP marketplace** (https://www.opencompute.org/products)

OPEN Compute Project ®



III. 소프드웨어/하드웨어 분리

- ❖ **Proprietary Network Products vs. Disaggregated Open Hardware**
- ❖ **Commercial Networking Products vs Open Source Hardware**



III. 소프드웨어/하드웨어 분리

- ❖ **Differentiate between proprietary devices**
- ❖ **Disaggregated devices**

Proprietary Network Device	Details
Cisco Catalyst 3750 Switch	A combination of hardware and Cisco IOS software
Juniper M10iRouter	A combination of hardware and Juniper Junos OS
Arista 7170 Switch	A combination of hardware and Arista EOS software

Disaggregated Network Device	Details
Edge-Core AS5712 (48 x 10G switch)	Comes with no software/OS. You can check compatibility and install OpenSwitch/Open Network Linux (ONL)/Cumulus Linux/Pica8/Big Switch, etc.
Mellanox SN2700	Comes with no software/OS. You can check compatibility and install Cumulus Linux.
Alpha Networks SNX-60x0-486F (48-port 10G SFP)	Comes with no software/OS. You can check compatibility and install ONL/OpenSwitch/Cumulus Linux, etc.
Inventec DCS7032Q28 32 x 100GB	Comes with no software/OS. You can check compatibility and install ONL/OpenSwitch/Cumulus Linux, etc.

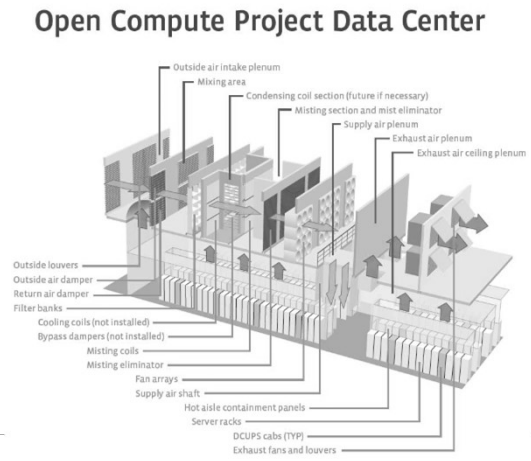
III. 소프드웨어/하드웨어 분리

❖ **Open Compute Project**

- **The Open Compute Project (OCP) was announced by Facebook, along with Intel, RackSpace, Goldman Sachs, and Andy Bechtolsheim, in April 2011. The effort was the result of a redesign of Facebook's data center in Prineville, Oregon. The aim of OCP is to create open source standards for high density and highly-efficient IT equipment for data centers, including server, storage, network, and security.**
- **OCP publishes the open hardware specifications for the data center and enterprise IT systems. There are multiple project workgroups within OCP, each including a project charter and a team working towards producing and enhancing the open source technologies within that project.**

III. 소프드웨어/하드웨어 분리

❖ **All Facebook data centers are 100% OCP**

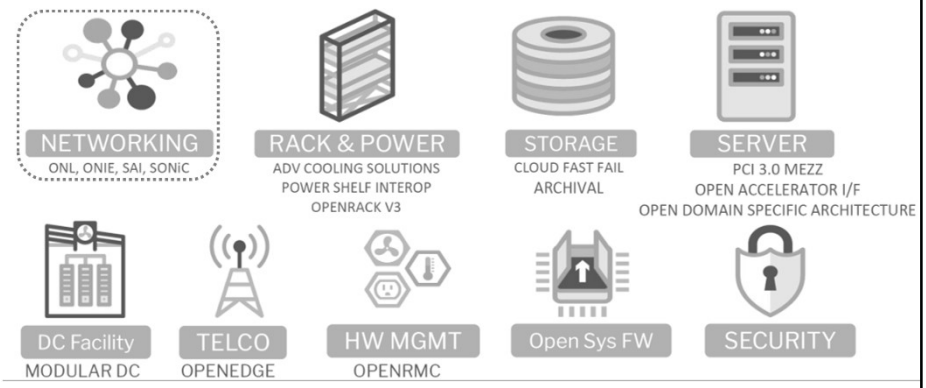


Facebook OCP Prineville DataCenter
PUE = 1.06

Typical DataCenter
PUE > 1.4

III. 소프드웨어/하드웨어 분리

❖ **OCP Projects**



III. 소프드웨어/하드웨어 분리

❖ **OCP Rack and Servers and Storage System**

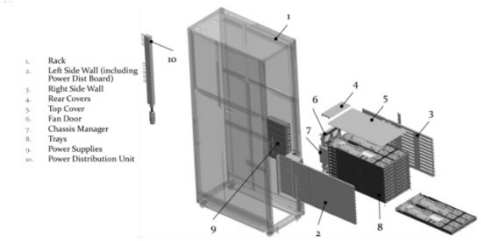
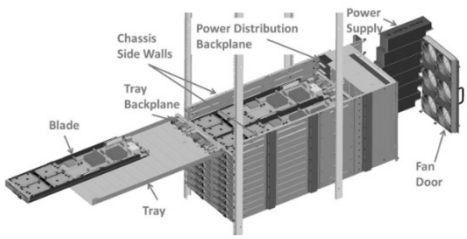
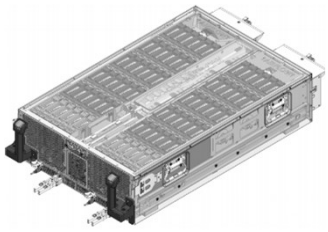


Figure 1: View of OCS with rack

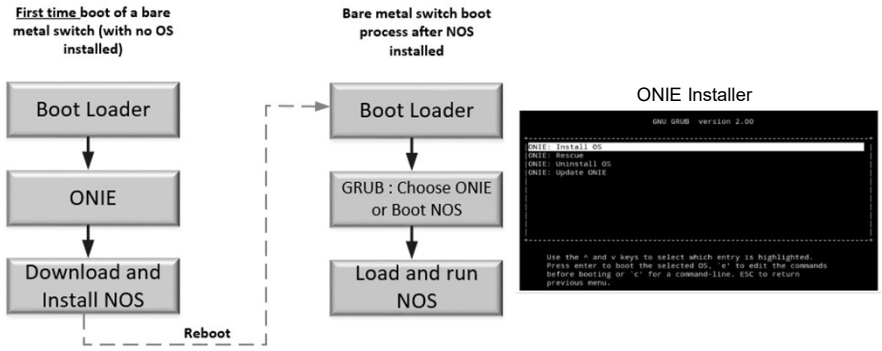




http://files.opencompute.org/oc/public.php?service=files&t=62279808dbea0cf380632c3042246979&path=/V2

III. 소프드웨어/하드웨어 분리

❖ **The Open Network Install Environment (ONIE)**
❖ **Bare Metal Switch to Boot Up Process and Execution of ONIE**



III. 소프드웨어/하드웨어 분리

❖ **How Ethernet Switches Are Built**

Chip Manufacturer	Chipset
Broadcom Inc.	• Strata SGX Family: ✓ Helix (1G) ✓ Trident 2, 2+, 3 (10G/40G) ✓ Tomahawk 2, 3 (100G/200G/400G) • Strata DNX Family (Large buffer): ✓ Qumran ✓ Jericho
Mellanox Technologies	Spectrum, Spectrum 2 (10G/40G/100G/200G/400G)
Cavium	XPliant (1G/10G/40G/100G)
Barefoot Networks	Tofino (10G/40G/100G)
Marvell Technology Group	Prestera switching family
Microsemi (Vitesse)	Gigabit switch chipsets
Intel Corporation	FM6000 series (10G/40G)

https://www.intel.com/content/www/us/en/ethernet-products/switch-silicon/ethernet-switch-fm5000-fm6000-datasheet.html

III. 소프드웨어/하드웨어 분리

❖ Ethernet Switches

•**Chassis**:

Just the metal part, with your choice of colors and assembly (For example, a pink or lipstick red would make it unique).

•**Power supplies**:

Two redundant AC-DC power supply systems, it's ready-made available from many factories.

•**Fans**:

Enough to cool down and dissipate the heat.

•**Control System**:

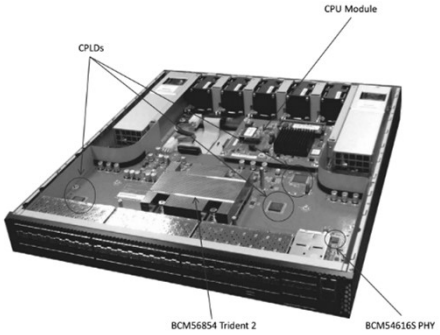
To control fans, system management.

•**CPU PCBA**:

An x86, Power PC ,or ARM-based processor with its RAM, flash, and PCIe, which runs the NOS.

•**Switch main board PCBA**:

The main board which hosts the main switch silicon, interface cages, CPLDs (Complex Programmable Logic Devices), and PHYs (in the case of RJ45 interfaces), this PCB normally has between 16 to 22 layers.



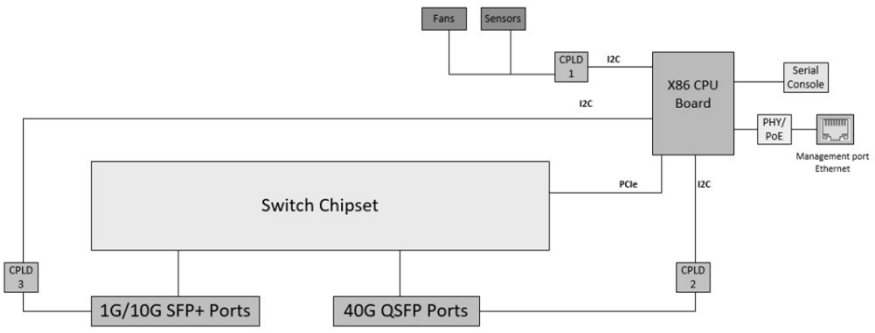
An Edge-Core AS5712 48 x 10G, 6 x 40G Switch

http://files.opencompute.org/oc/public.php?service=files&t=b96cbe20907b1a99edbecbeac3e92c4d

III. 소프드웨어/하드웨어 분리

❖ Ethernet Switches

Opensource Networking
james@jslab.kr



http://files.opencompute.org/oc/public.php?service=files&t=b96cbe20907b1a99edbecbeac3e92c4d

III. 소프드웨어/하드웨어 분리

❖ **Chipset Manufacturer**

Chip Manufacturer	Chipset
Broadcom Inc.	Strata SGX Family: • Helix (1G) • Trident 2, 2+, 3 (10G/40G) • Tomahawk 2, 3 (100G/200G/400G) Strata DNX Family (Large buffer): • Qumran • Jericho
Mellanox Technologies	Spectrum, Spectrum 2 (10G/40G/100G/200G/400G)
Cavium	XPliant (1G/10G/40G/100G)
Barefoot Networks	Tofino (10G/40G/100G)
Marvell Technology Group	Prestera switching family
Microsemi (Vitesse)	Gigabit switch chipsets
Intel Corporation	FM6000 series (10G/40G)

III. 소프드웨어/하드웨어 분리

❖ **Ethernet Switches**

Type	You can install any NOS	What is included with purchase	Hardware Support	NOS	NOS Support
Bare metal switch	Yes	Switch hardware only	Hardware manufacturer	Purchased separately	NOS vendor
White-box switch	Yes	Switch hardware only	Hardware manufacturer	Purchased separately	NOS vendor
Brite-box switch	Not supported by its vendor	Switch hardware and a NOS	Company selling the brite-box switch	Already included	Company selling the brite-box switch

https://www.dell.com/ae/business/p/open-networking-switches/pd

III. 소프드웨어/하드웨어 분리

❖ **Edge-Core Switches**

Switch Model	Main Ports	Switch Chipset
AS4610	48 x 1G RJ45	Broadcom Helix 4
AS5712	48 x 10G	Broadcom Trident 2
AS5812	48 x 10G	Broadcom Trident 2+
AS5912	48 x 10G	Broadcom Qumran-MX
AS6712	32 x 40G	Broadcom Trident 2
AS6812	32 x 40G	Broadcom Trident 2+
AS7816	64 x 100G	Broadcom Tomahawk 2
AS7712	32 x 100G	Broadcom Tomahawk
AS7512	32 x 100G	Cavium Xpliant
AS7900	32 x 400G	Broadcom Tomahawk 3

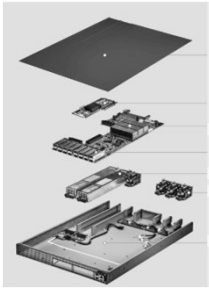
http://files.opencompute.org/oc/public.php?service=files&t=b96cbe20907b1a99edbecbeac3e92c4d

III. 소프드웨어/하드웨어 분리

❖ **Facebook Switches:**

Switch Model	Main Ports	Switch Chipset
Wedge	16 x 40G	Broadcom Trident 2
Wedge 100	32 x 100G	Broadcom Tomahawk
Backpack (Chassis-based)	128 x 100G	Broadcom Tomahawk
Wedge 100C	32 x 100G	Cavium Xpliant
Wedge 100B	32 x 100G / 65 x 100G	Barefoot Tofino T10

Wedge System Building Blocks



http://files.opencompute.org/oc/public.php?service=files&t=80dde82c251e1c3f5ad8ff7fd2675061

Facebook Wedge 100B based on Barefoot Networks Tofino Chip

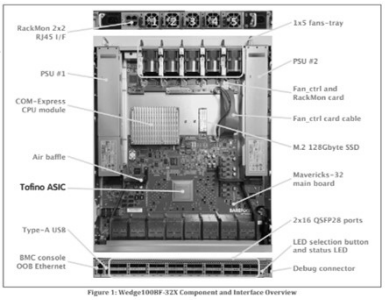


Figure 1: Wedge100BF-32X Component and Interface Overview

http://files.opencompute.org/oc/public.php?service=files&t=226d6609f6a58925148ed01289a8c517

III. 소프드웨어/하드웨어 분리

- ❖ **Edgecore Networks Wedge100-32X 100GbE**
- ❖ **Facebook - Wedge-100 Switch (19-in vs 21-in)**

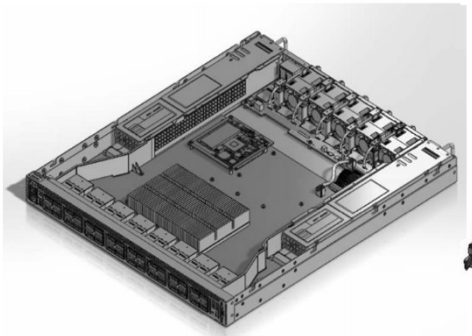


Figure 2: ISO view of Standard 19-in SKU

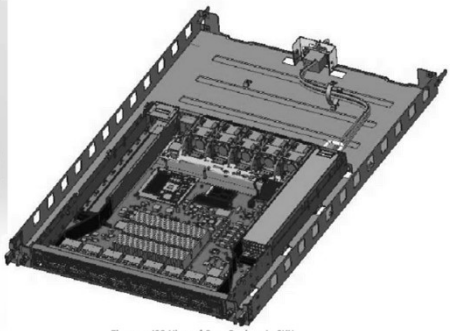
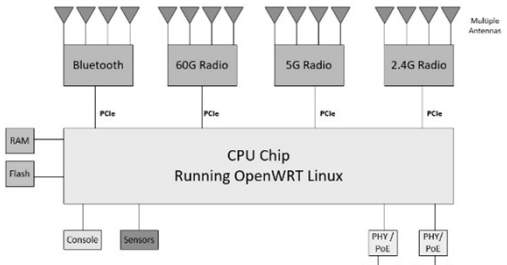


Figure 4: ISO View of OpenRack 21-in SKU

III. 소프드웨어/하드웨어 분리

- ❖ **Bare Metal Wireless Access Points**



Top View of ECW7212-L

Front View of ECW7212-L

http://files.opencompute.org/oc/public.php?service=files&t=cc7242c818159bfa2c1d69825eb6bb1b

III. 소프드웨어/하드웨어 분리

❖ **Mellanox Bare Metal Switches:**

Switch Model	Main Ports	Switch Chipset
MSX1410	48 x 10G / 12 x 40G	Mellanox Switch-X2
MSX1710	48 x 10G / 36 x 40G	Mellanox Switch-X2
SN 270	48 x 10G / 32 x 100G	Mellanox Spectrum

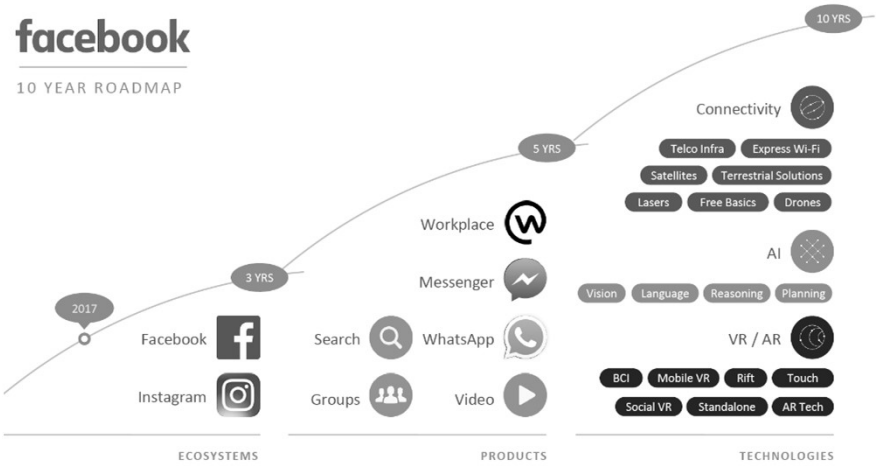
❖ **Barefoot Tofino-Based Switches:**

Switch Model	Main Ports	Switch Chipset
Wedge 100B	32 x 100G / 65 x 100G	Barefoot Tofino T10

http://files.opencompute.org/oc/public.php?service=files&t=80dde82c251e1c3f5ad8ff7fd2675061

III. 소프드웨어/하드웨어 분리

❖ **페이스북 로드맵:**



Opensource Networking
james@jslab.kr

Opensource Networking
james@jslab.kr

IV. IO 추상화와 Datapath

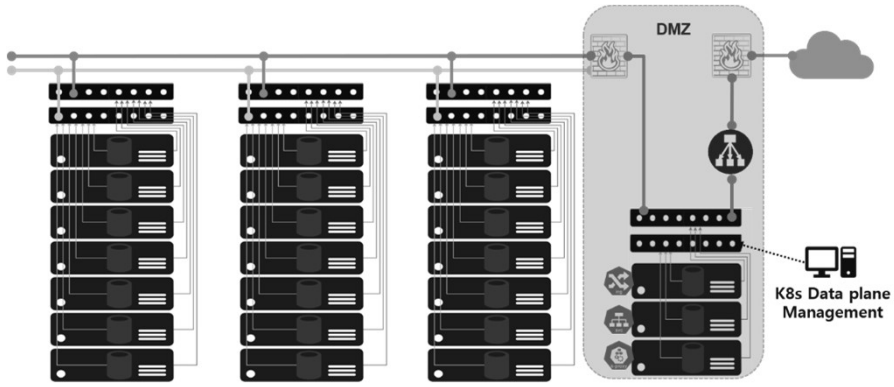
- **Types of Planes in a Network Device**
- **DPDK** (Data Plane Development Kit)
- **FD.io** (The Fast Data Project)
- **IO Visor Project**
- **Open vSwitch** (OVS)
- **OpenDataPlane** (ODP)
- **Open Container Initiative** (OCI)
- **Open Container Initiative and Open Virtualization Format**
- **SmartNICs**
- **Working with FPGAs**
- **Barefoot Networks Tofino Programmable Switch Silicon**

Opensource Networking
james@jslab.kr

https://www.linuxfoundation.org/projects/networking/

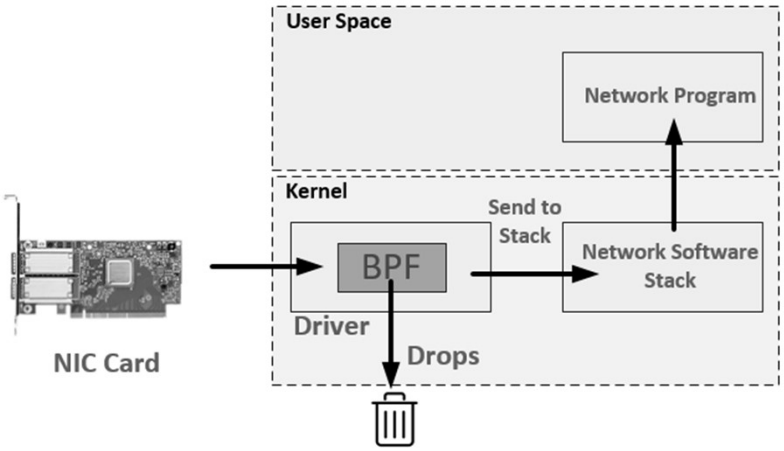
IV. IO 추상화와 Datapath

❖ **Data plane for cloud**



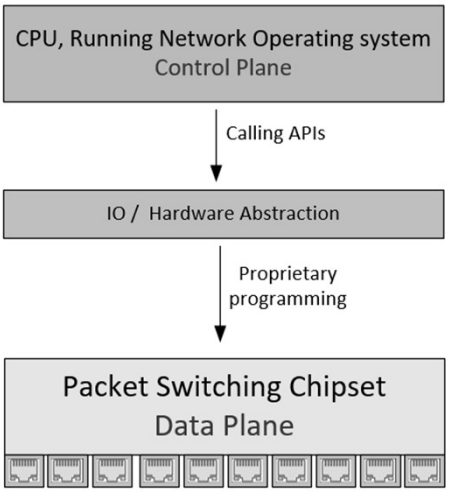
IV. IO 추상화와 Datapath

❖ **IO Visor - XDP**



IV. IO 추상화와 Datapath

❖ **Types of Planes in a Network Device:**



IV. IO 추상화와 Datapath

❖ **Types of Planes in a Network Device (Continued):**

Task	API call from Control Plane to hardware abstraction	Call from hardware abstraction to chipset
Create new VLAN ID 500	`vlan_id=500;` `Create_vlan(vlan_id);`	0x8721; 0x8734; 0x876772829283;

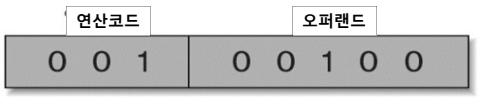
Add port eth1, eth2 to vlan 500	`vlan_id=500;`
	`port=1;`
	`vlan_add_port(vlan_id,port)`
	`port=2;`
	`vlan_add_port(vlan_id,port)`
		0x8722;
		0x8973;
		0x2389202;
		0x8722;
		0x8973;
		0x2389201;

IV. IO 추상화와 Datapath

❖ 기계명령어 형식 - 연산 코드(Opcode) / 오퍼랜드(Operand)

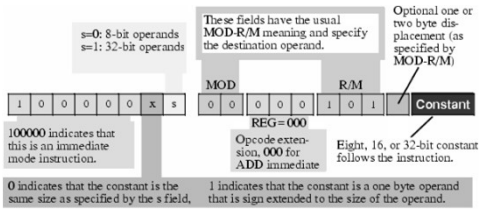
- Opcode: CPU가 수행할 연산을 지정해 주는 비트들 (Operation code)
- Operand: 데이터가 저장된 기억장치 주소 혹은 연산에 사용될 데이터 비트

8bit 기계어(예)



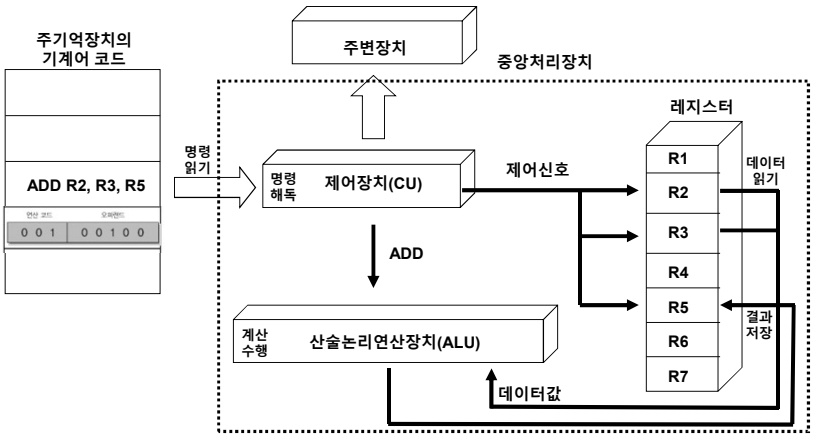
기계 명령어 형식

x86 Instructions



IV. IO 추상화와 Datapath

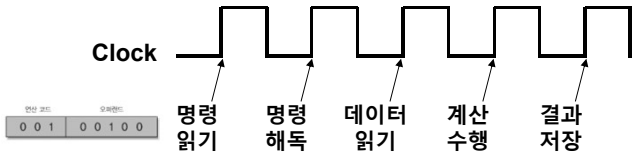
❖ 프로그램에 의한 중앙 처리 장치 동작 과정



IV. IO 추상화와 Datapath

❖ 중앙 처리 장치

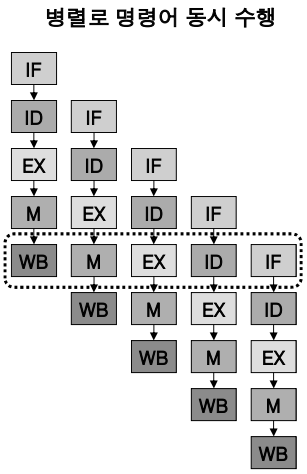
- 클럭 (Clock)
- 컴퓨터 동작을 위한 진동
- 중앙 처리 장치가 작업을 수행하는 단위
- 같은 종류의 CPU라면 초당 클럭 수가 많을 수록 속도가 빠름
- Opcode별 필요 클럭 수는 다를 수 있음



IV. IO 추상화와 Datapath

❖ **Pipeline 병렬 처리** (5 단계 예)

- **1 단계 : 명령읽기 IF(Instruction Fetch)**
  - 명령어를 메모리에서 가져옴
- **2 단계 : 명령해독 ID(Instruction Decode)**
  - 명령어를 해석
- **3 단계 : 계산수행 EX(Execution)**
  - 명령어 실행
- **4 단계 : 데이터 읽기 M(Memory access)**
  - 읽거나 쓸 메모리 특정 위치에 접근
- **5 단계 : 결과저장 WB(Write Back)**
  - 레지스터에 다시 씀



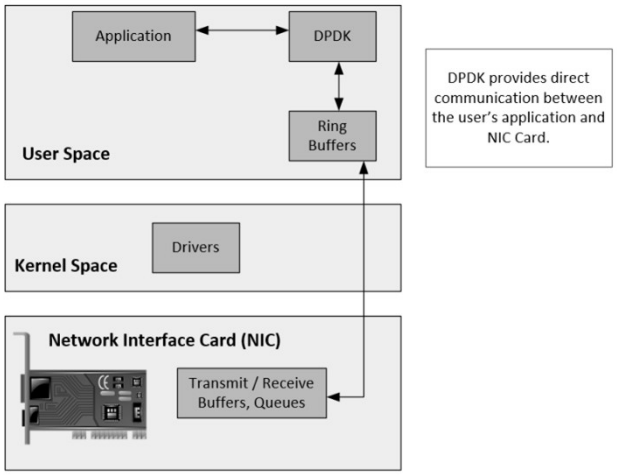
IV. IO 추상화와 Datapath

❖ **DPDK (Data Plane Development Kit):**

DPDK - Quick Summary	
Name	Data Plane Development Kit
By	The Linux Foundation
Where it runs	Linux, any x86, VM, or embedded system
What it does	Packet processing, routing, switching, encapsulation on the NIC card
Features	Environment-independent. Mostly used on appliances or x86 servers
What it can do out-of-the-box	You can build networking applications using DPDK libraries that can process packets at a high speed. DPDK APIs are very comprehensive, start from NIC functions such as bonding and network protocol parsing (Ethernet, ARP, ICMP, IPv4, IPv6, TCP, UDP, etc.), classification, QoS, ACL, etc.
Pros	DPDK is a low-level library and one of the most robust frameworks for packet processing. DPDK can achieve very high speeds in packet processing.
Caveats	Coding and using DPDK requires low-level C programming skills, requires the programmer to consider numerous parameters when using it.

