

# 클라우드 네이티브 네트워킹

2020. 11.

안 종 석  
james@jslab.kr  
JS Lab

## 목차

- 1장. 개요
- 2장. 가상화 (Virtualization)
- 3장. 클라우드 서비스 (Cloud Services)
- 4장. 컨테이너 (Containers)
- 5장. DevOps와 CI/CD
- 6장. 도구(Tools)
- 7장. 서비스 메시 (Service Mesh)
- 8장. 서버리스 (Serverless Computing)
- 9장. 관리 (Management)
- 10장. 보안 (Security)
- 11장. 디자인 패턴 (Design Pattern)
- 12장. Use Case
- 실습 자료 (별도)

JS Lab

James@jlab.kr

---

## 1장. 개요

- 개요
- 레퍼런스
- Cloud Native

JS Lab

James@jlab.kr

---

## 1장. 개요

□ 클라우드 네이티브 (Cloud Native)

- 클라우드 네이티브 컴퓨팅 재단(CNCF)에서 정의
- CNCF(Cloud Native Computing Foundation)는 오픈소스와 제조사 중립 프로젝트 생태계 육성/유지하기 위해 리눅스 재단에서 관리 (홈페이지: <https://www.cncf.io/> )
- 자동화와 결합하여 엔지니어가 최소한의 노력으로 빈번하고 예측 가능한 변경 작업을 수행 할 수 있는 기술과 쉬운 관리 뛰어난 복원력을 제공 할 수 있음
- 클라우드 고유 기술을 사용하며 동적 환경에서 구축하고 실행하여 서비스를 확장하기 위해 컨테이너, 서비스 메시, 마이크로서비스, 변경 불가능 인프라(Immutable Infrastructure) 및 선언적 API를 사용하는 접근 방식을 보여줌

JS Lab

# 1장. 개요

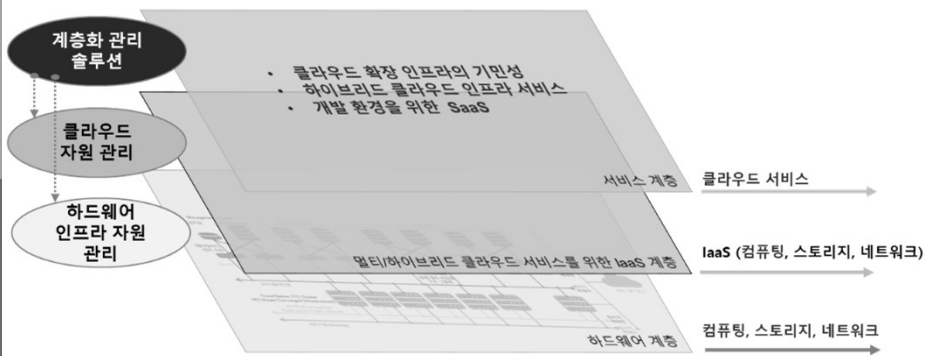
- 가상 네트워크 환경 (Overlay SDN)
- 물리 네트워크 환경 (Underlay SDN)



JS Lab

# 1장. 개요

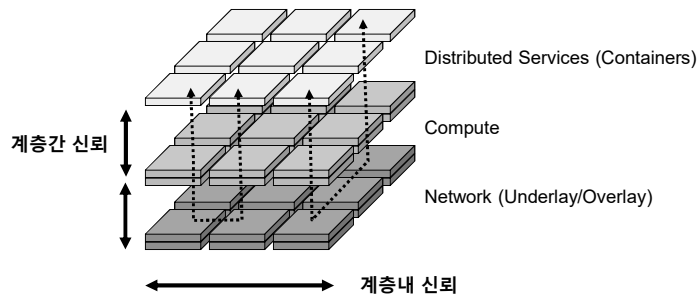
- 클라우드 네트워킹: 독립적 추상화
- 가상 네트워크 환경 (Overlay SDN): VMware의 NSX 등
- 물리 네트워크 환경 (Underlay SDN): Cisco의 ACI 등



JS Lab

# 1장. 개요

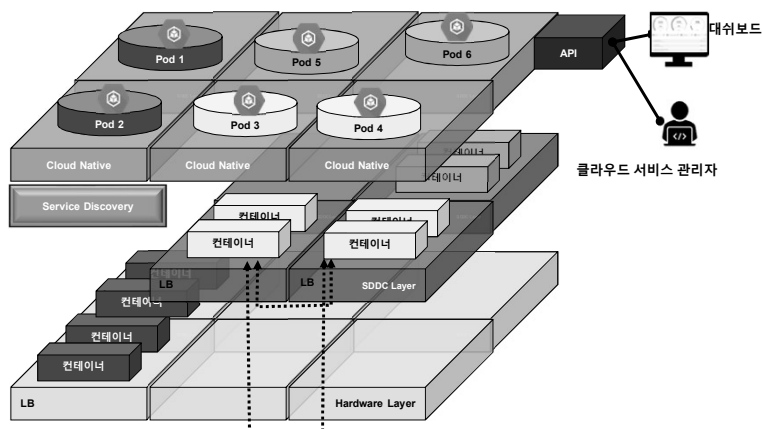
- 계층간 보안 고려
- 계층간 신뢰 필요: 계층간 인증/터널링 연결, 성능 이슈
- 계층내 신뢰 필요: 계층 내의 폐쇄그룹 구성



JS Lab

# 1장. 개요

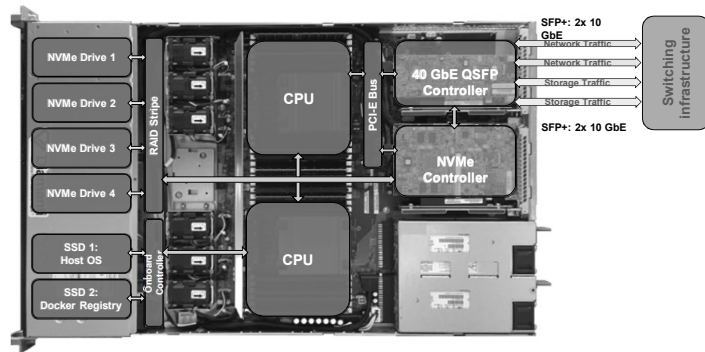
- Kubernetes는 자원 배포 단위로 Pod를 사용하여 추상화
- Pod는 1개 이상의 Container로 구성



