# 5G 네트워크

## (Day 4. 5G 네트워크 인프라 가상화 기술)

2022년 7월
안종석
james@jslab.kr



1

- Day 1: 5G 네트워크 기술
- Day 2: Enhanced Mobile Broadband
- Day 3: Private 5G와 테스트베드
- Day 4: 5G 네트워크 인프라 가상화 기술
- Day 5: Cloud Native 5G 인프라
- (별도) Day 4~5 실습교재



2

## ➢ Day 4: 5G 네트워크 인프라 가상화 기술

- Review Day 1 / Day 2 / Day3
- 통신 환경의 가상화 기술
- 5G Network 가상화
- 부록. 오픈스택
- 실습. 5G 네트워크 가상화 (별도 교재)





## ➢ (별도) Day 4~5 시연/실습

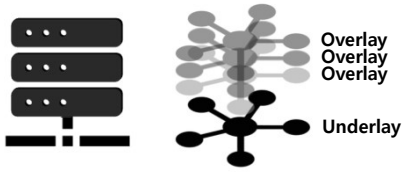
- 개요
- 5 Nodes in One-box Lab (시연)
- 2 Box Lab (실습)
- Cloud Native 5G Network Testbed (실습)

# DAY 4: 5G 네트워크 인프라 가상화 기술

5

❖ TTA: 네트워크 가상화, -假想化, Network Virtualization

- 하나의 물리적 네트워크가 마치 여러 개의 다른 기종 프로토콜이 운영되는 논리적 오버레이 네트워크로 운용되는 것을 가리키는 말. 클라우드 컴퓨팅과 미래 인터넷에서 핵심기술로 대두되고 있다. 인터넷에서 요구하는 가상화 기술에는 호스트 가상화, 링크 가상화, 라우터 가상화, 스위치 가상화 등이 있다.




Source: http://terms.tta.or.kr/dictionary/dictionaryView.do?subject=%EB%84%A4%ED%8A%B8%EC%9B%8C%ED%81%AC+%EA%B0%80%EC%83%81%ED%99%94

JS Lab

5

# DAY 4: 5G 네트워크 인프라 가상화 기술

6

❖ 위키: 네트워크 가상화

- 네트워크 가상화는 물리적인 네트워크를 하나 이상의 논리적 네트워크로 세분화하는 것을 의미하며, 네트워크 인프라에 대한 투자 대비 네트워크 인프라 자원 활용의 극대화를 목표로 한다. 해당 산업은 스마트폰, 노트북, 태블릿 등의 수요 증가, 클라우드 서비스 급증, 수십 GB 크기의 비디오 영상 제공 등 외부 환경 요인에 영향을 받고 있으며 기존 네트워크 구조의 문제점을 해결하며, 미래의 네트워크 환경(IoT, 5G)을 대비할 수 있는 산업으로 주목받고 있다. 하나의 물리적 네트워크 장비에서 다수의 가상네트워크 인터페이스 기능을 지원해주는 링크 가상화, 하나의 물리적인 라우터에서 자원을 분리하여 다수의 가상 라우터를 구성하는 라우터 가상화 등이 포함된다. 최근 모바일, 태블릿 등 사용량 및 클라우드 서비스의 증가, 데이터 전송량 증가에 따라 소프트웨어 정의 네트워크(Software Defined Network, SDN)와 네트워크 기술 가상화(Network Function Virtualization, NFV)가 네트워크 가상화의 핵심기술로 조명되고 있다.
- 등장배경:
  - ✓ 모빌리티 수요 증가
  - ✓ 클라우드 서비스 급증
  - ✓ 트래픽 패턴의 변화
  - ✓ 새로운 네트워크 아키텍처에 대한 수요

Source: 네트워크 가상화 - 해시넷 (hash.kr)

JS Lab

6

# DAY 1. 네트워크 가상화 개요

7

❖ VMware: 네트워크 가상화란?

- 네트워크 가상화(NV)는 기존에 하드웨어로 제공되던 네트워크 리소스를 소프트웨어로 추상화하는 것을 말합니다. NV는 여러 물리적 네트워크를 하나의 소프트웨어 기반 가상 네트워크로 결합하거나 하나의 물리적 네트워크를 별도의 독립적인 가상 네트워크로 나눌 수 있습니다.

❖ VMware: 네트워크 가상화가 필요한 이유

- NV는 기반 하드웨어에서 네트워크 서비스를 분리하고 네트워크 전체의 가상 프로비저닝을 지원합니다. 누구나 중앙 집중식 관리 시스템을 통해 스위치와 라우터 같은 물리적 네트워크 리소스를 풀링하고 이에 액세스할 수 있습니다. 또한 NV는 여러 관리 작업을 자동화하여 수작업에 따른 오류를 줄이고 프로비저닝 시간을 단축합니다. 이를 통해 네트워크 생산성과 효율성을 높일 수 있습니다.


Source: https://www.vmware.com/kr/topics/glossary/content/network-virtualization.html

JS Lab

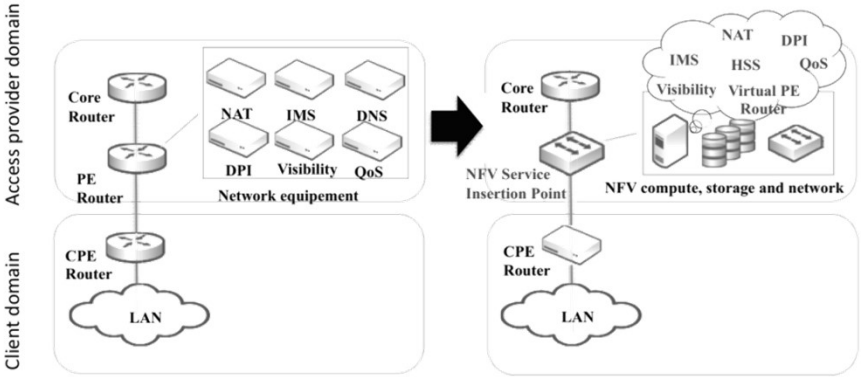
7

# DAY 4: 5G 네트워크 인프라 가상화 기술

8

❖ 코어 네트워크의 가상화

- Virtualization of a core network




Source: Software Networks (Virtualization, SDN, 5G and Security), by ISTE Press Ltd and John Wiley & Sons, Inc. 2020

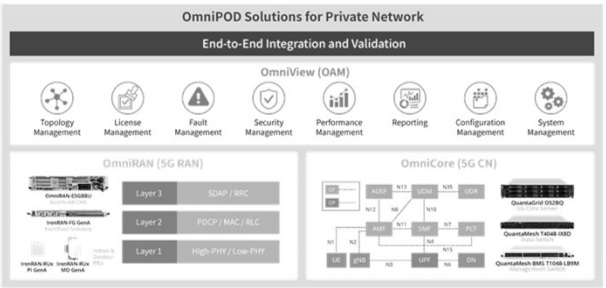
JS Lab

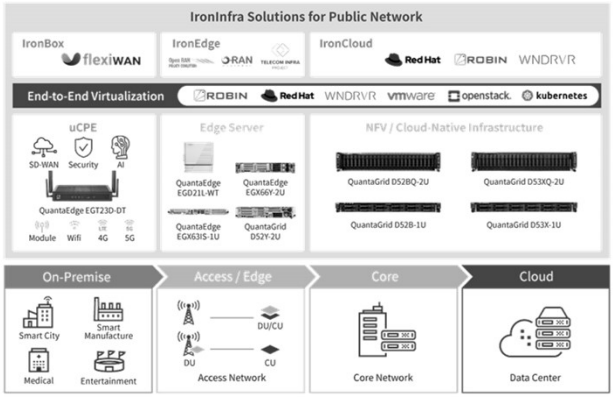
8

# DAY 4: 5G 네트워크 인프라 가상화 기술

9

❖ 제조사 (예): QCT




Source: https://go.qct.io/telco/

JS Lab

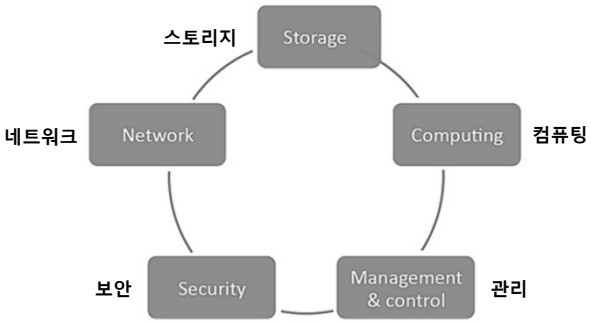
9

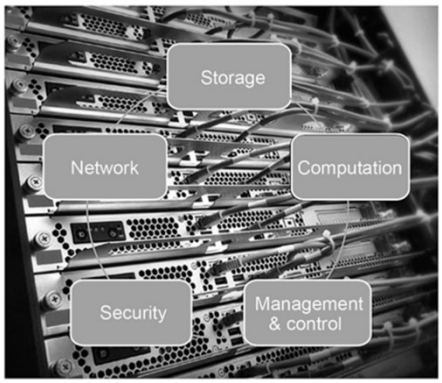
# DAY 4: 5G 네트워크 인프라 가상화 기술

10

❖ 가상화 기반 SDN의 확장

- The five domains necessary for the life of a company
- Virtualization of the five domains




Source: Software Networks (Virtualization, SDN, 5G and Security), by ISTE Press Ltd and John Wiley & Sons, Inc. 2020

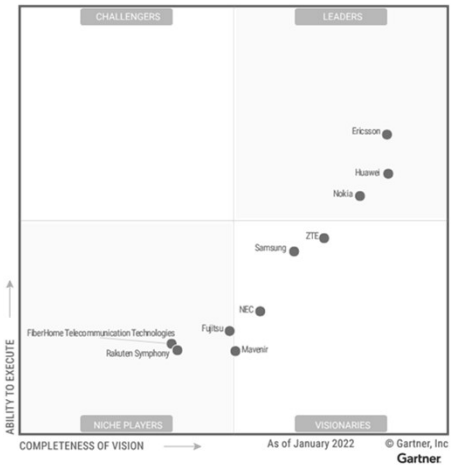
JS Lab

10

# DAY 4: 5G 네트워크 인프라 가상화 기술

11

- ❖ Gartner's Magic Quadrant for 5G Network Infrastructure (2022년 1월)
- ❖ Communications Service Providers
  - Radio access network equipment, including radio units (RUs), baseband units (BBUs) for 5G new radio and 4G LTE. Examples include the following:
    - Passive antennas, RU, AAU, virtualized BBU (vBBU), BBU, DU, CU, vDU, vCU and small cell
  - Core network equipment, including 5G next-generation core and evolved packet core (EPC):
    - UPF, AMF, SMF, PCF, AUSF, UDM, NSSF, NRF, NEF, NWDAF for 5G
    - MME, S-GW, P-GW, IMS, HSS, PCRF, EPC/virtualized EPC (vEPC) for 4G LTE
  - Transport network equipment: Fronthaul, midhaul, backhaul and wireless backhaul
  - Network infrastructure services: Design, build, run and support




Source: https://www.gartner.com/resources/746400/746462/Figure_1_Magic_Quadrant_for_5G_Network_Infrastructure_for_Communications_Service_Providers.png?reprintKey=1-299M6Q7A

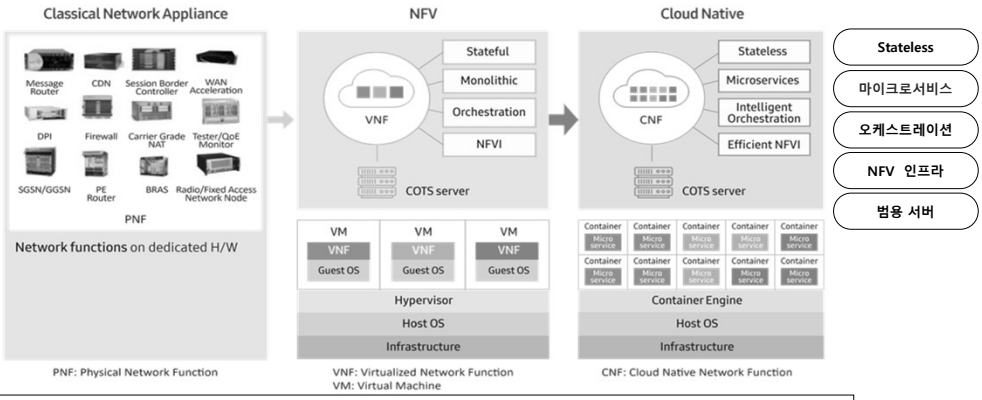
JS Lab

11

# DAY 4: 5G 네트워크 인프라 가상화 기술

12

- ❖ PNF(Physical Network Function) *하드웨어 어플라이언스 기반
- ❖ VNF(Virtualized Network Function) *가상머신 VM(Virtual Machine) 기반
- ❖ CNF(Cloud-Native Network Function) *클라우드 네이티브는 컨테이너/VM수용



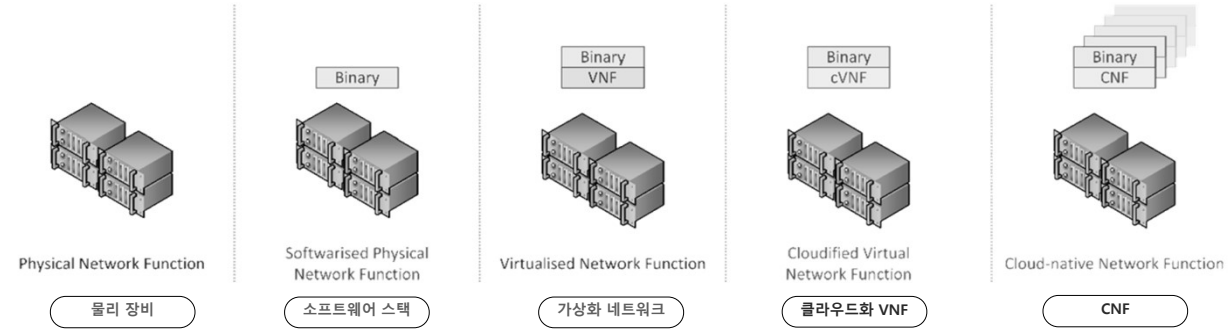

JS Lab

12

# DAY 4: 5G 네트워크 인프라 가상화 기술

13

❖ 5G 인프라 기술의 발전: Revised evolution of PNFs to CNFs and cVNFs




Source: 'Cloud Native Enabling Future Telco Platforms' by NGMN Alliance, Version 5.2, 17-May-2021

JS Lab

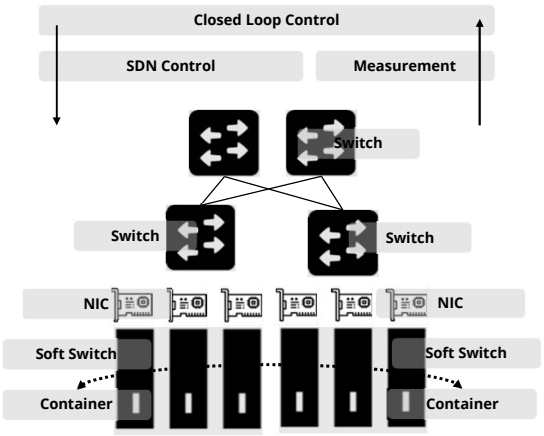
13

# DAY 4: 5G 네트워크 인프라 가상화 기술

14

❖ Needs of the New Datacenter

- Lines between servers and networking are blurring
- Developer optimized
- Needs to be cloud managed
- Need to consider traffic end-to-end
- Need to orchestrate all these components