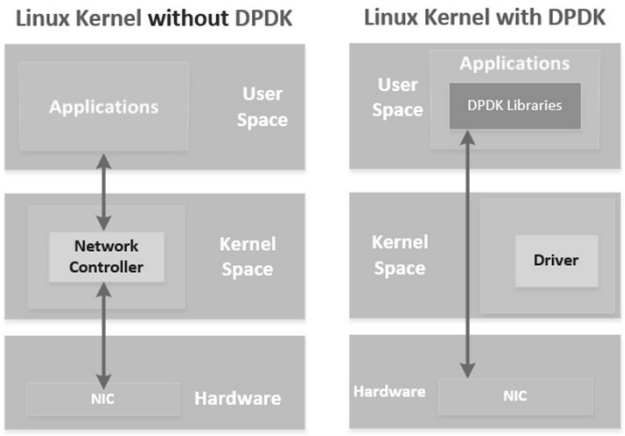
IV. IO 추상화와 Datapath

❖ **DPDK Setup and Architecture:**



IV. IO 추상화와 Datapath

❖ **DPDK Usage**



Differences between a System with and without DPDK

IV. IO 추상화와 Datapath

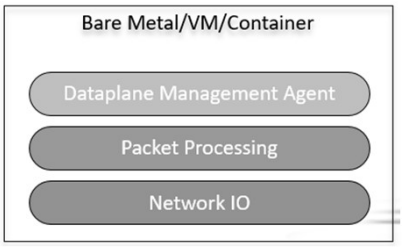
❖ **FD.io (The Fast Data Project)**

FD.io - Quick Summary	
Name	The Fast Data Project (FD.io)
By	The Linux Foundation
Where it runs	Linux, any x86, VM, or embedded system (Runs in the User Space)
What it does	Packet processing, routing, switching, NAT
Features	It uses vector packet processing (VPP) mechanisms to achieve high performance
What it can do out-of-the-box	FD.io's VPP provides a command line tool called vppctl, which can be used to interface with VPP to create interconnects, manage routing tables, create tunnel interfaces, manage hardware acceleration, etc. You can use FD.io APIs to build virtual switches, virtual routers, virtual firewalls, or other packet processing applications.

IV. IO 추상화와 Datapath

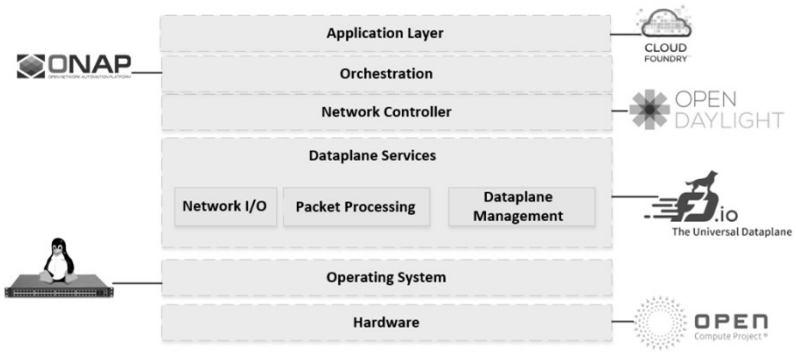
❖ **FD.io Components:**

- **Data Plane Management Agent:** An agent software that allows a Control Plane software or an SDN controller (such as OpenDaylight) to control and communicate with FD.io.
- **Packet Processing:** The packet processing engine of FD.io to classify, transform, prioritize, forward, terminate packets.
- **Network IO:** The hardware acceleration driver, connecting FD.io with the network hardware (for example, DPDK).



IV. IO 추상화와 Datapath

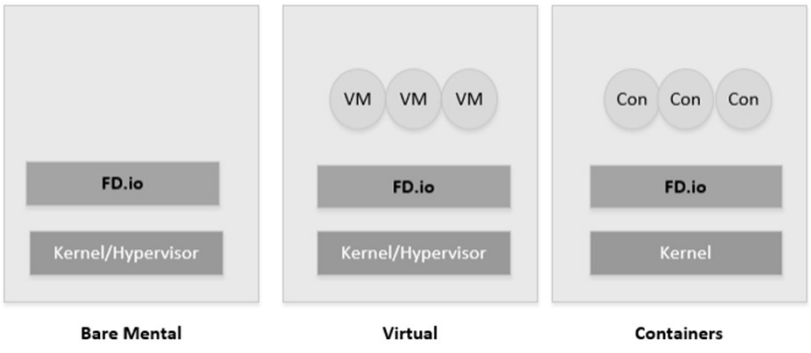
❖ **FD.io Communication with Other Networking Subsystems:**



IV. IO 추상화와 Datapath

❖ **FD.io can be used in servers to provide data plane functions to:**

- **Bare metal servers directly**
- **Virtual machines**
- **Containers**



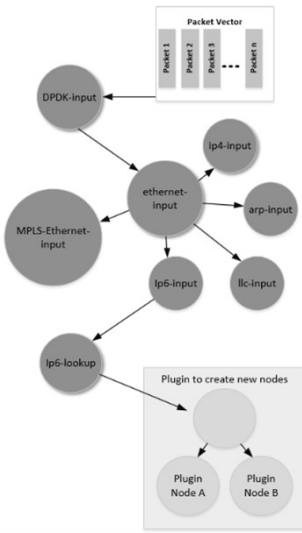
IV. IO 추상화와 Datapath

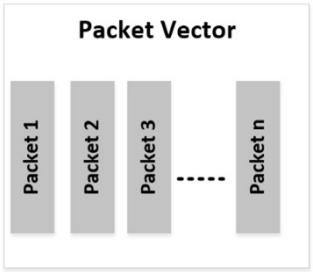
❖ **FD.io - Vector Packet Processing:** Vector Packet Processor (VPP) is the core component of FD.io. VPP can use DPDK for network IO and hardware-accelerated packet processing. VPP is a packet processing platform that can perform the following (below is a summarized list):

- **Interfaces:** VPP supports the following interfaces to be used as input an out: DPDK, TunTap, vhost.
- **Tunnels/Encapsulation:** VPP supports the following tunneling technologies: GRE, VXLAN, IPSec, MPLS over Ethernet or GRE, deep label stacks. User space applications can directly perform such encapsulation and decapsulation without the need to use the Linux kernel network controller.
- **Routing and switching (no routing protocol):** VPP provides direct access to many routing and switching features. Networking programs such as routing protocols (BGPD, OSPFD) will be able to use VPP to modify routing tables and other tasks: IPv4/IPv6, hierarchical FIB, VRFs, multi-paths, source RPF, segment routing, VLAN support, MAC learning, inbound ACL, proxy ARP.
- **Network services and security:** VPP supports NAT and other networking features and filters which can be used by security applications (source NAT, per-interface filters, DHCP, LLDP, BFD, policer, mirror/SPAN ports, IP flow export).

IV. IO 추상화와 Datapath

❖ **FD.io - Vector Packet Processing:**





IV. IO 추상화와 Datapath

❖ **How Does FD.io Relate to DPDK?**

- **FD.io provides an abstract environment for building virtual routers, switches and packet processors. FD.io can use DPDK to communicate directly with NIC cards. DPDK provides hardware acceleration to FD.io. However, hardware acceleration and usage of DPDK is optional - an FD.io-based application can still run without DPDK.**

IV. IO 추상화와 Datapath

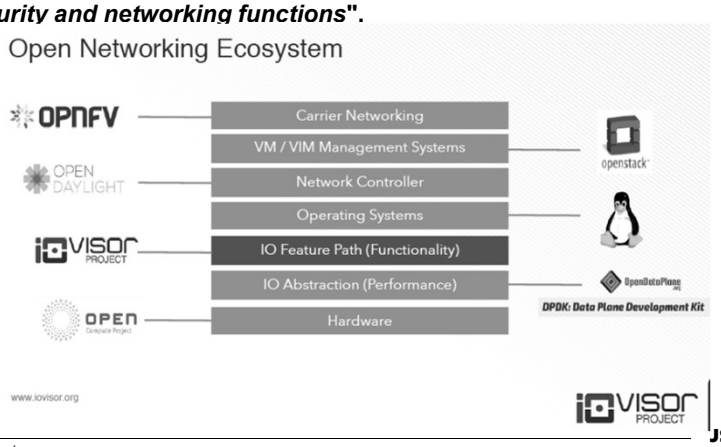
❖ **How Does FD.io Relate to IO Visor?**

- **The FD.io project and IO Visor (more about IO Visor in a little bit) are seen as complementary projects. The IO Visor Project focuses on dynamic runtime extensibility of data plane capabilities in the kernel. IO Visor aims to create a repository of IO modules that are portable across multiple possible data planes (like eBPF in the Linux kernel) and frameworks (like FD.io).**

IV. IO 추상화와 Datapath

❖ **IO Visor Project**

- "***The IO Visor Project is an open source project and a community of developers to accelerate the innovation, development, and sharing of virtualized in-kernel IO services for tracing, analytics, monitoring, security and networking functions***".



https://www.iovisor.org/

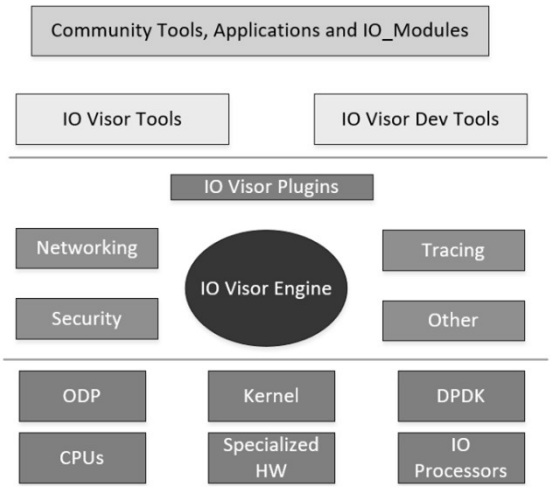
IV. IO 추상화와 Datapath

❖ **IO Visor Project**

IO Visor - Quick Summary	
Name	IO Visor Project
By	The Linux Foundation
Where it runs	Linux, any x86, VM, or embedded system (in-kernel)
What it does	Packet processing, routing, switching, NAT
Features	Supports three data path methods: XDP (eXpress Data Path), BCC (BPF Compiler Collection), and eBPF
What it can do out-of-the-box	Since IO Visor is mainly user in an in-kernel mode, it can help build robust networking applications for networking within a host, between a host and virtual machines or containers. It accelerates the in-host networking functions.

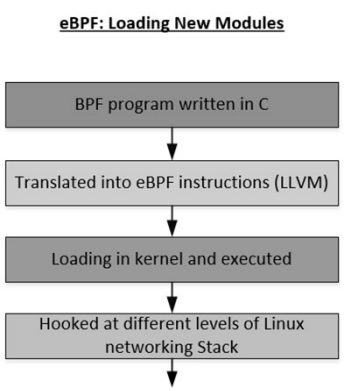
IV. IO 추상화와 Datapath

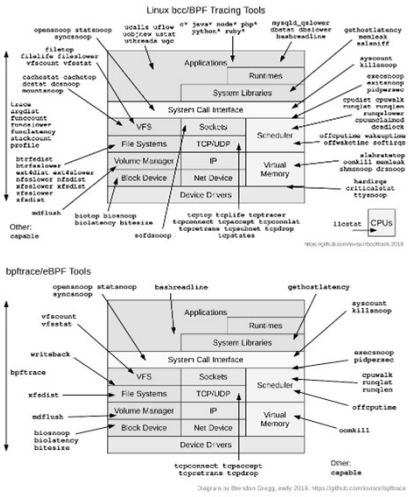
❖ **IO Visor Components**



IV. IO 추상화와 Datapath

❖ **IO Visor - eBPF**





http://www.brendangregg.com/ebpf.html

Opensource Networking
james@jslab.kr

IV. IO 추상화와 Datapath

❖ **eBPF:** extended Berkeley Packet Filter. eBPF 는 x86-64 와 arm64 의 공통점을 따온 것처럼 보이는 별도의 어셈블리 언어입니다. 실제로 이런 코드를 사람이 직접 작성하지는 않고, 성능 측정을 위한 C 언어 코드를 작성하면 이를 eBPF 프로그램으로 트랜스파일

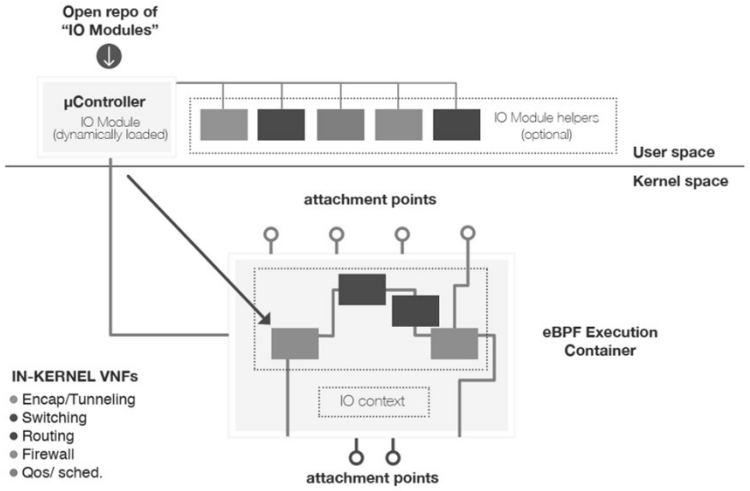
❖ **IO Visor – eBPF** (개발자의 생각 @ blog)**:** 운영 체제 수준에서 제공하는 기능인 eBPF와 이를 사용한 BCC 툴킷을 써서 실행 중인 서비스를 멈추거나 수정하지 않고도 성능을 측정하고 문제의 원인을 찾을 수 있음

- **가설 #1: 어디선가 CPU를 많이 쓴다.**
- **가설 #2: 데이터베이스가 충분히 빠른가?**
- **가설 #3: 다른 대기 코드가 있다**
- **가설 #4: DNS 조회가 오래 걸리고 있다**

https://blog.ifunfactory.com/2018/03/29/linux-%EA%B2%8C%EC%9E%84-%EC%84%9C%EB%B2%84-%EC%84%B1%EB%8A%A5-%EB%B6%84%EC%84%9D%EC%97%90-ebpf-bcc-%ED%99%9C%EC%9A%A9%ED%95%98%EA%B8%B0/?fbclid=IwAR1EaKA5gT9CbLWPBxOqSDzi9DDcEjbiFU8kO9dDfq1yycHtSa9ZSH3vWFA

IV. IO 추상화와 Datapath

❖ **Networking Based on IO Visor**



http://www.iovisor.org/wp-content/uploads/sites/8/2016/09/use_case_condensed.pdf

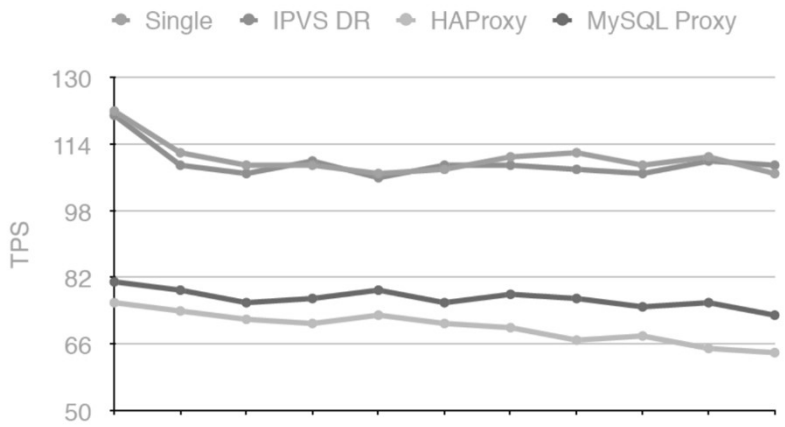
JS Lab

Opensource Networking

james@jslab.kr

IV. IO 추상화와 Datapath

❖**IP Virtual Server** (IPVS)
❖**리눅스 기반에 설치되는 서버로 L4 기능을 대체**



IPVS: (IP Virtual Server): 리눅스 커널에 내장한 트랜스포트 계층의 로드밸런싱

JS Lab

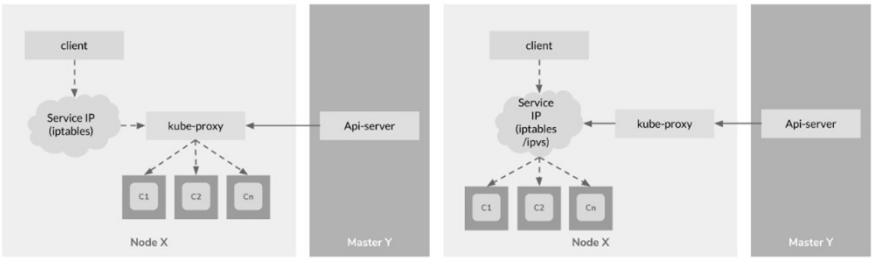
Opensource Networking

james@jslab.kr

IV. IO 추상화와 Datapath

❖ **Kubernetes Service Proxy Modes (Userspace, iptables, & ipvs)**

- **kubectl expose deployment** <application-name> **--type=**LoadBalancer **--name=**<service-name>

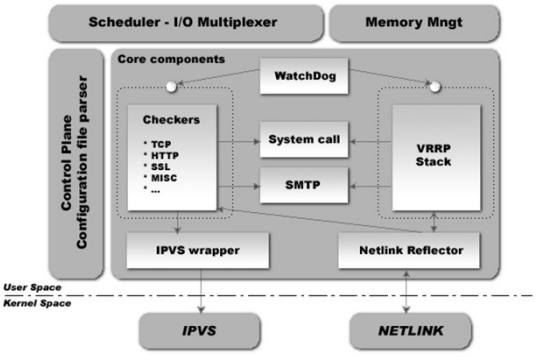


https://medium.com/containermind/a-beginners-guide-to-kubernetes-7e8ca56420b6

IV. IO 추상화와 Datapath

❖**IP Virtual Server** (IPVS)
❖**지원 기능**

- **HTTP_GET :** HTTP로 요청을 보내서 응답 확인
- **SSL_GET :** HTTPS로요청을 보내서 응답 확인
- **TCP_CHECK :** TCP로 접속할 수 있는지 여부를 확인
- **SMTP_CHECK :** SMTP로 HELO 명령을 보내서 응답 확인
- **MISC_CHECK :** 외부 명령을 실행해서 종료코드를 확인

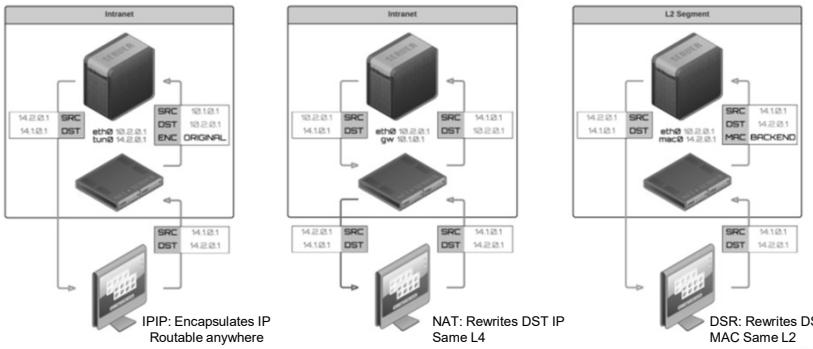


http://www.softfactory.org/architecture/loadbalancing/ipvs-ip-virtual-server

IV. IO 추상화와 Datapath

❖IP Virtual Server @ Docker

- Works inside the Linux Kernel, based onNetfilter.
- Supports TCP, SCTP & UDP, v4 and v6.
- 8+ methods: WRR, WLC, LBLCR, SH and much more – plugins.
- NAT, Tunneling, Direct Routing.
- Address bundling via FWMark services.



IPIP: Encapsulates IP Routable anywhere

NAT: Rewrites DST IP Same L4

DSR: Rewrites DST MAC Same L2

Round Robin(rr) Weighted Round Robin(wrr) Least Connection(lc) Weighted Least Connection(wlc) Locality-Based Least Connection(lblc) Source Hashing (sh)

JS Lab

Opensource Networking

james@jslab.kr

IV. IO 추상화와 Datapath

❖ Open vSwitch (OVS)

OVS - Quick Summary	
Name	Open Virtual Switch (OVS)
By	The Linux Foundation
Where it runs	Linux
What it does	Virtual switch/router with physical and virtual interfaces
Features	L2 switching, VXLAN encapsulation, distributed switches controlled by a co ntroller
What it can do out-of-the-box	OVS is one of the core components of virtualization hypervisors

https://www.openvswitch.org/

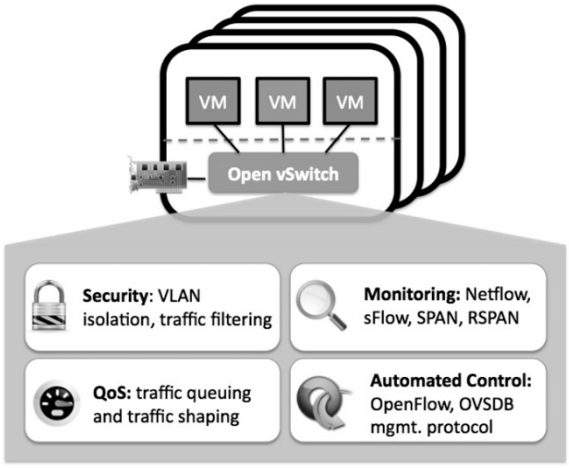
JS Lab

Opensource Networking

james@jslab.kr

IV. IO 추상화와 Datapath

❖ **Open vSwitch (OVS)**



https://www.openvswitch.org/

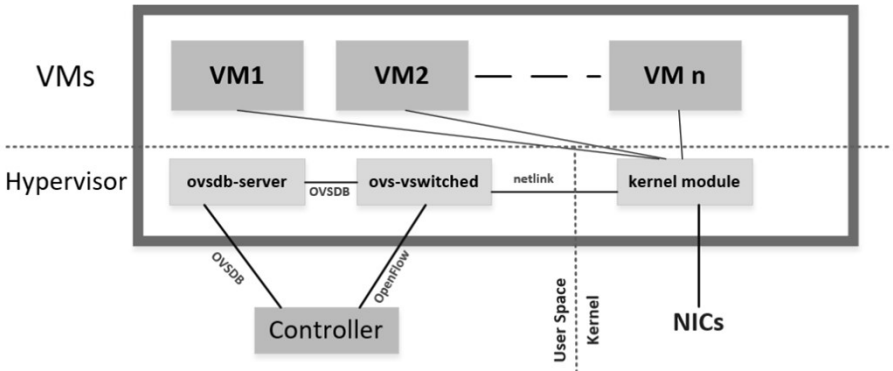
JS Lab

Opensource Networking

james@jslab.kr

IV. IO 추상화와 Datapath

❖ **Open vSwitch (OVS) Architecture**



https://www.openvswitch.org/

JS Lab

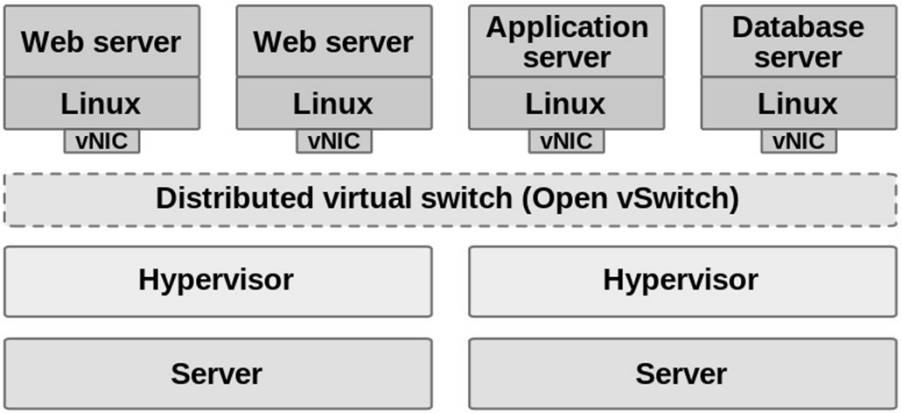
Opensource Networking

james@jslab.kr

IV. IO 추상화와 Datapath

❖ **Open vSwitch (OVS)**

❖ **Distributed Open vSwitch Instance**



https://www.openvswitch.org/

IV. IO 추상화와 Datapath

❖ **Open vSwitch (OVS)**

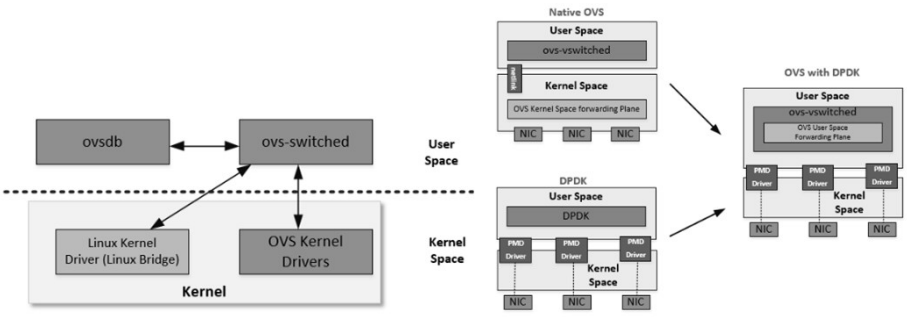
❖ **OVS - Supported Features**

- Visibility into inter-VM communication via NetFlow, sFlow(R), IPFIX, SPAN, RSPAN, and GRE-tunnelled mirrors
- LACP (IEEE 802.1AX-2008)
- Standard 802.1Q VLAN model with trunking
- Multicast snooping
- IETF Auto-Attach SPBM and rudimentary required LLDP support
- BFD and 802.1ag link monitoring
- STP (IEEE 802.1D-1998) and RSTP (IEEE 802.1D-2004)
- Fine-grained QoS control
- Support for HFSC qdisc
- Per VM interface traffic policing
- NIC bonding with source-MAC load balancing, active backup, and L4 hashing
- OpenFlow protocol support (including many extensions for virtualization)
- IPv6 support
- Multiple tunnelling protocols (GRE, VXLAN, STT, and Geneve, with IPsec support)
- Remote configuration protocol with C and Python bindings
- Kernel and user-space forwarding engine options
- Multi-table forwarding pipeline with flow-caching engine
- Forwarding layer abstraction to ease porting to new software and hardware platforms.