JS Lab

# 1장. 개요

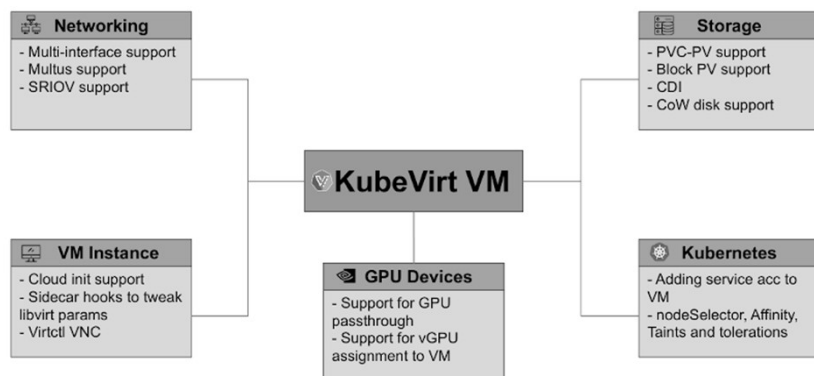
- 클라우드 네이티브화를 위한 HCI(Hyper Converged Infrastructure)
- Kubernetes CNI(Container Network Interface)와 CSI(Container Storage Interface) 가속 필요



JS Lab

# 1장. 개요

- KubeVirt

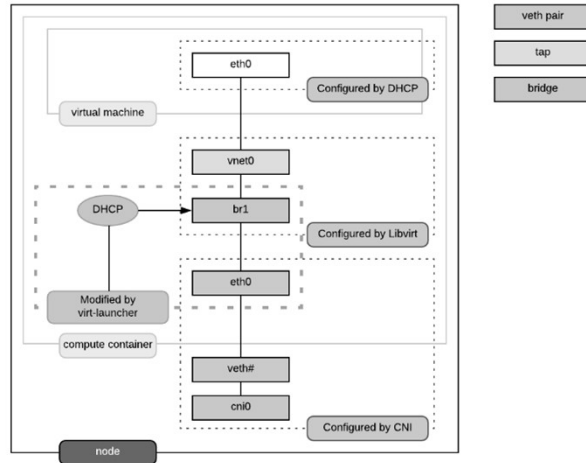


[https://kubevirt.io/2020/KubeVirt\\_deep\\_dive-virtualized\\_gpu\\_workloads.html](https://kubevirt.io/2020/KubeVirt_deep_dive-virtualized_gpu_workloads.html)

JS Lab

# 1장. 개요

## □ KubeVirt Networking

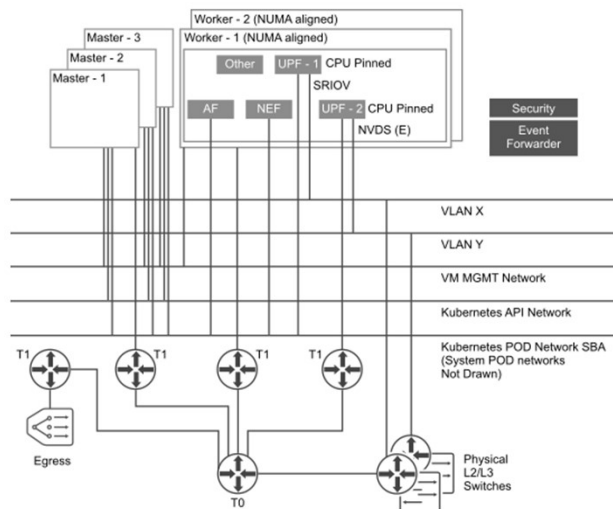


<https://kubevirt.io/2018/KubeVirt-Network-Deep-Dive.html>

JS Lab

# 1장. 개요

## □ Edge using vCloud NFV

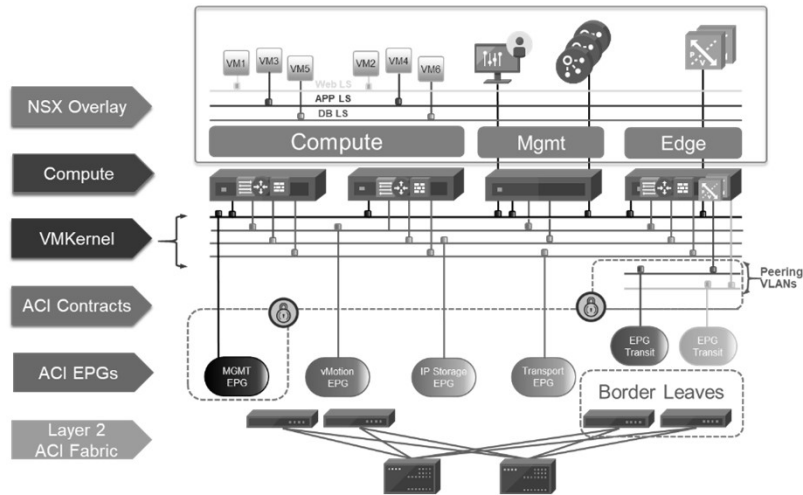


<https://docs.vmware.com/en/VMware-vCloud-NFV-OpenStack-Edition/3/vcloud-nfv-cloud-native-es-33/GUID-6B8C1471-41A0-4436-B9C2-6A9CD7488D9.html#?tblid=hwAR2wJPd8yA/ywCADh20WQLngsh4Llmu51H3cCSOPNGB...>

JS Lab

# 1장. 개요

## □ NSX Data Center Deployed on an ACI Underlay

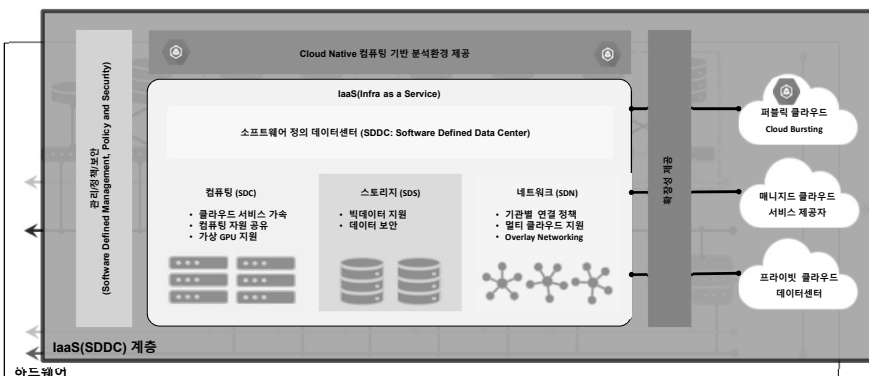


<https://nsx.techzone.vmware.com/resource/deploying-nsx-data-center-cisco-aci-underlay-design-guide-20#sec4-sub1>

JS Lab

# 1장. 개요

- 하드웨어의 추상화 기반 IaaS 인프라 운영 (SDDC)
- 가상화 자원 제공
- 하이브리드(Hybrid)/멀티(Multi) 클라우드 인프라 서비스