JS Lab

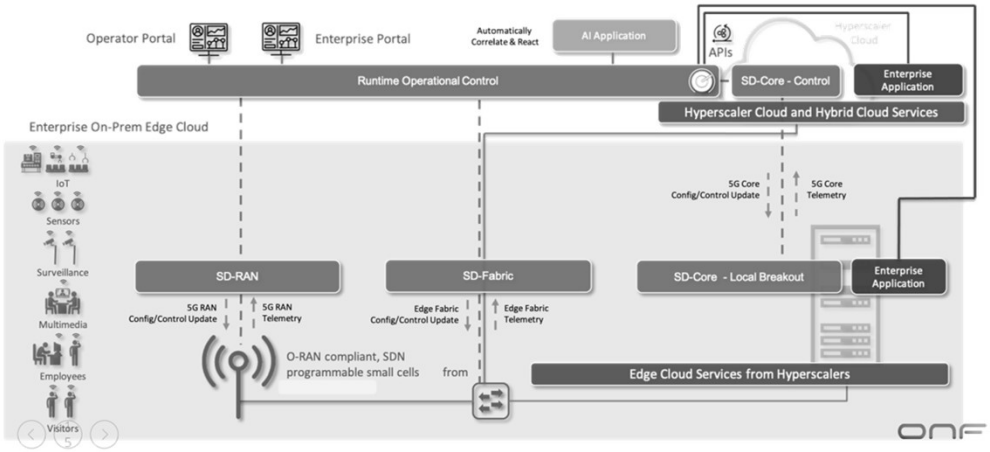
14

# DAY 4: 5G 네트워크 인프라 가상화 기술

15

❖ **Aether: 엔터프라이즈의 DT을 위한 5G 커넥티드 에지 플랫폼 오픈소스**

Aether




Source: https://opennetworking.org/wp-content/uploads/2020/12/Spotlight-OPs-1.pdf

JS Lab

15

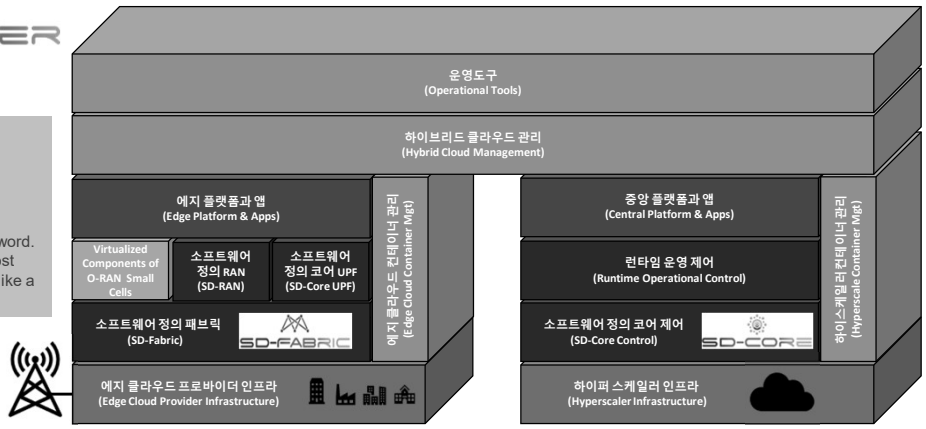
# DAY 4: 5G 네트워크 인프라 가상화 기술

16

❖ **SD-Core and SD-Fabric are an Integral Part of Aether**

**Setting Up Aether-in-a-Box**

- Ubuntu 18.04 clean install
- Kernel 4.15 or later
- Haswell CPU or newer
- At least 4 CPUs and 12GB RAM
- Ability to run "sudo" without a password. Due to this requirement, AiaB is most suited to disposable environments like a VM or a CloudLab machine.




Source: https://docs.aetherproject.org/master/developer/aiab.html

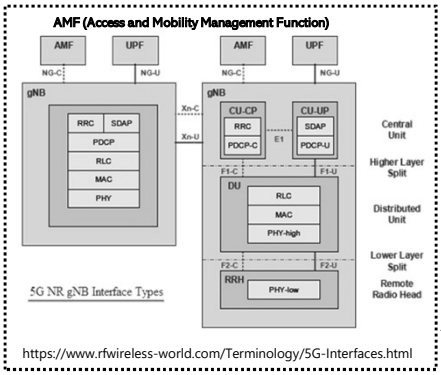
JS Lab

16

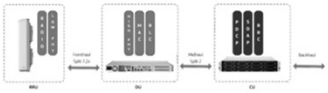
# DAY 4: 5G 네트워크 인프라 가상화 기술

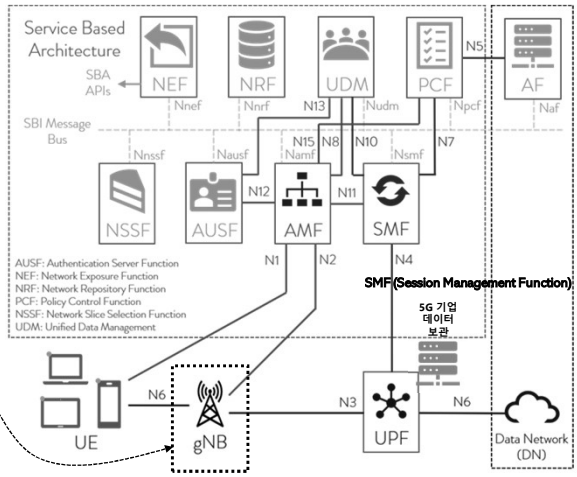
17

❖ 5G Service-Based Architecture




https://www.rfwireless-world.com/Terminology/5G-Interfaces.html






Source: https://www.metaswitch.com/knowledge-center/reference/what-is-the-5g-session-management-function-smf

JS Lab

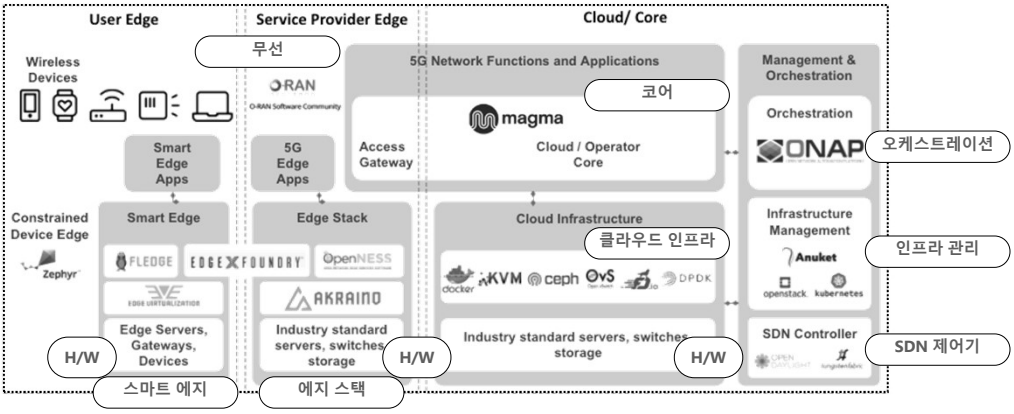
17

# DAY 4: 5G 네트워크 인프라 가상화 기술

18

❖ 리눅스재단(LF): ONAP's Cloud Native Journey (오케스트레이션)

## LF Open Source Component Projects for 5G




Source: https://www.lfnetworking.org/5g-super-blueprint/

JS Lab

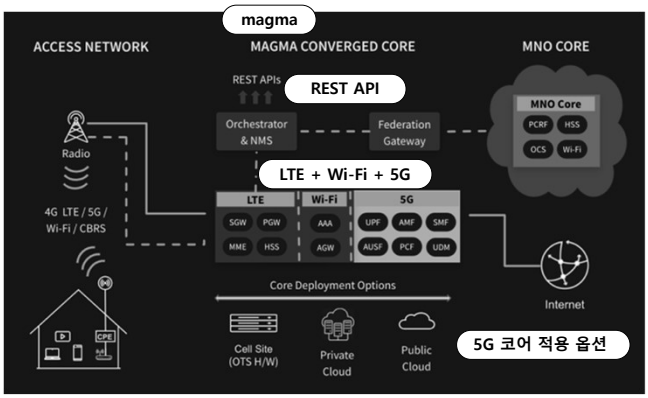
18

# DAY 4: 5G 네트워크 인프라 가상화 기술

19

❖ **5G 오픈소스 사용 제조사 (예): FreedomFI 사의 5G Core 오픈소스 magma 사용**

- **Access Gateway - AGW:** It is a distributed data plane component of the mobile network directly connected to an LTE or to a 5G radio. The Access Gateways are connected to a centralized orchestrator.


Source: https://levelup.gitconnected.com/opensource-5g-core-with-service-mesh-bba4ded044fa

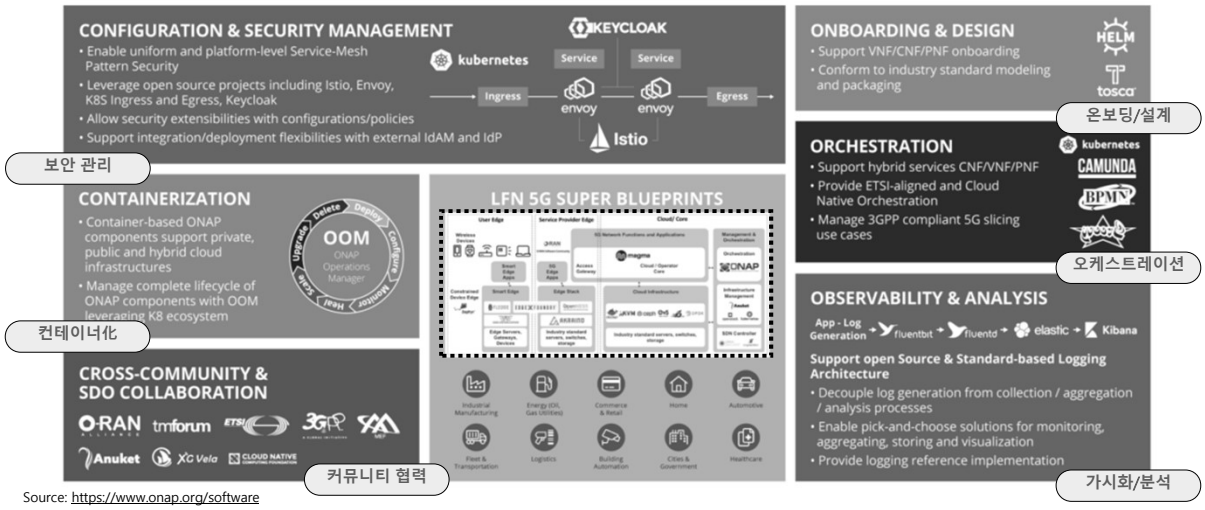
JS Lab

19

# DAY 4: 5G 네트워크 인프라 가상화 기술

20

❖ **리눅스재단(LF): LFN 5G Super Blueprints**




Source: https://www.onap.org/software

JS Lab

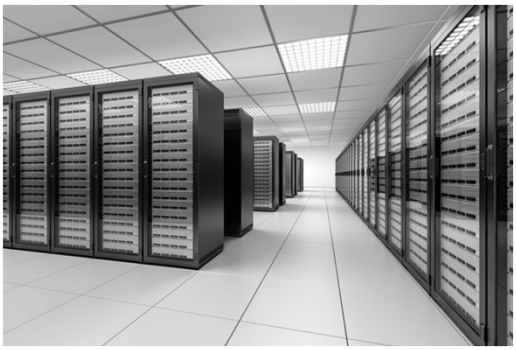
20

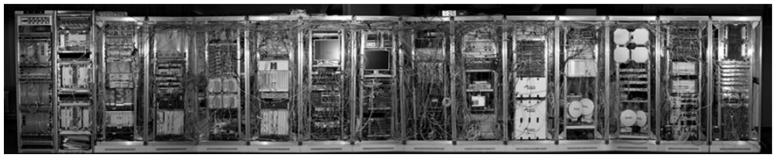
# DAY 4: 5G 네트워크 인프라 가상화 기술

21

❖ **SDN/NFV**

- **SDN:** 소프트웨어 정의 네트워킹(Software Defined Networking)이라고도 하며, 오픈 API를 통해 네트워크의 트래픽 전달 동작을 소프트웨어 기반 컨트롤러에서 제어 및 관리하는 접근방식
- **NFV**: Network Functions Virtualization, 네트워크기능 가상화





JS Lab

21

# DAY 4: 5G 네트워크 인프라 가상화 기술

22

❖ **News: Google, Linux Foundation Launch Nephio to Automate 5G**

- **네피오 프로젝트: The Nephio project is designed to provide Kubernetes-based cloud-native intent automation and automation templates to make it easier for telecom operators to deploy and manage multi-vendor cloud infrastructure and network functions across large-scale edge deployments. It sits on top of a Kubernetes substrate either directly controlled from an operator or a hyperscaler-based platform like Google Config Connector, Amazon Web Services (AWS) Controllers for Kubernetes, and Azure Service Operator**

Dan Meyer | Executive EditorApril 13, 2022 4:00 AM

Source: https://www.sdxcentral.com/articles/news/google-linux-foundation-launch-nephio-to-automate-5g/2022/04/

JS Lab

22

# DAY 4: 5G 네트워크 인프라 가상화 기술

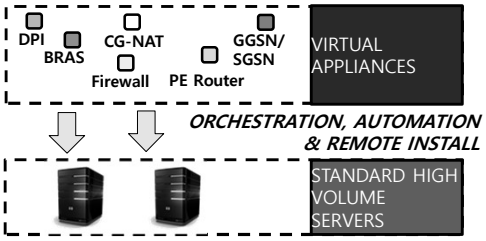
23

- ❖ 2012년 11월 ETSI는 ISG 'NFV' 설립
- ❖ NFV(Network Functions Virtualization): 하드웨어 의존성을 최소화 하는 네트워크 가상화로 독립적이며 유연하고 단순한 네트워크를 만드는 도구 제공

**Traditional Network Model: APPLIANCE APPROACH**


- Network Functions are **based on specific HW&SW**
- **One physical node per role**

**Virtualised Network Model: VIRTUAL APPLIANCE APPROACH**




- Network Functions are **SW-based over well-known HW**
- **Multiple roles** over **same HW**

European Telecommunications Standards Institute **(ETSI)**

Industry specification group **(ISG)**

**JS Lab**

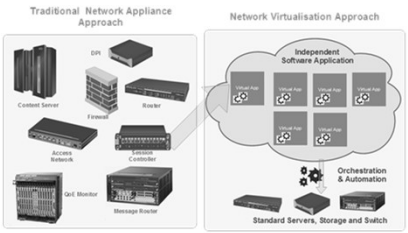
23

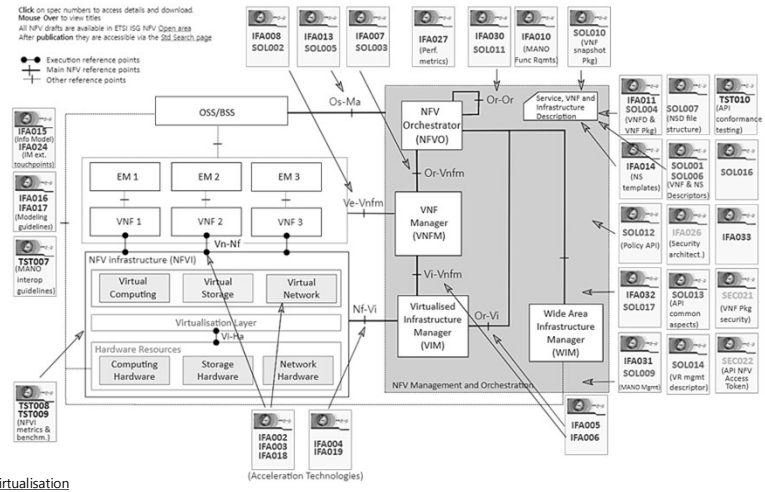
# DAY 4: 5G 네트워크 인프라 가상화 기술

24

- ❖ ETSI: NFV - Network Functions Virtualisation (ETSI – Standards), 2012년

- 2019-2020: NFV Release 4
- 2017-2018: NFV Release 3
- 2015-2016: NFV Release 2
- 2013-2014