Opensource Networking
james@jslab.kr
JS Lab
https://www.openvswitch.org/

IV. IO 추상화와 Datapath

- ❖ **OVS vs Linux Bridge**
- ❖ **OVS supports advanced features and distributed environment comparing to standard Linux bridge.**



Integration of DPDK Data Plane with Open vSwitch

https://www.openvswitch.org/

IV. IO 추상화와 Datapath

- ❖ **OpenDataPlane (ODP)**

ODP - Quick Summary	
Name	OpenDataPlane
By	The independent open source community is mainly driven by the Linaro Network Group
Where it runs	An abstraction layer, runs on x86, ARM and other embedded SoCs
What it does	Robust network programming for embedded systems
Features	Provides networking APIs
What it can do out-of-the-box	ODP is similar to DPDK in x86, but it supports embedded systems and SoCs. ODP provides a standard network abstraction for developers to write their applications. Applications can be ported between different hardware by re-compiling them. You can implement applications such as NAT, switching, routing, classifications, IPsec encapsulation using ODP over the supported platforms.

https://www.opendataplane.org/about/

IV. IO 추상화와 Datapath

❖ OpenDataPlane (ODP)

❖ ODP is supported on the following chipsets:

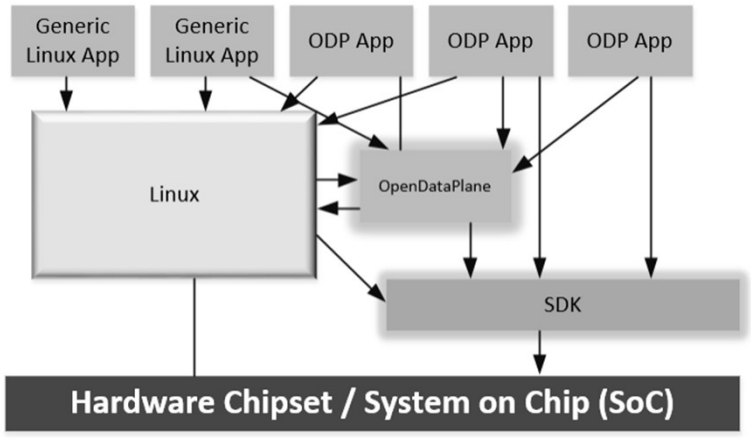
- Cavium Octeon™ SoCs
- Cavium ThunderX™ SoCs
- Kalray MPPA (Massively Parallel Processor Array)
- Freescale QorIQ – ARM-based DPAA2 architecture LS2080, LS2085
  QorIQ – ARM and PowerPC-based DPAA architecture LS1043
- Texas Instruments (TI): Keystone II SoCs
- Marvell ARMADA SoC implementation
- Linaro ODP-DPDK - software-optimized implementation using DPDK
- NXP QorIQ SoCs
- HiSilicon platforms.

https://www.opendataplane.org/about/

IV. IO 추상화와 Datapath

❖ OpenDataPlane (ODP)

❖ ODP and other components:

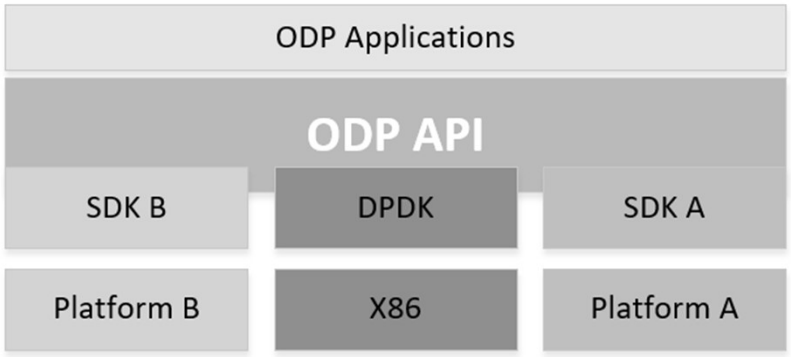


Relationship between OpenDataPlane and other Components

https://www.opendataplane.org/about/

IV. IO 추상화와 Datapath

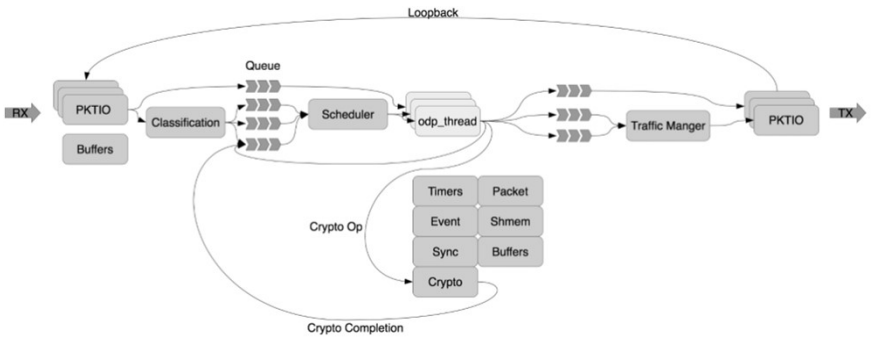
- ❖ **OpenDataPlane (ODP)**
- ❖ **ODP Implementations:**



https://www.opendataplane.org/about/

IV. IO 추상화와 Datapath

- ❖ **OpenDataPlane (ODP)**
- ❖ **ODP Use Cases:** an example ODP flow, from receiving a packet, to processing and sending out.



https://github.com/Linaro/odp-dpdk/blob/master/doc/images/packet_flow.svg

IV. IO 추상화와 Datapath

❖ **Open Container Initiative (OCI)**

OCI - Quick Summary	
Name	Open Container Initiative
By	The Linux Foundation
Where it runs	Linux
What it does	Standardizes the packaging and running of containers
Features	Currently, provides specs for exporting containers and running containers
What it can do out-of-the-box	OCI includes a software called runC, which can be used to interact with containers in different container platforms, such as Docker, Kata or Linux containers.

Opensource Networking
james@jslab.kr

https://www.opencontainers.org/

IV. IO 추상화와 Datapath

❖ **Open Container Initiative (OCI)**

❖ **Two specifications defined and developed by OCI:**

- **Runtime Specification (runtime-spec)**
- **Image Specification (image-spec)**

Opensource Networking
james@jslab.kr

JS Lab

https://www.opencontainers.org/

IV. IO 추상화와 Datapath

- Open Container Initiative (OCI)
- OCI - Filesystem Bundle
  - The filesystem bundle specification defines a format for encoding and saving a container's set of files. This bundle includes all the files, data, and metadata required to run a container.
  - The standard container bundle contains all the files and information needed to load and run a container package. This includes a configuration file that references the locations of the container root filesystem and other files related to the container, as well as the container's main filesystem root.

IV. IO 추상화와 Datapath

- Open Container Initiative (OCI)
- OCI - Runtime Bundle
  - The OCI runtime bundle is a standard specification for creating a bundle directory that includes all of the files required to launch an application as a container. The bundle contains an OCI configuration file where it can specify host-independent details such as which executable to launch and host-specific settings such as mount locations, hook paths, Linux namespaces and cgroups. Because the configuration includes host-specific settings, application bundle directories copied between two hosts may require configuration adjustments.

IV. IO 추상화와 Datapath

❖ **Open Container Initiative (OCI)**

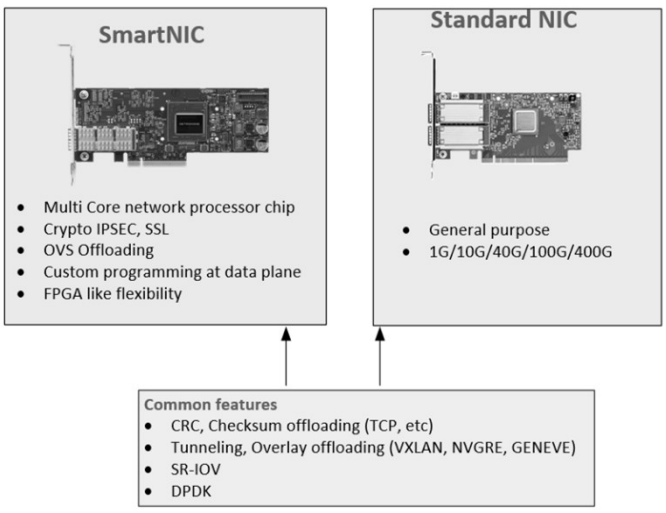
❖ **Open Container Initiative and Open Virtualization Format**

- **You may find that the Open Container Initiative provides a similar function as the Open Virtualization Format (OVF) initiative did a few years ago. With OVF, you could export a virtual machine from a hypervisor (for example Xen) as an OVF file, and import the OVF file into another hypervisor, such as VMware.**
  - ✓ OVF defined open standards for virtual machines and hypervisors.
  - ✓ OCI defines open standards for containers and container engines.

https://www.opencontainers.org/

IV. IO 추상화와 Datapath

❖ **SmartNICs**



https://www.opencontainers.org/

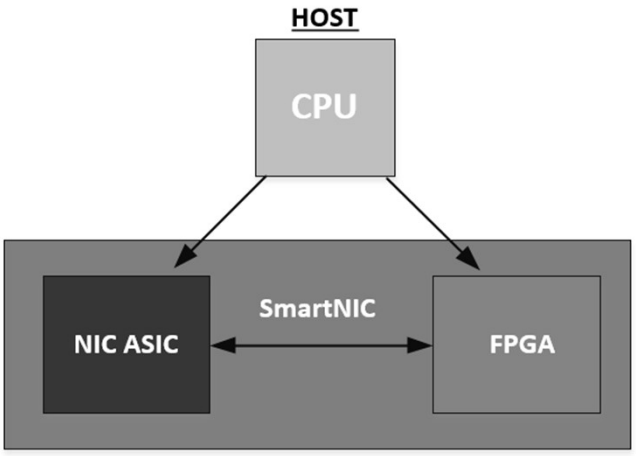
IV. IO 추상화와 Datapath

❖ **SmartNICs**

SmartNICs - Quick Summary	
Name	Smart Network Interface Cards (SmartNICs)
By	Multiple manufacturers, such as Netronome, Napatech, Mellanox Technologies, etc.
Where it runs	It's a physical PCIe card
What it does	Offloads packet processing from the host, processes packets at hi gh speed in NICs
Features	L2, L3 packet processing, custom application, etc.
What it can do o ut-of-the-box	You can build custom applications that run on SmartNIC chipsets (for example, a DDOS protector, IPS or packet encapsulators for I Psec, SSL, firewall - NGFW -, load balancing, etc.

IV. IO 추상화와 Datapath

❖ **SmartNICs**



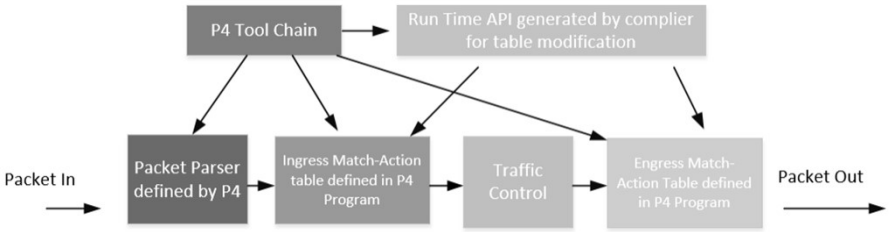
Opensource Networking
james@jslab.kr
JS Lab
https://www.opencontainers.org/

IV. IO 추상화와 Datapath

❖ **SmartNICs**

❖ **SmartNICs Programmability**

- **SmartNICs can be programmed using their SDK or using the standard P4 language. Most of SmartNICs now support P4 programming. However, they need the vendor's compiler and tool chain in order to compile the P4 program. P4 programs are portable - you can compile the same P4 application for different SmartNIC models by compiling the P4 application using the relevant vendors' compiler.**



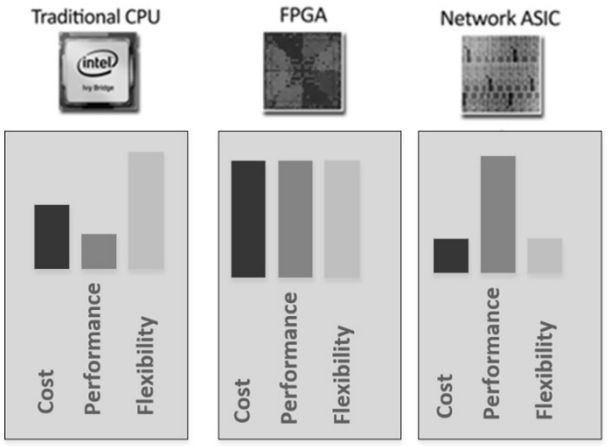
IV. IO 추상화와 Datapath

❖ **FPGAs and Xilinx SDNet**

SDNet - Quick Summary	
Name	SDNet
By	Xilinx Inc.
Where it runs	On Xilinx FPGA
What it does	SDNet is a library used to build networking applications in Xilinx FPGA to implement fast packet processing in FPGA at different speeds, such as 1Gbps, 10Gbps, 100Gbps, etc.
Features	Any packet processing function, such as switching, routing, classification, packet manipulation, ACL, etc.
What it can do out-of-the-box	You can use SDNet (which now supports P4 languages) to build your packet processing application inside a Xilinx FPGA chip. You can build a router, stateful firewall, IPs, IDs, DDOS protector, MPLS encapsulator, VXLAN encapsulator and router, etc.
Pros	You can build packet processing applications that run at very high speed, from 1Gbps to 400Gbps. Support of P4 language makes it easy to work with FPGA.
Caveats	FPGAs that can process packets at high speeds are generally expensive. Building applications for FPGA requires FPGA skills and knowledge, and the coding requires attention to numerous aspects.

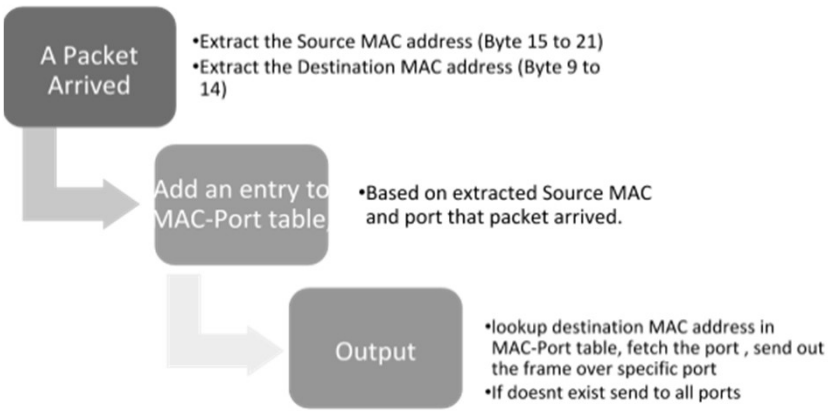
IV. IO 추상화와 Datapath

- ❖ FPGAs and Xilinx SDNet
- ❖ Key Differences between CPU, ASIC and FPGA



IV. IO 추상화와 Datapath

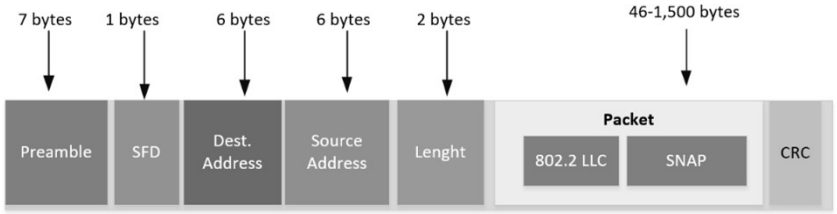
- ❖ Working with FPGAs
- ❖ Building a Basic Layer 2 Switch Using an FPGA



IV. IO 추상화와 Datapath

❖ **Working with FPGAs**

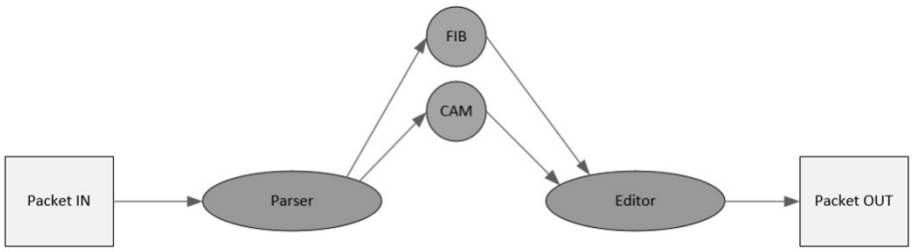
- **To build an FPGA-based network application, you don't need to extract all fields of Ethernet, IP, etc. You only need to extract and compare what is relevant for you. For example, if you are building a DDOS protection application, you need to extract only fields such as the Source & Destination IP address and TCP ports (or, in some advanced methods, a pattern), and compare them against a predefined list in your FPGA. If the packet matches that pattern, FPGA should just drop the packet.**



IV. IO 추상화와 Datapath

❖ **Xilinx SDNet:** Xilinx has created a framework called SDNet, which allows developers to build data plane programs to run in their FPGA chips. Initially, it was only supporting the Xilinx's proprietary SDK (SDNet). However, it now supports the standard P4 language.

- **To design a data plane application that runs on an FPGA, you can start by drawing the state machine or a simple flow diagram showing how you would like your application to work. You can take a look at the following illustration of a simple routing application:**



Simple Routing Application

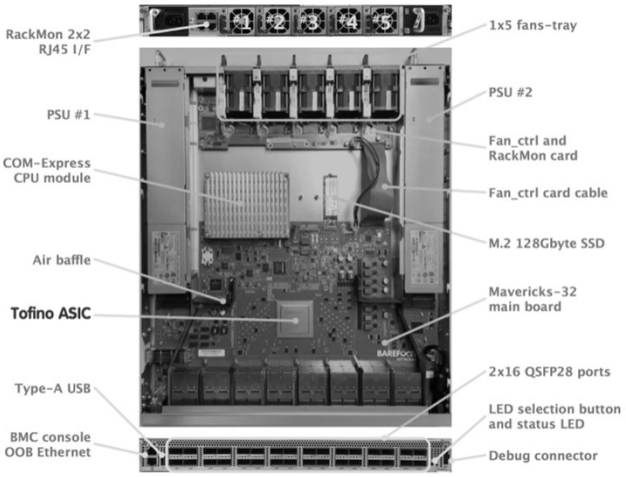
IV. IO 추상화와 Datapath

❖ Barefoot Networks Tofino Programmable Switch Silicon

Barefoot Networks Tofino Switch Chipset - Quick Summary	
Name	Barefoot Network Tofino Switch Chipset
By	Barefoot Networks
Where it runs	A physical chipset (ASIC) for Ethernet switches
What it does	Apart from the standard features of an Ethernet switch silicon, it adds flexibility for data plane programming in a switch by supporting the P4 language and allowing to directly program the data plane
Features	Up to 6.5 Tbps; supports 32 x 100G ports; supports P4 language
What it can do out-of-the-box	Tofino is a switch chipset which can be used to build an Ethernet switch. Hardware vendors can use this chipset to build an off-the-shelf Ethernet switch. Software vendors can use the Barefoot software tools to build networking software that can run and drive the chipset, as well as use the chipset's advanced packet processing features.

IV. IO 추상화와 Datapath

❖ Barefoot Networks Tofino Programmable Switch Silicon



Opensource Networking
james@jslab.kr
JS Lab
https://www.opencontainers.org/

Opensource Networking
james@jslab.kr

V. Network Operating Systems

- **OpenSwitch (OPX)**
- **Disaggregated Network Operating System (DANOS)**
- **Stratum**
- **Open Network Linux (ONL)**
- **Free Range Routing (FRR)**
- **P4 Language**
- **SONiC (Software for Open Networking in the Cloud)**
- **FBOSS (Facebook Open Switching System)**
- **Cumulus Linux**

Opensource Networking
james@jslab.kr

https://www.linuxfoundation.org/projects/networking/

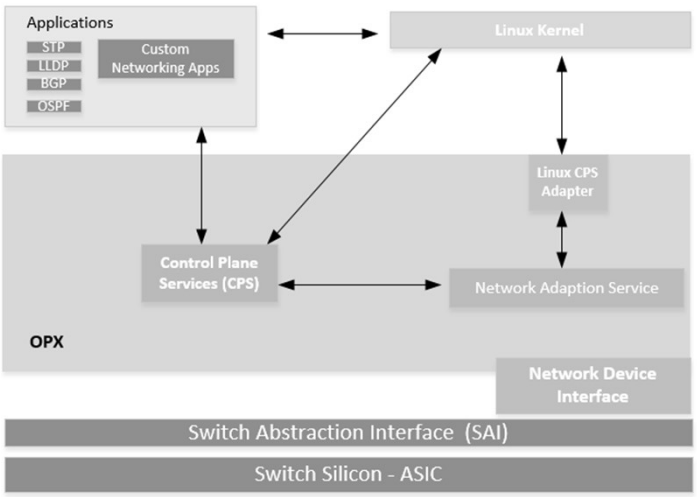
V. Network Operating Systems

❖ **OpenSwitch (OPX) Architecture:**

OPX - Quick Summary	
Name	OpenSwitch (OPX)
By	The Linux Foundation
Where it runs	On a compatible bare metal switch, mostly Dell EMC bare metal switches
What it does	Layer 2, Layer 3 routing and switching
Features	Layer 2 and Layer 3 features, BGP, OSPFv2/3, VRRP, VRF, ACL, QoS policing, shaping, automation with Ansible, Puppet, Chef, Python
What it can do out-of-the-box	You can load and use OpenSwitch on a compatible hardware and use it as a standard L2/L3 switch. It supports automation out-of-the-box, which can help you build an infrastructure as a code platform. OpenSwitch has some command line utilities, but it uses standard Linux commands for routing configuration.

V. Network Operating Systems

❖ **OpenSwitch (OPX) Architecture:**



Opensource Networking
james@jslab.kr
JS Lab
https://www.openswitch.net/

V. Network Operating Systems

❖ OpenSwitch (OPX) Commands:

Create VLAN 500, add Port 1 and Port 5 as tagged interfaces:

```
$ brctl addbr br500
$ ip link add link e101-001-0 name e101-001-0.500 type vlan id 500
$ ip link add link e101-005-0 name e101-005-0.500 type vlan id 500
$ brctl addif br500 e101-001-0.500
$ brctl addif br500 e101-005-0.500
```

Create a static route:

```
$ ip route add 10.0.0.0/24 via 192.168.1.2
$ ip route show
```

OpenSwitch has also its own utilities, such as:

- Enable logging and debugging
  opx_logging_cli enable
- Show system alarms
  opx-show-alms
- Show version
  opx-show-version
- Enable SAI-specific logging
  opx-switch-log

V. Network Operating Systems

❖ OpenSwitch (OPX) Supports Automation:

```
node 'Switch1.LNF.org' {
$int_enabled = true
$int_loopback = '10.0.0.1'
$int_layer3 = {
e101-001-0 => {'int'=>'e101-001-0', 'address' => '10.1.1.1', 'netmask' =>
'255.255.255.0', 'cidr_netmask' => 24},
e101-002-0 => {'int'=>'e101-002-0', 'address' => '10.2.2.2', 'netmask' =>
'255.255.255.0', 'cidr_netmask' => 24},
}
$bgp = {
myasn => 65002,
peergroupv4 => [ { name => 'Router-101', asn => 65000, peers => [ '192.168.0.2','192.168.10.2' ]
} ]
}
include ibgp::switch
}
```

V. Network Operating Systems

❖ **Disaggregated Network Operating System (DANOS):**

DANOS - Quick Summary	
Name	Disaggregated Network Operating System (DANOS)
By	The Linux Foundation
Where it runs	Not yet disclosed, but expected ti run on a compatible bare metal switch
What it does	As per the AT&T whitepaper, it supports Layer 2 and Layer 3 routing and sw itching
Features	Not yet disclosed, but expected to have Layer 2, Layer 3, BGP, OSPFv2/3, VRRP, VRF, ACL, QoS, MPLS, MP-BGP, and other BGP extensions
What it can do out-of-the-box	Not yet disclosed or available

https://www.danosproject.org/

V. Network Operating Systems

❖ **Disaggregated Network Operating System (DANOS):**

- **Supports ONIE for installation environment**
- **Supports the P4 language, and possibly, data plane programming**
- **Support SDN agents such as OpenFlow**
- **Supports automation and YANG models**
- **Supports a CLI**
- **Uses Forwarding Abstraction Layer (FAL) for communication with switch silicon**
- **Will be orchestrated with ONAP**

https://www.openswitch.net/

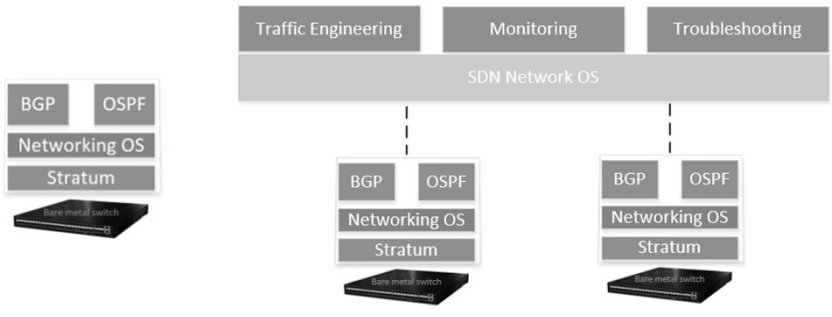
V. Network Operating Systems

❖ **Stratum:**

Stratum - Quick Summary	
Name	Stratum
By	The Linux Foundation - Open Networking Foundation
Where it runs	On a compatible bare metal switch
What it does	Not fully disclosed. Stratum launched in March 2018 - it is a NOS that can ru n a switch as an L2/L3 or fully managed by a network controller
Features	Traditional L2/L3, data plane programming using P4, OpenConfig, SDN age nt
What it can do out-of-the-box	Although Stratum is still in an incubation stage, it is a NOS running on bare metal switches. Stratum is considered as a NOS that you can use to slowly migrate your network to a fully SDN, traffic-engineered network. Stratum can run traditional networking protocols such as BGP, STP, OSPF, etc., while all owing you to program the switches using OpenFlow.

V. Network Operating Systems

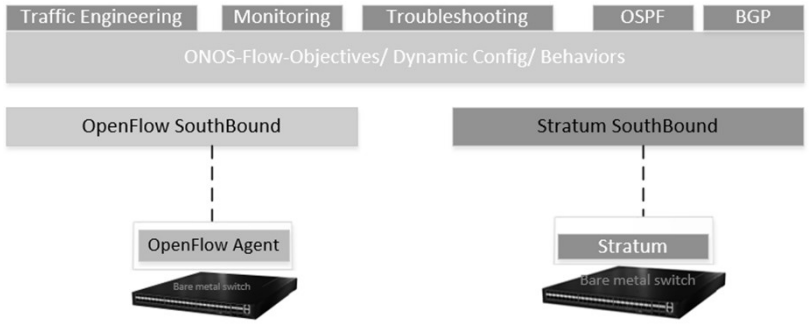
❖ **Stratum:**



Stratum Supports Hybrid Mode

V. Network Operating Systems

❖ **Stratum:**



Stratum Supports a Full SDN-based System

https://stratumproject.org/

V. Network Operating Systems

❖ **Open Network Linux (ONL):**

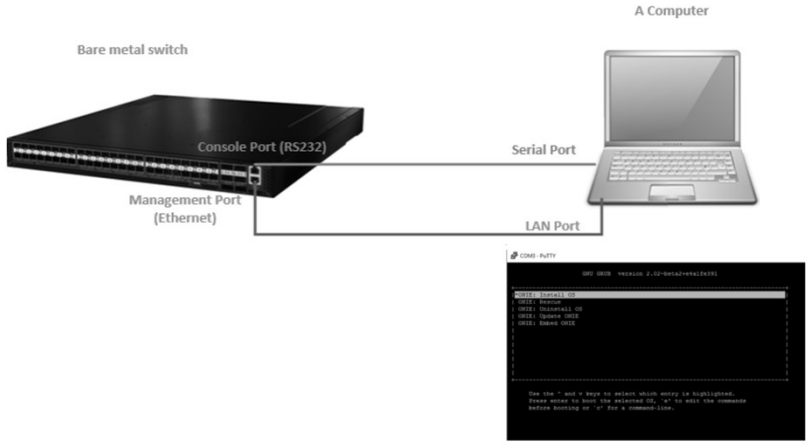
ONL - Quick Summary	
Name	Open Network Linux (ONL)
By	OCP and Open Network Linux Community, backed by Big Switch Networks, Inc.
Where it runs	On a compatible bare metal switch
What it does	ONL is a base-level operating system that can boot a full fledged Linux on a bare metal switch
Features	ONL is based on Debian Wheezy. It allows the installation of any Linux-compatible applications or the ASIC driver
What it can do out-of-the-box	ONL includes a comprehensive HCL that allows you to install it on most bare metal switches. After installing ONL, you can download the switch silicon drivers from the switch chipset manufacturer's website, and install them on ONL. This will enable ONL to control the switch ASIC. In Broadcom-based bare metal switches, a very common method is to download Broadcom's OFDPA (OpenFlow Data Plan Architecture) or OpenNSL (Open Network Switch Library) SDKs and deploy them on ONL. This is a simple way to make an OpenFlow-capable switch.

http://opennetlinux.org/

V. Network Operating Systems

❖ **Installing Open Network Linux (ONL):**

```
install_url tftp://172.17.172.103/latest-amd64.installer
```



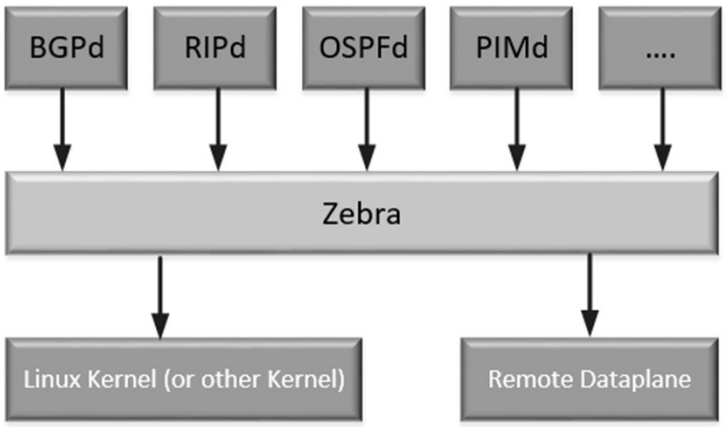
V. Network Operating Systems

❖ **Free Range Routing (FRR):**

FRRouting - Quick Summary	
Name	Free Range Routing (FRR)
By	The Linux Foundation
Where it runs	On a Linux host, or a bare metal switch that runs Linux
What it does	IP routing protocol suite for Linux
Features	In addition to the Quagga features, it provides VRF, EVPN, LSP, BFD, LDP for MPLS, CLI, support for JSON outputs
What it can do out-of-the-box	You can use FRR out-of-the-box on a Linux system to build a full-fledged router. There are other use cases as well, such as: 1. Routing on a host (on a compute node) to establish BGP with data center switches 2. Using FRR on an ONL to build an L3 Ethernet switch (requires additional components to program the switch chipset)

V. Network Operating Systems

❖ **Free Range Routing (FRR) Architecture :**



https://frrouting.org/

V. Network Operating Systems

❖ **P4 Language:**

P4 - Quick Summary	
Name	P4
By	P4 Community, joined the Open Networking Foundation and The Linux Foundation (March 2018)
What it does	Language to define the behavior of data planes
Features	Supports the match-and-action model with flexibility of matching different headers in a packet or a frame
What it can do out-of-the-box	P4 is a language used to describe the behavior of a data plane. You can build various applications in a data plane, such as DDOS protections, firewalls, IPS, IDs, etc.