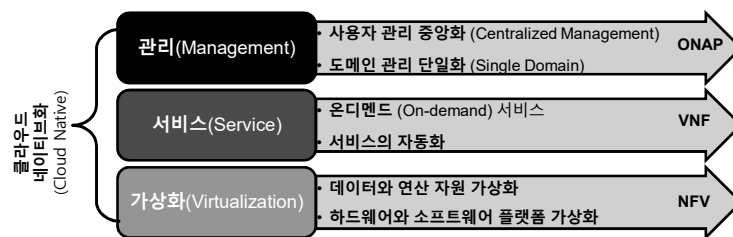


JS Lab

# 1장. 개요

## □ 서비스의 클라우드 네이티브화 (Cloud Native)

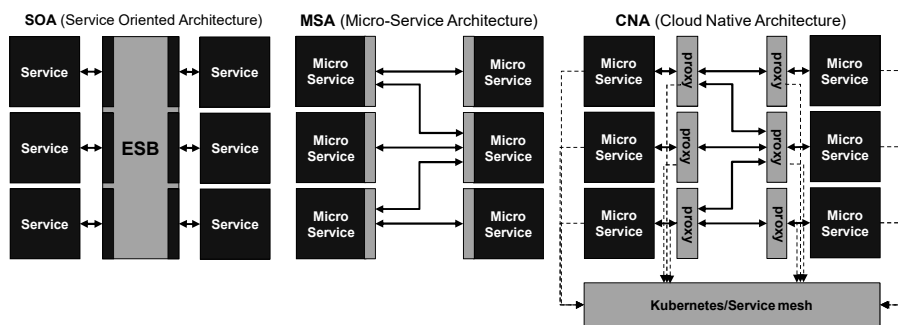
- 서비스 중단없는 배포, 자가 복구 (Self-healing)
- 컴퓨팅/스토리지/네트워크/DB/소프트웨어 자원 등의 유연한 사용
- 호환성 이슈 해결과 사용의 편리 (User Friendly)
- 빠른 서비스 제공



JS Lab

# 1장. 개요

- Monolithic
- Service Oriented Architecture (SOA)
- Microservice architectures (MSA)
- Cloud Native Architecture (CNA)



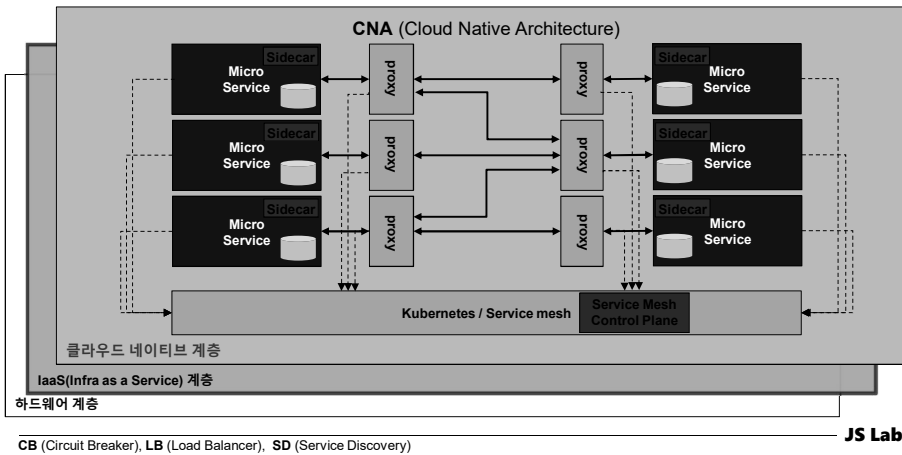
JS Lab



# 1장. 개요

## □ CNA 의 서비스 메쉬 관리

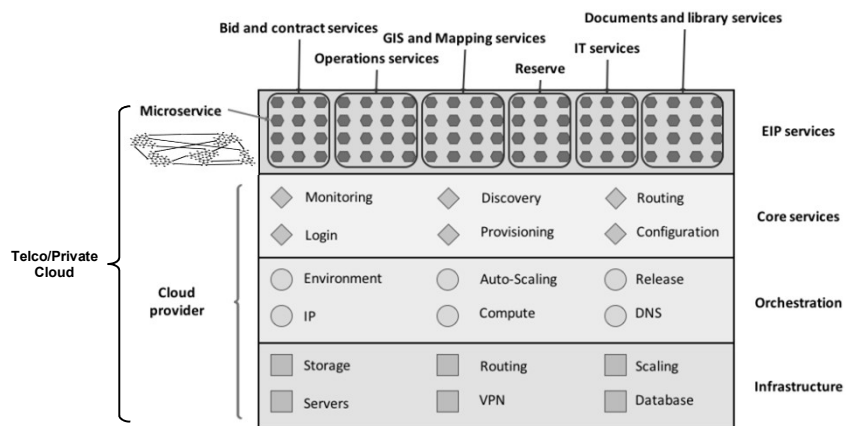
- Sidecar Design Pattern: 라우터 내장 CB, LB, SD 내장
- CNCF의 'Istio'는 정책 강화 Telemetry 제공



# 1장. 개요

## □ Microservice Implementation on Cloud Infrastructure

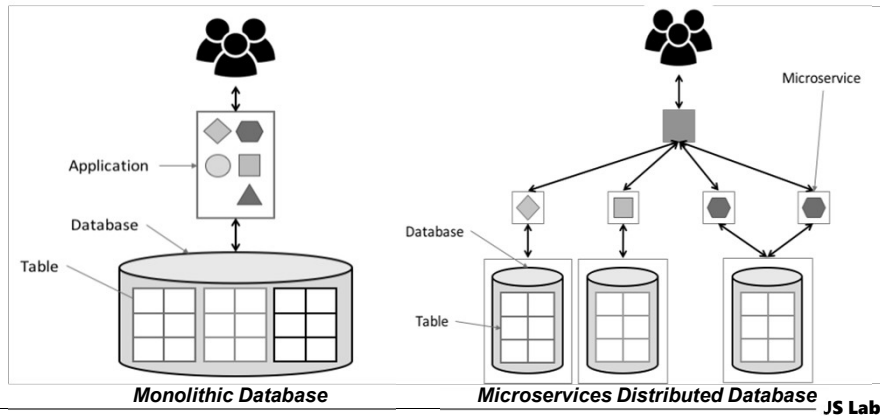
- 퍼블릭 클라우드 환경의 IT 팀들은 하드웨어/소프트웨어 관리 환경에서 마이크로 서비스 기반 서비스 계층을 관리하는 것으로 변화



# 1장. 개요

## □ Refactoring (from Monolithic Module to MSA)

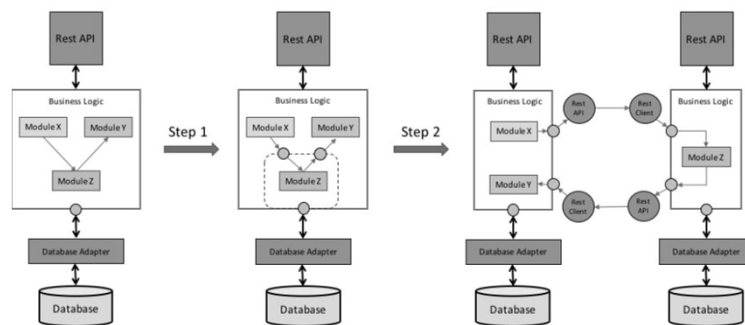
- Decentralized business and messaging rules
- Decentralized governance
- Decentralized data management