Source: https://www.etsi.org/technologies/689-network-functions-virtualisation

**JS Lab**

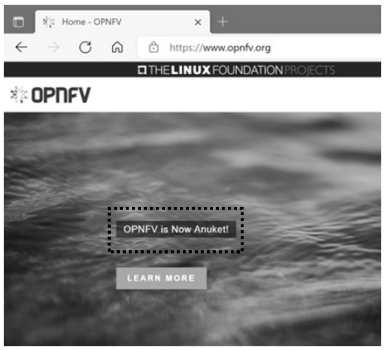
24

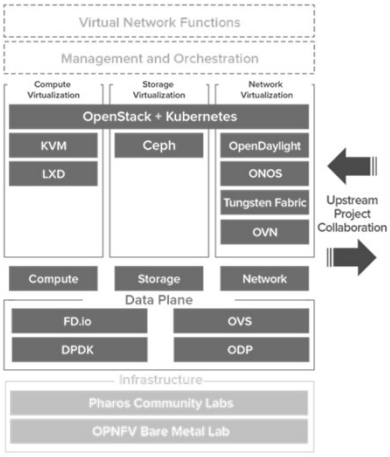
# DAY 4: 5G 네트워크 인프라 가상화 기술

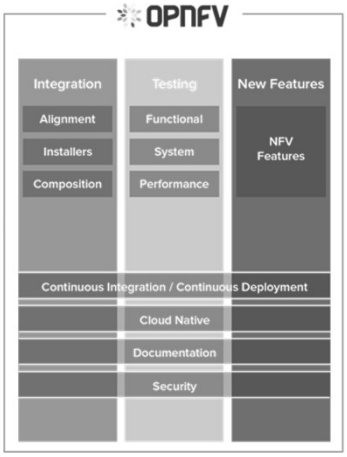
25

❖ **OPNFV: Open Platform for NFV (2014년~2021년)**

- **2021년 Anuket 변경**




Source: https://wiki.opnfv.org/

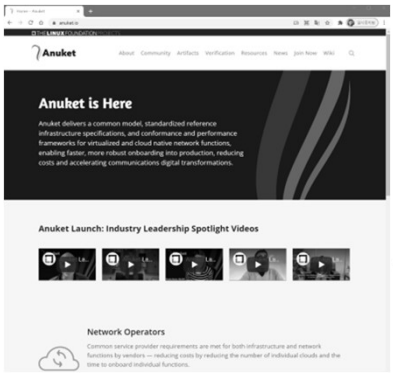
JS Lab

25

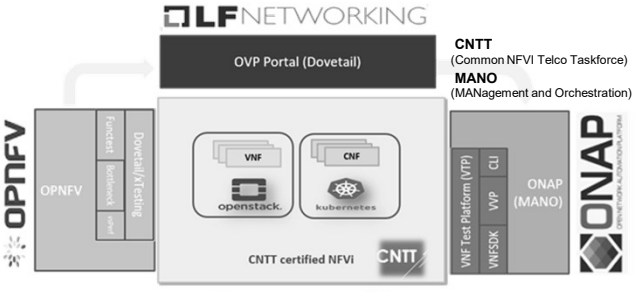
# DAY 4: 5G 네트워크 인프라 가상화 기술

26

❖ **Anuket (=OPNFV+CNTT)**




- Egyptian Goddess of the Nile
- Responsible for Annual Flooding of the Nile
- Associated with gazelles, arrows, and other swift-moving entities





Initially organized early in 2019, the Common Network Function Virtualization Infrastructure Telecom Taskforce (CNTT)

Source: https://wiki.onap.org/display/DW/OVP+Introduction

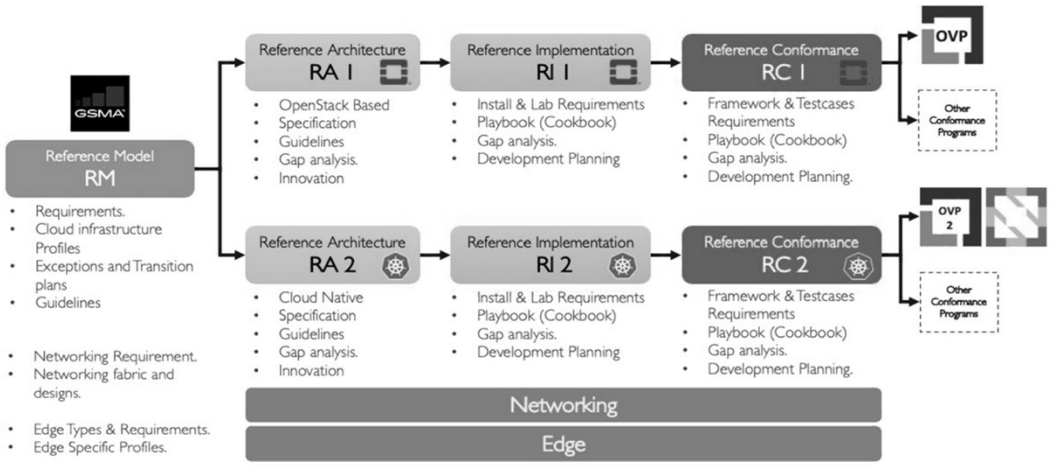
JS Lab

26

# DAY 4: 5G 네트워크 인프라 가상화 기술

27

❖ **CNTT Reference framework, Workstreams and Focus Groups**



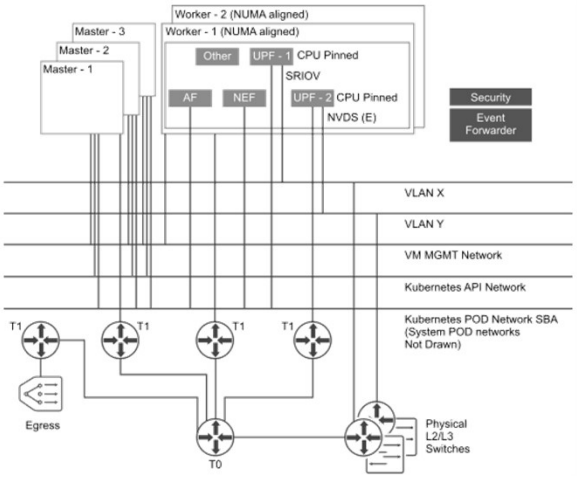

JS Lab

27

# DAY 4: 5G 네트워크 인프라 가상화 기술

28

❖ **5G Edge using vCloud NFV(Network functions virtualization)**




Source: https://docs.vmware.com/en/VMware-vCloud-NFV-OpenStack-Edition/3.3/vcloud-nfv-cloud-native-ra-33/GUID-6B8C1471-41A0-4436-B9C2-6A9CD74989D9.html?fbclid=IwAR2wJPdkyAyjwCADbH2OWQLingsh4LImzG1H3cCSOPNGBJeqT-erTBJC9Jg

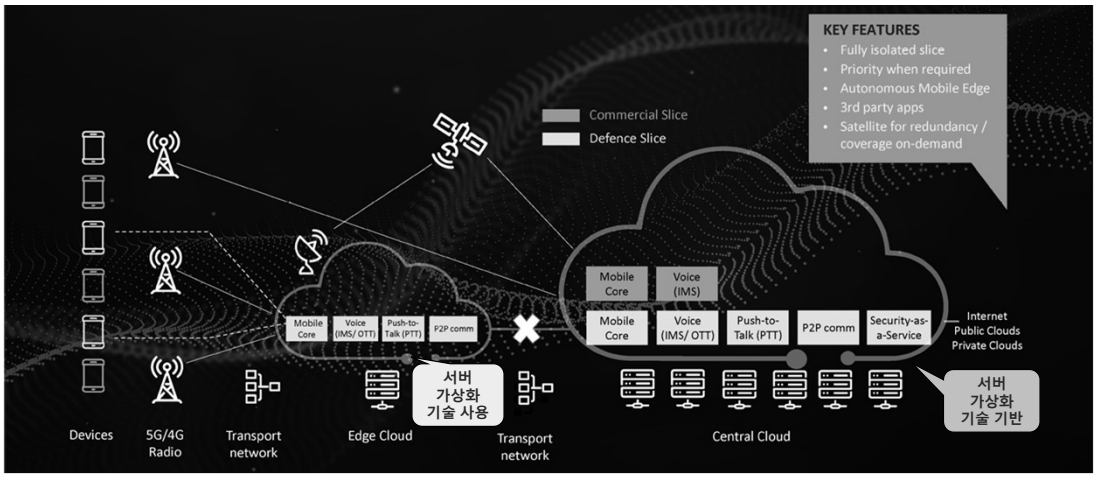
JS Lab

28

# DAY 4: 5G 네트워크 인프라 가상화 기술

29

❖ **Implementation of a Dedicated Network Slice** (with the Norwegian Defense)




Source: https://dataresponse.com/5g-is-a-game-changer-for-the-military/

JS Lab

29

# DAY 4: 5G 네트워크 인프라 가상화 기술

30

❖ **5G Enabling Technologies:**

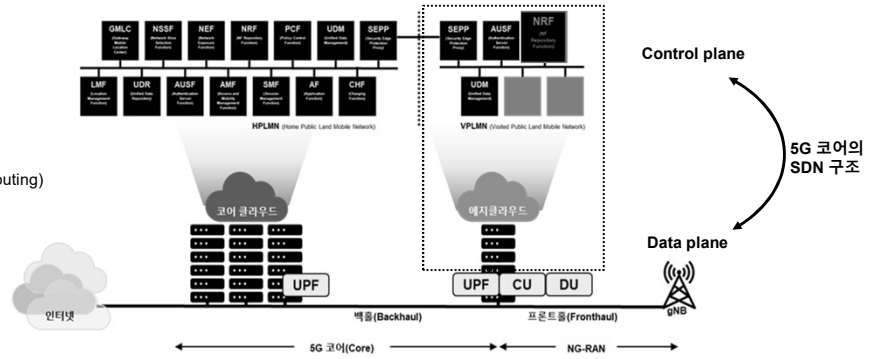
- **CUPS (제어/사용자 플레인 분리)**
- **Cloud Scale (클라우드 스케일)**
- **MEC (모바일 에지 컴퓨팅)**
- **Network Slicing (네트워크 슬라이싱)**

**RAN**(Radio Access Network)

**MEC**(Mobile | Multi-access Edge Computing)

**AF**(Application Function)

**UPF**(User Plane Function)




JS Lab

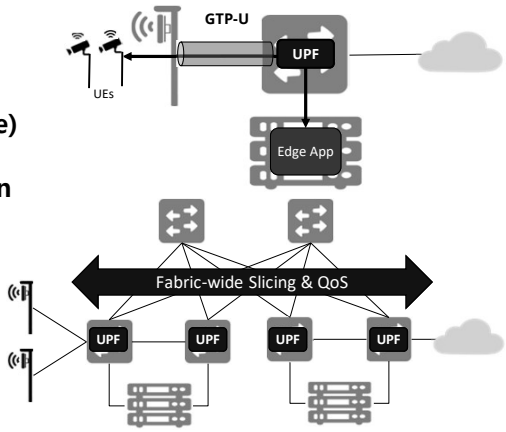
30

# DAY 4: 5G 네트워크 인프라 가상화 기술

31

❖ Embedded 5G UPF

- Free up CPU resources, run UPF at hardware speeds
  - Meet 5G requirements for ultra-low latency/jitter and high-throughput
- Tailored for enterprise and IoT use cases
  - GTP-U termination
  - Slicing & QoS
  - Usage reporting
  - Idle-mode buffering (cloud-native service)
- Distributed implementation
  - Any leaf can terminate any GTP-U session
  - Allows fabric-wide slicing and QoS
- INT visibility for SLA validation
  - Monitor flows inside GTP-U tunnels




JS Lab

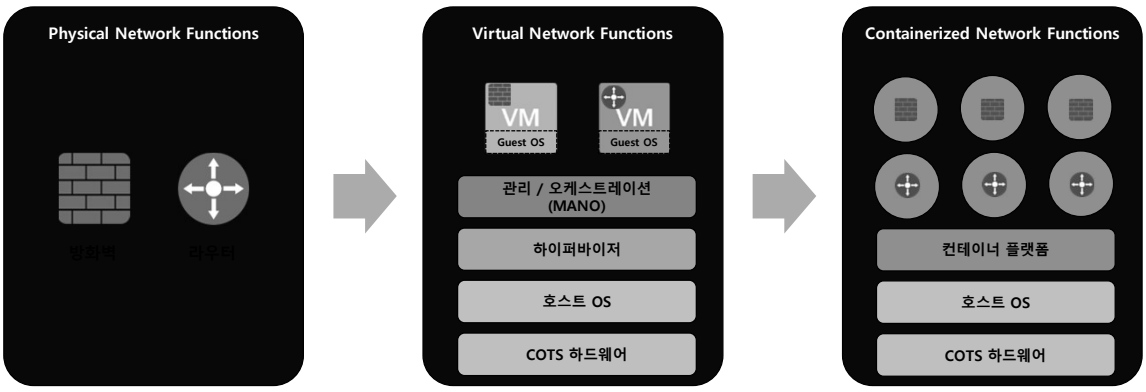
31

# DAY 4: 5G 네트워크 인프라 가상화 기술

32

❖ PNFs / VNFs / CNFs

- PNF → PNF+VNF → PNF+VNF+CNF




Physical Network Function (PNF), Virtual Network Function (VNF), Container Network Function (CNF)