V. Network Operating Systems

❖ **P4 Language:**

Headers

```
header ethernet_t {
 macAddr_t dstAddr;
 macAddr_t srcAddr;
 bit<16>   etherType;
}
header ipv4_t {
 bit<4>    version;
 bit<4>    ihl;
 bit<8>    diffserv;
 bit<16>   totalLen;
 bit<16>   identification;
 bit<3>    flags;
 bit<13>   fragOffset;
 bit<8>    ttl;
 bit<8>    protocol;
 bit<16>   hdrChecksum;
 ip4Addr_t srcAddr;
  ip4Addr_t dstAddr;
 }
```

Parsers

```
parser MyParser(packet_in packet,
  out headers hdr,
  inout metadata meta,
  inout standard_metadata_t std_meta) {
 state start {
   packet.extract(hdr.ethernet);
   transition accept;
  }
}
```


V. Network Operating Systems

❖ **P4 Language:**

Tables

```
table ipv4_lpm {
  key = {
   hdr.ipv4.dstAddr: lpm;
  }
  actions = {
   ipv4_forward;
   drop;
   NoAction;
  }
  size = 1024;
  default_action = NoAction();
}
```

Actions

```
control MyIngress(inout headers hdr,
          inout metadata meta,
          inout standard_metadata_t std_meta)
{
 action swap_mac(inout bit<48> src,
          inout bit<48> dst) {
  bit<48> tmp = src;
     src = dst;
     dst = tmp;
  }
 apply {
     swap_mac(hdr.ethernet.srcAddr,
              hdr.ethernet.dstAddr);
     std_meta.egress_spec =
std_meta.ingress_port;
  }
}
```

V. Network Operating Systems

❖ **SONiC (Software for Open Networking in the Cloud):**

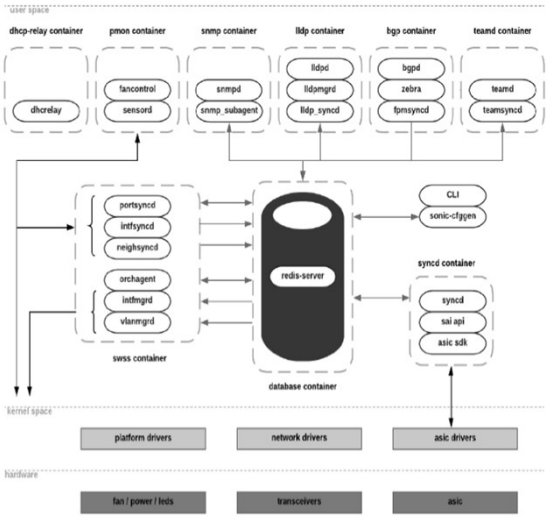
SONiC - Quick Summary	
Name	Software for Open Networking in the Cloud (SONiC)
By	Created by Microsoft, open source NOS
Where it runs	On a supported bare metal switch
What it does	Layer 2 and Layer 3 networking on a bare metal switch
Features	Supports a variety of switch chipsets. Has a CLI, which makes the interactio n with the software easier.
What it can do out-of-the-box	You can download and install SONiC on your bare metal switch and start us ing it as a leaf, spine, or other interconnects for routing and switching.

SONiC is based on Linux and uses SAI to manage and drive the switch chipset.

https://azure.github.io/SONiC/

V. Network Operating Systems

❖ **SONiC (Software for Open Networking in the Cloud):**



JS Lab

Opensource Networking
james@jslab.kr

V. Network Operating Systems

❖ **SONiC (Software for Open Networking in the Cloud):**

❖ **The redisDB has multiple tables. For example:**

- **BGP Table**: Stores configuration related to BGP neighbors, advertisements.
- **Port:** Contains configuration related to interfaces, speeds.
- **PortChannel:** Contains configuration of Link Aggregations (bonding/etherchannel, port channel).
- **VLAN:** Contains VLAN IDs, member ports.
- **VLAN members:** Contains configuration for individual ports for 802.1q tagging.
- **Layer 3 tables:** INTERFACE, PORTCHANNEL_INTERFACE, and VLAN_INTERFACE to store IP address details of Layer 3 interfaces.
- **ACL_RULE:** Contains configuration of access lists.

V. Network Operating Systems

❖ **SONiC (Software for Open Networking in the Cloud) Commands:**

Show MAC:

```
admin@sonic:~$ show mac
No.    Vlan MacAddress         Port
-----  ------ -----------------  -----------
 1   1000 E2:8C:56:85:4A:CD  Ethernet12
```

Configure VLANs and member ports:

```
root@sonic:/# config vlan add 1200
root@sonic:/# config vlan member add 1200 Ethernet1
root@sonic:/# config vlan member add 1200 Ethernet9 -u   (Defining Untagged)
root@sonic:/# config vlan member add 1200 Ethernet8
```

BGP **show** commands:

```
admin@sonic:~$ show bgp neighbors 192.168.1.161
admin@sonic:~$ show bgp neighbors 192.168.1.161 advertised-routes
admin@sonic:~$ show bgp neighbors 192.168.1.161 received-routes
admin@sonic:~$ show bgp neighbors 192.168.1.161 routes
```

V. Network Operating Systems

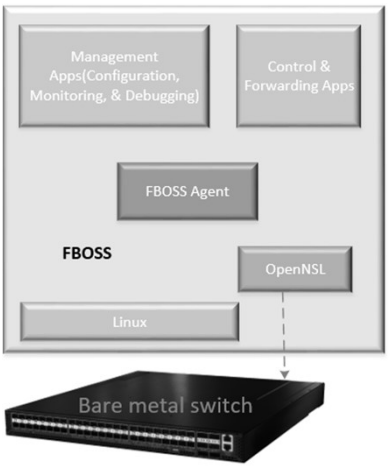
❖ **FBOSS (Facebook Open Switching System):**

FBOSS - Quick Summary	
Name	Facebook Open Switching System (FBOSS)
By	Facebook
Where it runs	On a Linux OS that is installed on a bare metal switch
What it does	FBOSS provides an agent software that can be remotely managed via API c alls to manage the switch chipset silicon. FBOSS should be used with an ext ernal controller
Features	FBOSS abstracts and provides APIs to remotely program a hardware switch (mainly based on Broadcom Trident and Tomahawk chipsets)
What it can do out-of-the-box	You can install FBOSS on a bare metal switch that is running ONL. With FB OSS, you can control the switch silicon using a controller software. You can build your own controller software to interact with FBOSS. Your controller sof tware can be a routing protocol software such as BIRD or Quagga, that can be connected to FBOSS

https://github.com/facebook/fboss

V. Network Operating Systems

❖ **FBOSS Daemons:**



FBOSS mainly uses the Broadcom OpenNSL library to communicate with a Broadcom chipset.

https://github.com/facebook/fboss

V. Network Operating Systems

❖ **FBOSS Daemons:**

Configuring a port:

```
{
  "ports": [
    {
      "logicalID": 1,
      "state": 2,
      "minFrameSize": 64,
      "maxFrameSize": 1500,
      "parserType": 1,
      "routable": true,
      "ingressVlan": 1000,
      "speed": 0
    }
}
```

Configure a routed interface:

```
"interfaces": [
   {
     "intfID": 1000,
     "routerID": 0,
     "vlanID": 1000,
     "ipAddresses": [
       "169.254.0.10/16",
       "2001:db:1111:1150::a/64"
     ],
     "ndp": {
       "routerAdvertisementSeconds": 4,
       "curHopLimit": 255,
       "routerLifetime": 1800,
       "prefixValidLifetimeSeconds": 2592000,
       "prefixPreferredLifetimeSeconds": 604800,
       "routerAdvertisementManagedBit": true,
       "routerAdvertisementOtherBit": true
     }
   }
```


https://github.com/facebook/fboss

V. Network Operating Systems

❖ **FBOSS Daemons:**

Configure a VLAN IP interface (SVI):

```
{
 "intfID": 3004,
  "routerID": 0,
  "vlanID": 3004,
  "ipAddresses": [
"10.11.24.111/31",
"2001:db:3336:e01:1000::aa/127"
  ]
}
```

Configure a VLAN and 802.1q (VLAN 1000 tagged on interface 1, untagged on interface 2):

```
"vlanPorts": [
   {
     "vlanID": 1000,
     "logicalPort": 1,
     "spanningTreeState": 2,
     "emitTags": false
   },
   {
     "vlanID": 1000,
     "logicalPort": 2,
     "spanningTreeState": 2,
     "emitTags": true
   }
```


https://github.com/facebook/fboss

V. Network Operating Systems

❖ **Cumulus Linux:**

Cumulus Linux - Quick Summary	
Name	Cumulus Linux NOS
By	Cumulus Networks
Where it runs	On supported bare metal switches, Linux hosts
What it does	Layer 2, Layer 3 automation
Features	Layer 2 and Layer 3 features, VRF, compatibility with automation tools such as Puppet and Ansible. Also, it provides a CLI tool.
What it can do out-of-the-box	Cumulus Linux is a commercial network operating system that can run on a supported bare metal switch. Cumulus use cases include mainly datacenter networks (CLOS-based leaf, spine).

V. Network Operating Systems

❖ **Cumulus Linux:**

❖ **To configure and manage Cumulus Linux, you need to use pure Linux networking commands and configuration. For example:**

- **Use ifupdown2 to manage interfaces, bonds (Ethernetchannels)**
- **Use brctl to configure and manage bridges (VLANs, tagging)**
- **Use cl-acltool (Cumulus Access List Tool) to manage access lists**
- **Use vtysh to manage routing daemons such as bgpd, ospfd, etc.**

V. Network Operating Systems

- ❖ **Cumulus Linux:**
- ❖ **To create VLANs:**
  - **Create VLAN 100 to 110:**
  - **switch# net add vlan 100-110**
  - **Add port 10 to VLAN 100 as untagged:**
  - **switch1# net add int swp10 bridge access 100**
  - **Add port 18 to VLAN 100 as 802.1q tagged:**
  - **net add int swp18 bridge trunk vlans 100**
  - **Create a bridge IP interface (AKA Switch Virtual Interface SVI or vlan interface):**
  - **net add vlan 100 ip address 172.16.17.1/24**

Opensource Networking
james@jslab.kr

https://www.opencontainers.org/

Opensource Networking
james@jslab.kr
JS Lab

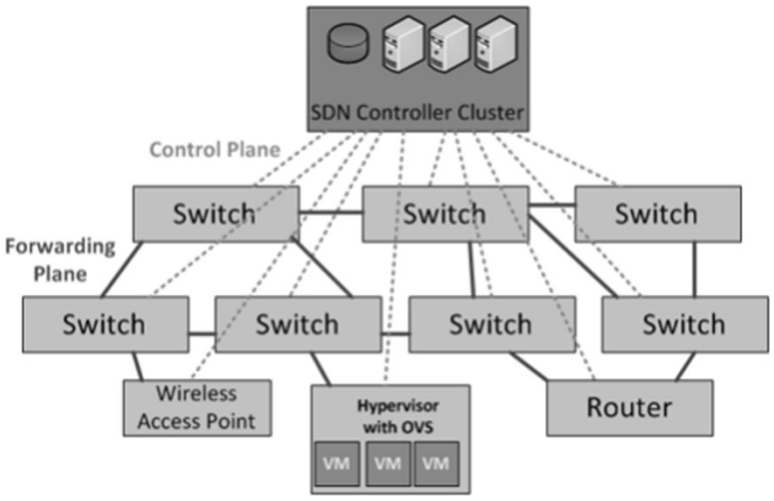
VI. 네트워크 제어(Network Control)

- **Introduction to SDN Controllers**
- **OpenDaylight** (ODL)
- **Open Network Operating System** (ONOS)
- **Tungsten Fabric** (formerly OpenContrail)
- **Project Calico**
- **Cisco Application-Centric Infrastructure** (ACI)
- **Floodlight**
- **Big Cloud Fabric** (BCF)

https://www.linuxfoundation.org/projects/networking/

VI. 네트워크 제어 (Network Control)

❖ **Introduction to SDN Controllers:**



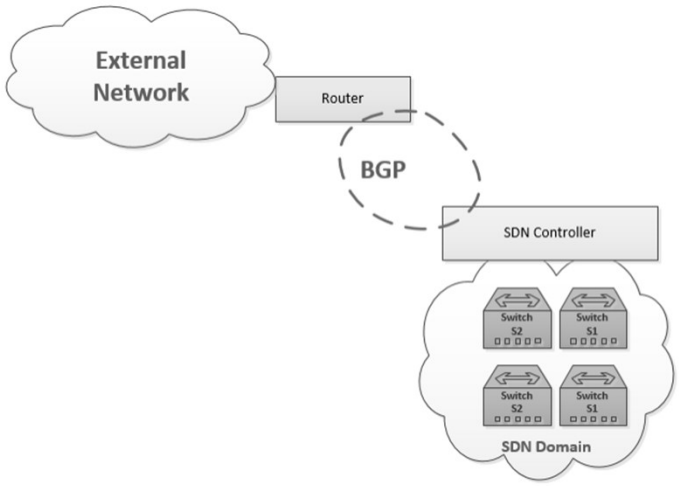
VI. 네트워크 제어 (Network Control)

❖ **Introduction to SDN Controllers:**

Feature	Direct Fabric Programming	Overlay
Can work and co-exist with existing IP network	NO	YES
Required to encapsulate packets	NO	YES
Scalable	YES	YES

VI. 네트워크 제어 (Network Control)

❖ **Introduction to SDN Controllers:**



Relationship between a Network Managed by an SDN Controller and External Networks

VI. 네트워크 제어 (Network Control)

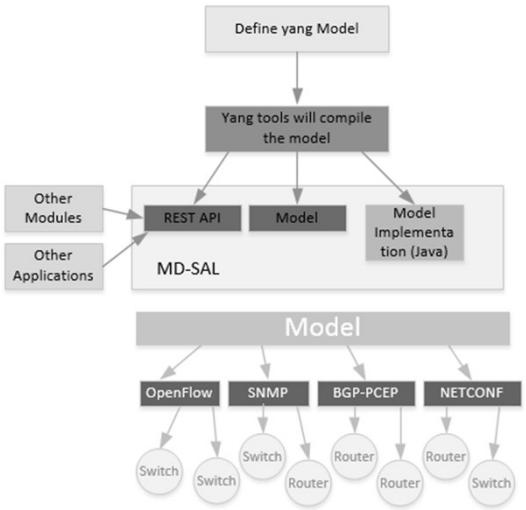
❖ **OpenDaylight (ODL):**

ODL - Quick Summary	
Name	OpenDaylight (ODL) SDN controller
By	The Linux Foundation
Where it runs	As an SDN controller, it runs on a separate host or a cluster of nodes to manage the underlying network
What it does	Manages and operates underlying network devices using southbound protocols
Features	It comes with a comprehensive library of protocols, ready made applications. It is a modular and flexible system, and it allows you to develop any networking application.
What it can do out-of-the-box	You can use ODL for networking with your cloud orchestration platform such as OpenStack. ODL can manage OpenFlow-based switches (it can be bare metal). You can develop a custom event-driven application on top of ODL. Normally, applications get activated on a specific event (for example, a packet arrived) and perform some actions.

https://www.opendaylight.org/what-we-do/odl-platform-overview

VI. 네트워크 제어 (Network Control)

❖ **OpenDaylight (ODL):**

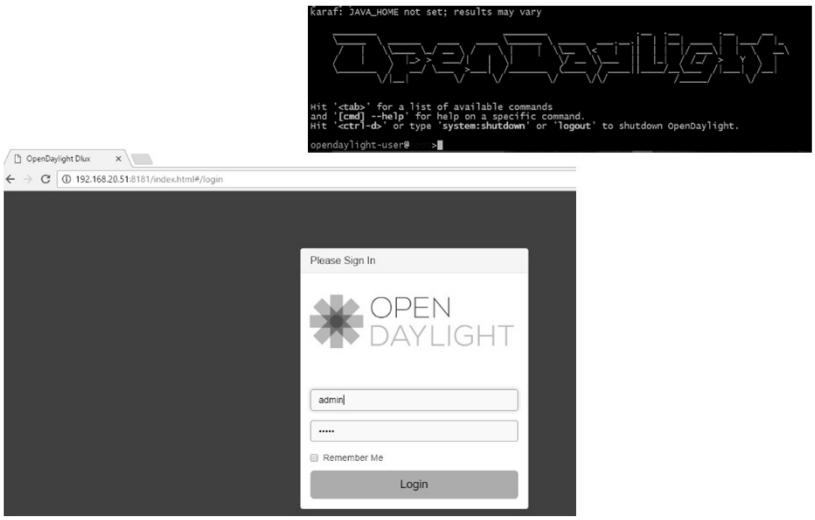


Opensource Networking
james@jslab.kr
JS Lab

https://www.opendaylight.org/what-we-do/odl-platform-overview

VI. 네트워크 제어 (Network Control)

❖ **OpenDaylight (ODL):**



https://www.opendaylight.org/what-we-do/odl-platform-overview

VI. 네트워크 제어 (Network Control)

❖ **Open Network Operating System (ONOS):**

ONOS - Quick Summary	
Name	Open Network Operating System (ONOS)
By	Hosted by The Linux Foundation, maintained by the Open Networking Foundation
Where it runs	As an SDN controller platform, it runs on a separate host to manage the underlying network
What it does	Manages and operates the underlying network devices using southbound protocols
Features	Modular software that allows building plugins and applications, built-in northbound APIs such as REST and gRPC, built-in southbound protocols such as P4, OpenFlow, NETCONF, TL1, SNMP, CLI, BGP, RESTCONF. GUI, YANG tool chain
What it can do out-of-the-box	ONOS can be used as a standard SDN controller to manage underlying networking devices such as routers and switches. ONOS supports different southbound protocols, such as OpenFlow, BGP, and NETCONF, which can configure and manage underlying routers and switches.

https://onosproject.org/

VI. 네트워크 제어 (Network Control)

❖ **ONOS vs ODL:**

	ODL	ONOS
Applications for datacenters, cloud environment	ODL provides applications to integrate with OpenStack (such as VTN Virtual Tenant Manager) or southbound protocols that are used in datacenters, such as OVSDB	SONA (Simplified Overlay Network Architecture) is an ONOS application which integrates with OpenStack and provides network virtualization and tenant isolation
Applications for service providers, telcos	Limited to some use cases and southbound protocols, such as BGP-PCEP	Has many use cases and applications for telcos, such as CORD, Packet Optical, IP RAN, SDN IP, VPLS, Carrier Ethernet, etc.
Performance and high availability	ODL supports clustering	ONOS clustering is mature and comprehensive

VI. 네트워크 제어 (Network Control)

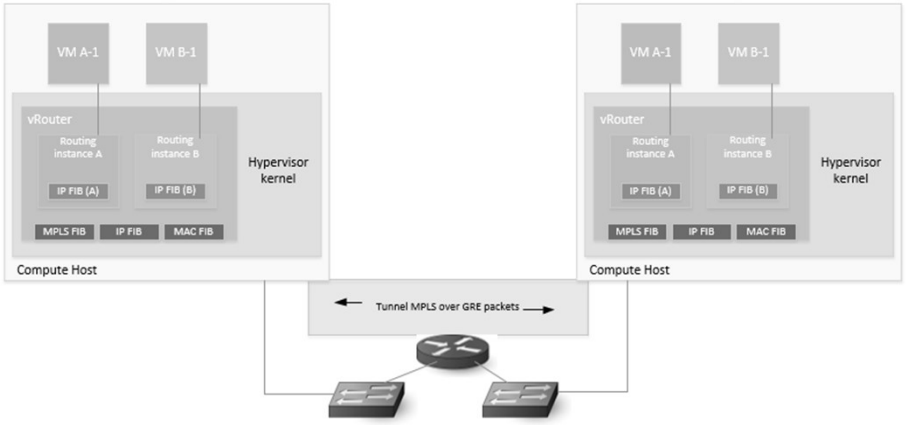
❖ **Tungsten Fabric (formerly OpenContrail):**

Tungsten Fabric - Quick Summary	
Name	Tungsten Fabric (formerly OpenContrail)
By	The Linux Foundation
Where it runs	On multiple nodes; it can be deployed on each compute host of a virtual environment
What it does	Creates and manages virtual overlay networks over any underlying fabric
Features	Layer2 and Layer3 overlays over a Layer3 underlay. Uses MPLS over GRE/MPLS over UDP. The underlying switches do not need to support MPLS
What it can do out-of-the-box	Tungsten Fabric works well in a cloud environment, such as OpenStack. You can deploy and integrate Tungsten Fabric with your cloud orchestration software to provide services such as service function chaining and tenant network isolation.

https://tungstenfabric.io/

VI. 네트워크 제어 (Network Control)

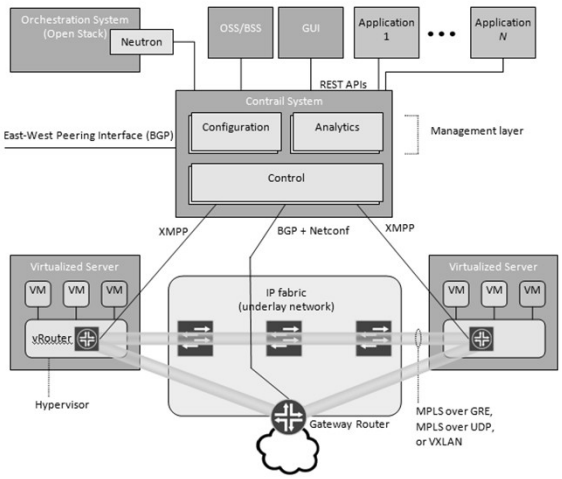
❖ **Tungsten Fabric (formerly OpenContrail):**



Virtual Overlay Networks Over A Standard Legacy L2/L3 Network

VI. 네트워크 제어 (Network Control)

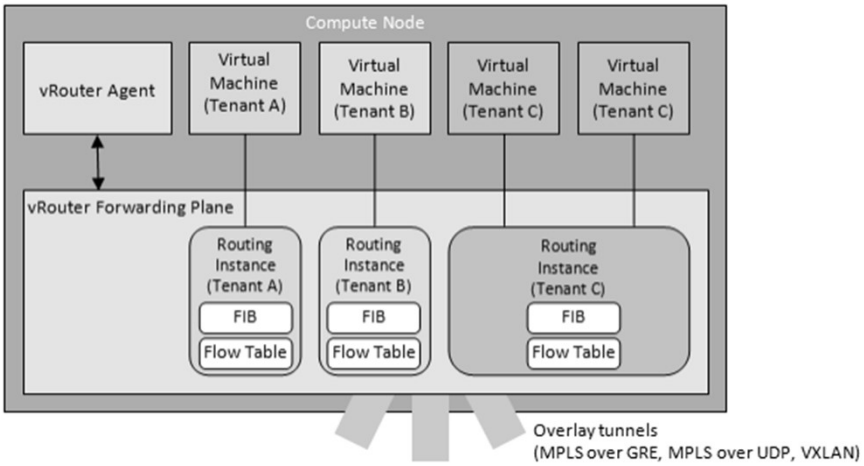
❖ **Tungsten Fabric (formerly OpenContrail):**



Tungsten Fabric Architecture

VI. 네트워크 제어 (Network Control)

❖ **Tungsten Fabric (formerly OpenContrail):**



Tungsten Fabric vRouter Architecture within a Compute Host

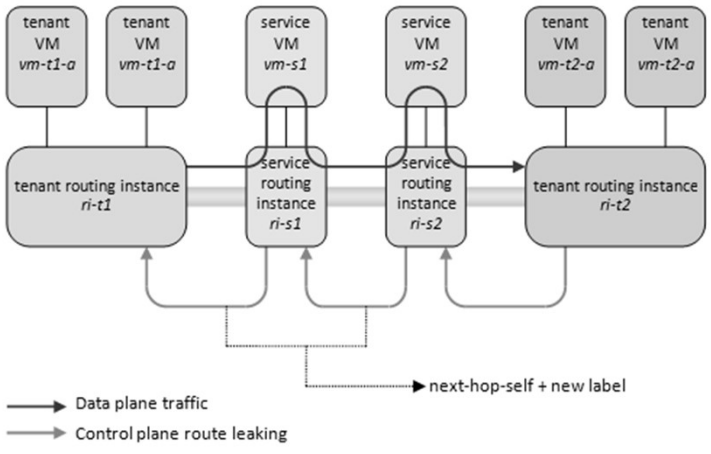
https://tungstenfabric.io/

JS Lab

Opensource Networking
james@jslab.kr

VI. 네트워크 제어 (Network Control)

❖ **Tungsten Fabric (formerly OpenContrail):**



Service Chaining Policy Applied for Traffic between the Two Virtual Machine Workloads

https://tungstenfabric.io/

JS Lab

Opensource Networking
james@jslab.kr

VI. 네트워크 제어 (Network Control)

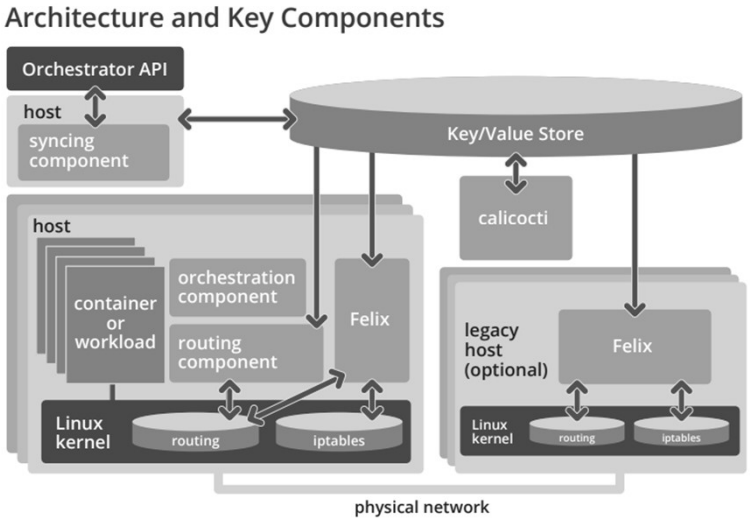
❖ **Project Calico:**

Project Calico - Quick Summary	
Name	Project Calico
By	Calico open source community, backed by Tigera, Inc.
Where it runs	The Calico controller runs on multiple nodes, and gets deployed on each compute host of a virtual environment
What it does	Creates and manages virtual networks for VMs, containers and bare metal
Features	Supports overlay and encapsulation, uses a built-in IPAM for allocation and assigning IP addresses to workloads
What it can do out-of-the-box	Project Calico can be used in OpenStack, Kubernetes, and other orchestration environments to provide networking

https://www.projectcalico.org/

VI. 네트워크 제어 (Network Control)

❖ **Project Calico:**



https://www.projectcalico.org/wp-content/uploads/2018/04/ProjectCalico.v3.1.datasheet.pdf