# 1장. 개요

## □ Refactoring (from Monolithic Module to MSA)

- 서비스 모듈간 메시지 송수신 필요 (API 사용)



Refactoring

- 외부 동작을 바꾸지 않으면서 내부 구조를 개선하는 방법으로, 소프트웨어 시스템을 변경하는 프로세스
- 소프트웨어의 기능은 바꾸지 않음



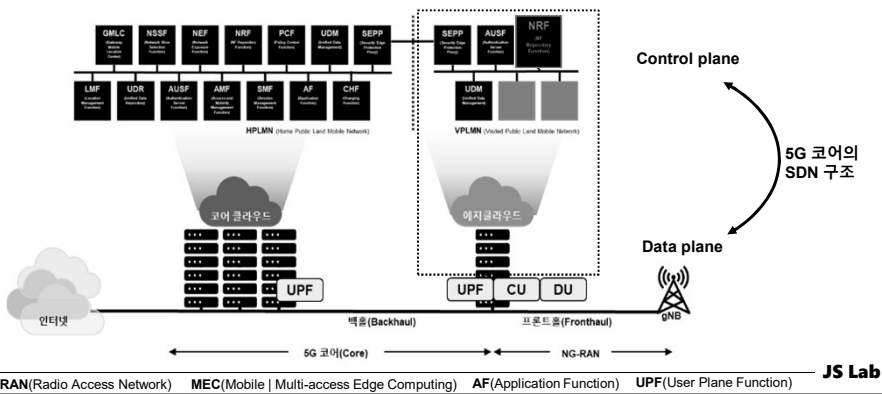
JS Lab

<https://haloworld.tistory.com/24> [Halo World:]

# 1장. 개요

## □ 5G Enabling Technologies:

- CUPS (제어/사용자 플레인 분리)
- 5G Core Control Plane의 Cloud Native화
- MEC (모바일 에지 컴퓨팅)
- Network Slicing (네트워크 슬라이싱)



## 1장. 개요

- 2장. 가상화 (Virtualization)
- 3장. 클라우드 서비스 (Cloud Services)
- 4장. 컨테이너 (Containers)
- 5장. DevOps와 CI/CD
- 6장. 도구(Tools)
- 7장. 서비스 메시 (Service Mesh)
- 8장. 서버리스 (Serverless Computing)
- 9장. 관리 (Management)
- 10장. 보안 (Security)
- 11장. 디자인 패턴 (Design Pattern)
- 12장. Use Case

**2장. 가상화 (Virtualization)**

- 하이퍼바이저
- 가상화와 클라우드 서비스
- 통신 시장의 NFV 환경

JS Lab

**2장. 가상화 (Virtualization)**

□ "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction".

JS Lab

## 2장. 가상화 (Virtualization)

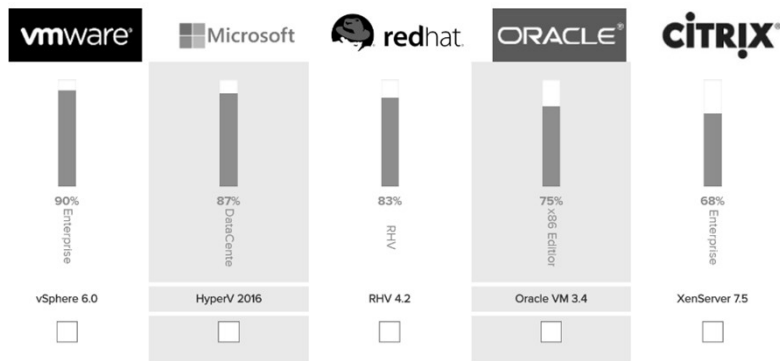
- Infrastructure as a Service (IaaS)
- Platform as a Service (PaaS)
- Software as a Service (SaaS)

	Private Cloud	Infrastructure (as a service)	Platform (as a service)	Function (as a service) (serverless arch)	Software (as a service)
	Functions	Functions	Function	Functions	Functions
	Data	Data	Data	Data	Data
	Application	Data	Application	Application	Application
	Runtime	Runtime	Runtime	Runtime	Runtime
	Backend Code	Backend Code	Backend Code	Backend Code	Backend Code
	OS	OS	OS	OS	OS
가상화	Virtualization	Virtualization	Virtualization	Virtualization	Virtualization
하드웨어	Server Machines	Server Machines	Server Machines	Server Machines	Server Machines
	Storage	Storage	Storage	Storage	Storage
	Networking	Networking	Networking	Networking	Networking

JS Lab

## 2장. 가상화 (Virtualization)

- 기업용 Private Cloud를 위한 가상화
- VMware, Microsoft, redhat, Oracle, Citrix
- 필요 기능에 따라 제조사 평가 결과가 다를 수 있음



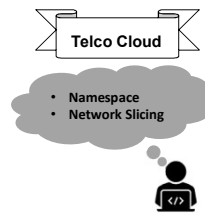
<https://www.whatmatrix.com/comparison/Virtualization>

JS Lab

## 2장. 가상화 (Virtualization)

### □ Key Features of Cloud Computing

- Speed and Agility
- Cost
- Easy Access to Resources
- Maintenance
- Multi-tenancy
- Reliability

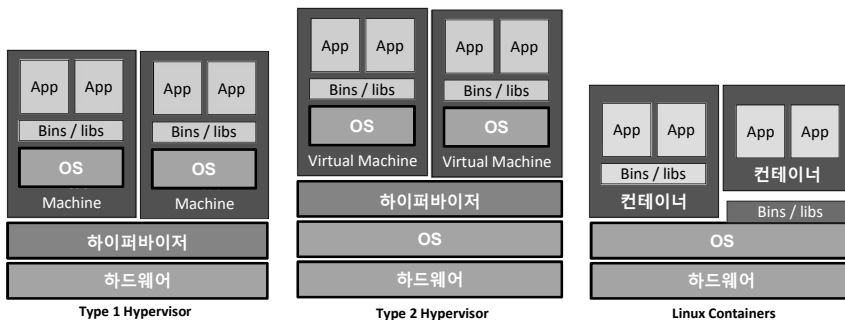


JS Lab

james@jlab.kr

## 2장. 가상화 (Virtualization)

- KVM
- Xen
- VMware
- VirtualBox
- Hyper-V

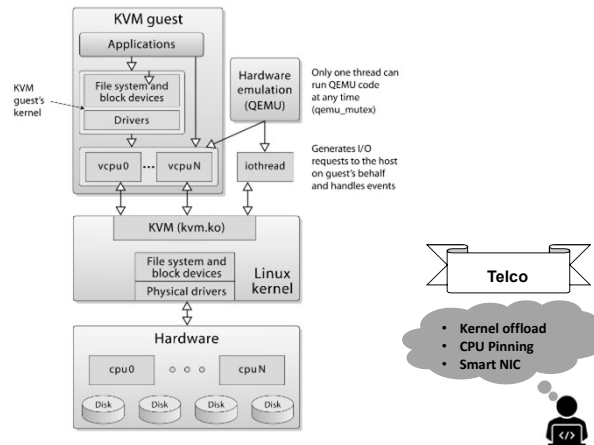


JS Lab

james@jlab.kr

## 2장. 가상화 (Virtualization)

### □ KVM



A High-Level Overview of the KVM/QEMU Virtualization Environment  
(by V4711, licensed under CC BY-SA 4.0, retrieved from Wikipedia)