JS Lab

32

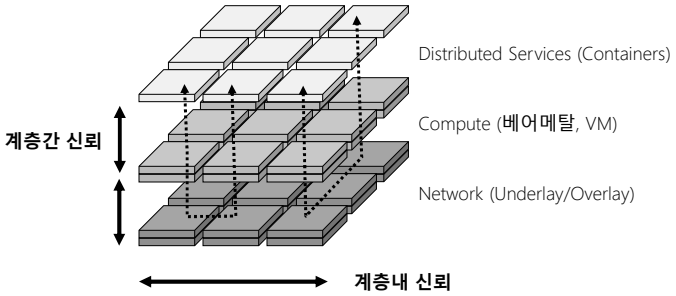
# DAY 4: 5G 네트워크 인프라 가상화 기술

33

❖ 클라우드 계층 고려 인프라 보안

❖ 클라우드 계층 기반 보안 체계

- 클라우드/SDDC/하드웨어 계층간 격리
- 외부 서비스의 노출 정책 지정 (Ingress, LB, DMZ)
- 계층내 신뢰 정책 강화




JS Lab

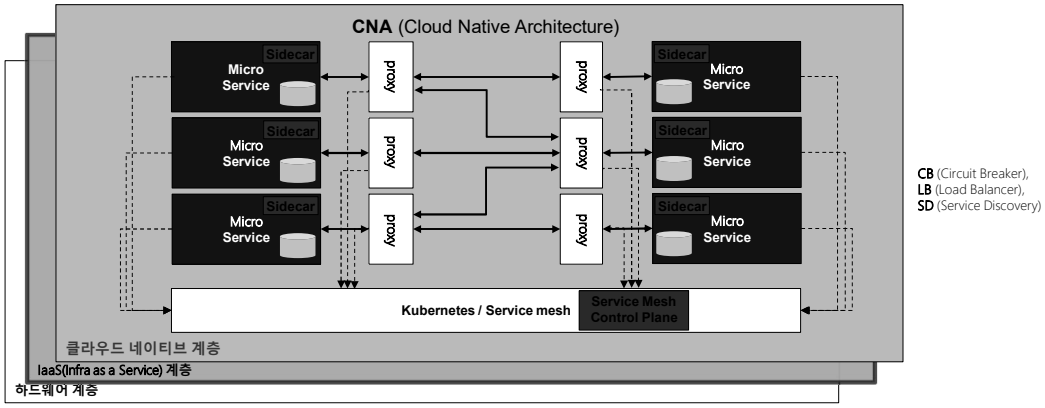
33

# DAY 4: 5G 네트워크 인프라 가상화 기술

34

❖ CNA 의 서비스 메시 관리

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



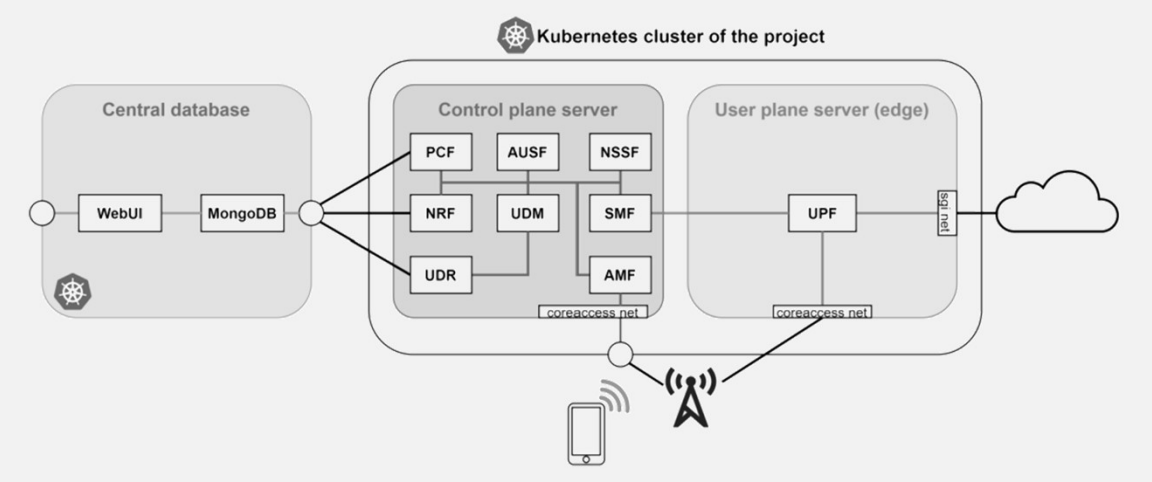

JS Lab

34

# DAY 4: 5G 네트워크 인프라 가상화 기술

35

❖ Containerized core network approach




Source: 5GPPP Architecture Working Group, 5G Architecture White Paper

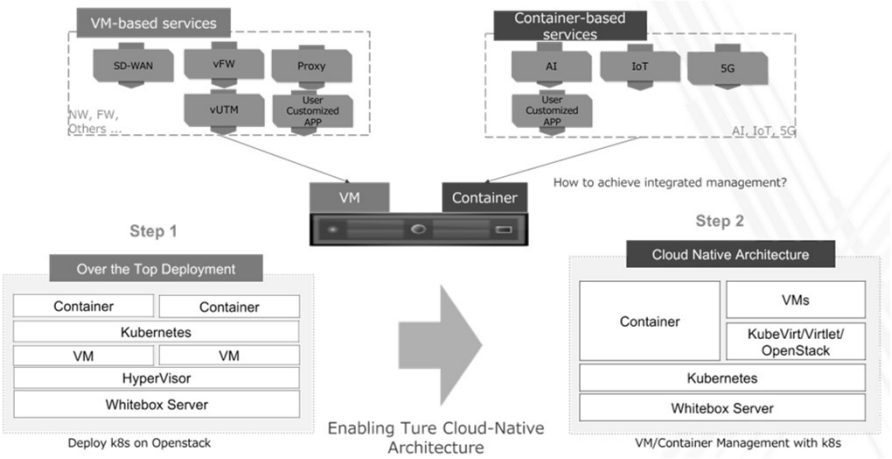
JS Lab

35

# DAY 4: 5G 네트워크 인프라 가상화 기술

36

❖ NFV (Network Functions Virtualization) and beyond




JS Lab

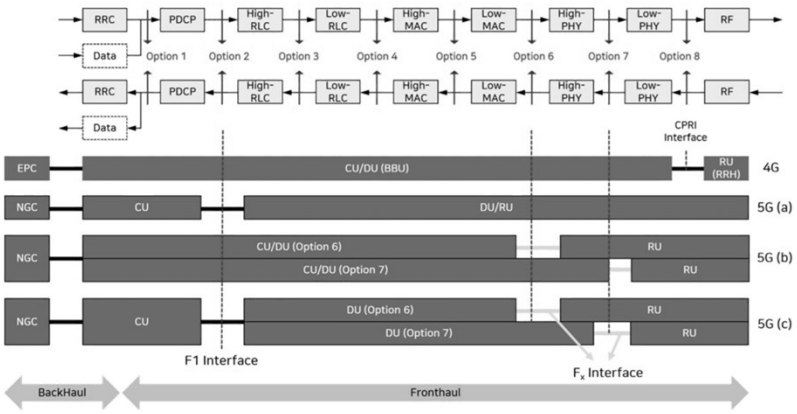
36

# DAY 4: 5G 네트워크 인프라 가상화 기술

37

❖ 5G 프로토콜과 기능적 분리 (TR 38.801)

- 지노드비, next generation Node B, gNodeB, gNB




Source: https://jb-story.tistory.com/346?msclkid=dbed5618b28311ec95bd530caef9a6a5

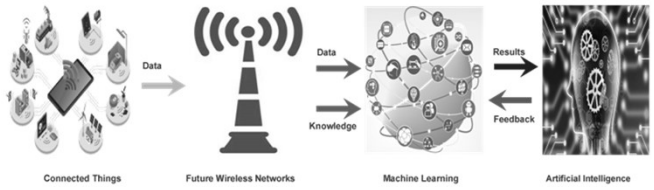
JS Lab

37

# DAY 4: 5G 네트워크 인프라 가상화 기술

38

❖ ML 진행(예): ML-based solution for 5G network.




| Author References | Key Contribution | ML Applied | Network Participants Component | | 5G Network Application Parameter | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | RAN | Core | LB | SDN | RAN | RA | SEC |
| Alave et al. [68] | Network traffic prediction | LSTM and DNN | ✓ | ✓ | 트래픽 예측 | | | ✓ | x |
| Bega et al. [15] | Network slice admission control algorithm | Machine Learning and Deep Learing | ✓ | ✓ | 슬라이스 제어 | | | ✓ | x |
| Suomalainen et al. [69] | 5G Security | Machine Learning | x | ✓ | 보안 | | | ✓ | ✓ |
| Bashir et al. [70] | Resource Allocation | Machine Learning | ✓ | ✓ | 자원 할당 | | | ✓ | x |
| Balevi et al. [71] | Low Latency communication | Unsupervised clustering | x | ✓ | 저지연 | | | ✓ | x |
| Tayyaba et al. [72] | Resource Management | LSTM, CNN, and DNN | ✓ | ✓ | DNN | | | ✓ | ✓ |
| Sim et al. [73] | 5G mmWave Vehicular communication | FML (Fast machine Learning) | x | ✓ | mmWave | | | ✓ | x |
| Li et al. [74] | Intrusion Detection System | Machine Learning | x | ✓ | 침입 탐지 | | | ✓ | ✓ |
| Kafle et al. [75] | 5G Network Slicing | Machine Learning | x | ✓ | 슬라이스 | | | ✓ | ✓ |
| Chen et al. [76] | Physical-Layer Channel Authentication | Machine Learning | x | ✓ | 물리 계층 | | | x | ✓ |
| Sevgican et al. [77] | Intelligent Network Data Analytics Function in 5G | Machine Learning | ✓ | x | 데이터 분석 | | | - | - |

Source: https://www.mdpi.com/1424-8220/22/1/26/htm

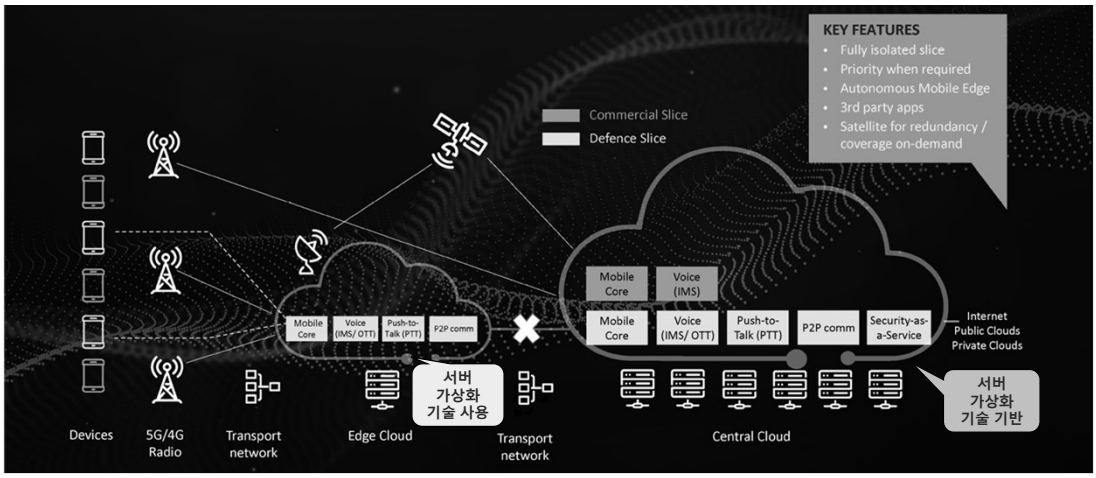
JS Lab

38

# DAY 4: 5G 네트워크 인프라 가상화 기술

39

❖ Implementation of a Dedicated Network Slice (with the Norwegian Defense)




Source: https://dataresponse.com/5g-is-a-game-changer-for-the-military/

JS Lab

39

# DAY 4: 5G 네트워크 인프라 가상화 기술

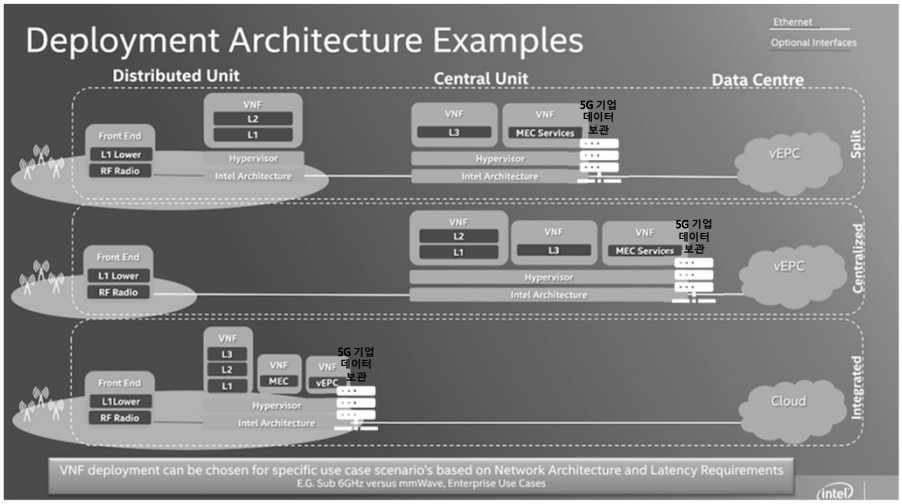
40

❖ 가상 RAN (vRAN): 인텔의 FlexRAN (예: 하드웨어와 독립적인 VNF 배포)

분할(Split)

중앙집중(Centralized)

통합(Integrated)



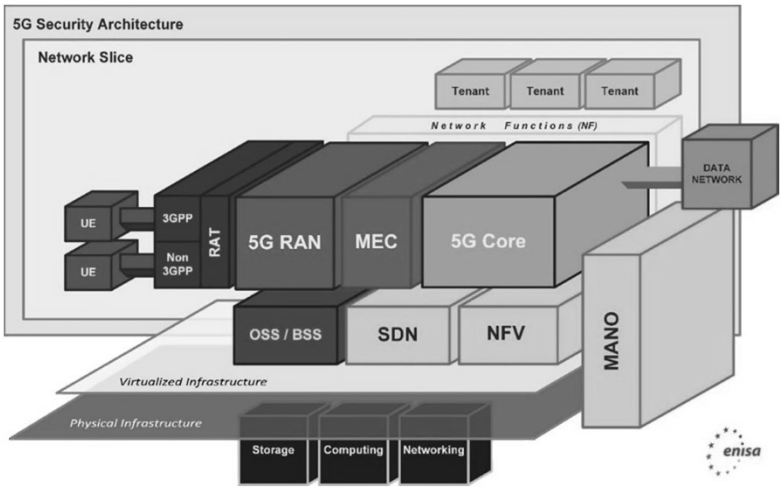

JS Lab

40

# DAY 4: 5G 네트워크 인프라 가상화 기술

41

❖ 네트워크 슬라이싱 (예): **ENISA**(European Union Agency for Cybersecurity)




Source: ENISA 5G Security Architecture

JS Lab

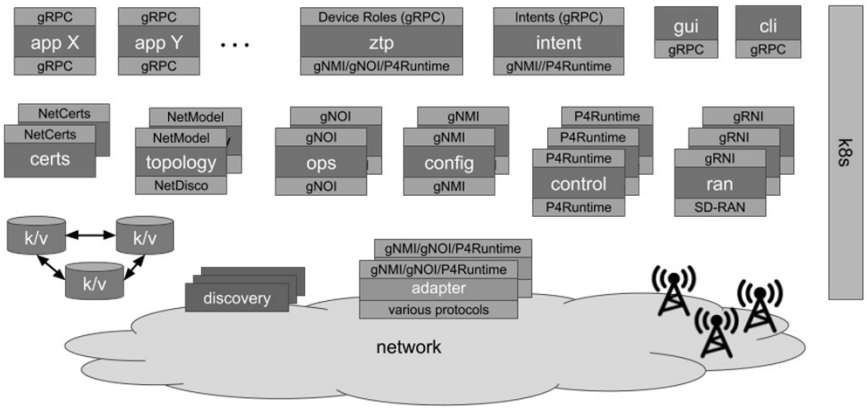
41

# DAY 4: 5G 네트워크 인프라 가상화 기술

42

❖ µONOS Deployment Architecture

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




Source: https://docs.onosproject.org/

JS Lab

42

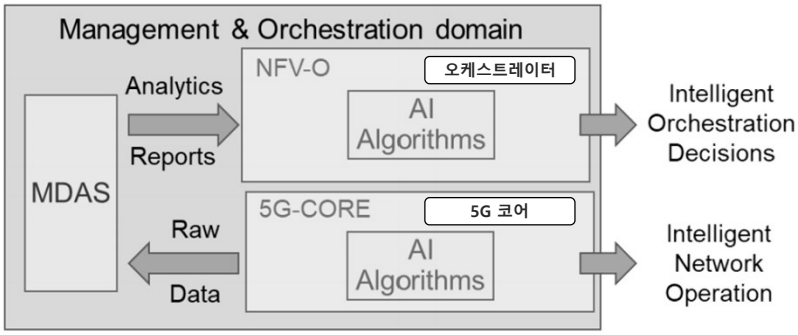
# DAY 4: 5G 네트워크 인프라 가상화 기술

43

❖ 가상화 관리의 AI 적용: **Management analytics function**

- **Management Data Analytics Service (MDAS)**

**The MDAS main functions**




Source: AI and ML – Enablers for Beyond 5G Networks' (URL http://doi.org/10.5281/zenodo.4299895) , 5G PPP Technology Board, 2021-05-11

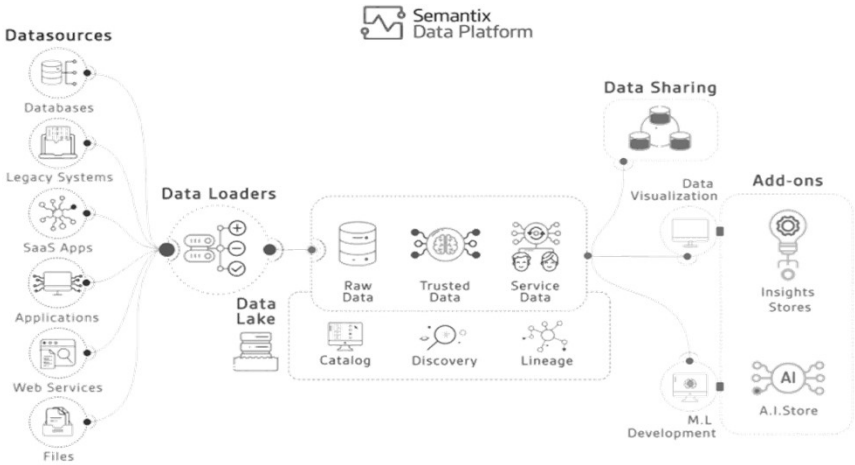
JS Lab

43

# DAY 4: 5G 네트워크 인프라 가상화 기술

44

❖ **Data movement in Data Lake**




Source: https://forum.huawei.com/enterprise/en/data-movement-in-data-lake-architecture/thread/763915-891

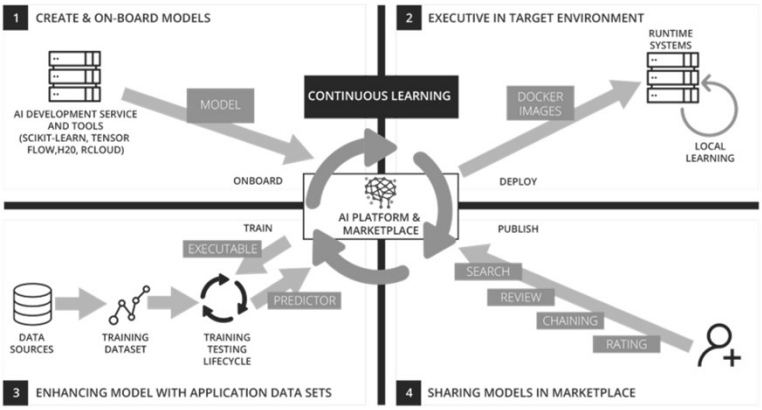
JS Lab

44

# DAY 4: 5G 네트워크 인프라 가상화 기술

45

❖ **Acumos: 오픈소스 AI/ML Platform** (By AT&T and The Linux Foundation)