VI. 네트워크 제어 (Network Control)

❖ **Cisco Application-Centric Infrastructure (ACI):**

ACI - Quick Summary	
Name	Cisco ACI SDN Controller
By	Cisco Systems
Where it runs	ACI/APIC runs on a cluster of servers to manage the underlying network of Nexus 9000 switches
What it does	Manages the SDN domain and its switches. Provides traffic isolation, security, and virtual networks.
Features	Manages single and multiple sites
What it can do out-of-the-box	You can integrate the ACI with your virtualization platform or cloud environment and manage your physical and virtual network switches. ACI provides automation features that can help enhance the processes and reduce the time for changes in the network.

https://www.opendaylight.org/what-we-do/odl-platform-overview

VI. 네트워크 제어 (Network Control)

❖ **Floodlight:**

Floodlight - Quick Summary	
Name	Floodlight Controller
By	Sponsored by Big Switch Networks
Where it runs	On a cluster of hosts to manage an underlay network
What it does	Floodlight is a modular SDN controller capable of having SDN applications. It manages the underlying physical and virtual switches via the OpenFlow protocol.
Features	Integrates with OpenStack; open source
What it can do out-of-the-box	Floodlight documentation is not fancy and might seem old. However, you can use Floodlight to manage any OpenFlow-capable physical switch.

https://www.bigswitch.com/tags/floodlight-controller

VI. 네트워크 제어 (Network Control)

❖ **Big Cloud Fabric (BCF):**

Big Cloud Fabric - Quick Summary	
Name	Big Cloud Fabric (BCF)
By	Big Switch Networks
Where it runs	On a cluster of servers. To manage the data plane, Big Cloud Fabric provides a lightweight NOS called SwitchLight (to be installed on a compatible hardware)
What it does	Manages a datacenter or tenant-based cloud environment
Features	Commercial SDN controller, works with bare metal switches, integrates with cloud orchestration tools such as OpenStack, VMware vSphere, and Kubernetes.
What it can do out-of-the-box	By deploying Big Cloud Fabric, you can integrate a network with your cloud orchestration platform. For example, BCF integrates with VMware vSphere. Any VLAN you create on VMware vSphere will be automatically presented in your underlying network managed by BCF. BCF can apply network policies and service chaining on your network traffic to enforce traffic to pass through a network analytics or IPS when it is being routed to the Internet.

https://www.bigswitch.com/products/big-cloud-fabric

VII. 클라우드와 가상화 관리

- **Open Network Automation Platform** (ONAP)
- **M-CORD** (Mobile CORD)
- **R-CORD** (Residential CORD)
- **E-CORD** (Enterprise CORD)
- **Trellis**
- **Open Source MANO**
- **Open Platform for NFV** (OPNFV)
- **Open Security Controller** (OSC)
- **Akraino Edge Stack**

Opensource Networking
james@jslab.kr

https://www.linuxfoundation.org/projects/networking/

VII. 클라우드와 가상화 관리

❖ **Open Network Automation Platform (ONAP):**

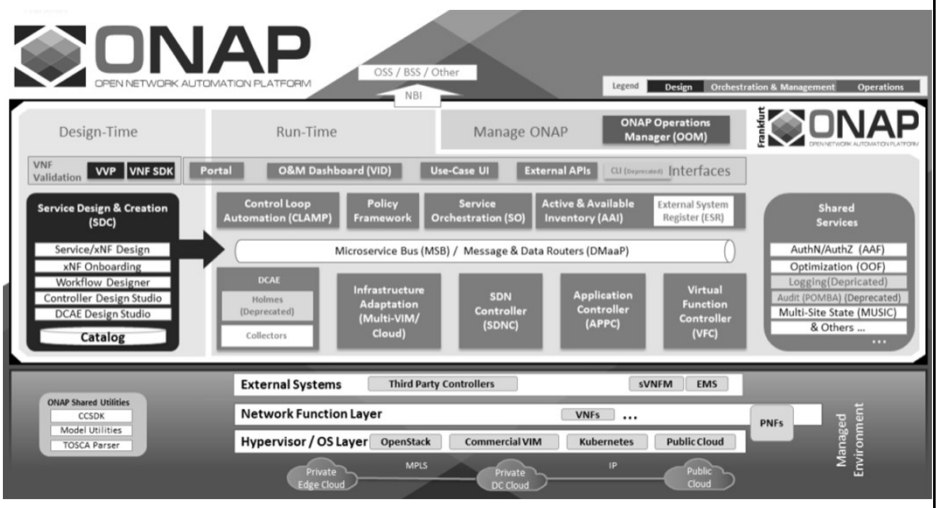
ONAP - Quick Summary	
Name	Open Network Automation Platform (ONAP)
By	The Linux Foundation
Where it runs	On separate hosts, recommended to run on Kubernetes or OpenStack
What it does	ONAP is a platform that orchestrates the lifecycle of virtual network services in a software defined networking environment
Features	Open source; creates services that consist of VNFs and policies. Runs and executes the services. Has a closed-loop.
What it can do out-of-the-box	ONAP relies on many other components and platforms; it will not be able to deliver the real functions without the other components out-of-the-box. There are multiple use cases for ONAP, such as uCPE, edge networking services, Voice-over-LTE, virtual firewall, virtual DHCP server, etc. You can deploy a full or minimum ONAP environment using OpenStack. A full ONAP environment requires numerous virtual machines and gigabytes of RAM.

Opensource Networking
james@jslab.kr

https://www.onap.org/ https://onap.readthedocs.io/en/dublin/

VII. 클라우드와 가상화 관리

❖ **Open Network Automation Platform (ONAP)**
❖ **ONAP Architecture**



https://www.onap.org/

JS Lab

Opensource Networking
james@jslab.kr

VII. 클라우드와 가상화 관리

❖ **ONAP Design Time Environment:**

- **Service Design and Creation (SDC)**
  - ✓ To define system assets and their policies.
- **VNF Software Development Kit (VNFSDK) and VNF Validation Program (VVP)**
  - ✓ For VNF packaging and validation of VNFs.
- **Policy Creation (POLICY)**
  - ✓ To define policies that need to be maintained or enforced.
- **Closed Loop Automation Management Platform (CLAMP)**
  - ✓ To design and manage closed control loops.
- **Optimization Framework (OOF)**
  - ✓ To optimize the application and services.

https://www.onap.org/

JS Lab

Opensource Networking
james@jslab.kr

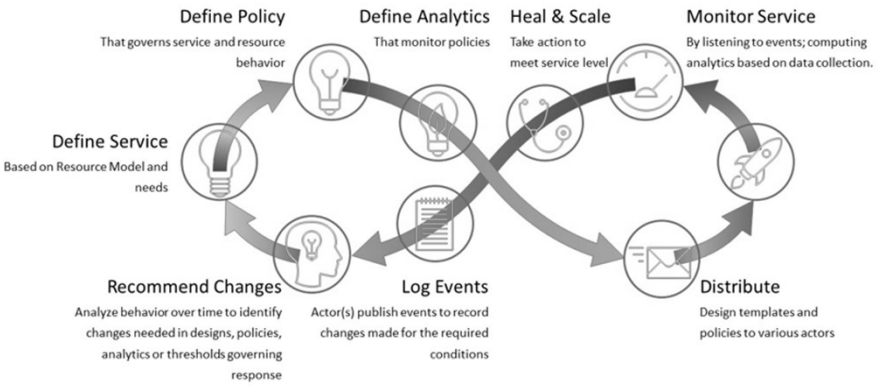
VII. 클라우드와 가상화 관리

❖ **ONAP Run Time Environment:**

- **Service Orchestrator (SO)**
  - ✓ The Service Orchestrator is an automation engine in the ONAP Run Time environment.
- **Software Defined Network Controller (SDNC)**
  - ✓ It is responsible for executing the network configuration.
- **Application Controller (APPC)**
  - ✓ It is responsible for executing and configuring the Virtual Network Functions (VNF).
- **Virtual Function Controller (VF-C)**
  - ✓ It is responsible for the lifecycle management of the Virtual Network Functions (VNF) which are run by the VNF manager.
- **Active and Available Inventory (A&AI)**
  - ✓ It is responsible for the real-time view of system resources, products and their relationships.

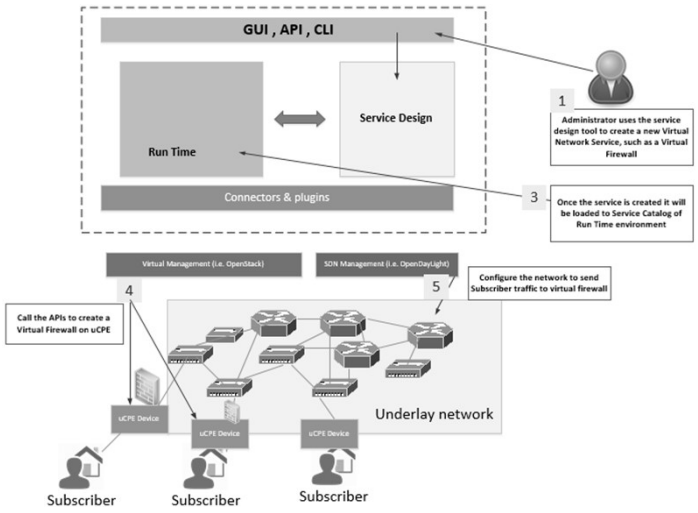
VII. 클라우드와 가상화 관리

❖ **ONAP: Closed Loop Automation:**



VII. 클라우드와 가상화 관리

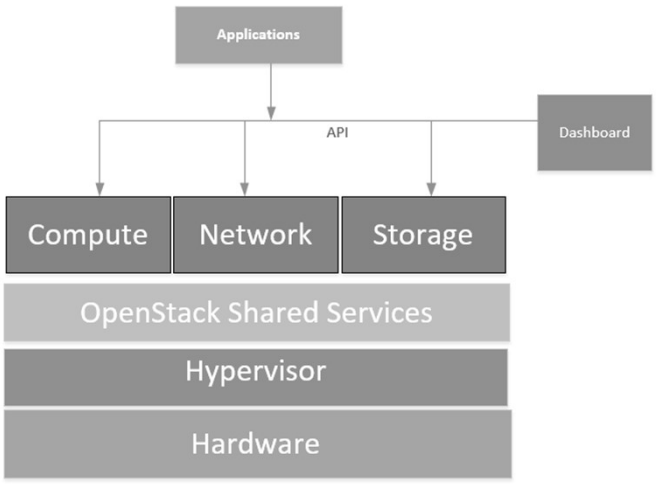
❖ **ONAP Example:**



https://www.onap.org/ https://onap.readthedocs.io/en/beijing/guides/onap-developer/settingup/onap_oom.html

VII. 클라우드와 가상화 관리

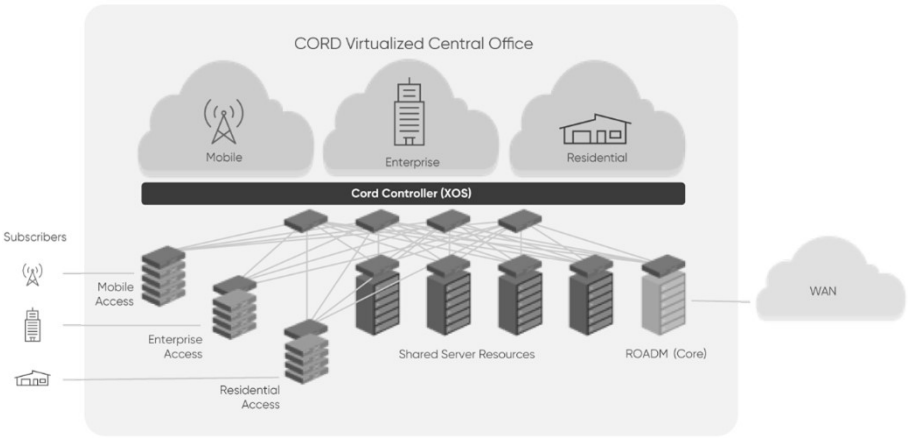
❖ **ONAP and OpenStack:**



https://www.onap.org/ https://onap.readthedocs.io/en/amsterdam/guides/onap-developer/settingup/fullonap.html

VII. 클라우드와 가상화 관리

❖ **ONAP and CORD:**

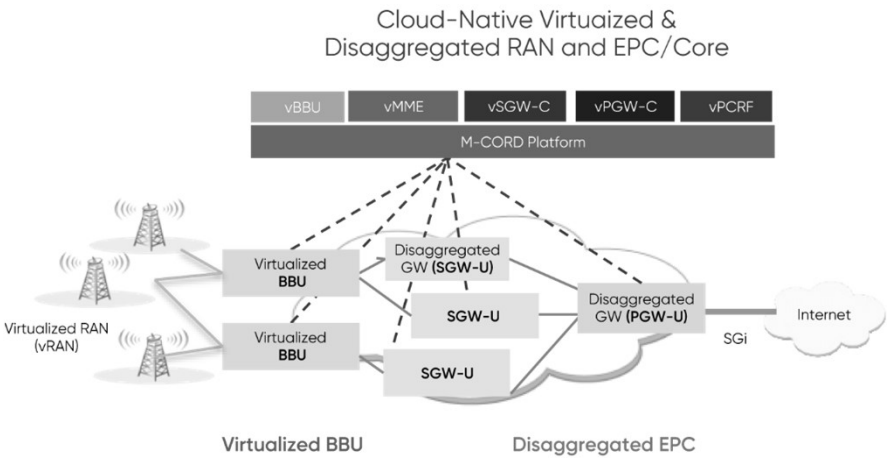


CORD Hardware Architecture

https://www.opennetworking.org/cord/

VII. 클라우드와 가상화 관리

❖ **M-CORD (Mobile CORD):**

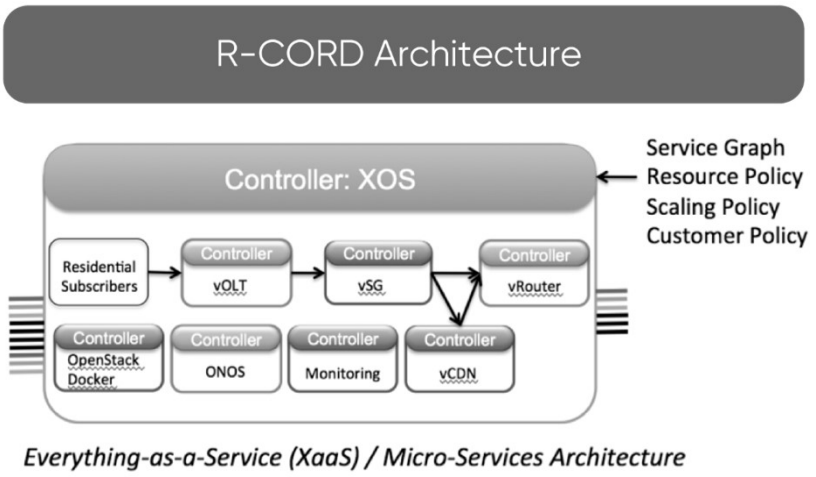


The Role of M-CORD in a Mobile Service Provider Infrastructure

https://www.bigswitch.com/products/big-cloud-fabric

VII. 클라우드와 가상화 관리

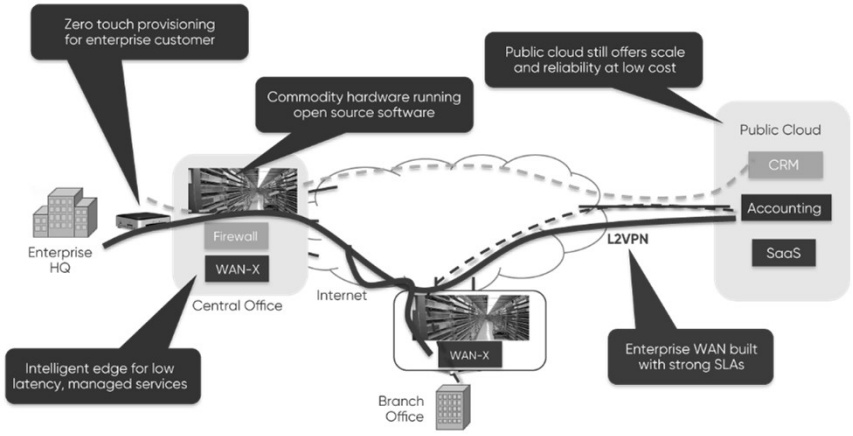
❖ **R-CORD (Residential CORD):**



https://www.opennetworking.org/r-cord/

VII. 클라우드와 가상화 관리

❖ **E-CORD (Enterprise CORD):**



https://www.opennetworking.org/e-cord/

VII. 클라우드와 가상화 관리

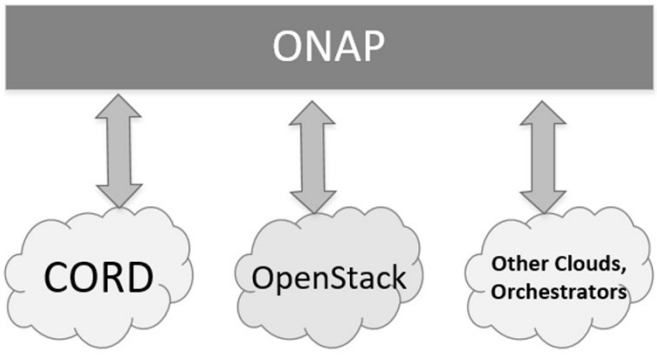
❖ **E-CORD (Enterprise CORD):**

- **VPN**
- **Internet Access**
- **Firewall and border protection**
- **CDN (Content Delivery Network)**
- **Network core functions such as DNS, DHCP, etc.**
- **SD-WAN**
- **Traffic optimization and enhanced QoS**
- **Zero Touch Provisioning of commodity hardware at customer premises and sites**
- **Correctly measured monitoring services to deliver an outcome-based SLAs and KPIs**
- **A platform that enables creation and delivery of innovative services.**

https://www.bigswitch.com/products/big-cloud-fabric

VII. 클라우드와 가상화 관리

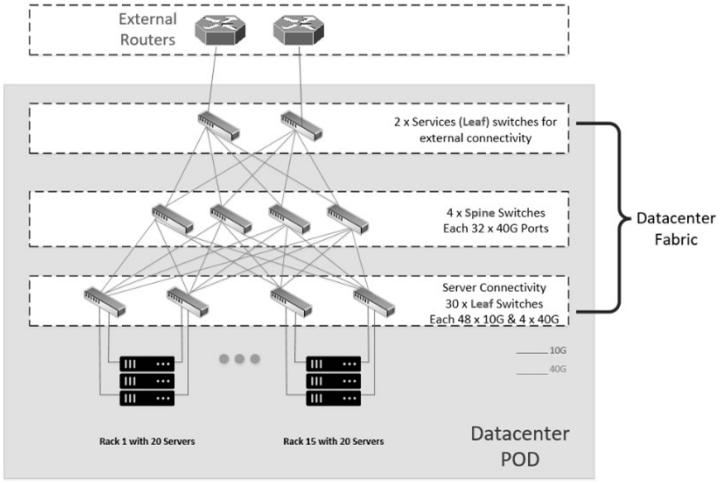
❖ **ONAP Communication with CORD:**



https://www.opennetworking.org/e-cord/

VII. 클라우드와 가상화 관리

❖ **Trellis:**

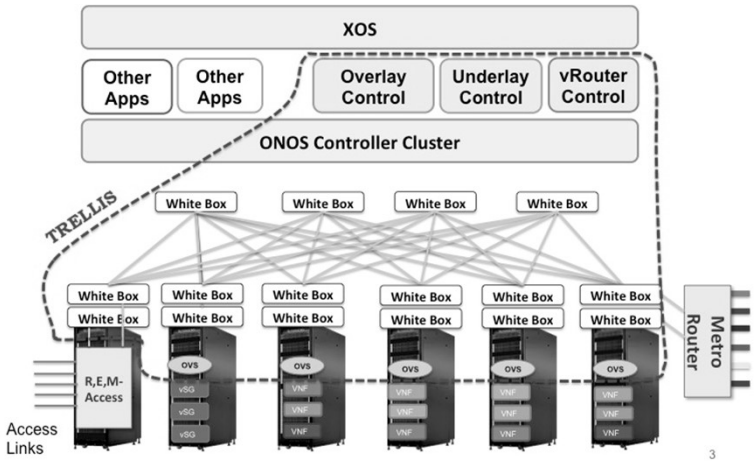


Leaf-Spine Design in a Datacenter

https://www.opennetworking.org/trellis/

VII. 클라우드와 가상화 관리

❖ **Trellis:**

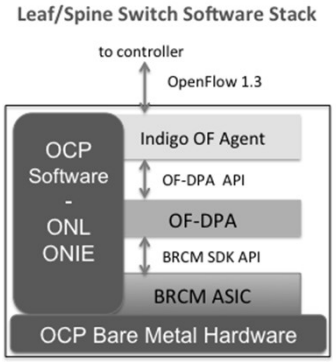


High-Level Architecture of Trellis in a Datacenter

https://wiki.opencord.org/display/CORD/Trellis%3A+CORD+Network+Infrastructure

VII. 클라우드와 가상화 관리

❖ **Trellis Architecture:**



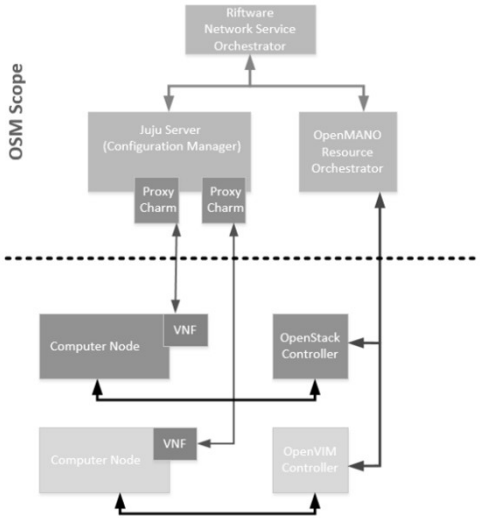
OCP: Open Compute Project
ONL: Open Network Linux
ONIE: Open Network Install Environment
BRCM: Broadcom Merchant Silicon ASICs
OF-DPA: OpenFlow Datapath Abstraction

Trellis Components on a White Box Switch

https://wiki.opencord.org/display/CORD/Trellis+Underlay+Fabric

VII. 클라우드와 가상화 관리

❖ **Open Source MANO Architecture:**



https://osm.etsi.org/

VII. 클라우드와 가상화 관리

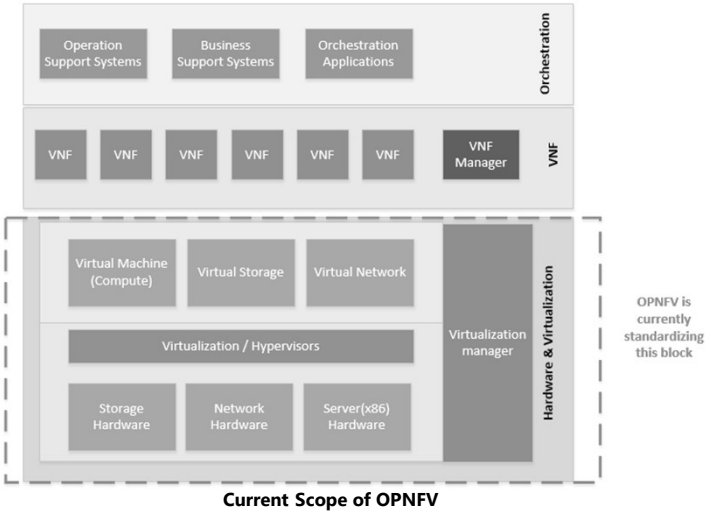
❖ **Open Source MANO:**

- **Service Orchestrator (SO)**
  SO is responsible for the end-to-end service orchestration and provisioning of VNFs and service chaining. SO manages the automation workflow for service deployment. OSM uses RIFT.io as orchestration engine.
- **Resource Orchestrator (RO)**
  RO is responsible to communicate with virtualization platforms such as OpenStack and VMware. RO provisions the NFV virtual workloads on virtualization platforms.
- **VNF Configuration and Abstraction (VCA)**
  VCA is responsible for the configuration of VNFs that are provisioned by the Resource Orchestrator. OSM uses Canonical's Juju Charms as an automation engine to apply the required configuration to the provisioned VNFs.

https://www.bigswitch.com/products/big-cloud-fabric

VII. 클라우드와 가상화 관리

❖ **Open Platform for NFV (OPNFV):**



Current Scope of OPNFV

https://www.opnfv.org/

VII. 클라우드와 가상화 관리

❖ **Open Security Controller (OSC):**

Open Security Controller - Quick Summary	
Name	Open Security Controller (OSC)
By	The Linux Foundation
Where it runs	On a Linux host
What it does	Provision virtual firewall, IPS, and other components. Communicate with SDN controllers to create service chaining and ensure traffic is routed via the provisioned security functions.
What it can do out-of-the-box	OSC has built-in capabilities to integrate with OpenStack and Kubernetes to provision firewalls and other virtual security services.

https://www.opensecuritycontroller.org/

VII. 클라우드와 가상화 관리

❖ **Open Security Controller (OSC):**

- **Call an API on virtual machine management platform (i.e. OpenStack) to provision a virtual machine using the base image of virtual firewall, along with the networks that it needs to connect to.**
- **Virtual machine management platform (i.e. OpenStack) provisions the virtual machine, reports back the Virtual Machine details, such as IP address, MAC address allocated for this VNF, etc.**
- **OSC calls the API to the network control layer (i.e OpenDaylight or Tungsten Fabric) to create a service chain that sends all traffic from VM A to this newly created VNF.**