JS Lab

## 2장. 가상화 (Virtualization)

### □ VirtualBox

- VirtualBox is an x86 and AMD64/Intel64 virtualization product from Oracle, which runs on Windows, Linux, Macintosh, and Solaris hosts and supports guest OSes from Windows, Linux families, and others, like Solaris, FreeBSD, DOS, etc.
- It is an easy-to-use multi-platform hypervisor. It is not part of the mainline kernel. So, to use it on Linux, we have to compile and insert the respective kernel module.
- VirtualBox is distributed under the GNU General Public License (GPL) version 2.

JS Lab

## 2장. 가상화 (Virtualization)

---

### □ Benefits of Using VirtualBox

- It is an open source solution.
- It is free to use.
- It runs on Linux, Windows, OS X, and Solaris.
- It provides two virtualization choices: software-based virtualization and hardware-assisted virtualization.
- It is an easy-to-use multi-platform hypervisor.
- It provides the ability to run virtualized applications side-by-side with normal desktop applications.
- It provides teleportation - live migration.

JS Lab

## 2장. 가상화 (Virtualization)

---

### □ Vagrant

- Reproducible environment
- Management of multiple projects in their restricted environment
- Sharing the environment with other teammates
- Keeping the development and deployment environments in sync
- Running the same VM on different OSes, with a hypervisor like VirtualBox.

```
$ vagrant init hashicorp/precise64
$ vagrant up
```

JS Lab



james@jlab.kr

---

- 1장. 개요
- 2장. 가상화 (Virtualization)
- 3장. 클라우드 서비스 (Cloud Services)
- 4장. 컨테이너 (Containers)
- 5장. DevOps와 CI/CD
- 6장. 도구(Tools)
- 7장. 서비스 메시 (Service Mesh)
- 8장. 서버리스 (Serverless Computing)
- 9장. 관리 (Management)
- 10장. 보안 (Security)
- 11장. 디자인 패턴 (Design Pattern)
- 12장. Use Case

JS Lab

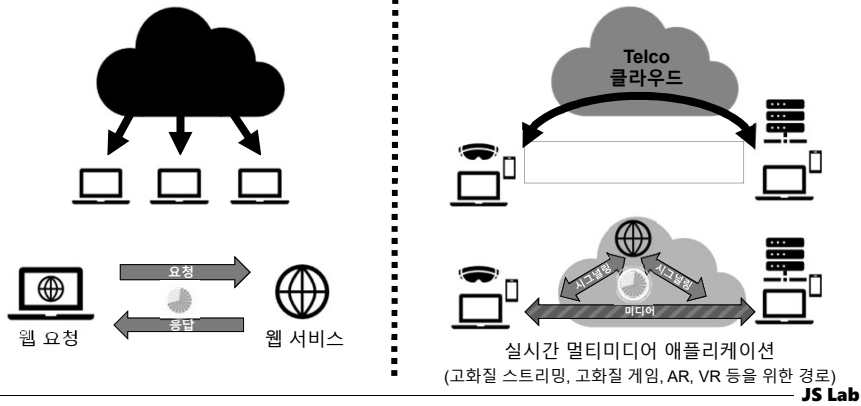
- james@jlab.kr
- 
- 3장. 클라우드 서비스 (Cloud Services)**

  - Domain 별 클라우드
  - IaaS
  - PaaS
  - Hybrid / Multi-Cloud
- JS Lab

### 3장. 클라우드 서비스 (Cloud Services)

#### □ Public vs. Telco 클라우드

- Public Cloud: 소프트웨어 정의 가상 인프라 기반 서비스 (웹서비스)
- Telco Cloud: 언더레이 인프라 기반 클라우드 서비스 (전송경로 제공)
- Telco Cloud는 하드웨어 인프라 환경 고려

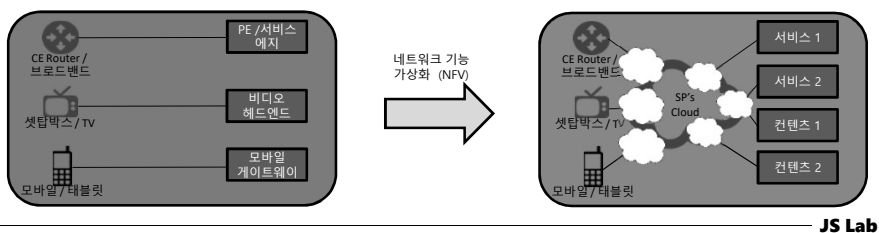


### 3장. 클라우드 서비스 (Cloud Services)

#### □ 엔터프라이즈는 애플리케이션 요구에 적합한 클라우드 자원

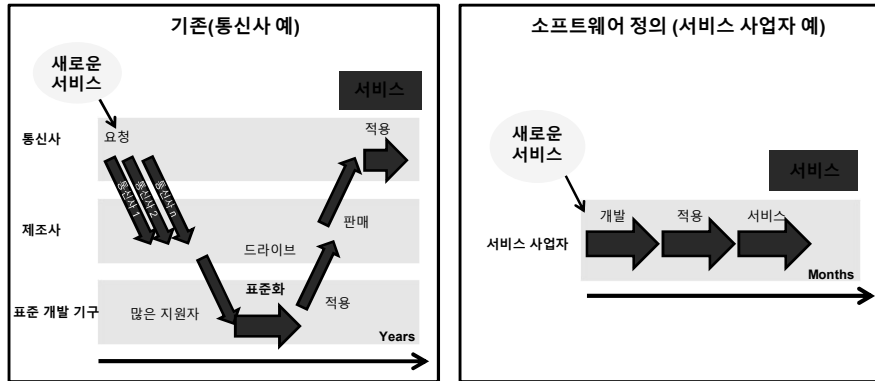


#### □ 텔코 (Telco)는 네트워크 기능 가상화 기반 데이터센터화



### 3장. 클라우드 서비스 (Cloud Services)

- 통신사와 서비스 사업자의 적용 환경
- 개발능력 미보유 엔터프라이즈는 서비스 제공 가능 오픈소스나 제조사의 SDx 솔루션 필요



JS Lab

### 3장. 클라우드 서비스 (Cloud Services)

#### □ Edge vs Core @ Telco

	Edge (에지) 클라우드	중앙 클라우드
App의 위치	노드의 물리적 위치에서 중요한 서비스	비교적 위치와 독립적인 서비스
워크로드의 이동성	워크로드가 노드간 이동	클라우드 노드 장애 이외에는 비교적 고정
워크로드의 역동성	다양한 App들이 다양한 시간에 크게 다른 요구를 함	서비스를 적용하면 대부분의 시간에 안정적인 워크로드
아키텍처	다른 형태의 많은 수의 노드와 다양한 용량과 기술	대부분 동일하며 차이가 작음 (예: AWS, OpenStack, Azure 등)
지연	지연과 거리는 종단 사용자들을 위한 주요 역할	대부분 지연에 민감하지 않음
자원 가용성	에지노드는 작고, App을 위한 자원의 가용성을 보장하지 않음	가용성 확보는 중요하며 주요 기능 중 1개