Source: https://www.acumos.org/platform/

JS Lab

45

# DAY 4: 5G 네트워크 인프라 가상화 기술

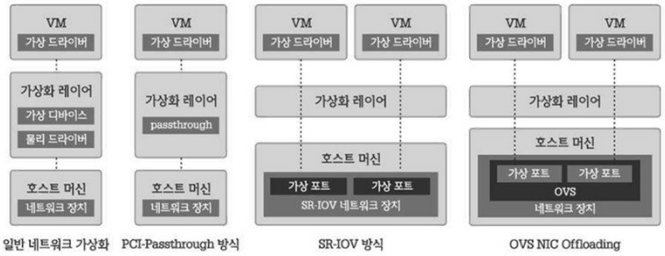
46

❖ **VNF (Virtual Network Function)의 마이크로 서비스화**

❖ **VNF의 MSA화를 통한 각 모듈의 간소화 및 각 모듈 간의 SBA에 기반한 통신**

❖ **VNF의 컨테이너화 / 마이크로 서비스화**

- **Kubernetes 도입에 따른 운영 환경 변화**
- **Kubernetes Networking: 물리 장비와 동일한 네트워크 성능을 보장 해야 하며, 이를 위해서 SR-IOV, OVS-DPDK와 같은 가속 기술 사용이 필요**




Source: SDN/NFV 기반 5G 통신망 인프라의 진화, 정보화진흥원, SKT 신상호 매니저

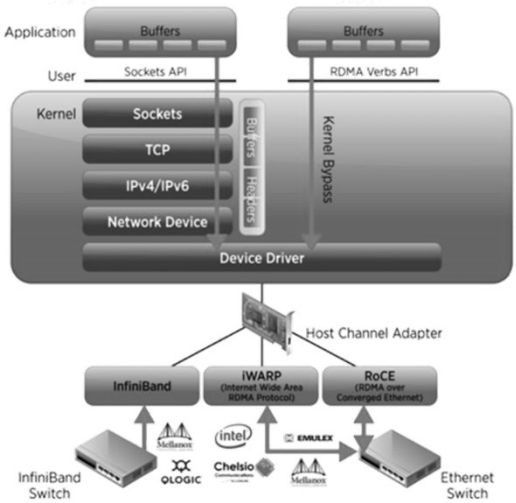
JS Lab

46

# DAY 4: 5G 네트워크 인프라 가상화 기술

47

❖ 가속을 위한 오프로드 하드웨어 구조





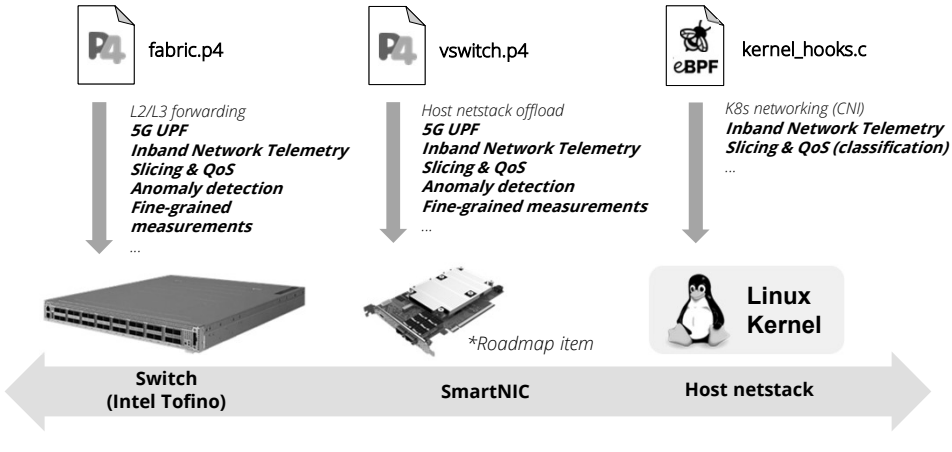

JS Lab



# DAY 4: 5G 네트워크 인프라 가상화 기술

48

❖ **End-to-End Programmable Data PlaneControl APIs**



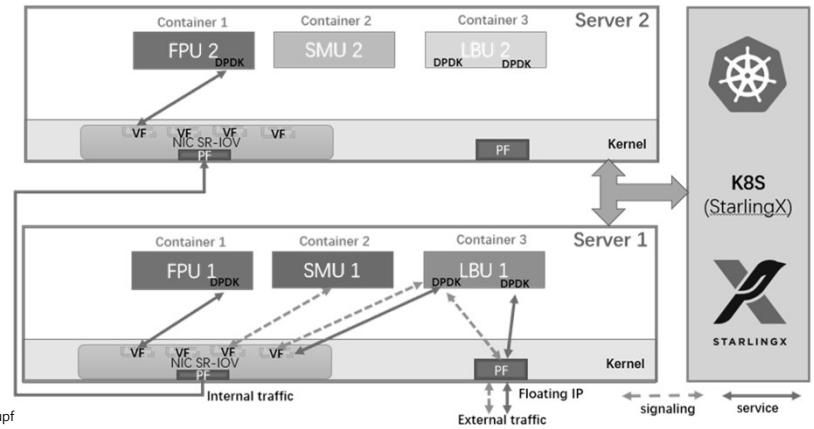

JS Lab

# DAY 4: 5G 네트워크 인프라 가상화 기술

49

❖ **OpenUPF Architecture**

- **three units: LBU (load balance unit), SMU (slow match unit) and FPU (fast pass unit).**




Source: https://bestofcpp.com/repo/5GOpenUPF-openupf



JS Lab

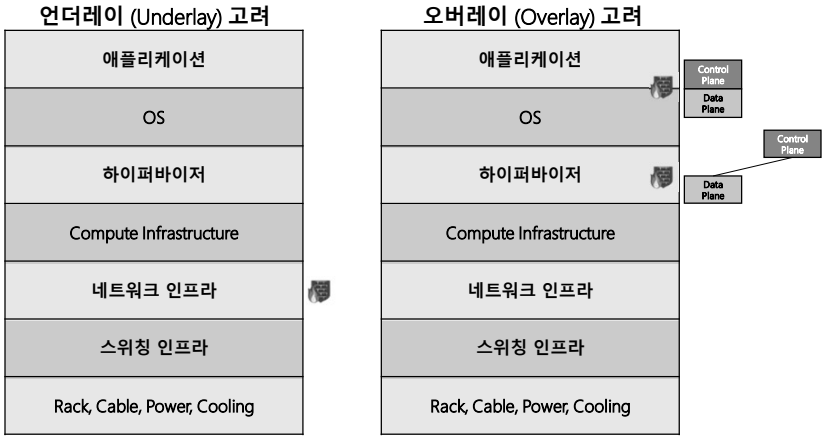


# DAY 4: 5G 네트워크 인프라 가상화 기술

50

❖ **오버레이 네트워크 인프라 보안**

- **가상기기 사용 보안: vFW, vIDS, vIPS (VM 제공)**
- **SDN 기반 분산 보안: dvFW, dvLB (w / dvRouter, dvSwitch)**



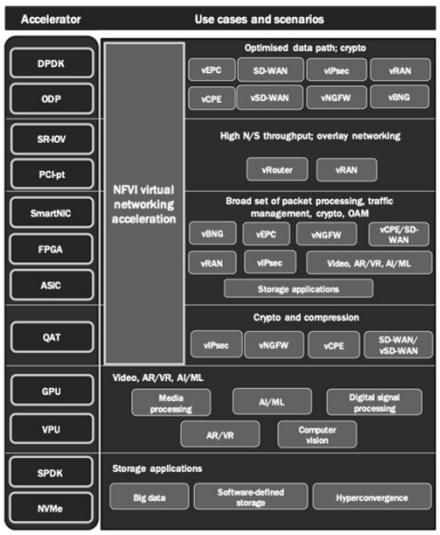

JS Lab

# DAY 4: 5G 네트워크 인프라 가상화 기술

51

❖ Acceleration technologies and use cases

- 가상화를 위한 가속 기술
- DPDK
- SR-IOV
- SmartNIC
- FPGA
- ASIC
- QAT
- GPU
- DPU
- IPU
- NVMe




Source: 5GPPP Architecture Working Group, 5G Architecture White Paper

JS Lab

51

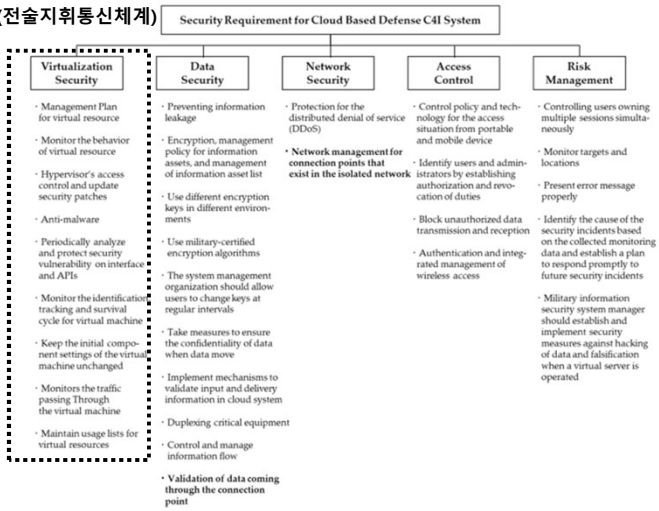
# DAY 4: 5G 네트워크 인프라 가상화 기술

52

❖ Security requirements for cloud-based defense C4I system. (가상화 보안)

1. 가상자원 관리
2. 가상자원 모니터
3. 하이퍼바이저 보안
4. Anti-malware
5. 주기적 보안 분석
6. VM 모니터
7. VM 세팅 유지
8. VM 트래픽 모니터
9. 가상 자원 사용 관리

C4I 체계(전술지휘통신체계)




Source: https://www.mdpi.com/2071-1050/13/24/13827/htm

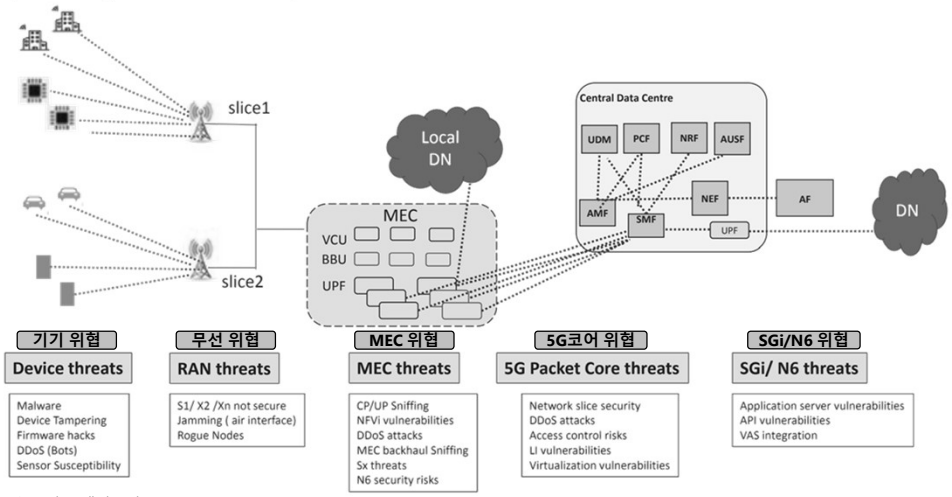
JS Lab

52

# DAY 4: 5G 네트워크 인프라 가상화 기술

53

❖ 5G 보안 위협 (Security threats in 5G)




Source: 5G Mobile Core Network, Rajaneesh Sudhakar Shetty, apres

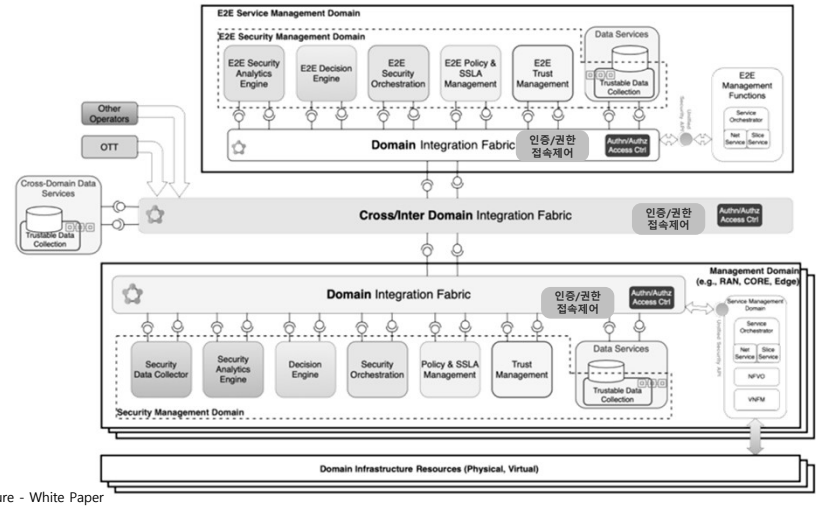
JS Lab

53

# DAY 4: 5G 네트워크 인프라 가상화 기술

54

❖ Security Framework High-Level Architecture




Source: 5GPPP Architecture Working Group 5G Architecture - White Paper

JS Lab

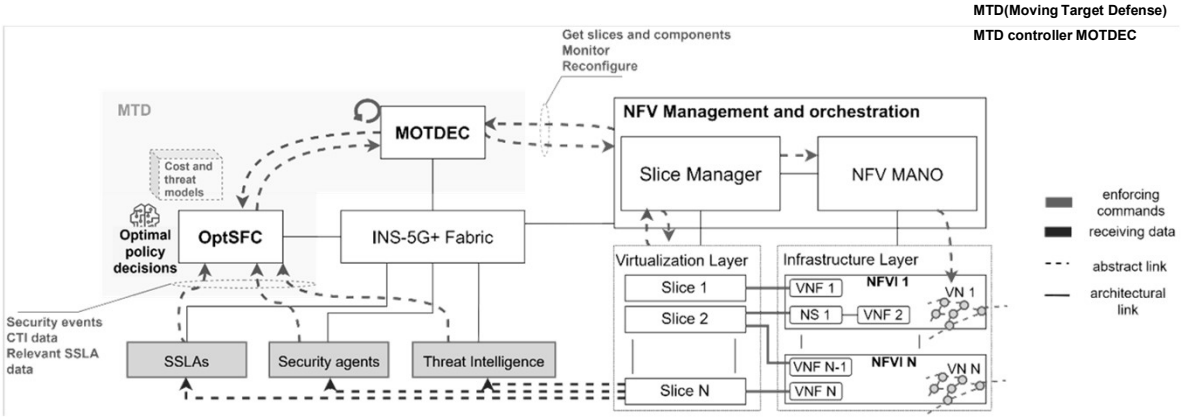
54

# DAY 4: 5G 네트워크 인프라 가상화 기술

55

❖ Security (보안)

- MTD for network slice protection




Source: AI and ML – Enablers for Beyond 5G Networks' (URL http://doi.org/10.5281/zenodo.4299895) , 5G PPP Technology Board, 2021-05-11

JS Lab

55

# DAY 4: 5G 네트워크 인프라 가상화 기술

56

❖ NSX Dashboard

- Home
- Networking
- Security
- Inventory
- Plan & Troubleshoot
- System

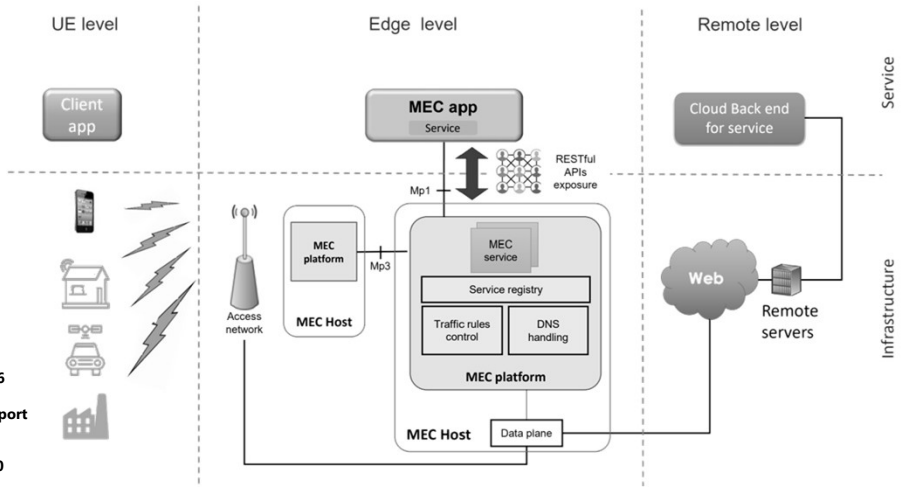

JS Lab

56

# DAY 4: 5G 네트워크 인프라 가상화 기술

57

❖ ETSI의 MEC 보안: MEC applications, interaction and service exposure




ETSI White Paper No. 46
MEC security:
Status of standards support
and future evolutions
1st edition – May 2021
ISBN No. 979109262040