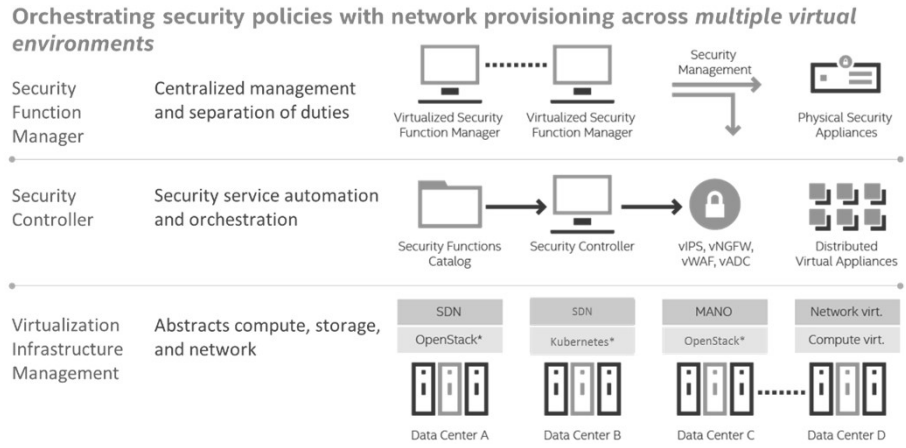
Opensource Networking
james@jslab.kr

https://www.opensecuritycontroller.org/

JS Lab

VII. 클라우드와 가상화 관리

❖ **OSC Architecture:**



Open Security Controller Conceptual Architecture

https://www.opensecuritycontroller.org/

JS Lab

VII. 클라우드와 가상화 관리

❖ **OSC Interactions:**

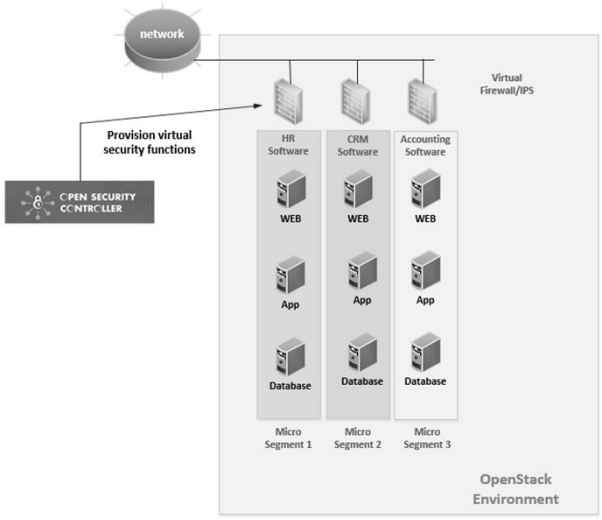
- **Virtualization management systems**
  OSC includes a connector to communicate with virtualization management systems such as OpenStack and Kubernetes. This connector directly calls the Virtual Infrastructure Manager (VIM) APIs in order to provision virtual network security functions. OSC Virtualization Management plugin also subscribes to notification events from the VIM system in order to receive information and status related to provisioned virtual network security workloads (virtual machines or containers).
- **SDN controllers**
  OSC supports communication with multiple networking and SDN controllers. OSC has a built-in connector that works with multiple SDN controllers. OSC uses the SDN controller plugin to implement traffic redirection or Service Function Chaining (SFC) to send specific traffic to the newly created network security function.
- **Security Function Managers**
  OSC uses this connector to communicate with security function systems such as IPS manager, firewall manager, security policy manager, etc. Using this plugin, OSC will be able to call the Security Function Manager APIs to apply specific policy updates, device group membership settings, etc., to the newly created virtual network function

https://www.opensecuritycontroller.org/

JS Lab

VII. 클라우드와 가상화 관리

❖ **OSC Use Case: Microsegmentation**



https://www.opensecuritycontroller.org/

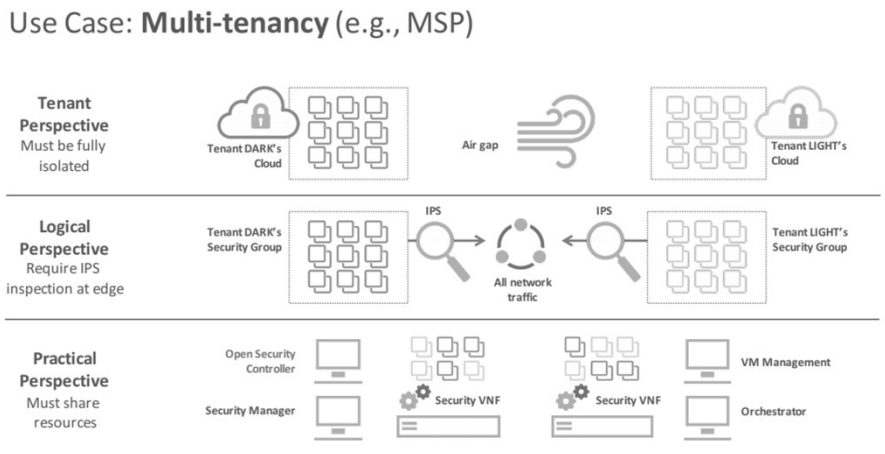
JS Lab

Opensource Networking

james@jslab.kr

VII. 클라우드와 가상화 관리

OSC Use Case: Segmentation within a Multi-Tenant Environment



https://www.opensecuritycontroller.org/

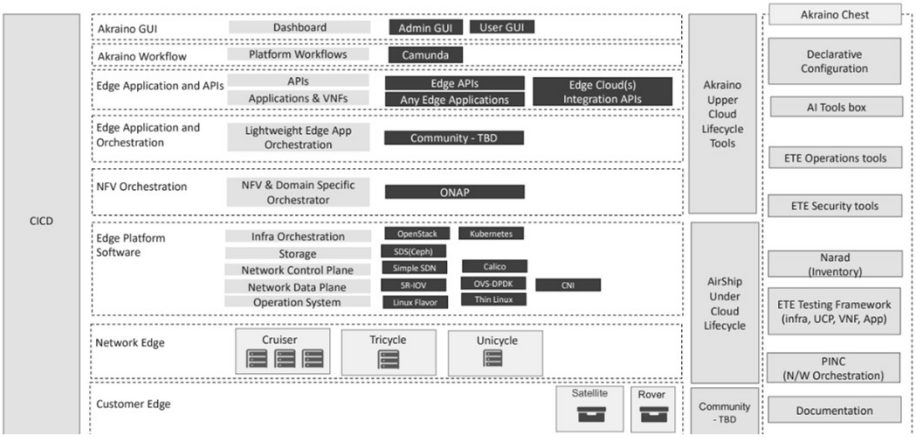
JS Lab

Opensource Networking

james@jslab.kr

VII. 클라우드와 가상화 관리

❖ **Akraino Edge Stack:**

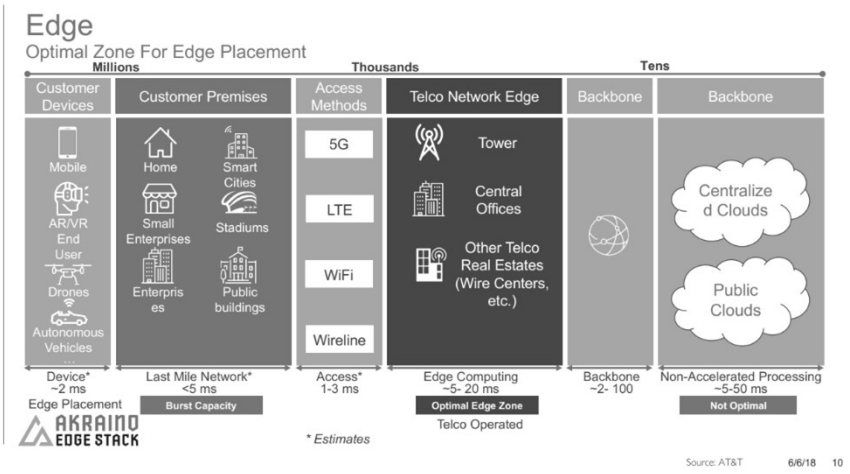


Akraino's Proposed Architecture

https://www.akraino.org/

VII. 클라우드와 가상화 관리

❖ **Akraino Edge Stack:**

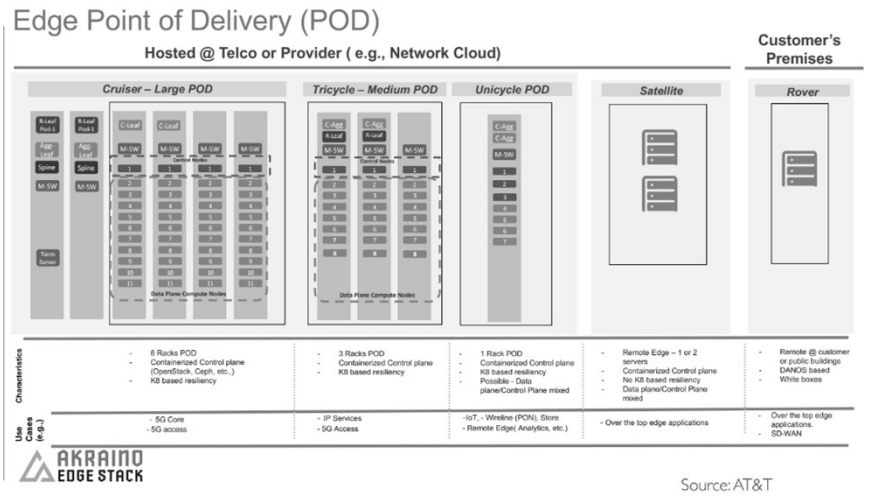


Optimal Zone for Edge Placement

https://www.akraino.org/

VII. 클라우드와 가상화 관리

❖ **Akraino Edge Stack:**



Edge Point of Delivery

https://www.akraino.org/

Opensource Networking
james@jslab.kr

Opensource Networking

james@jslab.kr

VIII.네트워크 가상화

- **Introduction to Network Virtualization**
- **Network Virtualization vs. Network Function Virtualization**
- **Vendor-Specific Virtualization: VMware NSX**
- **Overlay Networks**
- **OpenStack Networking**
- **Hardware Acceleration**
- **Containers and Networking**
- **Docker Networking for Containers**
- **Kubernetes** (K8s)
- **Service Mesh** (Istio)

JS Lab

https://www.linuxfoundation.org/projects/networking/

Opensource Networking

james@jslab.kr

VIII. 네트워크 가상화

❖ **Introduction to Network Virtualization:**

Network virtualization is a key concept for both open source networking, as well as new cloud technologies. Virtualization technologies use network virtualization to allow communication between virtual machines or containers within a compute host, or across multiple compute hosts. Network virtualization includes virtual networks that only exist within a host (such as Linux bridge, IO Visor, etc.), as well as technologies that allow communication between Linux bridges of multiple hosts (encapsulation and overlay networks).

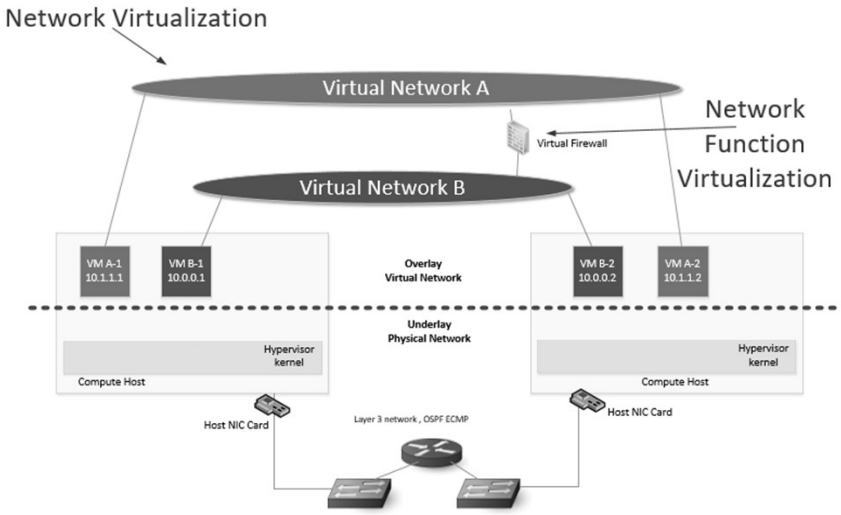
Virtual networks that have been connecting the virtual machines within a host have existed for many years. Most of them use a host-based virtual bridge to connect the virtual interface of the virtual machines (a Linux bridge is a virtual Layer 2 switch). Virtual machines have been the main use case for Linux bridges for a long time. In recent years, with the introduction of containers, virtual switches started being used to connect containers, as well as virtual machines.

Container systems such as Docker use Linux bridge as their main method of connectivity.

JS Lab

VIII. 네트워크 가상화

❖ **Network Virtualization vs. Network Function Virtualization:**



VIII. 네트워크 가상화

❖ **Vendor-Specific Virtualization: VMware NSX:**

VMware NSX - Quick Summary	
Name	VMware NSX
By	VMware
Where it runs	Over a virtualized environment, in VMware vSphere
What it does	Creates and manages virtual networks, firewalls, load balancers, routers. Secures the East-West traffic by protecting VM-to-VM traffic at the host level. Provides security compliance and auditing. Provides a platform to implement microsegmentation using its ready-made firewall, load balancer, router and networking features. NSX has its own virtual firewall and virtual load balancers, which can be used for any workload.
What you can do out-of-the-box	You can deploy VMware NSX in your existing VMware vSphere environment (version should be supported). After installation, you will get a new section in your VMware vCenter to manage your network and security. You can create virtual networks, provision firewalls and load balancers, create virtual routers and peers with external networks, create traffic filtering between VMs and many other features out-of-the-box.

https://www.vmware.com/products/nsx.html

VIII. 네트워크 가상화

❖ **Vendor-Specific Virtualization: VMware NSX:**

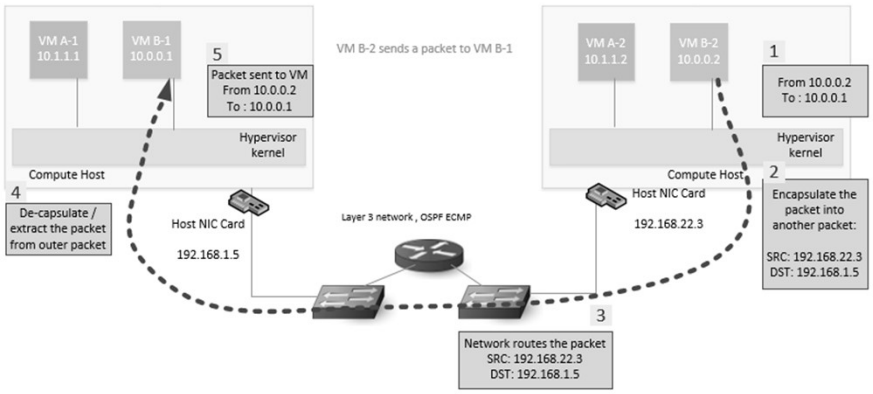
❖ **VMware NSX enhances the VMware's distributed virtual switches and adds many networking and security features. VMware NSX provides the following virtual functions:**

- **Switching**
- **Routing**
- **Firewalling**
- **Load balancing**
- **VPN**
- **Access control**
- **Quality of Service management**

https://www.vmware.com/products/nsx.html

VIII. 네트워크 가상화

❖ **Vendor-Specific Virtualization: VMware NSX:**

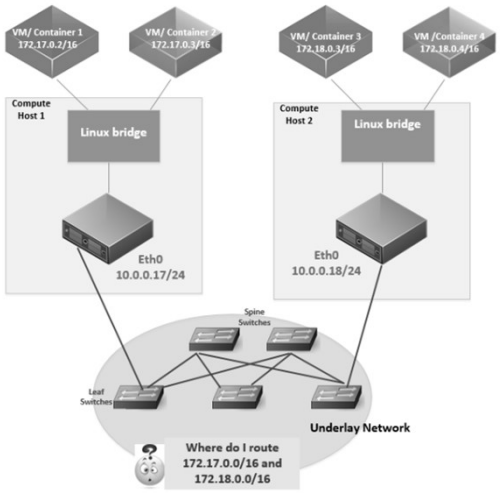


Basic Protocol-Independent Encapsulation and Overlay

https://www.vmware.com/products/nsx.html

VIII. 네트워크 가상화

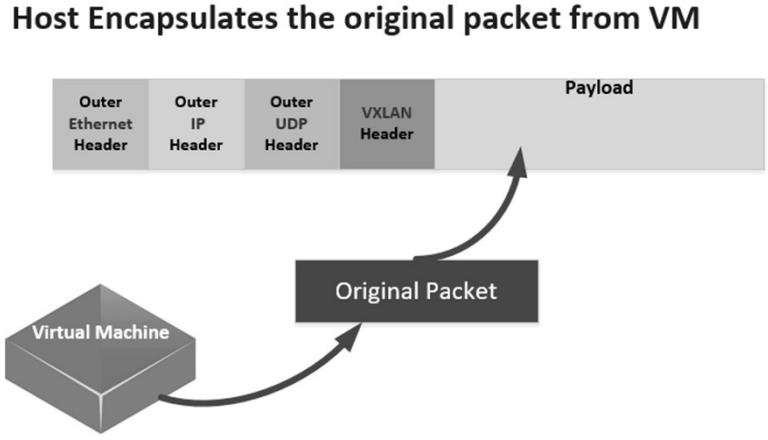
❖ **Overlay Networks:**



Underlay Network Unaware of IP Addresses and Virtual Networks Existing on Compute Hosts

VIII. 네트워크 가상화

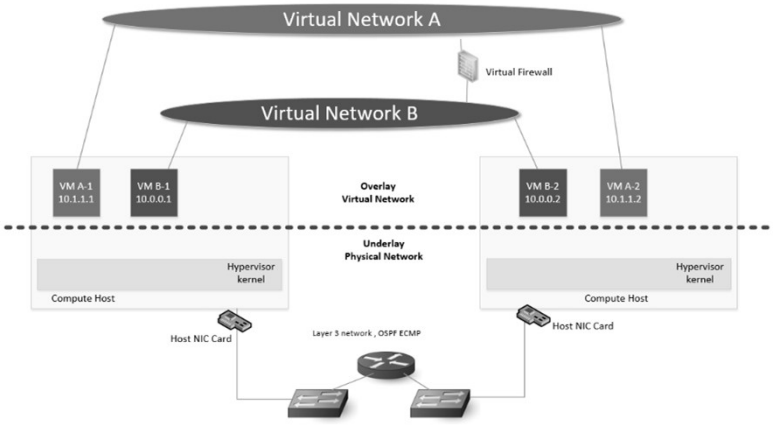
❖ **Overlay Networks:**



Host Encapsulates the Original Packet from the VM

VIII. 네트워크 가상화

❖ **Overlay Networks:**

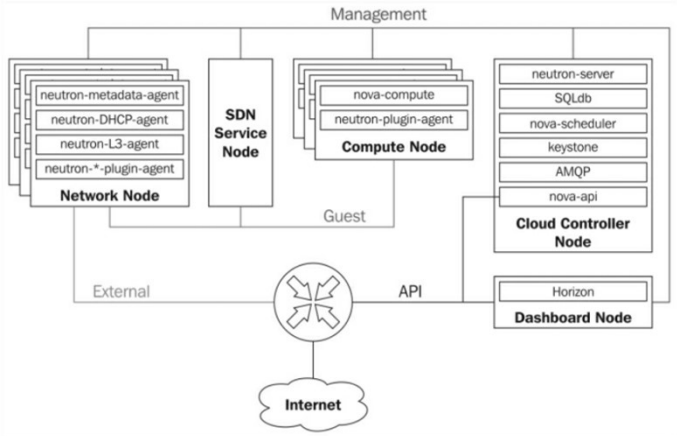


Overlay Network

VIII. 네트워크 가상화

❖ **오픈스택(OpenStack) 네트워크**

- **네트워크를 분리 (관리/Guest/외부/API)**
- **VLAN 사용시 트렁크 모드 사용을 피하거나 네트워크를 물리적으로 나누는 것이 필요**



Locati, Fabio Alessandro. OpenStack Cloud Security

VIII. 네트워크 가상화

❖ **Hardware Acceleration:**

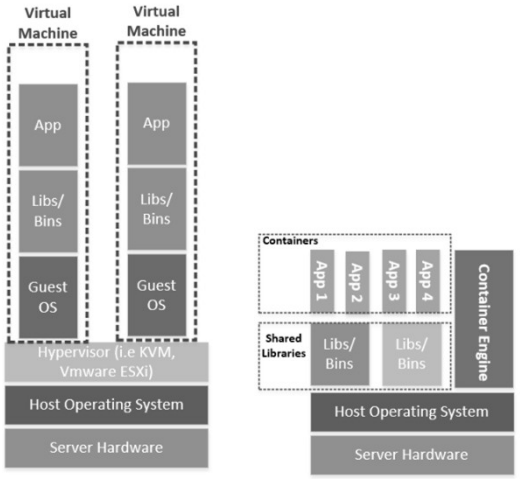
Encapsulation and decapsulation of traffic add additional work on the host's CPU to process each packet from and to virtual workloads. This process adds extra overhead to the system, reducing the host's CPU time which should be used for running applications.

The good news is that, these days, most NIC cards support VXLAN offloading, which means the work of encapsulation and decapsulation can be offloaded to the NIC card chipset, instead of the CPU.

When we talked about DPDK, NIC cards and SmartNICs, the process of traffic encapsulation and decapsulation can be hardware-accelerated. Most NIC cards have built-in capabilities to offload the VXLAN Encapsulation/Decapsulation works from the CPU, without requiring special configuration. If a NIC card doesn't support native VXLAN offloading, you can use DPDK to build an application to accelerate the VXLAN encapsulation and decapsulation process.

VIII. 네트워크 가상화

❖ **Containers and Networking:**



Virtual Machines VS Containers

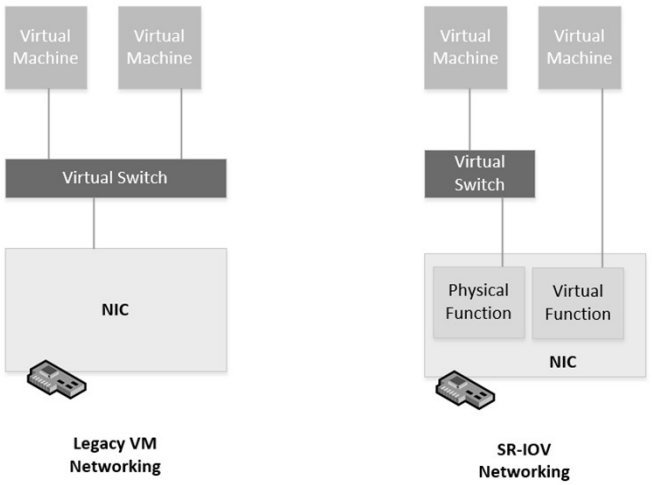
VIII. 네트워크 가상화

❖ **Containers and Networking:**

	Virtual Machines	Containers
Running Engine	Hypervisor	Container Engine
Workload's OS/Kernel	Each VM has its own OS and kernel.	All containers use the host machine's OS and kernel.
Density	Host machine needs to run multiple full operating systems.	Host machines run multiple containers using a common kernel and groups of common binaries and libraries. Less memory and CPU utilization than VMs. A host can run more containers compared to VMs. Higher density.
Networking	Relies on host virtual switches or direct NIC connectivity using SR-IOV* (Single Root Input Output Virtualization).	Relies on host virtual switches or direct NIC connectivity using SR-IOV.
Storage	Host presents a virtual block storage to the VM.	Container Engine manages the container storage.
Guests	Can be any OS (such as Linux, BSD, Windows, etc.).	Only the same as the host's kernel. You can still run an Ubuntu container on a CentOS host.

VIII. 네트워크 가상화

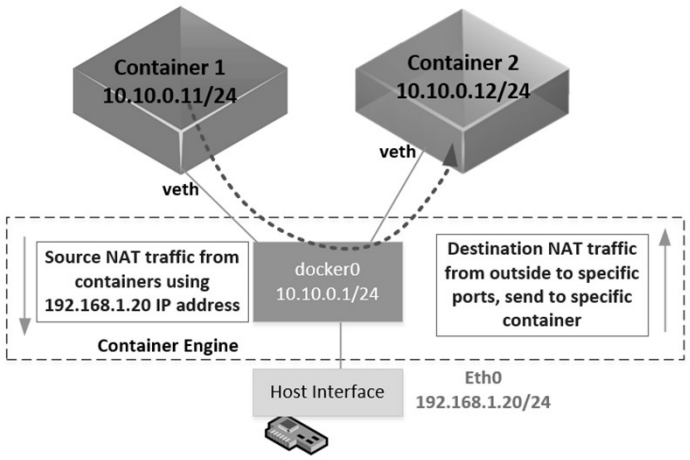
❖ **Containers and Networking:**



Legacy vs SR-IOV Networking

VIII. 네트워크 가상화

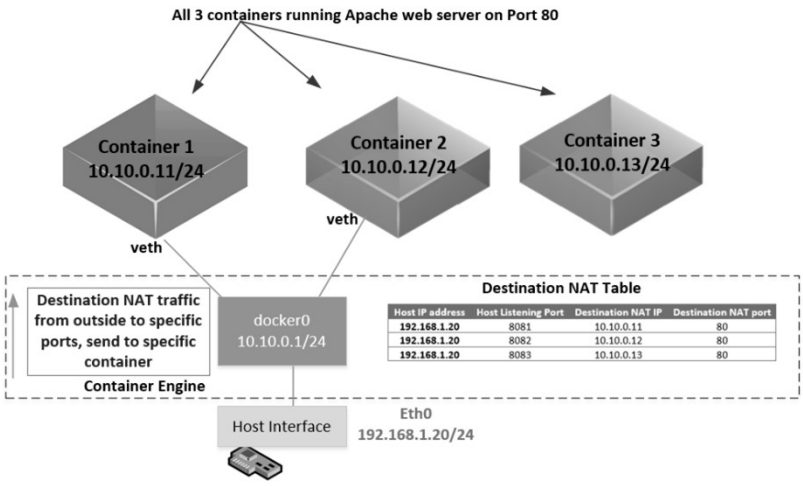
❖ **Docker Networking for Containers:**



Docker Networking Function and NATing

VIII. 네트워크 가상화

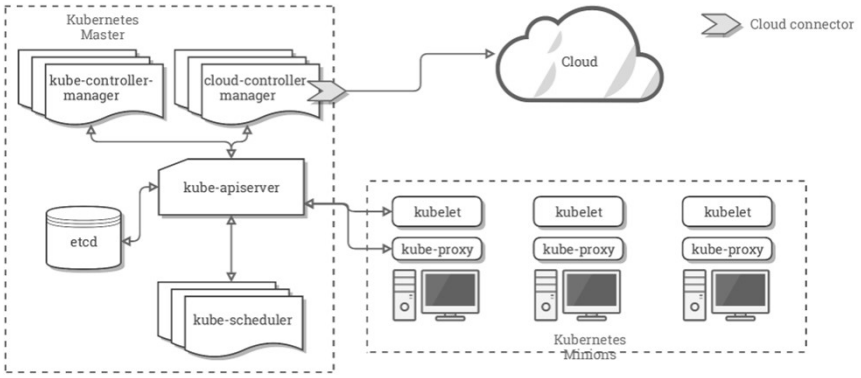
❖ **Connecting to Containers from the Outside:**



Destination NAT for Containers

VIII. 네트워크 가상화

❖ **Kubernetes:**



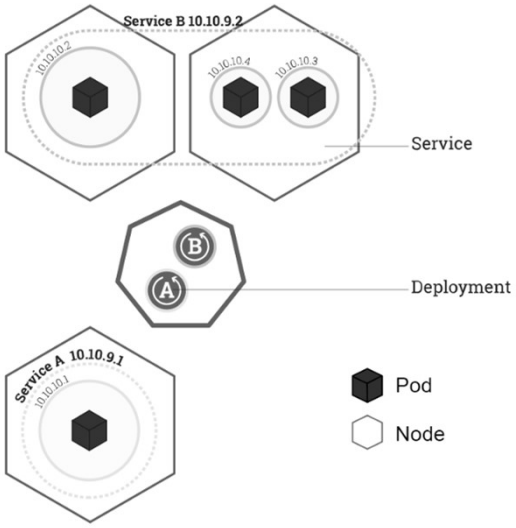
Kubernetes Architecture

https://kubernetes.io/

Opensource Networking
james@jslab.kr

VIII. 네트워크 가상화

❖ **Kubernetes Services:**



https://kubernetes.io/

Opensource Networking
james@jslab.kr

VIII. 네트워크 가상화

❖ **Kubernetes Pod: Istio-telemetry**(예)

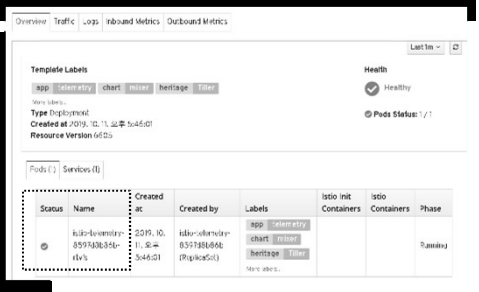
VIII. 네트워크 가상화

❖ **Kubernetes Pod: Istio-telemetry and Proxy @ Docker** (예)

- **docker ps**
- **docker exec CONTAINER ifconfig**

```
root@worker72:/home/jslab# docker ps
CONTAINER ID        IMAGE                     COMMAND                  CREATED             STATUS              PORTS               NAMES
9ac429cc8e5a        f31df53bffed              "/usr/local/bin/mixs…"   25 hours ago        Up 25 hours                             k8s_mixer_istio-telemetry-8597d8b86b-rtvfs_istio-system_8d5af400-7c2e-4cab-8fee-64050ac3de76_3
385fecb3572d        rancher/pause:3.1         "/pause"                 25 hours ago        Up 25 hours                             k8s_POD_istio-telemetry-8597d8b86b-rtvfs_istio-system_8d5af400-7c2e-4cab-8fee-64050ac3de76_0
4d87223a4af2        rancher/istio-proxyv2     "/usr/local/bin/pilo…"   25 hours ago        Up 25 hours
root@worker72:/home/jslab#
```

```
root@worker72:/home/jslab# docker exec 4d87223a4af2 ifconfig
eth0      Link encap:Ethernet  HWaddr b2:2d:11:40:1c:03
          inet addr:10.42.1.9  Bcast:0.0.0.0  Mask:255.255.255.255
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:174489 errors:0 dropped:0 overruns:0 frame:0
          TX packets:167359 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:20185982 (20.1 MB)  TX bytes:582220802 (582.2 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:30864 errors:0 dropped:0 overruns:0 frame:0
          TX packets:30864 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:551008609 (551.0 MB)  TX bytes:551008609 (551.0 MB)

root@worker72:/home/jslab#
```