JS Lab

### 3장. 클라우드 서비스 (Cloud Services)

Cloud vendor managed service offerings

- Databases
- Hadoop
- Directory services
- Load balancers
- Caching systems
- Data warehouses
- Code repositories
- Automation tools
- Elastic searching tools

JS Lab

### 3장. 클라우드 서비스 (Cloud Services)

Cloud Vendor's Services 비교 (예: AWS vs GCP)

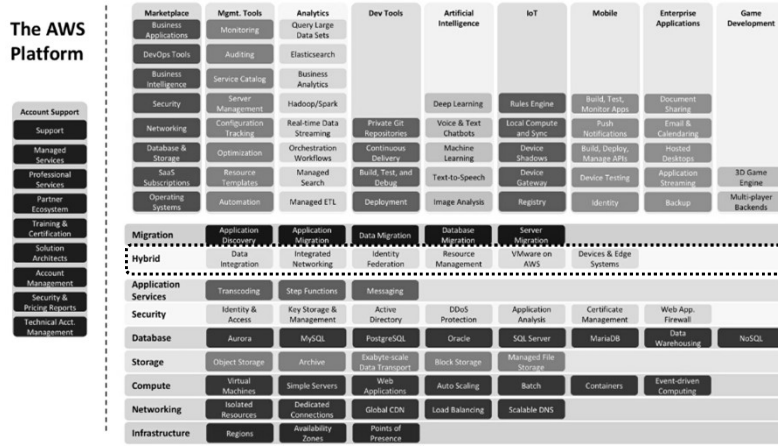
서비스 카테고리	서비스	AWS	Google Cloud Platform
컴퓨팅	IaaS	Amazon Elastic Compute Cloud	Compute Engine
	PaaS	AWS Elastic Beanstalk	App Engine
	컨테이너	Amazon Elastic 컨테이너 서비스	Google Kubernetes Engine
	서버리스 함수	AWS Lambda	Cloud Functions
	관리형 일괄 컴퓨팅	AWS Batch	해당 없음
네트워크	가상 사설 클라우드	Amazon Virtual Private Cloud	가상 사설 클라우드
	부하 분산기	Elastic Load Balancer	Cloud Load Balancing
	전용 상호 연결	Direct Connect	Cloud Interconnect
	도메인 및 DNS	Amazon Route 53	Cloud DNS
	CDN	Amazon CloudFront	Cloud CDN
저장소	개체 저장소	Amazon S3	Cloud Storage
	블록 저장소	Amazon EBS	Cloud Storage
	제한 가용성 저장소	Amazon S3 Glacier	Cloud Storage
	아카이브 저장소	Amazon S3 Glacier Deep Archive	Cloud Storage
	파일 저장소	Amazon S3 Glacier Select	Cloud Storage
데이터베이스	RDBMS	Amazon RDS	Cloud SQL
	NoSQL: 키-값	Amazon DynamoDB	Cloud Firestore
	NoSQL: 색인 생성	Amazon ElastiCache	Cloud Spanner
빅데이터 및 분석	일괄 데이터 처리	Amazon EMR	Cloud Dataproc
	스트림 데이터 처리	Amazon Kinesis	Cloud Dataflow
	스트림 내부 데이터화	Amazon Kinesis Data Analytics	Cloud Dataflow
애플리케이션 서비스 관리 서비스	분석	Amazon Athena	Cloud BigQuery
	워크플로 조정	Amazon Step Functions	Cloud Composer
	메시지	Amazon SNS	Cloud Pub/Sub
머신러닝	모니터링	Amazon CloudWatch	Cloud Monitoring
	로깅	Amazon CloudWatch Logs	Cloud Logging
	배분	AWS CloudFormation	Cloud Deployment Manager
	음성	Amazon Transcribe	Cloud Text-to-Speech
	Vision	Amazon Rekognition	Cloud Vision AI
자연 언어 처리	Amazon Comprehend	Cloud Natural Language API	
번역	Amazon Translate	Cloud Translation	
대화 인터페이스	Amazon Lex	Dialogflow Enterprise 버전	
Video Intelligence	Amazon Rekognition Video	Cloud Video Intelligence	
자동 생성 모델	해당 없음	Cloud AutoML (베타)	
완전 관리형 ML	Amazon SageMaker	AI Platform	

JS Lab

### 3장. 클라우드 서비스 (Cloud Services)

#### Cloud Vendor's Services

##### Amazon Web Service(AWS) 클라우드 플랫폼(예)

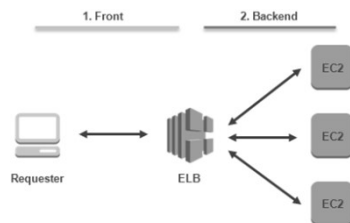


JS Lab

### 3장. 클라우드 서비스 (Cloud Services)

#### Amazon EC2 has many features, allowing you to:

- Create an Elastic IP for remapping the Static IP address automatically
- Provision a Virtual Private Cloud for isolation. Amazon Virtual Private Cloud provides secure and robust networking for Amazon EC2 instances
- Use CloudWatch for monitoring resources and applications
- Use Auto Scaling to dynamically resize your resources, etc.