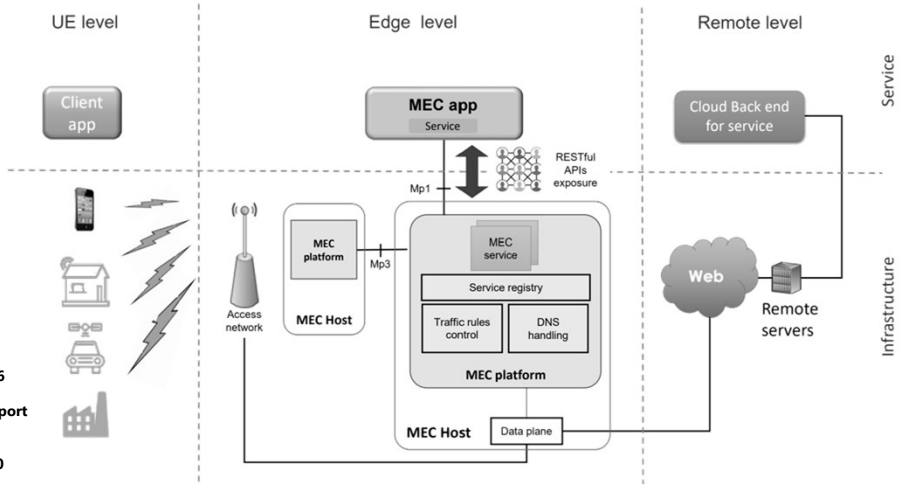
JS Lab

57

# DAY 4: 5G 네트워크 인프라 가상화 기술

58

❖ ETSI의 MEC 보안: MEC applications, interaction and service exposure




ETSI White Paper No. 46
MEC security:
Status of standards support
and future evolutions
1st edition – May 2021
ISBN No. 979109262040

JS Lab

58

# DAY 4: 5G 네트워크 인프라 가상화 기술

59

❖ 5G 네트워크 가상화 실습 (별도 교재)

## 실습 (별도교재)

JS Lab

59

60


JS Lab

60

# ❖ 부록. OpenStack

- 개요
- 기술

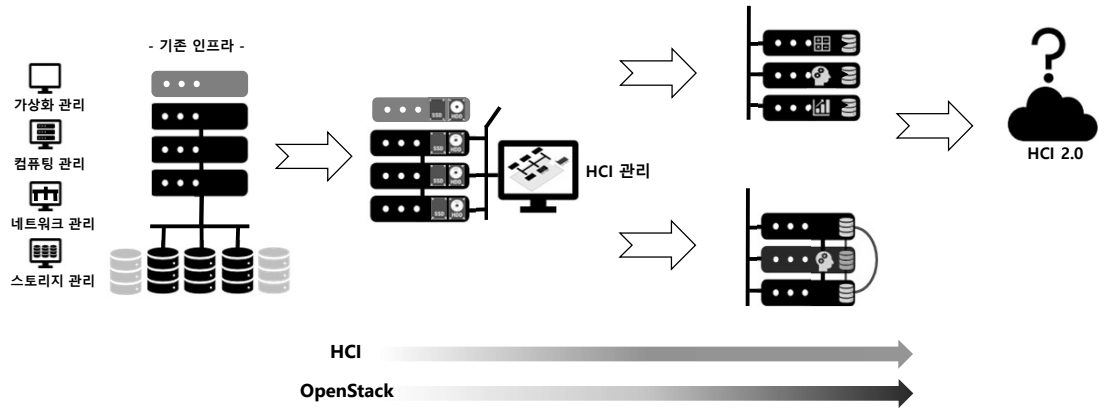


61



# 부록. OPENSTACK

❖ HCI의 발전과 OpenStack



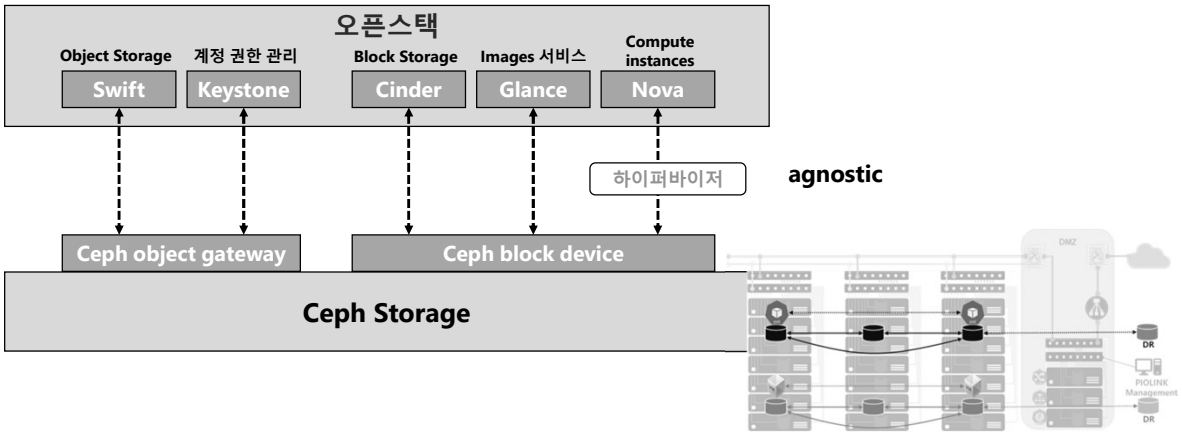



62

# 부록. OPENSTACK

63

❖ **Ceph Storage Architecture for OpenStack**




JS Lab

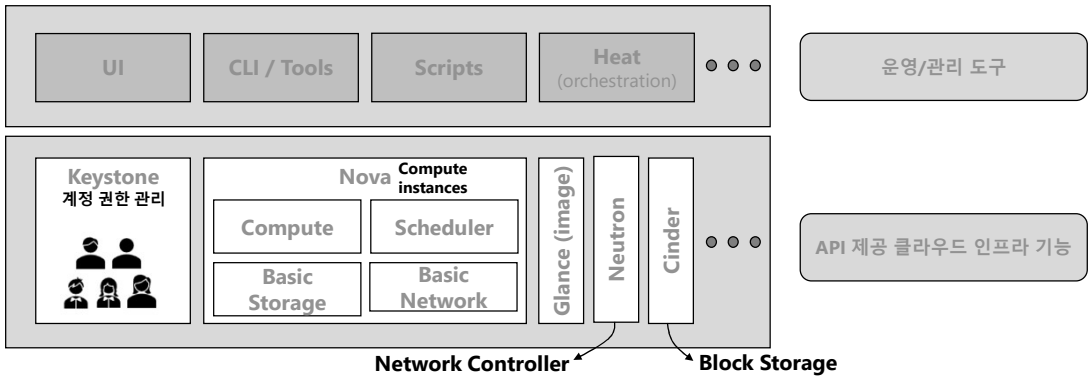
63

# 부록. OPENSTACK

64

❖ **OpenStack Architecture Overview**

OpenStack is fundamentally agnostic to the underlying infrastructure and integrates well with various compute, virtualization, network and storage technologies.




OpenStack architecture is loosely coupled, and extensible to support any hypervisor / container, storage, and network system

JS Lab

64

# 부록. OPENSTACK

65

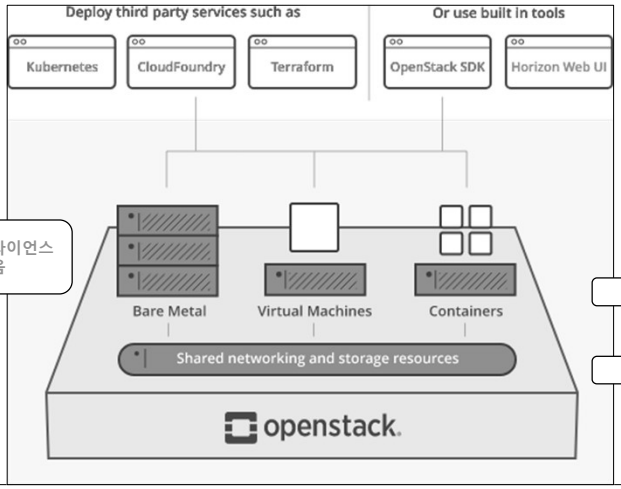
❖ 오픈스택: Cloud Infrastructure for Virtual Machines, Bare Metal, and Containers

서드파티 배포 서비스

(쿠버네티스, 클라우드파운드리, 테라폼)

Red Hat은 Ansible 사용

논리적인 구성이며 하드웨어 어플라이언스 구성은 제조사별로 차이 있음




내장 도구

(SDK, Horizon)

베어메탈, 가상머신, 컨테이너

공유 네트워킹과 스토리지 자원

openstack.

Open Infrastructure FOUNDATION

Source: https://www.openstack.org/

JS Lab

65

# 부록. OPENSTACK

66

❖ 최신 릴리즈: OpenStack Wallaby

OPENSTACK WALLABY

| Series | Status | Initial Release Date | Next Phase (project) | EOL Date |
|---|---|---|---|---|
| Xena | Development | 2021-10-06 *estimated* (schedule) | Development *estimated 2021-04-14* | |
| Wallaby | Maintained | 2021-04-14 | Extended Maintenance *estimated 2022-10-14* | |
| Victoria | Maintained | 2020-10-14 | Extended Maintenance *estimated 2022-04-18* | |
| Ussuri | Maintained | 2020-05-13 | Extended Maintenance *estimated 2021-11-12* | |
| Train | Extended Maintenance (see note below) | 2019-10-16 | Unmaintained *TBD* | |
| Stein | Extended Maintenance (see note below) | 2019-04-10 | Unmaintained *TBD* | |
| Rocky | Extended Maintenance (see note below) | 2018-08-30 | Unmaintained *TBD* | |
| Queens | Extended Maintenance (see note below) | 2018-02-28 | Unmaintained *TBD* | |
| Pike | Extended Maintenance (see note below) | 2017-08-30 | Unmaintained *TBD* | |
| Ocata | Extended Maintenance (see note below) | 2017-02-22 | Unmaintained *estimated 2020-06-04* | |
| Newton | End Of Life | 2016-10-06 | | 2017-10-25 |
| Mitaka | End Of Life | 2016-04-07 | | 2017-04-10 |
| Liberty | End Of Life | 2015-10-15 | | 2016-11-17 |
| Kilo | End Of Life | 2015-04-30 | | 2016-05-02 |
| Juno | End Of Life | 2014-10-16 | | 2015-12-07 |
| Icehouse | End Of Life | 2014-04-17 | | 2015-07-02 |
| Havana | End Of Life | 2013-10-17 | | 2014-09-30 |
| Grizzly | End Of Life | 2013-04-04 | (+ horizon, keystone, cinder, quantum) | 2014-03-29 |
| Folsom | End Of Life | 2012-09-27 | (+ horizon, keystone, cinder, quantum) | 2013-11-19 |
| Essex | End Of Life | 2012-04-05 | (+ horizon, keystone) | 2013-05-06 |
| Diablo | End Of Life | 2011-09-22 | (nova, swift, glance) | 2013-05-06 |
| Cactus | End Of Life | 2011-04-15 | (nova, swift, glance) | |
| Bexar | End Of Life | 2011-02-03 | (nova, swift, glance) | |
| Austin | End Of Life | 2010-10-21 | (nova, swift) | |

Source: https://www.openstack.org/

JS Lab

66

# 부록. OPENSTACK

67

❖ **Loosely coupled architecture**

- **Default ports**

서비스 배포는 API를 사용하는 컨테이너 또는
쿠버네티스의 POD로 구현 가능 구조

**Default ports that secondary services related to OpenStack components use**

| Service | Default port | Used by |
|---|---|---|
| HTTP | 80 | OpenStack dashboard (Horizon) when it is not configured to use secure access. |
| HTTP alternate | 8080 | OpenStack Object Storage (swift) service. |
| HTTPS | 443 | Any OpenStack service that is enabled for SSL, especially secure-access dashboard. |
| rsync | 873 | OpenStack Object Storage. Required. |
| iSCSI target | 3260 | OpenStack Block Storage. Required. |
| MySQL database service | 3306 | Most OpenStack components. |
| Message Broker (AMQP traffic) | 5672 | OpenStack Block Storage, Networking, Orchestration, and Compute. |

| OpenStack components use | Default ports |
|---|---|
| Application Catalog (murano) | 8082 |
| Backup Service (Freezer) | 9090 |
| Big Data Processing Framework (sahara) | 8386 |
| Block Storage (cinder) | 8776 |
| Clustering (senlin) | 8777 |
| Compute (nova) endpoints | 8774 |
| Compute ports for access to virtual machine consoles | 5900-5999 |
| Compute VNC proxy for browsers (openstack-nova-novncproxy) | 6080 |
| Compute VNC proxy for traditional VNC clients (openstack-nova-xvpvncproxy) | 6081 |
| Container Infrastructure Management (Magnum) | 9511 |
| Container Service (Zun) | 9517 |
| Data processing service (sahara) endpoint | 8386 |
| Database service (Trove) | 8779 |
| DNS service (Designate) | 9001 |
| High Availability Service (Masakari) | 15868 |
| Identity service (keystone) endpoint | 5000 |
| Image service (glance) API | 9292 |
| Key Manager service (Barbican) | 9311 |
| Loadbalancer service (Octavia) | 9876 |
| Networking (neutron) | 9696 |
| NFV Orchestration service (tacker) | 9890 |
| Object Storage (swift) | 6000, 6001, 6002 |
| Orchestration (heat) endpoint | 8004 |
| Orchestration AWS CloudFormation-compatible API (openstack-heat-api-cfn) | 8000 |
| Orchestration AWS CloudWatch-compatible API (openstack-heat-api-cloudwatch) | 8778 |
| Placement API (placement) | 8003 |
| Proxy port for HTML5 console used by Compute service | 6082 |
| Rating service (Cloudkitty) | 8889 |
| Registration service (Adjutant) | 5050 |
| Resource Reservation service (Blazar) | 1234 |
| Root Cause Analysis service (Vitrage) | 8999 |
| Shared File Systems service (Manila) | 8786 |
| Telemetry alarming service (Aodh) | 8042 |
| Telemetry event service (Panko) | 8977 |
| Workflow service (Mistral) | 8989 |

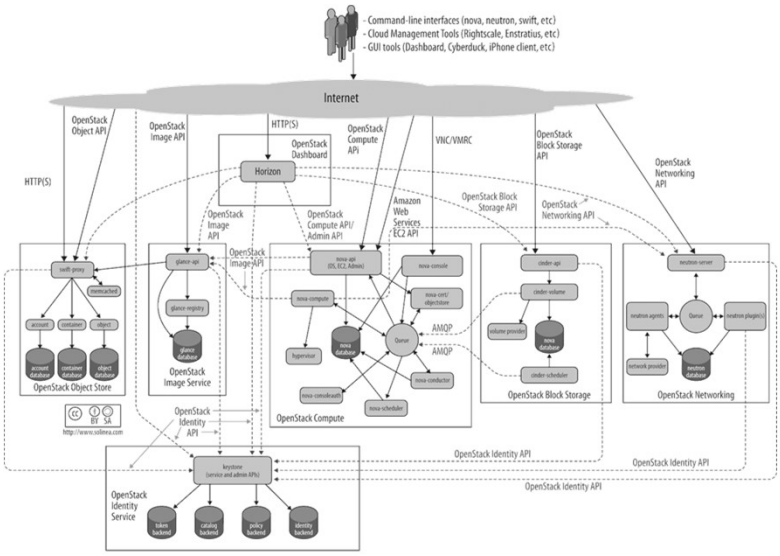
JS Lab

67

# 부록. OPENSTACK

68

❖ **OpenStack Design**




JS Lab

68

# 부록. OPENSTACK

69

## ❖ 오픈스택 구성 (OPENSTACK COMPONENTS)

- 운영도구 (Operations tooling: Those services deliver APIs primarily targeted to cloud admins and deployers, to help with cloud operations.)