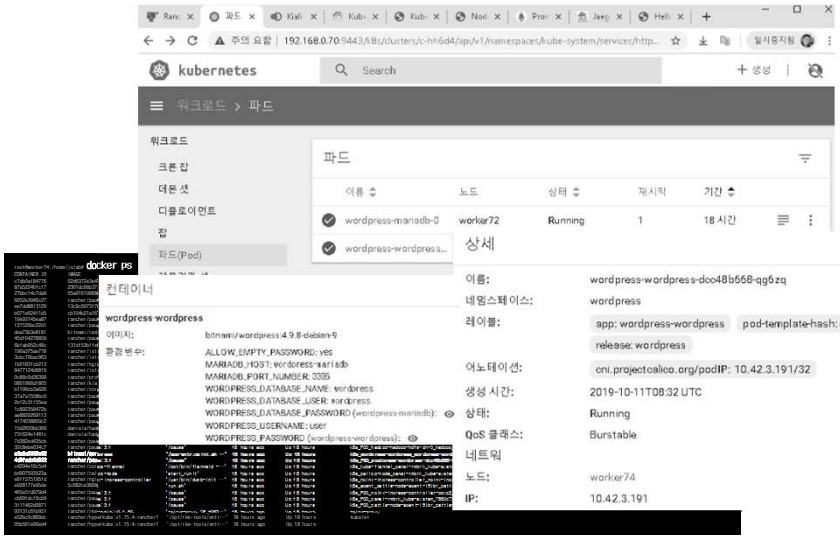
VIII. 네트워크 가상화

❖ **Kubernetes Pod: Istio-telemetry and Proxy @ iptables** (예)

```
root@worker72:/home/jslab# iptables -t nat -L -n

Chain KUBE-NODEPORTS (1 references)
target     prot opt source               destination
KUBE-MARK-MASQ  tcp  --  0.0.0.0/0            0.0.0.0/0            /* kubeflow/ambassador:ambassador */ tcp dpt:30138
KUBE-SVC-STYH4L3MNMUUEPBQ  tcp  --  0.0.0.0/0            0.0.0.0/0            /* kubeflow/ambassador:ambassador */ tcp dpt:30138
KUBE-MARK-MASQ  tcp  --  0.0.0.0/0            0.0.0.0/0            /* docker-registry/docker-registry:registry */ tcp dpt:31486
KUBE-SVC-5PSZEDW4UNLDD4A6  tcp  --  0.0.0.0/0            0.0.0.0/0            /* docker-registry/docker-registry:registry */ tcp dpt:31486
KUBE-MARK-MASQ  tcp  --  0.0.0.0/0            0.0.0.0/0            /* wordpress/wordpress-wordpress:http */ tcp dpt:31904
KUBE-SVC-2FUFTE55KQIFOMSM  tcp  --  0.0.0.0/0            0.0.0.0/0            /* wordpress/wordpress-wordpress:http */ tcp dpt:31904
KUBE-MARK-MASQ  tcp  --  0.0.0.0/0            0.0.0.0/0            /* wordpress/wordpress-wordpress:https */ tcp dpt:30526
KUBE-SVC-2YER72ONBESPQKXP  tcp  --  0.0.0.0/0            0.0.0.0/0            /* wordpress/wordpress-wordpress:https */ tcp dpt:30526

Chain KUBE-SEP-G4VQDTVXSGZWTZQE (1 references)
target     prot opt source               destination
KUBE-MARK-MASQ  all  --  10.42.1.9            0.0.0.0/0
DNAT       tcp  --  0.0.0.0/0            0.0.0.0/0            tcp to:10.42.1.9:9091

Chain KUBE-SEP-UVXKT3JO7F7QKUSS (1 references)
target     prot opt source               destination
KUBE-MARK-MASQ  all  --  10.42.1.9            0.0.0.0/0
DNAT       tcp  --  0.0.0.0/0            0.0.0.0/0            tcp to:10.42.1.9:42422

Chain KUBE-SEP-WI4Q7OH56TS2COK3 (1 references)
target     prot opt source               destination
KUBE-MARK-MASQ  all  --  10.42.1.9            0.0.0.0/0
DNAT       tcp  --  0.0.0.0/0            0.0.0.0/0            tcp to:10.42.1.9:15014

Chain KUBE-SEP-Z6QXVYMQAAERW6O5 (1 references)
target     prot opt source               destination
KUBE-MARK-MASQ  all  --  10.42.1.9            0.0.0.0/0
DNAT       tcp  --  0.0.0.0/0            0.0.0.0/0            tcp to:10.42.1.9:15004

Chain KUBE-SERVICES (2 references)
KUBE-MARK-MASQ  tcp  -- !10.42.0.0/16        10.43.7.130          /* istio-system/istio-telemetry:grpc-mixer-mtls cluster IP */ tcp dpt:15004
KUBE-SVC-POFVSRMRNLJ5KKAQ  tcp  --  0.0.0.0/0            10.43.7.130          /* istio-system/istio-telemetry:grpc-mixer-mtls cluster IP */ tcp dpt:15004
KUBE-SVC-TUZA6CYEP5VUF2XG  tcp  --  0.0.0.0/0            10.43.24.19          /* istio-system/istio-policy:grpc-mixer-mtls cluster IP */ tcp dpt:15004
KUBE-SVC-U2XTOAGOXQJP3ONI  tcp  --  0.0.0.0/0            10.43.24.19          /* istio-system/istio-policy:grpc-mixer cluster IP */ tcp dpt:9091
KUBE-MARK-MASQ  tcp  -- !10.42.0.0/16        10.43.7.130          /* istio-system/istio-telemetry:grpc-mixer cluster IP */ tcp dpt:9091
KUBE-SVC-LTOKVKL3D46W1GR3  tcp  --  0.0.0.0/0            10.43.7.130          /* istio-system/istio-telemetry:grpc-mixer cluster IP */ tcp dpt:9091
```

VIII. 네트워크 가상화

❖ **Kubernetes Pod: Wordpress @ K8s Dashboard**



https://kubernetes.io/

VIII. 네트워크 가상화

❖ **Kubernetes Pod: Wordpress @ K8s Dashboard**

- **Pod Information @ iptables**
- **docker ps**

Opensource Networking

james@jslab.kr

```
root@worker74:/home/jslab# iptables -t nat -L -n

Chain KUBE-NODEPORTS (1 references)

target     prot opt source               destination
KUBE-MARK-MASQ  tcp  --  0.0.0.0/0            0.0.0.0/0            /* wordpress/wordpress-wordpress:https */ tcp dpt:30526
KUBE-SVC-2YER72GNBESPQKXP  tcp  --  0.0.0.0/0            0.0.0.0/0            /* wordpress/wordpress-wordpress:https */ tcp dpt:30526
KUBE-MARK-MASQ  tcp  --  0.0.0.0/0            0.0.0.0/0            /* wordpress/wordpress-wordpress:http */ tcp dpt:31904
KUBE-SVC-2FUFTE55KQIFOMSM  tcp  --  0.0.0.0/0            0.0.0.0/0            /* wordpress/wordpress-wordpress:http */ tcp dpt:31904

Chain KUBE-SERVICES (2 references)

target     prot opt source               destination
KUBE-MARK-MASQ  tcp  -- !10.42.0.0/16        10.43.0.91           /* wordpress/wordpress-wordpress:https cluster IP */ tcp dpt:443
KUBE-SVC-2YER72GNBESPQKXP  tcp  --  0.0.0.0/0            10.43.0.91           /* wordpress/wordpress-wordpress:https cluster IP */ tcp dpt:443
KUBE-MARK-MASQ  tcp  -- !10.42.0.0/16        10.43.205.38         /* wordpress/wordpress-mariadb:mysql cluster IP */ tcp dpt:3306
KUBE-SVC-DEV3U2PZRMWXYVGH  tcp  --  0.0.0.0/0            10.43.205.38         /* wordpress/wordpress-mariadb:mysql cluster IP */ tcp dpt:3306
KUBE-MARK-MASQ  tcp  -- !10.42.0.0/16        10.43.0.91           /* wordpress/wordpress-wordpress:http cluster IP */ tcp dpt:80
KUBE-SVC-2FUFTE55KQIFOMSM  tcp  --  0.0.0.0/0            10.43.0.91           /* wordpress/wordpress-wordpress:http cluster IP */ tcp dpt:80
```

```
root@worker74:/home/jslab# docker ps
CONTAINER ID        IMAGE                  COMMAND                  CREATED             STATUS              PORTS               NAMES
a3a3a5386c92        bitnami/wordpress      "/app-entrypoint.sh …"   18 hours ago        Up 18 hours                             k8s_wordpress-wordpress_wordpress-wordpress-dcc48b668-qg6zq_wordpress_8d0edcc3-f572-4230-b148-af24c4fe8ae8_0
4d8fadc6d822        rancher/pause:3.1      "/pause"                 18 hours ago        Up 18 hours                             k8s_POD_wordpress-wordpress-dcc48b668-qg6zq_wordpress_8d0edcc3-f572-4230-b148-af24c4fe8ae8_0
```

JS Lab

VIII. 네트워크 가상화

❖ **Kubernetes Pod: Scaleout for Wordpress @ K8s Dashboard**

Opensource Networking

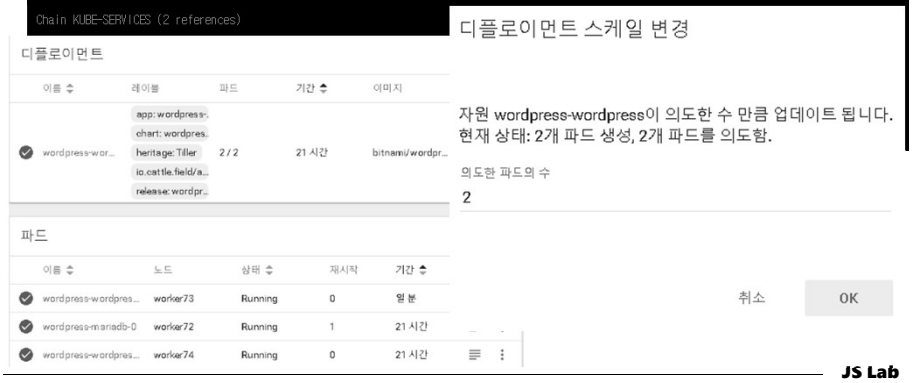
james@jslab.kr

root@worker74:/home/jslab# iptables -t nat -L -n

Chain KUBE-NODEPORTS (1 references)

target prot opt source destination
KUBE-MARK-MASQ tcp -- 0.0.0.0/0 0.0.0.0/0 /* wordpress/wordpress-wordpress:https */ tcp dpt:30526
KUBE-SVC-2YER72GNBESPQKXP tcp -- 0.0.0.0/0 0.0.0.0/0 /* wordpress/wordpress-wordpress:https */ tcp dpt:30526
KUBE-MARK-MASQ tcp -- 0.0.0.0/0 0.0.0.0/0 /* wordpress/wordpress-wordpress:http */ tcp dpt:31904
KUBE-SVC-2FUFTE55KQIFOMSM tcp -- 0.0.0.0/0 0.0.0.0/0 /* wordpress/wordpress-wordpress:http */ tcp dpt:31904

Chain KUBE-SERVICES (2 references)



JS Lab

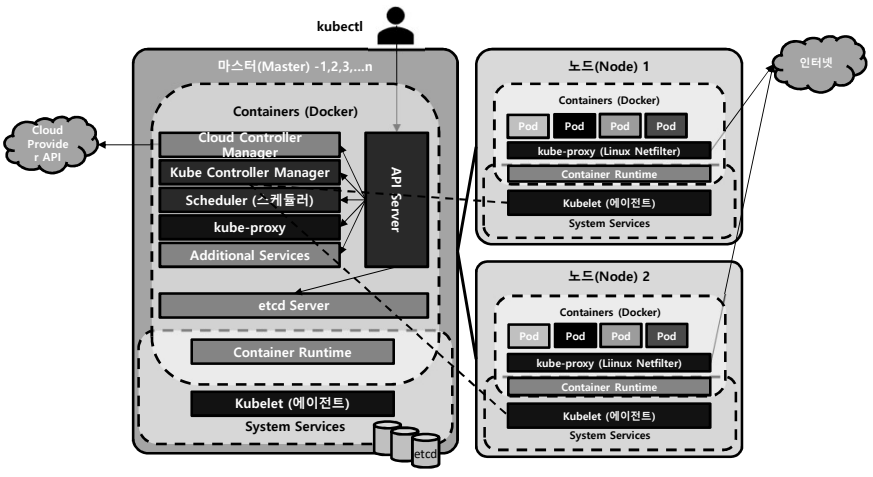
VIII. 네트워크 가상화

❖ **Kubernetes Pod: Wordpress w/ LB @ iptables**

```
root@worker74:/home/jslab# iptables -t nat -L -n

Chain KUBE-NODEPORTS (1 references)
target     prot opt source               destination
KUBE-MARK-MASQ  tcp  --  0.0.0.0/0            0.0.0.0/0            /* wordpress/wordpress-wordpress:https */ tcp dpt:30526
KUBE-SVC-2YER72GNBESPQKXP  tcp  --  0.0.0.0/0            0.0.0.0/0            /* wordpress/wordpress-wordpress:https */ tcp dpt:30526
KUBE-MARK-MASQ  tcp  --  0.0.0.0/0            0.0.0.0/0            /* wordpress/wordpress-wordpress:http */ tcp dpt:31904
KUBE-SVC-2FUFTE55KQIFOMSM  tcp  --  0.0.0.0/0            0.0.0.0/0            /* wordpress/wordpress-wordpress:http */ tcp dpt:31904

Chain KUBE-SERVICES (2 references)
target     prot opt source               destination
KUBE-MARK-MASQ  tcp  -- !10.42.0.0/16         10.43.0.91           /* wordpress/wordpress-wordpress:https cluster IP */ tcp dpt:443
KUBE-SVC-2YER72GNBESPQKXP  tcp  --  0.0.0.0/0            10.43.0.91           /* wordpress/wordpress-wordpress:https cluster IP */ tcp dpt:443
KUBE-MARK-MASQ  tcp  -- !10.42.0.0/16         10.43.205.38         /* wordpress/wordpress-mariadb:mysql cluster IP */ tcp dpt:3306
KUBE-SVC-DEV3U2PZRMWXYVGH  tcp  --  0.0.0.0/0            10.43.205.38         /* wordpress/wordpress-mariadb:mysql cluster IP */ tcp dpt:3306
KUBE-MARK-MASQ  tcp  -- !10.42.0.0/16         10.43.0.91           /* wordpress/wordpress-wordpress:http cluster IP */ tcp dpt:80
KUBE-SVC-2FUFTE55KQIFOMSM  tcp  --  0.0.0.0/0            10.43.0.91           /* wordpress/wordpress-wordpress:http cluster IP */ tcp dpt:80
```

VIII. 네트워크 가상화

❖ **Kubernetes Architecture:**



https://kubernetes.io/

VIII. 네트워크 가상화

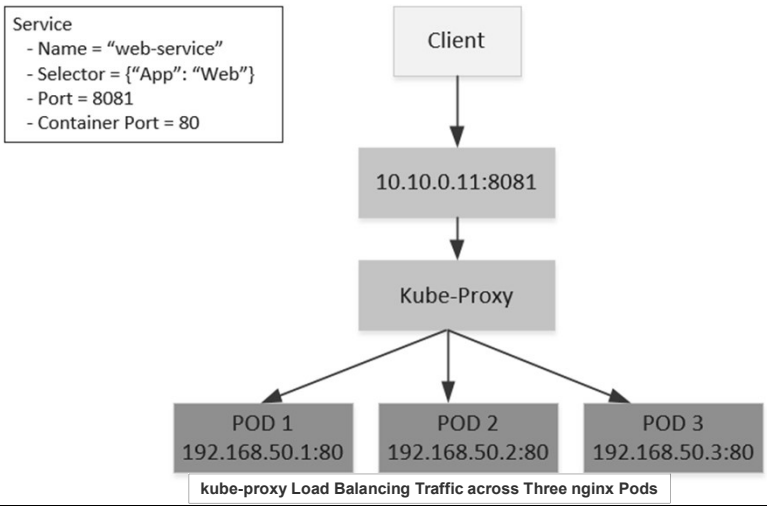
❖ **Kubernetes Networking:**

❖ **Master Server Components:**

- **kube-apiserver**
  Provides frontend and controllable APIs to control the Kubernetes environment via an orchestration or management platform.
- **etcd**
  Kubernetes data store.
- **kube-scheduler**
  Responsible for allocating worker nodes for newly created pods with no nodes assigned.
- **kube-controller-manager**
  Includes four controllers as:
  - **Node Controller**: Responsible for finding out when a node goes down
  - **Replication Controller**: Responsible for replication in the system
  - **Endpoints Controller**: Populates the endpoints objects (i.e, joins services & pods)
  - **Service Account & Token Controllers**: Create default accounts and API access tokens for new namespaces
- **cloud-controller-manager**
  Responsible to run controllers that interact with the underlying cloud providers, such as public cloud (AWS, Azure) or on-prem cloud (such as OpenStack).

VIII. 네트워크 가상화

❖ **Kubernetes Networking:**



kube-proxy Load Balancing Traffic across Three nginx Pods

VIII. 네트워크 가상화

❖ **Kubernetes Networking:**

Kubernetes is compatible with the following networking solutions to create networking clusters:

- **Cisco ACI**
- **Big Switch Big Cloud Fabric**
- **Tungsten Fabric (OpenContrail)**
- **VMware NSX-T**
- **OpenVSwitch (OVS)**
- **Project Calico**

Opensource Networking
james@jslab.kr

https://kubernetes.io/

VIII. 네트워크 가상화

❖ **Service Mesh (서비스 메시):**

분산 시스템 및 클라우드 네이티브 애플리케이션에 주로 사용되는 서비스 간 통신에 중점을 둔 인프라 계층

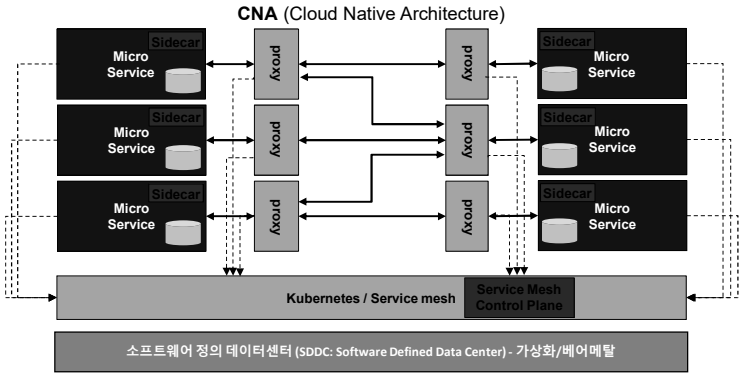
- **서비스 또는 마이크로 서비스의 모음으로 컨테이너형 애플리케이션이 구축되면 서비스 메시가 형성**
- **서비스간 연결하고 서비스의 상호 작용을 관리하기 위해 IP 주소와 포트 위에 계층을 만들고 로드 밸런싱, 모니터링, 서비스 간 인증 등을 제공**
- **예: Istio, linkerd**

Opensource Networking
james@jslab.kr

VIII. 네트워크 가상화

❖ 서비스 메쉬 관리 체계

- Sidecar Design Pattern: 라우터 내장 CB, LB, SD 내장
- 오픈소스 CNCF의 'Istio'는 정책 강화 Telemetry 제공



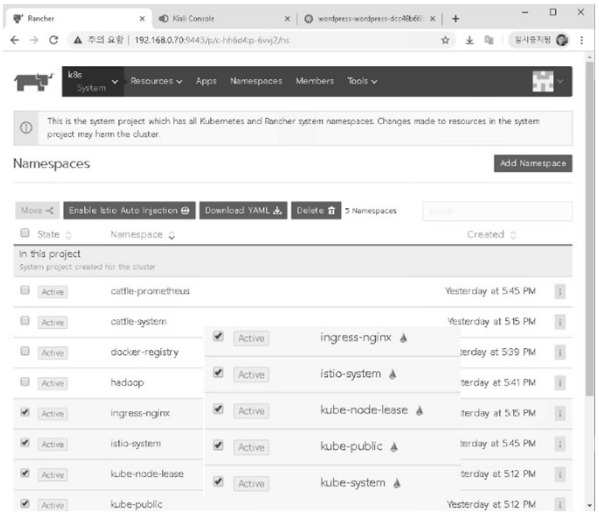
Circuit Breaker(CB), Load Balancer(LB), Service Discovery(SD)

JS Lab

Opensource Networking
james@jslab.kr

VIII. 네트워크 가상화

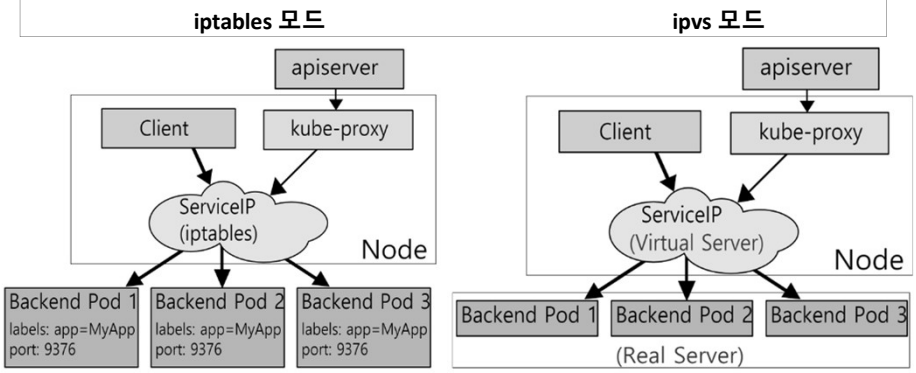
❖ 서비스 메시 관리 체계



JS Lab

VIII. 네트워크 가상화

- ❖ **IP Virtual Server** (IPVS)
- ❖ **리눅스 기반에 설치되는 서버로 L4 기능을 대체**
- ❖ **ipvs(IP Virtual Server) 모드는 리눅스 커널에 있는 L4 로드밸런싱 기술로 Netfilter에 포함**
- ❖ **ipvs는 커널스페이스에서 작동하고 데이터 구조를 해시테이블로 저장해서 가지고 있기 때문에 iptables보다 빠르고 좋은 성능**



출처: https://arisu1000.tistory.com/27839

IX. NFV(Network Function Virtualization)

- **Introduction to Network Function Virtualization**
- **Virtual Firewalls**
- **pfSense Open Source Virtual Firewall**
- **Snort Open Source Virtual IPS/IDS**
- **Virtual Load Balancers**
- **Katran Open Source Load Balancer**
- **HAproxy Open Source Virtual Load Balancer**
- **Virtual Routers**
- **VyOS Open Source Virtual Router/Firewall**
- **Service Chaining**
- **uCPE** (Universal Customer Premises Equipment)

Opensource Networking
james@jslab.kr

https://www.linuxfoundation.org/projects/networking/

IX. NFV(Network Function Virtualization)

❖ **Introduction to Network Function Virtualization:**

❖ **Some important NFV use cases are:**

- **Service Providers**
  - ✓ At the core, virtualizing routers
  - ✓ At the edge, virtualizing CPE (Customer Premises Equipment)
- **Enterprises**
  - ✓ Decentralizing network functions, such as firewall and load balancer
  - ✓ Building micro-segments and microservices
- **Cloud and Datacenter Providers**
  - ✓ Building a virtualized infrastructure for tenants
  - ✓ Decentralizing network functions, such as firewall and load balancer.

Opensource Networking
james@jslab.kr

IX. NFV(Network Function Virtualization)

❖ **Introduction to Network Function Virtualization:**

Virtual Router / Switch	Virtual Firewall / IPS / IDS	Virtual Load Balancer	SD-WAN	UC-IP Telephony
VyOS, open source router	pfSense, open source firewall	HAproxy, open source	Silver Peak Virtual Unity	Cisco CallManager
Vyatta, commercial router	Juniper vSRX, commercial firewall	F5 vLTM, commercial	Riverbed SteelConnect	Asterisk-based systems
Cisco Nexus 1000v, commercial	Snort, open source IDS	Loadbalancer.org, commercial	Cisco Viptela	Avaya
Cisco CSR	Cisco ASAv	Avi Networks, commercial	VeloCloud Networks	Skype for Business
VMware NSX	VMware NSX	VMware NSX	Versa	Freeswitch

IX. NFV(Network Function Virtualization)

❖ **Virtual Firewalls:**

Vendor	Name	Commercial or Open Source
Juniper	vSRX	Commercial
Cisco	ASAv	Commercial
FortiGate	Virtual Appliances	Commercial
Sophos	Virtual Appliance	Commercial
VMware	NSX Firewall	Commercial
Stonegate (ForcePoint)	Virtual Appliance	Commercial
Rubicon Communications	pfSense	Open Source
Palo Alto Networks	Virtual Appliance	Commercial

IX. NFV(Network Function Virtualization)

❖ **pfSense Open Source Virtual Firewall:**

pfSense - Quick Summary	
Name	pfSense
By	Rubicon Communications, LLC (Netgate)
Where it runs	On a dedicated hardware appliance or on a virtual machine
What it does	pfSense is a stateful firewall with industry standard capabilities and features
Features	Firewalling, logging, Layer 2 transparent firewalling, state table control, NAT, high availability clustering, multi-WAN load balancing, server load balancing, IP Sec VPN, SSL VPN, PPPOE Server, reporting and graphs, captive portal, DHCP server
What it can do out-of-the-box	You can install pfSense on a hypervisor, assign virtual interfaces, and start using it as a firewall. pfSense can be used as a virtual firewall in a microsegmentation environment, or can be used as CPE, or for NAT configuration.

Opensource Networking

james@jslab.kr

JS Lab

IX. NFV(Network Function Virtualization)

❖ **Snort Open Source Virtual IPS/IDS:**

Snort - Quick Summary	
Name	Snort
By	Currently hosted and developed by Cisco
Where it runs	On a dedicated hardware appliance or on a virtual machine
What it does	Snort is a network Intrusion Detection/Prevention System
Features	Traffic logging, detecting and matching packet header information (L2-L7), finds patterns and executes actions such as Alert, Block, Replace, etc. Has flexible rules and policies
What it can do out-of-the-box	You can install Snort on a virtual machine and have it connected to monitor and check the traffic of a network segment or even the traffic that is going to a specific host. Snort comes with a predefined attack signature database; you can register to receive the updated attack signature database regularly, which is a paid subscription service.