#### aws EC2 instance types

	General Purpose	Compute Optimized	Memory Optimized	Accelerated Computing	Storage Optimized				
Type	t2	m5	c5	r4	x1e	p3	h1	i3	d2
Description	versatile, great for changing workloads	fastest, great for compute-bound workloads	high ratio of compute to memory	Good for memory-intensive applications	Good for full-stack applications	Good for graphics processing and other GPU uses	HDD-backed, balance of compute and memory	SSD-backed, balance of compute and memory	SSD-backed, highest disk IOPS
Resources	1 v for 1GB of RAM	1 v for 2GB of RAM	4 v for 8GB	1 v for 8GB	1 v for 16GB	1 v for 16GB	1 v for 16GB	1 v for 16GB	1 v for 16GB

JS Lab

## 3장. 클라우드 서비스 (Cloud Services)

### □ Benefits of Using Amazon EC2:

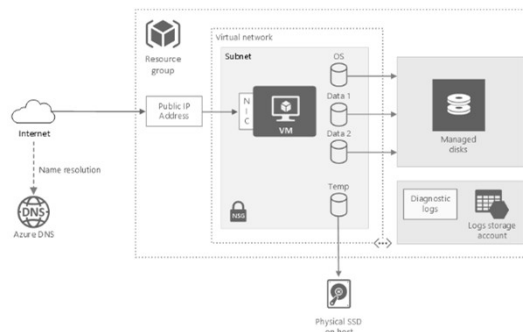
- It is an easy-to-use IaaS solution.
- It is flexible and scalable.
- It provides a secure and robust functionality for your compute resources.
- It enables automation.
- It is cost-effective: you only pay for the time and resources you use.
- It is designed to work in conjunction with other AWS components.
- It promises 99.99% uptime.
- It provides specialized instances for workloads, such as floating point operations, high graphics capability, high input/output (I/O), High Performance Computing (HPC), etc.

JS Lab

## 3장. 클라우드 서비스 (Cloud Services)

### □ Azure Virtual Machine

- We can manage Virtual Machines from Azure's web interface.
- Azure also provides a command line utility to manage resources and applications on the Azure cloud.



JS Lab

## 3장. 클라우드 서비스 (Cloud Services)

### □ The benefits of using Azure virtual machine are:

- It is an easy-to-use IaaS solution.
- It is flexible and scalable.
- It provides a secure and robust functionality for your compute resources.
- It enables automation.
- It is cost-effective: you only pay for the time and resources you use.
- It is designed to work in conjunction with other Azure services.

JS Lab

## 3장. 클라우드 서비스 (Cloud Services)

### □ Google Compute Engine

- Google Cloud Platform is Google's Cloud offering, which has many products in different domains, like compute, storage, networking, big data, and others. Google Compute Engine provides the compute service. We can manage the instances through GUI, APIs or command line. Access to the individual VM's console is also available



JS Lab

### 3장. 클라우드 서비스 (Cloud Services)

---

- GCE supports different machine types, which we can choose from depending on our need. They are categorized in the following types:
  - Standard machine types
  - High-CPU machine types
  - High-memory machine types
  - Shared-core machine types
  - We can also configure custom machine types.

JS Lab

### 3장. 클라우드 서비스 (Cloud Services)

---

- Benefits of Using Google Compute Engine:
  - It is flexible and allows you to scale your applications easily.
  - Fast boot time.
  - It is very secure, encrypting all data stored.
  - It enables automation.
  - It is cost-effective: you only pay for the time and resources you use.
  - It supports custom machine types.
  - It supports Virtual Private Cloud, Load Balancers, etc.

JS Lab



### 3장. 클라우드 서비스 (Cloud Services)

- Introduction to OpenStack
- With OpenStack, we can offer a cloud computing platform for public and private clouds. OpenStack was started as a joint project between Rackspace and NASA in 2010. In 2012, a non-profit corporate entity, called the OpenStack Foundation, was formed and it is managing it since then. It is now supported by more than 500 organizations. OpenStack is an open source software platform, which is released under an Apache 2.0 License.

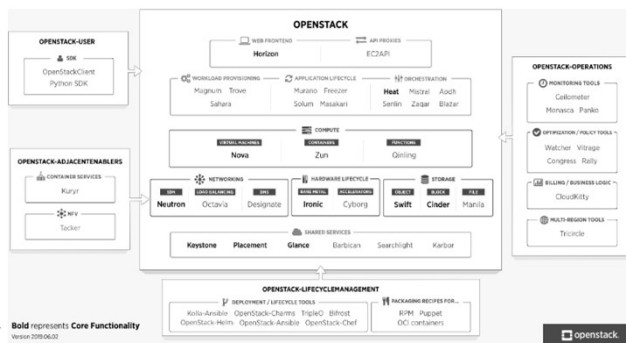
james@jlab.kr

JS Lab

### 3장. 클라우드 서비스 (Cloud Services)

- OpenStack
- Component/Features

- Keystone
- Nova
- Horizon
- Neutron
- Glance
- Swift
- Cinder
- Heat
- Ceilometer



james@jlab.kr

JS Lab

<https://www.openstack.org/software/>

## 3장. 클라우드 서비스 (Cloud Services)

### □ Benefits of Using OpenStack

- It is an open source solution.
- It is a cloud computing platform for public and private clouds.
- It offers a flexible, customizable, vendor-neutral environment.
- It provides a high level of security.
- It facilitates automation throughout the stages of the cloud lifecycle.
- By reducing system management overhead and avoiding vendor lock-in, it can be cost-effective.

JS Lab

## 3장. 클라우드 서비스 (Cloud Services)

### □ Platform as a Service (PaaS)

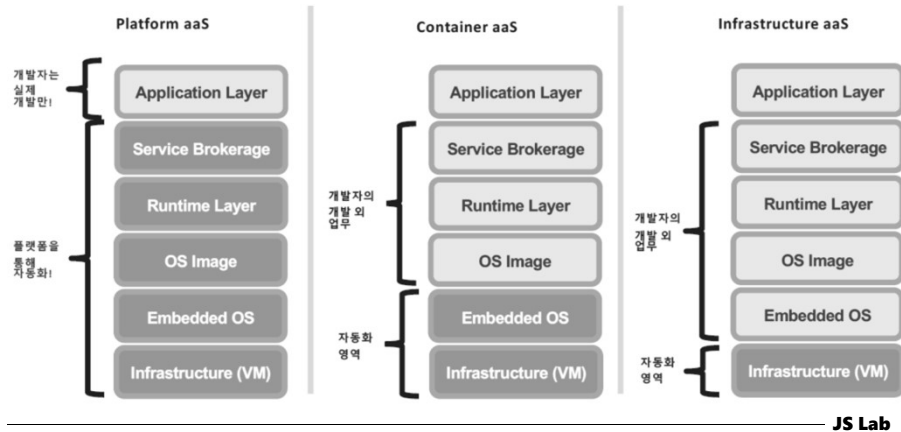
- A class of cloud computing services which allows its users to develop, run, and manage applications without worrying about the underlying infrastructure. With PaaS, users can simply focus on building their applications, which is a great help to developers.
- We can either use PaaS services offered by different cloud computing providers like Amazon, Google, Azure, etc., or deploy it on-premise, using software like OpenShift Origin.
- PaaS can be deployed on top of IaaS, or, independently on VMs, bare metal, and containers.
- In this chapter, we will take a closer look at some of the PaaS providers and their features. We will also provide a demo video for each one of them.