**Compute**

**NOVA** Compute Service
**ZUN** Containers Service

**Hardware Lifecycle**

**IRONIC** Bare Metal Provisioning Service
**CYBORG** Lifecycle management of accelerators

**Storage**

**SWIFT** Object store
**CINDER** Block Storage
**MANILA** Shared filesystems

**Networking**

**NEUTRON** Networking
**OCTAVIA** Load balancer
**DESIGNATE** DNS service

Source: https://www.openstack.org/

**Shared Services**

**KEYSTONE** Identity service
**PLACEMENT** Placement service
**GLANCE** Image service
**BARBICAN** Key management

**Orchestration**

**HEAT** Orchestration
**SENLIN** Clustering service
**MISTRAL** Workflow service
**ZAQAR** Messaging Service
**BLAZAR** Resource reservation service
**AODH** Alarming Service

**Workload Provisioning**

**MAGNUM** Container Orchestration Engine Provisioning
**SAHARA** Big Data Processing Framework Provisioning
**TROVE** Database as a Service

**Application Lifecycle**

**MASAKARI** Instances High Availability Service
**MURANO** Application Catalog
**SOLUM** Software Development Lifecycle Automation
**FREEZER** Backup, Restore, and Disaster Recovery

**API Proxies**

**EC2API** EC2 API proxy

**Web Frontend**

**HORIZON** Dashboard

JS Lab

69

# 부록. OPENSTACK

70

## ❖ 오픈스택 구성 (OPENSTACK COMPONENTS)

- 오픈스택 서비스 (OpenStack Services: An OpenStack deployment contains a number of components providing APIs to access infrastructure resources. This page lists the various services that can be deployed to provide such resources to cloud end users.)

**Monitoring services**

**CEILOMETER** Metering & Data Collection Service
**PANKO** Event, Metadata Indexing Service
**MONASCA** Monitoring

**Billing / Business Logic**

**ADJUTANT** Operations processes automation
**CLOUDKITTY** Billing and chargebacks

**Resource optimization**

**WATCHER** Optimization Service
**VITRAGE** Root Cause Analysis service

**Testing / Benchmark**

**RALLY** Benchmarking tool
**TEMPEST** The OpenStack Integration Test Suite
**PATROLE** The OpenStack RBAC Integration Test Suite

**Application Lifecycle**

**MASAKARI** Instances High Availability Service
**MURANO** Application Catalog
**SOLUM** Software Development Lifecycle Automation
**FREEZER** Backup, Restore, and Disaster Recovery

- **서비스를 위한 애드온** **(Add-Ons to Services:** This software runs as an add-on or plug-in into other OpenStack services.**)**

**Swift add-ons**

**STORLETS** Computable object storage

- **통합** (**Integration enablers**: Software in this section facilitates integration of OpenStack components in adjacent open infrastructure stacks.)

**NFV**

**TACKER** NFV Orchestration

**Containers**

**KURYR** OpenStack Networking integration for containers

JS Lab

70

# 부록. OPENSTACK

71

❖ **클라이언트 도구(CLIENT TOOLS)**

- 클라이언트 도구 (Client tools: Client-side tools and libraries for interacting with OpenStack APIs.)

**SDKs**

**OPENSTACKSDK** Official Python SDK for OpenStack APIs

**CLIs**

**OPENSTACKCLIENT** Command-line interface for all OpenStack services

❖ **배포 도구 (DEPLOYMENT TOOLS)**

- 라이프사이클 관리 (Lifecycle management: Tools and packaging recipes to help install and maintain the lifecycle of OpenStack deployments.)

**Packaging recipes for popular frameworks**

**LOCI** Lightweight OCI containers
**PUPPET-OPENSTACK** Puppet modules to deploy OpenStack
**RPM-PACKAGING** RPM package specs to deploy OpenStack

**Frameworks for lifecycle management**

**TRIPLEO** Deploys OpenStack using OpenStack itself
**OPENSTACK-HELM** Deploys OpenStack in containers using Helm
**KOLLA-ANSIBLE** Deploys OpenStack in containers using Ansible
**KAYOBE** Deployment of containerised OpenStack to bare metal
**OPENSTACK-ANSIBLE** Ansible playbooks to deploy OpenStack
**OPENSTACK-CHARMS** Deploys OpenStack in containers using Charms and Juju
**BIFROST** Ansible playbooks using ironic
**OPENSTACK-CHEF** Chef cookbooks to build, operate and consume OpenStack

Source: https://www.openstack.org/

JS Lab

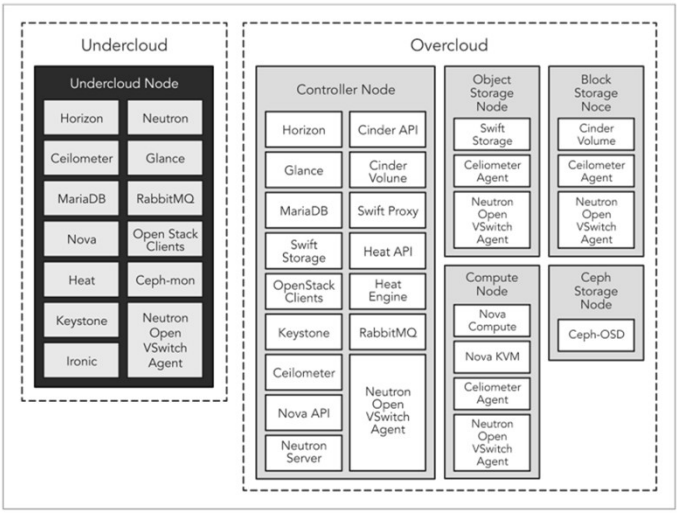
71

# 부록. OPENSTACK

72

❖ **Undercloud / Overcloud**

- 운영자 관점 Undercloud: an operator-facing deployment cloud
- 테넌트 관점 Overcloud: tenant-facing workload cloud
- 배포와 관리: Undercloud will contain the necessary OpenStack components to deploy and manage Overcloud




JS Lab

72

# 부록. OPENSTACK

73

❖ **Service Projects**

- 시장의 요구 수용
- 우선순위 개발 필요

| | Release Summary | | | |
|---|---|---|---|---|
| Deliverable | Earliest Version | Most Recent Version | Stable Status | Notes |
| adjutant | 2.0.0 | 2.0.0 | Maintained | |
| aodh | 12.0.0 | 12.0.0 | Maintained | release notes |
| barbican | 12.0.0 | 12.0.0 | Maintained | release notes |
| blazar | 7.0.0 | 7.0.0 | Maintained | release notes |
| ceilometer | 16.0.0 | 16.0.0 | Maintained | release notes |
| cinder | 18.0.0 | 18.0.0 | Maintained | release notes |
| cloudkitty | 14.0.0 | 14.0.0 | Maintained | release notes |
| cyborg | 6.0.0 | 6.0.0 | Maintained | release notes |
| designate | 12.0.0 | 12.0.0 | Maintained | release notes |
| ec2-api | 12.0.0 | 12.0.0 | Maintained | |
| freezer | 10.0.0 | 10.0.0 | Maintained | |
| glance | 22.0.0 | 22.0.0 | Maintained | release notes |
| heat | 16.0.0 | 16.0.0 | Maintained | release notes |
| horizon | 19.0.0 | 19.2.0 | Maintained | release notes |
| ironic | 16.1.0 | 17.0.3 | Maintained | release notes |
| keystone | 19.0.0 | 19.0.0 | Maintained | |
| magnum | 12.0.0 | 12.0.0 | Maintained | |
| manila | 12.0.0 | 12.0.0 | Maintained | release notes |
| masakari | 11.0.0 | 11.0.0 | Maintained | release notes |
| mistral | 12.0.0 | 12.0.0 | Maintained | release notes |
| monasca-api | 6.0.0 | 6.0.0 | Maintained | release notes |
| monasca-events-api | 3.0.0 | 3.0.0 | Maintained | |
| murano | 11.0.0 | 11.0.0 | Maintained | release notes |
| neutron | 18.0.0 | 18.1.0 | Maintained | release notes |
| nova | 23.0.0 | 23.0.2 | Maintained | release notes |
| octavia | 8.0.0 | 8.0.0 | Maintained | release notes |
| panko | 10.0.0 | 10.0.0 | Maintained | release notes |
| placement | 5.0.0 | 5.0.1 | Maintained | release notes |
| sahara | 14.0.0 | 14.0.0 | Maintained | release notes |
| senlin | 11.0.0 | 11.0.0 | Maintained | release notes |
| solum | 10.0.0 | 10.0.0 | Maintained | release notes |
| storlets | 7.0.0 | 7.0.0 | Maintained | release notes |
| swift | 2.27.0 | 2.27.0 | Maintained | release notes |
| tacker | 5.0.0 | 5.0.0 | Maintained | release notes |
| trove | 15.0.0 | 15.0.0 | Maintained | release notes |
| vitrage | 7.4.0 | 7.4.0 | Maintained | release notes |
| watcher | 6.0.0 | 6.0.0 | Maintained | |
| zaqar | 12.0.0 | 12.0.0 | Maintained | |
| zun | 7.0.0 | 7.0.0 | Maintained | release notes |

**Alarm (모니터링 Backend)**

**Metering**

**Block Storage**

**Images 서비스**

**OpenStack Orchestration (Autoscaling)**

OpenStack's Dashboard

**계정 권한 관리**

**Network**

**Provision compute instances (GPU 서비스)**

**Load balancer 가상화**

**Event (모니터링 Backend)**

**Object Storage**

| | |
|---|---|
| 기능 테스트 tool | Tempest |
| 성능측정 테스트 tool | Rally |

Source: https://releases.openstack.org/wallaby/index.html

JS Lab

73

# 부록. OPENSTACK

74

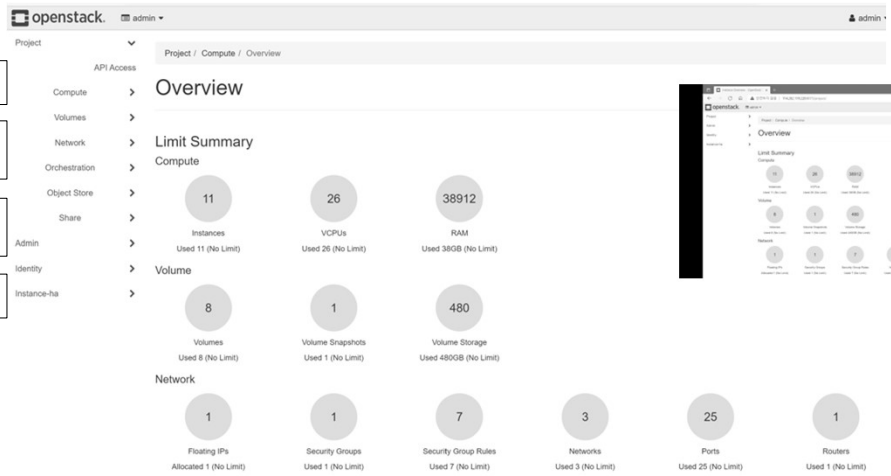
❖ **OpenStack Horizon (Web Frontend)**

**Compute**
**NOVA** Compute Service

**Networking**
**NEUTRON** Networking
**OCTAVIA** Load balancer

**Storage**
**SWIFT** Object store
**CINDER** Block Storage

**Orchestration**
**HEAT** Orchestration




JS Lab

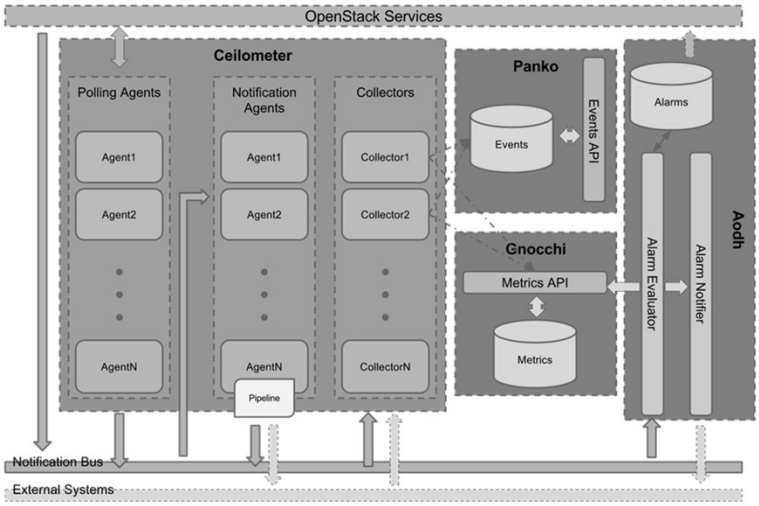
74

# 부록. OPENSTACK

75

❖ **운영 관리를 위한 필요 정보 제공: Ceilometer, Panko and Aodh**

- 시장의 요구 수용
- 우선순위 필요




JS Lab

75

# 부록. OPENSTACK

76

❖ **시장의 HCI 요구 기능**

- 서버 가상화
- 스토리지 가상화
- 네트워크 가상화
- 데이터 관리
- 가용성
- 확장성
- 개방성
- 사용자 포탈
- 관리자 포탈
- 백업/복구

| | |
|---|---|
| 서버 가상화 | VM 생성 및 관리 |
| | VM Thin Provisioning |
| | VM Export/Import |
| | VM 템플릿 |
| | VM 가용존 지정 |
| | VM MAC 주소 지정 |
| | VM 스케줄링 |
| | VM 메타데이터 조회 |
| 스토리지 가상화 | 블록 스토리지 생성 및 관리 |
| | 멀티 블록 스토리지 지원 |
| | 블록 스토리지 리사이징 |
| | 오브젝트 스토리지 생성 및 관리 |
| 네트워크 가상화 | 네트워크 분산 처리 |
| | 프라이빗 네트워크 생성 및 관리 |
| | 퍼블릭 네트워크 생성 및 관리 |
| | 가상 네트워크 영역별 테넌트 분리 |
| 데이터 관리 | 데이터 중복 제거 |
| | 데이터 압축 |
| | 데이터 리빌드 |
| | 데이터 재분배 |
| | 데이터 복제 |
| | 데이터 저장 방식 |
| 가용성 | 라이브 마이그레이션 |
| | 컨트롤러 삼중화 |
| 확장성 | 단일 서버 지원 |
| | 무중단 서버 확장 |
| | 다양한 타입의 서버 확장 |
| 개방성 | 오픈 아키텍처 지원 |
| | 멀티 하이퍼바이저 지원 |
| 사용자 포탈 | VM 관리 |
| | VM 볼륨 관리 |
| | VM 볼륨 스냅샷 관리 |
| | VM 템플릿 관리 |
| | 보안그룹 관리 |
| 관리자 포탈 | 통합 관리자 포탈 제공 |
| | 통합 모니터링 대시보드 제공 |
| | 하드웨어 모니터링 |
| | 오픈 API 제공 |
| | 가상, 물리 네트워크 관리를 위한 프로토 콜 제공(OVS sflow, 가상 Tap 등) |
| 백업/복구 | VM 백업 및 복구 |
| | 블록 스토리지 백업 및 복구 |

JS Lab

76

# 부록. OPENSTACK

77

❖ **시장의 HCI 확장 요구 기능**

- 스토리지 가상화
- 네트워크 가상화
- 부하분산
- 가용성
- GPU 가상화
- 컨테이너 관리
- 재해복구
- 백업/복구
- 기타 (하이브리드 클라우드 등)

| | |
|---|---|
| **스토리지 가상화** | 파일 스토리지 생성 및 관리 |
| | 파일 스토리지 리사이징 |
| **네트워크 가상화** | 언더레이 네트워크와 연동 가능 |
| **부하분산** | 다양한 프로토콜을 통한 부하 분산 |
| | 서버 상태 검사 |
| | 세션 유지 |
| | 다양한 부하 분산 알고리즘 |
| **가용성** | 서버 이베큐에이션 |
| **GPU 가상화** | 그래픽 가속 지원 |
| **컨테이너 관리** | 다중 사용자 지원 |
| | 클러스터 템플릿 관리 |
| | 컨테이너 클러스터 관리 |
| | 컨테이너 클러스터 사이즈 변경 |
| | 컨테이너 클러스터 모니터링 |
| | 다양한 컨테이너 엔진 지원 |
| **재해복구** | 분산 백업 |
| | 다양한 백업 방식 지원 |
| | 압축 백업 |
| | 암호화 백업 |
| | 스케줄링 자동 복구 |
| **백업/복구** | 파일 스토리지 백업 및 복구 |
| | 볼륨 백업 예약 |