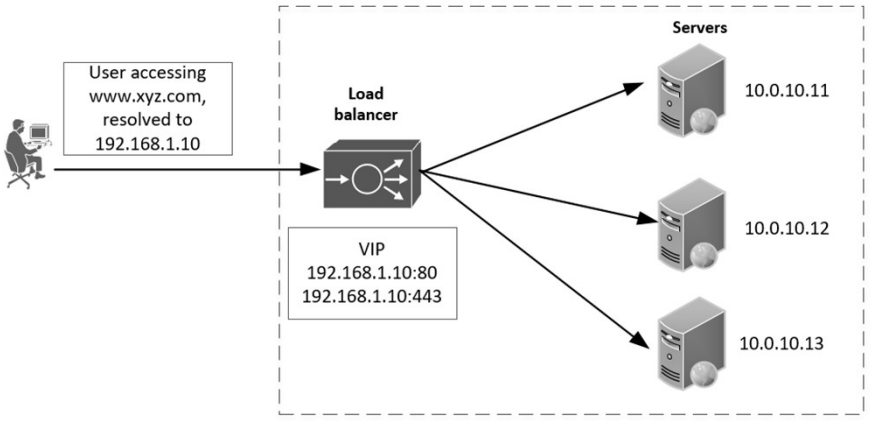
Opensource Networking

james@jslab.kr

JS Lab

IX. NFV(Network Function Virtualization)

❖ **Virtual Load Balancers:**



IX. NFV(Network Function Virtualization)

❖ **Virtual Load Balancers:**

Vendor	Name	Commercial or Open Source
F5 Networks	Virtual LTM	Commercial
Citrix	Virtual Load Balancer	Commercial
Avi Networks	Virtual Load Balancer	Commercial
Barracuda	Virtual Load Balancer	Commercial
Kemp	Kemp Virtual	Commercial
Fortigate	Virtual Load Balancer	Commercial
HAproxy Technologies	HA Proxy	Open Source
Facebook	Katran	Open Source

Opensource Networking
james@jslab.kr

JS Lab

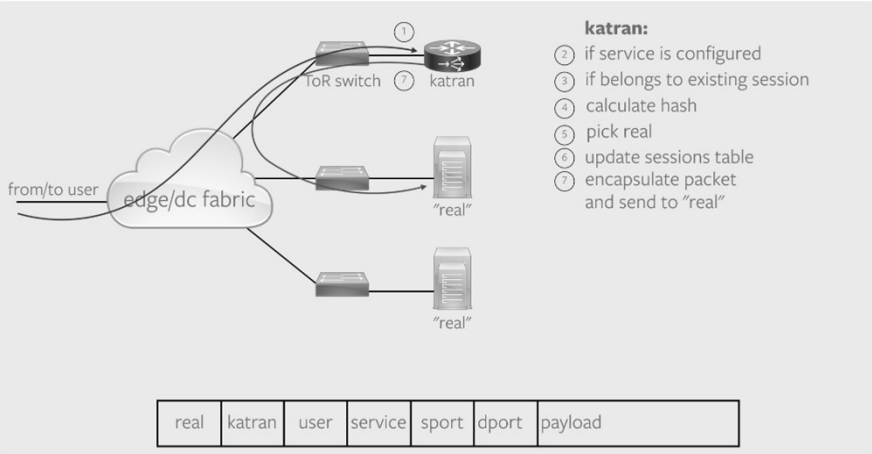
IX. NFV(Network Function Virtualization)

❖ **Katran Open Source Load Balancer:**

Katran - Quick Summary	
Name	Katran
By	Facebook
Where it runs	On dedicated virtual machines
What it does	Hy performance load balancing
Features	Open source, fast (especially with XDP in the driver mode), performance scales linearly with a number of NIC RX queu es, RSS (Received Side Scaling) friendly encapsulation.
What it can do out-of-the-box	You can use it for load balancing in high volume environme nts

IX. NFV(Network Function Virtualization)

❖ **Katran Open Source Load Balancer:**

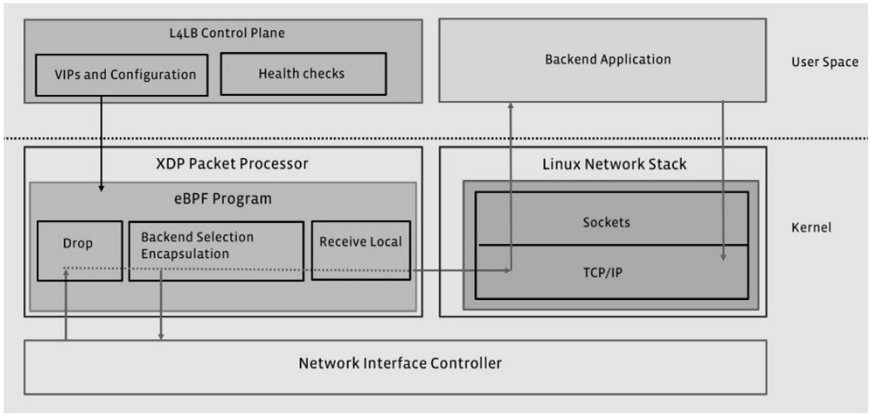


Opensource Networking
james@jslab.kr

JS Lab

IX. NFV(Network Function Virtualization)

❖ **Katran Open Source Load Balancer:**



Katran Architecture

IX. NFV(Network Function Virtualization)

❖ **HAproxy Open Source Virtual Load Balancer:**

HAProxy - Quick Summary	
Name	HAProxy
By	HAPROXY Community
Where it runs	On a host
What it does	High performance L4-L7 load balancing
Features	L4-L7 load balancing, SSL
What it can do out-of-the-box	High performance load balancing, used in high profile websites such as Git Hub, Vimeo, Stack Overflow, etc.

IX. NFV(Network Function Virtualization)

❖ **Virtual Routers:**

Vendor	Name	Commercial or Open Source
Cisco	CSR (Cloud Service Router)	Commercial
Cisco	ISRv (Integrated Services Virtual Router)	Commercial
Juniper	vMX	Commercial
Brocade (acquired)	Vyatta	Commercial
Alcatel Lucent	VSR	Commercial
VMware	NSX	Commercial
Cloud Router	Cloud Router	Open Source
VyOS	VyOS	Open Source
Quagga	Linux Router (Quagga)	Open Source

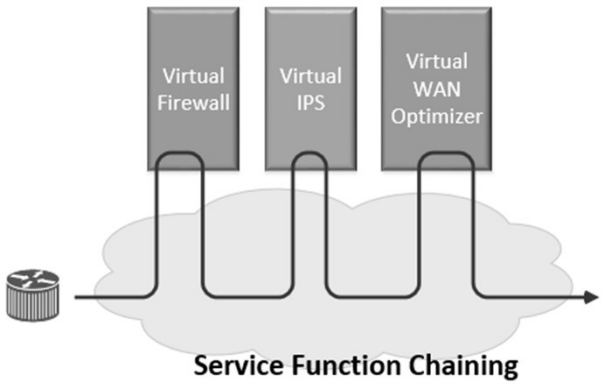
IX. NFV(Network Function Virtualization)

❖ **VyOS Open Source Virtual Router/Firewall:**

VyOS - Quick Summary	
Name	VyOS
By	Open Source
Where it runs	As a virtual appliance or on a x86 server
What it does	Routing and firewalling
Features	Layer 2, VLANs, 802.1q, QinQ, Layer 3, BGP, OSPF, RIP, PBR, ECMP, zone-based firewalling, tunneling, PPPOE, GRE, L2TP, VXLAN, IPSec VPN, SSL VPN, NAT, DHCP server, VRRP, sFlow, web proxy, QoS and traffic shaping. Uses a CLI for configuration without GUI
What it can do out-of-the-box	You can just load VyOS on a virtual machine and use it as a router to connect to an ISP or route between networks, or use it as VPN server or a firewall within your network.

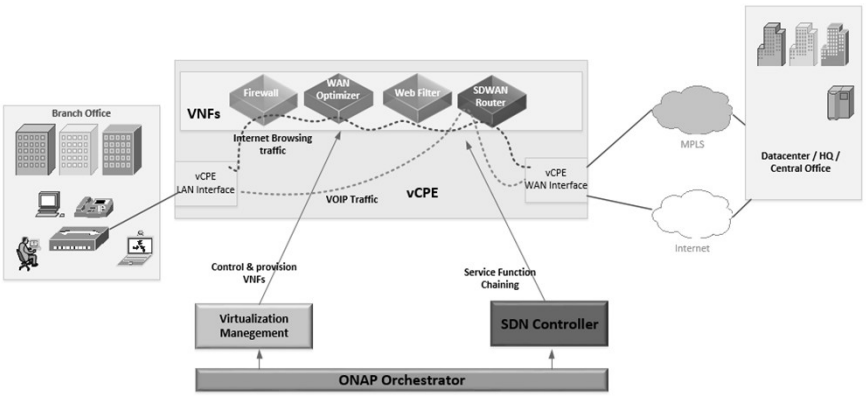
IX. NFV(Network Function Virtualization)

❖ **Service Chaining:**



IX. NFV(Network Function Virtualization)

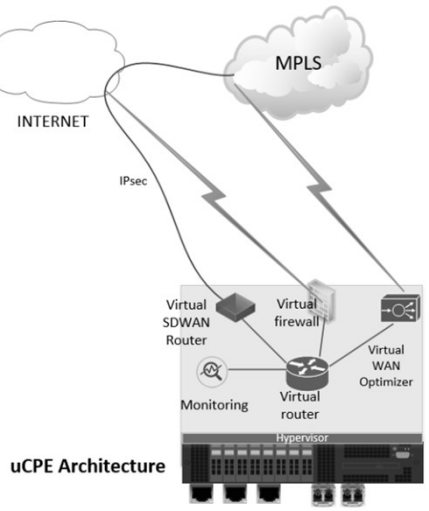
❖ **Service Chaining:**



NFV and Service Chaining in a Service Provider Network

IX. NFV(Network Function Virtualization)

❖ **uCPE (Universal Customer Premises Equipment):**



uCPE Architecture

IX. NFV(Network Function Virtualization)

❖ **uCPE (Universal Customer Premises Equipment):**

•Virtual routers

These are standard packet-forwarding systems. They can route and run routing protocols and other features, such as NAT, Policy-Based Routing, and so on.

•Virtual firewalls

These are standard stateful or stateless firewalls with L3-L7 filtering capabilities. They may be equipped with deep packet inspection engines to provide features such as IDS and IPS.

•Virtual load balancers

L4 to L7 load balancers with capability of hosting virtual IPs (VIP) and forwarding (and NAT) the traffic to real servers. They may be equipped with Web Application Firewall (WAF) features.

•Virtual WAN optimizers

These include caching, TCP optimization, and protocol acceleration.

•SD-WAN routers

They are used to logically bound multiple WAN and Internet connections and build VPN tunnels back to the SD-WAN head-end units in data centers. They are used for intelligent link measurement and application-based routing.

Opensource Networking
james@jslab.kr

X. 네트워크 자동화

- Introduction to Network Automation
- Ansible
- Puppet
- Chef
- Python
- Using Netmiko to Connect to a Networking Device
- NETCONF

Opensource Networking
james@jslab.kr

https://www.linuxfoundation.org/projects/networking/

X. 네트워크 자동화

❖ **Introduction to Network Automation:**

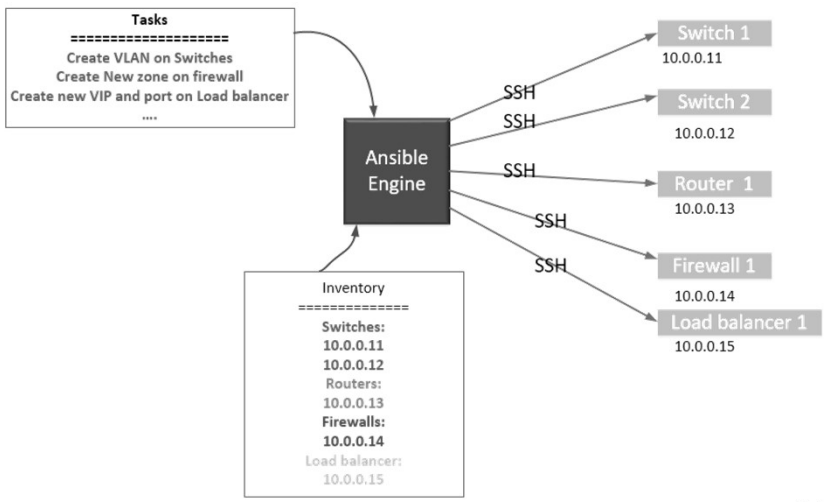
Next, we will explore the network automation tools that can help you build automated workflows to execute tasks in a network. Network automation works on both next-generation SDN and legacy networks. Even if you have an aging network equipment, you are still able to use automation tools to make faster and reliable changes.

Automation tools that we are exploring in this chapter are:

- Ansible
- Puppet
- Chef.

X. 네트워크 자동화

❖ **Ansible:**



X. 네트워크 자동화

❖ **Ansible:**

Ansible - Quick Summary	
Name	Ansible
By	Red Hat
Where it runs	On a workstation or a server
What it does	You can build automation playbooks to execute repeatable tasks on multiple devices
Features	Supports SSH/telnet access to networking devices. Ansible has multiple ready-made plugins for networking products, such as Cisco IOS, Arista, F5, Juniper, Cumulus, etc.
What it can do out-of-the-box	You can simply create Ansible scripts to tell Ansible to execute specific tasks on your equipment.

Opensource Networking
james@jslab.kr

X. 네트워크 자동화

❖ **Puppet:**

Puppet - Quick Summary	
Name	Puppet
By	PuppetLabs
Where it runs	On a server. Requires agents to be installed on all managed devices (except Cisco IOS devices)
What it does	Manages the configuration of network devices and other servers
What it can do out-of-the-box	You can use Puppet to manage your Cisco IOS-based catalyst switches without the need of agents (as of June 2018). Puppet also supports other products, from Juniper, Cumulus Networks, OpenSwitch, Cisco ACI, etc.

Opensource Networking
james@jslab.kr

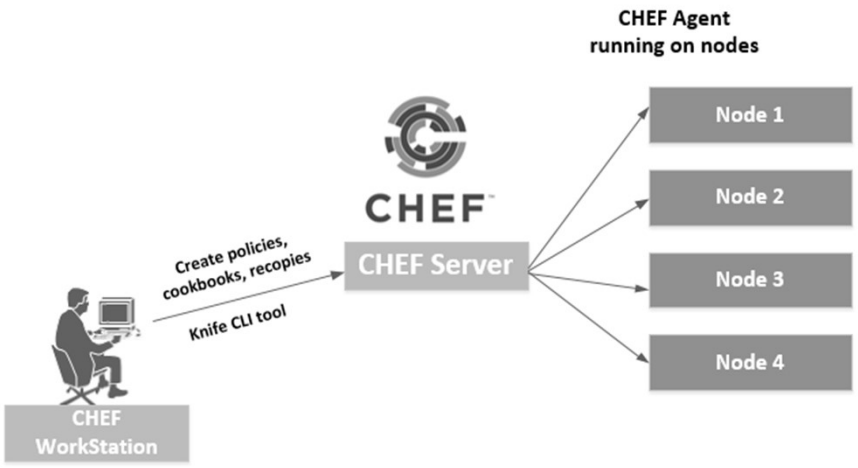
X. 네트워크 자동화

❖ **Chef:**

Chef - Quick Summary	
Name	Chef
By	Chef
Where it runs	On a separate server. Requires Chef agents to be installed on all managed nodes
What it does	Automates tasks for managing the infrastructure network, servers, application servers, as well as building and deploying applications
What it can do out-of-the-box	You can manage and automate the configuration of networking devices that come with a Chef agent or allow you to install a Chef agent. This includes all networking tools running on a host-based such as OVS, as well as other networking appliances such as Cisco Nexus switches.

X. 네트워크 자동화

❖ **Chef Architecture:**



X. 네트워크 자동화

❖ **Python:**

Python is a very popular modern programming language. Python is used for network automation, especially when you need to write complex, customized rules that are needed to perform specific sets of tasks. Python has many networking libraries that you can use to manage your network. Using Python, you can build programs that can connect to a network device, execute commands, grab the outputs, and show you the results.

Python supports multiple protocols such as SSH, SNMP, Telnet and APIs to communicate with a networking device. If you have an aging device that only supports older versions of SSH, you will be able to use Python's SSH library (Paramiko) to manage that device.

When dealing with legacy networking devices that only support CLI (via SSH or Telnet), the main issue is to parse the outputs. In a CLI environment, the device always returns a stream of characters which is formatted for human reading. For example, when you issue a "show run interface Gigabit 1" command, the output is a set of characters that define the interface speed, its mode (Layer 2 without VLANs, Layer 2 with 802.1q, or Layer 3), VLANs, etc.:

https://www.python.org/

X. 네트워크 자동화

❖ **Using Netmiko to Connect to a Networking Device:**

The following example shows how to define a networking device and use Netmiko to connect to the device and execute some configurations.

```
from netmiko import ConnectHandler
cisco_881 = {
    'device_type': 'cisco_ios',
    'ip':   '10.10.10.10',
    'username': 'test',
    'password': 'password',
    'port' : 8022,          # optional, defaults to 22
    'secret': 'secret',     # optional, defaults to ''
    'verbose': False,       # optional, defaults to False
}
```

Establish an SSH connection to the device by passing in the device dictionary.

```
net_connect = ConnectHandler(**cisco_881)
```

Execute **show** commands.

```
output = net_connect.send_command('show ip int brief')
print(output)
Interface                  IP-Address OK? Method Status                Protocol
FastEthernet0              unassigned YES unset down                  down
FastEthernet1              unassigned YES unset down                  down
FastEthernet2              unassigned YES unset down                  down
FastEthernet3              unassigned YES unset down                  down
FastEthernet4              10.10.10.10 YES manual up                    up
Vlan1
```


https://kubernetes.io/

X. 네트워크 자동화

❖ **NETCONF:**

X. 네트워크 자동화

❖ **NETCONF:**

NETCONF Command	Description
`<get>`	Retrieve running configuration and device state information
`<get-config>`	Retrieve all or part of a specified configuration datastore
`<edit-config>`	Edit a configuration datastore by creating, deleting, merging, or replacing content
`<copy-config>`	Copy an entire configuration datastore to another configuration datastore
`<delete-config>`	Delete a configuration datastore
`<lock>`	Lock an entire configuration datastore of a device
`<unlock>`	Release a configuration datastore lock previously obtained with the `<lock>` com mand
`<close-session>`	Request graceful termination of a NETCONF session
`<kill-session>`	Force the termination of a NETCONF session

Opensource Networking
james@jslab.kr

XI. Network Data Analytics

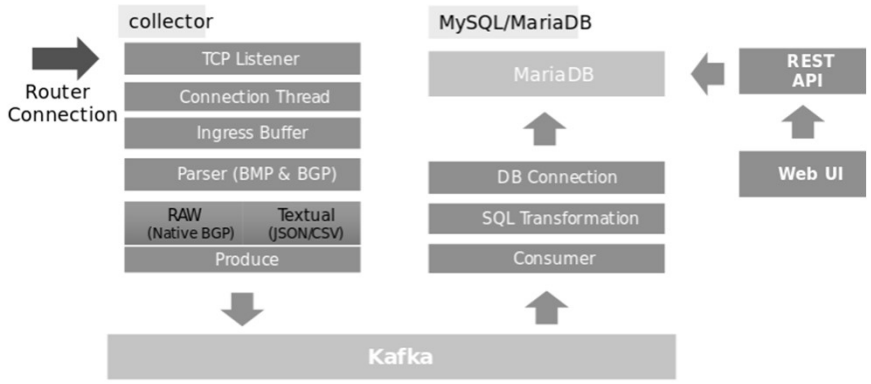
- **SNAS (Streaming Network Analytics System)**
- **PNDA**
- **PNDA Principles and Benefits**
- **BGP Analytics Application (Example)**

https://www.linuxfoundation.org/projects/networking/

Opensource Networking
james@jslab.kr

XI. Network Data Analytics

❖ **SNAS (Streaming Network Analytics System):**

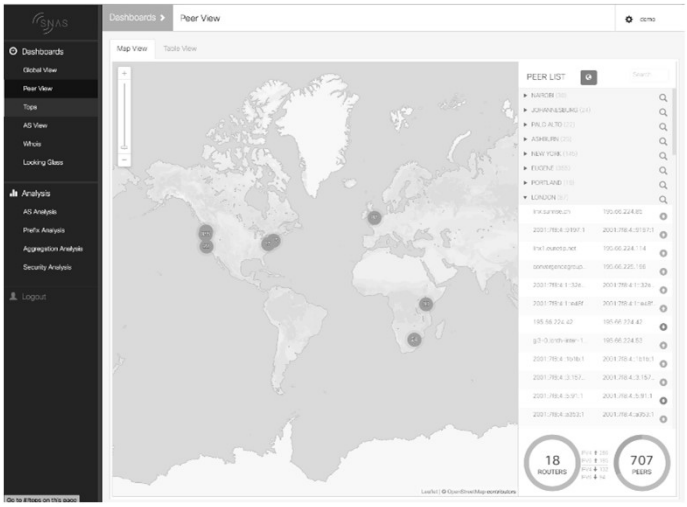


SNAS Architecture

https://www.snas.io/

XI. Network Data Analytics

❖ **SNAS (Streaming Network Analytics System):**



SNAS Browser UI

http://www.snas.io/demo/demo-ui-rv/

Opensource Networking
james@jslab.kr

XI. Network Data Analytics

❖ **SNAS (Streaming Network Analytics System):**

Using Grafana with SNAS

http://www.snas.io/demo/demo-grafana/

Opensource Networking
james@jslab.kr

XI. Network Data Analytics

❖ **PNDA:**

PNDA features:

- Open source platform for Network Data Analytics
- Aggregates data like logs, metrics and network telemetry
- Scales up to consume millions of messages per second
- Efficiently distributes data with a publisher and a subscriber model
- Processes bulk data in batches, or streams data in real-time
- Manages the lifecycle of applications that process and analyze data
- Lets you explore data using interactive notebooks.

PNDA has a 3-tier architecture:

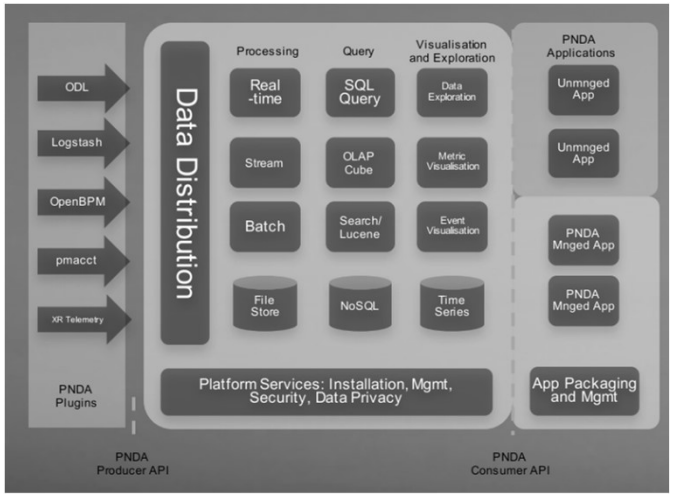
- Log ingestion plugin to get data into PNDA
- Analysis engine, which includes data distribution, parsers, storage, big data queries, and data visualization
- Consumer application - PNDA applications that utilize the PNDA analytical information to produce specific use cases for the user.

http://pnda.io/

Opensource Networking
james@jslab.kr

XI. Network Data Analytics

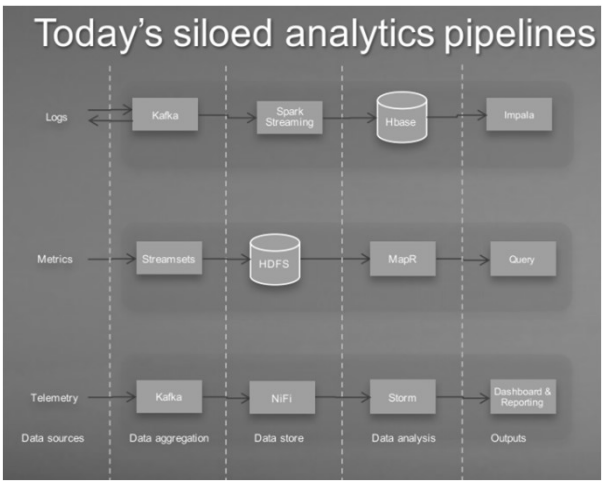
❖ **PNDA:**



http://pnda.io/overview

XI. Network Data Analytics

❖ **PNDA Principles and Benefits:**

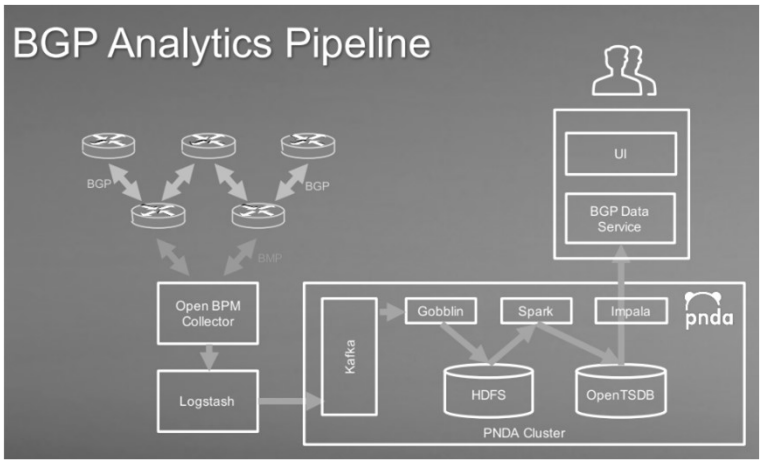


Siloed Analytics Pipelines

https://kubernetes.io/

XI. Network Data Analytics

❖ **BGP Analytics Application (Example):**



BGP Analytics Pipeline

https://kubernetes.io/

Opensource Networking
james@jslab.kr

XII. Use Case

- All Projects on One Page
- Use Case: Service Provider Transit Network
- Use Case: Service Provider Core Network
- Use Case: Service Provider uCPE
- Use Case: Cloud Providers and Enterprise Datacenters

https://www.linuxfoundation.org/projects/networking/

XII. Use Case

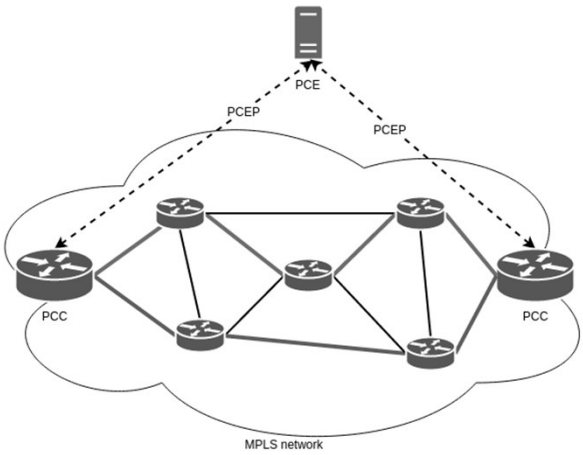
❖ Use Case: Service Provider Transit Network

Example: Using open source technologies, service providers can build a software defined transit network:

- Bare metal switches
  Bare metal (1G/10G/40G/100G/400G) switches can be used to build the transit network connected to a distributed fiber infrastructure.
- Bare metal optical
  Using new bare metal ODTN (Open Disaggregated Transport Network), service providers can leverage the software defined optical networking.
- Switch operating system
  Service providers can load open source networking operating system such as ONL (Open Network Linux), OpenSwitch or Trellis.
- SDN Controller
  Service providers can use open source SDN controllers such as OpenDaylight or ONOS as a platform to manage and control the transit network. The SDN controllers will be able to automatically populate the flow tables of the transit switches in order to deliver packets to customers and subscribers.
- Orchestration
  Using ONAP as an orchestration platform can help service providers to create an automated orchestration system that can manage not only the network, but also interact with other OSS and BSS services.
- Analytics
  Leveraging PNDA and SNAS can help service providers build a monitoring and analytics platform for their network.

XII. Use Case

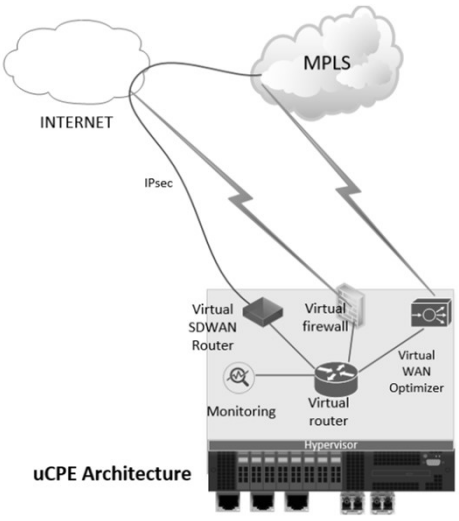
❖ **Use Case: Service Provider Core Network**



PCEP-Based Architecture

XII. Use Case

❖ **Use Case: Service Provider uCPE**



uCPE Architecture

XII. Use Case

❖ **Use Case: Cloud Providers and Enterprise Datacenters**

•Bare metal switches

Bare metal (1G/10G/40G/100G/400G) switches can be used to deploy a Clos-based leaf-spine switch architecture.

•Switch operating system

Service providers can load open source networking operating systems such as ONL (Open Network Linux), OpenSwitch or Trellis.

•SDN Controller

Service providers can use open source SDN controllers such as OpenDaylight or ONOS as a platform to manage and control the transit network. The SDN controllers will be able to automatically populate the flow tables of the transit switches in order to deliver packets to customers and subscribers.

Opensource Networking
james@jslab.kr

JS Lab


Community for KOREN AI Network Lab
James Ahn @ james@jslab.kr
Q & A
JS Lab