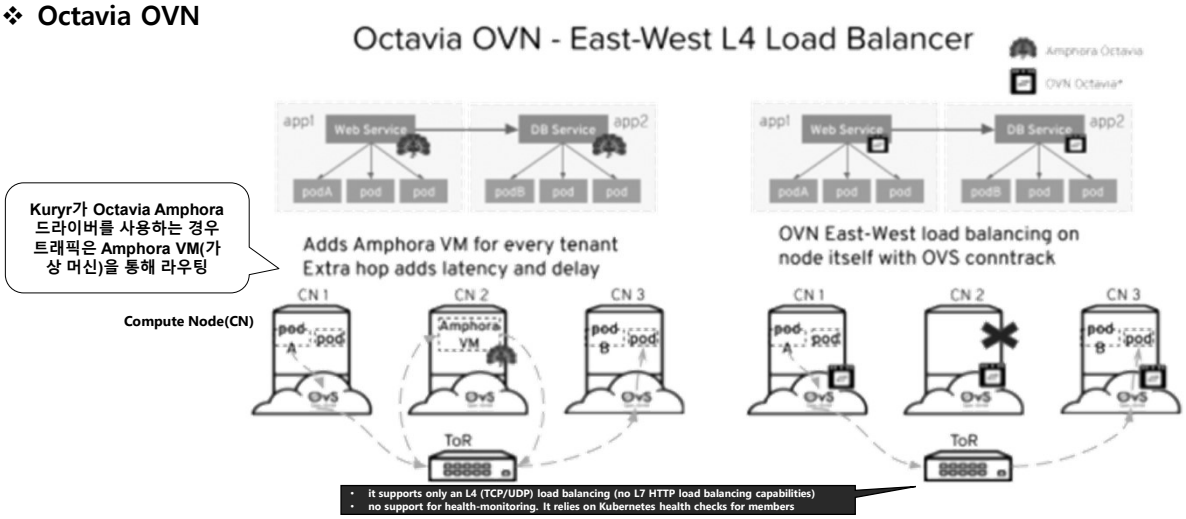
JS Lab

77

# 부록. OPENSTACK

78

❖ **Octavia OVN**




Source: https://superuser.openstack.org/articles/run-your-kubernetes-cluster-on-openstack-in-production/

JS Lab

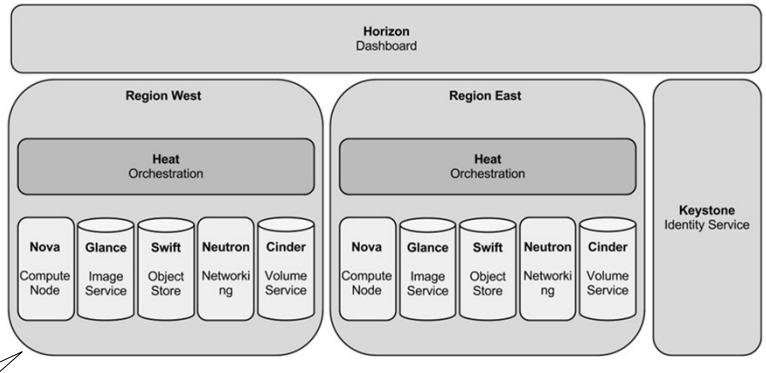
78

# 부록. OPENSTACK

79

❖ **Heat**

- Orchestration
- Heat/Blueprints/Multi Region Support for Heat
- A Heat template describes the infrastructure for a cloud application in text files which are readable and writable by humans, and can be managed by version control tools




OpenStack Native Control by Heat

Source: https://wiki.openstack.org/wiki/Heat/Blueprints/Multi_Region_Support_for_Heat

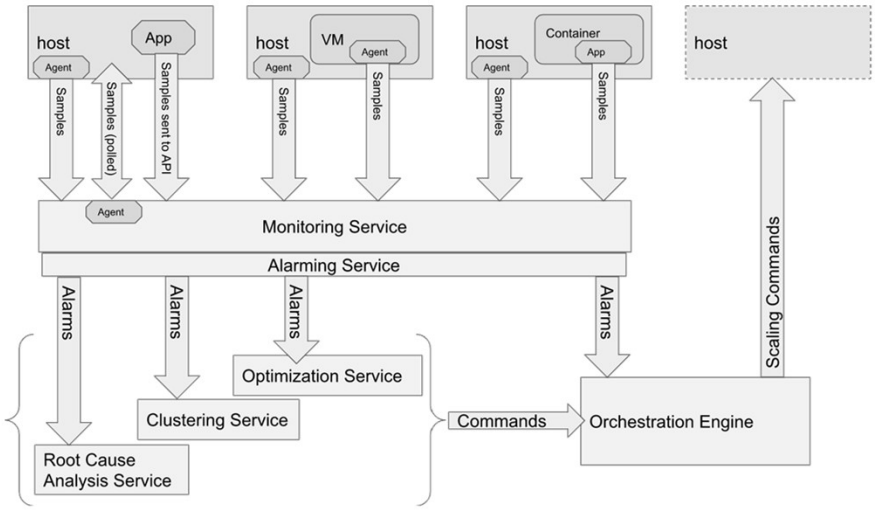
JS Lab

79

# 부록. OPENSTACK

80

❖ **Autoscaling**




Source: https://docs.openstack.org/auto-scaling-sig/latest/theory-of-auto-scaling.html

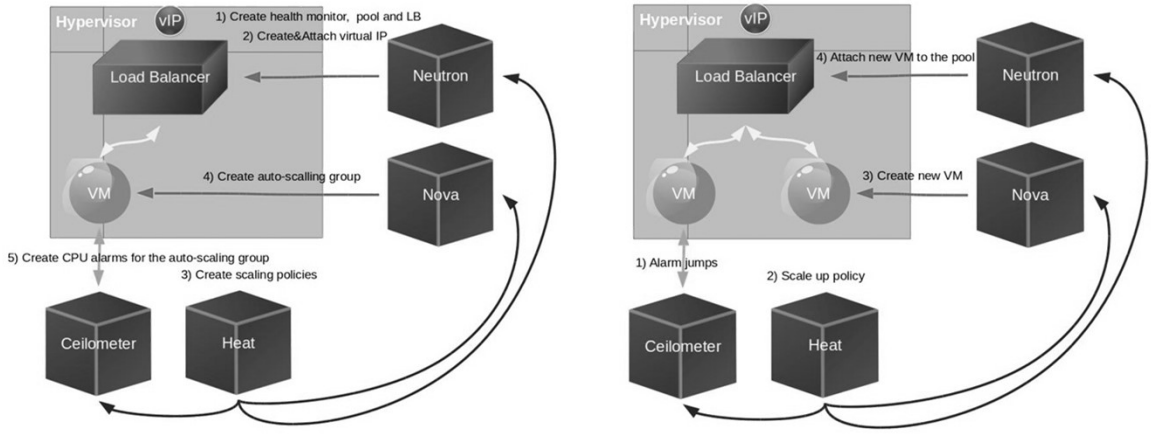
JS Lab

80

# 부록. OPENSTACK

81

❖ **Heat for autoscaling**




Source: https://ibm-blue-box-help.github.io/help-documentation/heat/autoscaling-with-heat/

JS Lab

81

# 부록. OPENSTACK

82

❖ **Distributed Routing**

**DVR** (Distributed Virtual Router)

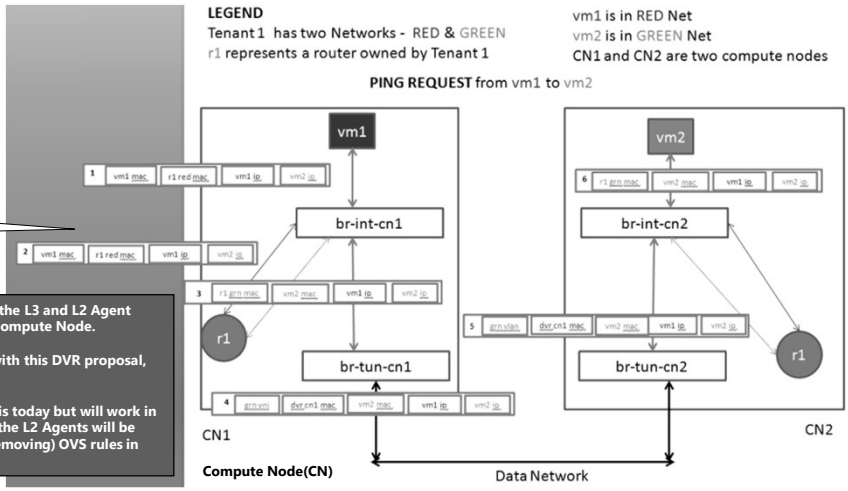
**라우팅 경로 단축**

**컴튜트 노드에 L3 에이전트 제공**

**In order to accomplish distributed routing, both the L3 and L2 Agent work will need to work hand-in-hand inside the Compute Node.**

**Today the L3 Agent runs in Network Node, but with this DVR proposal, the L3 Agent will run in Compute Nodes as well.**

**The L2 Agent on compute nodes will continue as is today but will work in an enhanced mode called the 'DVR Mode' where the L2 Agents will be additionally responsible for managing (adding/removing) OVS rules in order to accomplish distributed routing.**




Source: https://wiki.openstack.org/wiki/Neutron/DVR_L2_Agent

JS Lab

82

# 부록. OPENSTACK

83

❖ **OpenStack* EPA (Enhanced Platform Awareness)**

- 적용 기능별 성능 개선 분석(Feature Breakdown and Analysis)
- EPA Features Covered
  - ✓ Host CPU feature request (20~40%)
  - ✓ PCI passthrough (~8%)
  - ✓ HugePages* support (10~20%)
  - ✓ NUMA awareness (~10%)
  - ✓ IO based NUMA scheduling (~25%)
  - ✓ CPU pinning (10~20%)
  - ✓ CPU threading policies (~50%)
  - ✓ OVS-DPDK, neutron (~900%)

| Feature Name | First OpenStack* Release | Description | Benefit | Performance Data |
|---|---|---|---|---|
| **Host CPU feature request** | Icehouse* | Expose host CPU features to OpenStack managed guests | Guest can directly use CPU features instead of emulated CPU features | ~20% to ~40% improvement in guest computation |
| **PCI passthrough** | Havana* | Provide direct access to a physical or virtual PCI device | Avoid the latencies introduced by hypervisor and virtual switching layers | ~8% improvement in network throughput |
| **HugePages* support** | Kilo* | Use memory pages larger than the standard size | Fewer memory translations requiring fewer cycles | ~10% to ~20% improvement in memory access speed |
| **NUMA awareness** | Juno* | Ensures virtual CPUs (vCPU)s executing processes and the memory used by these processes are on the same NUMA node | Ensures all memory accesses are local to the node and thus do not consume the limited cross-node memory bandwidth, adding latency to memory | ~10% improvement in guest processing |
| **IO based NUMA scheduling** | Kilo* | Creates an affinity that associates a VM with the same NUMA nodes as the PCI device passed into the VM | Delivers optimal performance when assigning PCI device to a guest | ~25% improvement in network throughput for smaller packets |
| **CPU pinning** | Kilo | Supports the pinning of VMs to physical processors | Avoids scheduling mechanism moving the guest virtual CPUs to other host physical CPU cores, improving performance and determinism | ~10 % to ~20% improvement in guest processing |
| **CPU threading policies** | Mitaka* | Provides control over how guests can use the host hyper thread siblings | More fine-grained deployment of guests on HT-enabled systems | Up to ~50% improvement in guest processing |
| **OVS-DPDK, neutron** | Liberty* | An industry standard virtual switch accelerated by DPDK | Accelerated virtual switching | ~900% throughput improvement |

EPA 적용시 네트워크와 CPU 처리등의 10% 이상 성능 개선 기대

Intel 리딩

Source: https://software.intel.com/content/www/us/en/develop/articles/openstack-epa-feature-breakdown-and-analysis.html

JS Lab

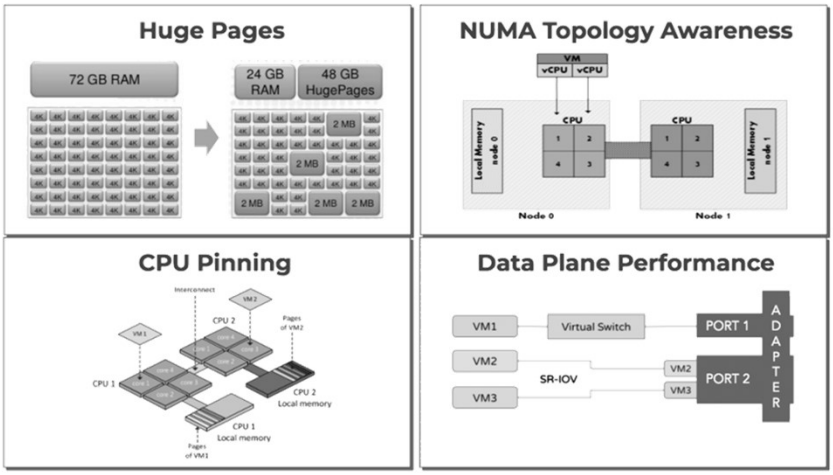
83

# 부록. OPENSTACK

84

❖ **OpenStack* EPA**

- Huge Pages
- NUMA awareness
- CPU pinning
- Data Plane




Source: https://superuser.openstack.org/articles/etsi-nfv-plugtests-vim/

JS Lab

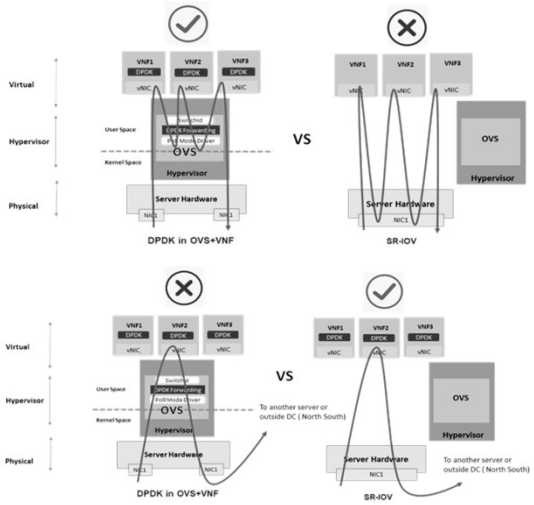
84

# 부록. OPENSTACK

85

❖ **OVS w/DPDK vs SR-IOV**

- OVS w/DPDK는 East-West 트래픽에 효과적
- SR-IOV는 North-South 트래픽에 효과적




Source: https://telcocloudbridge.com/blog/dpdk-vs-sr-iov-for-nfv-why-a-wrong-decision-can-impact-performance/

JS Lab

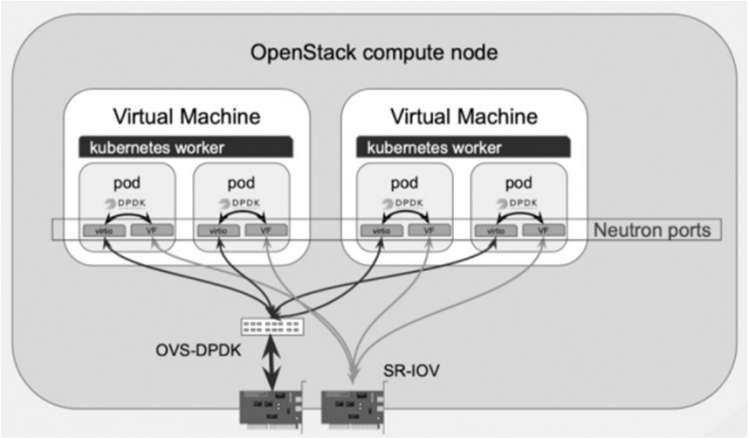
85

# 부록. OPENSTACK

86

❖ **OVS w/DPDK vs SR-IOV**

- Telco use cases – in particular 5G




Source: https://superuser.openstack.org/articles/run-your-kubernetes-cluster-on-openstack-in-production/ (March 16, 2021)

JS Lab

86

# 부록. OPENSTACK



❖ **Hadoop OpenStack Support: Swift Object Store**

- Apache Hadoop Release 3.3.1
- OpenStack offers infrastructure services such as VM hosting (Nova), authentication (Keystone) and storage of binary objects (Swift)
- This module enables Apache Hadoop applications -including MapReduce jobs, read and write data to and from instances of the OpenStack Swift object store

Source: https://hadoop.apache.org/docs/stable/hadoop-openstack/index.html