JS Lab

### 3장. 클라우드 서비스 (Cloud Services)

#### □ Cloud Foundry

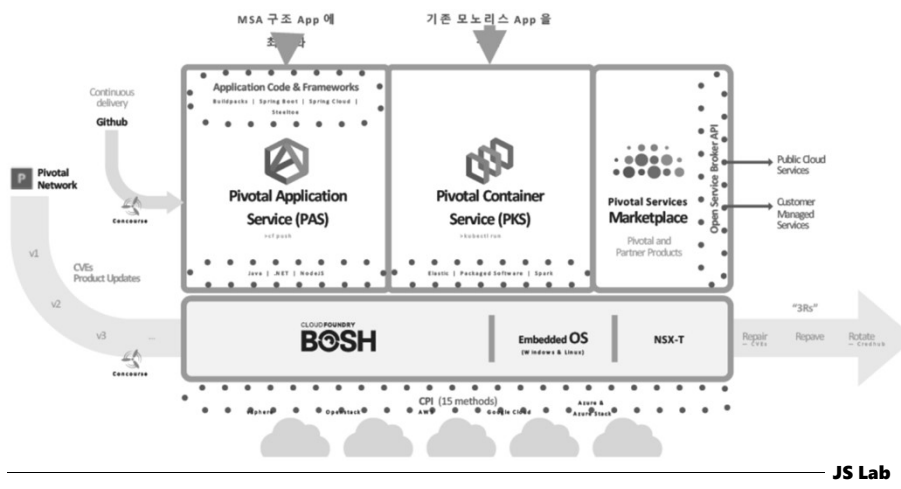
- PCF(Pivotal Cloud Foundry)
- 대규모 서비스에는 별도 플랫폼 담당자가 있을 수 있음



### 3장. 클라우드 서비스 (Cloud Services)

#### □ Cloud Foundry

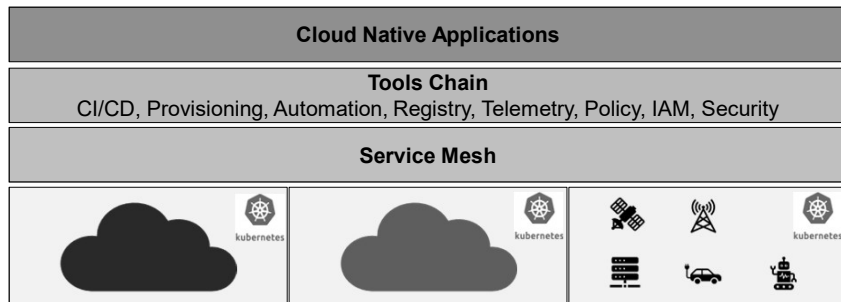
- MSA 개발/운영클라우드플랫폼모델



### 3장. 클라우드 서비스 (Cloud Services)

□ 멀티 클라우드 Federation w/K8s

- High Availability
- Application Migration
- Policy Enforcement
- Vendor Lock-in Avoidance
- Capacity Overflow



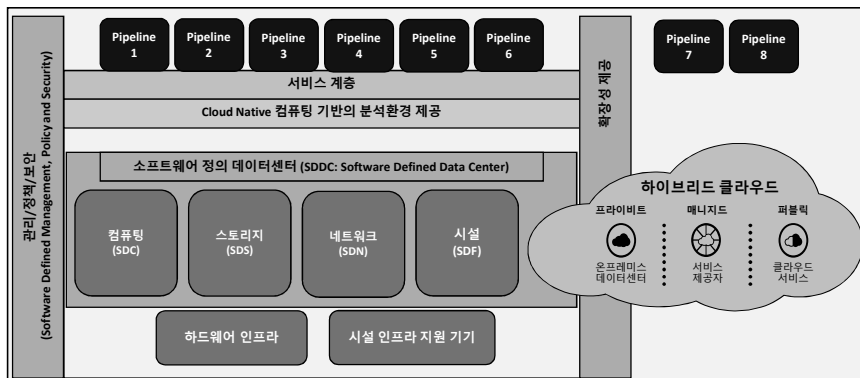
JS Lab

james@jlab.kr

### 3장. 클라우드 서비스 (Cloud Services)

□ SDDC 기반 인프라의 클라우드 수용 (for Enterprise)

- Cloud Native 등의 기술 발전 수용 SDDC 데이터센터 구축 체계
- 데이터센터 자원의 추상화 관리 (SDDC)



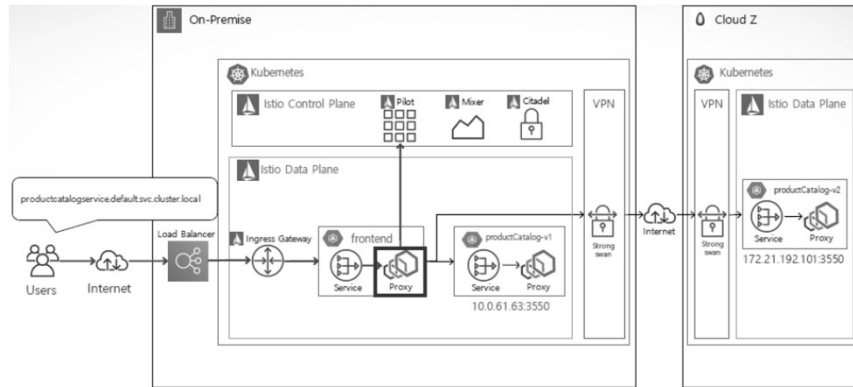
JS Lab

james@jlab.kr

### 3장. 클라우드 서비스 (Cloud Services)

#### □ Hybrid/Multi-Cloud Service Mesh Architecture (예)

- 서로 다른 Cloud(Cross Cluster)간 Service Mesh 연결/확장
- AWS – DirectConnect, Azure - ExpressRoute



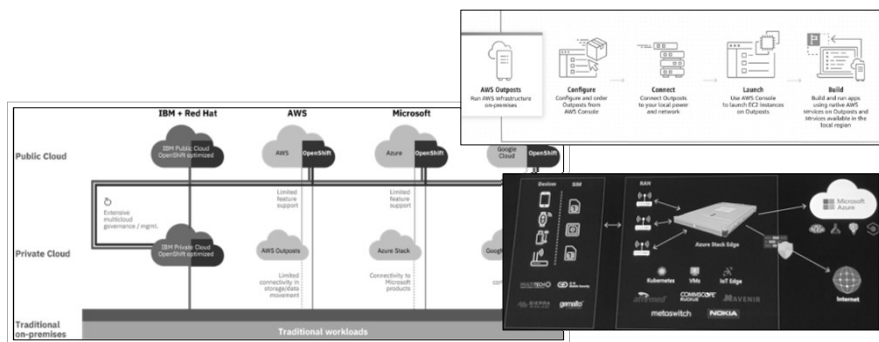
JS Lab

james@jstlab.kr

### 3장. 클라우드 서비스 (Cloud Services)

#### □ 하이브리드/멀티 클라우드

- IBM Cloud Pak
- AWS Outpost
- MS Azure Stack (Edge)
- Google Anthos



JS Lab

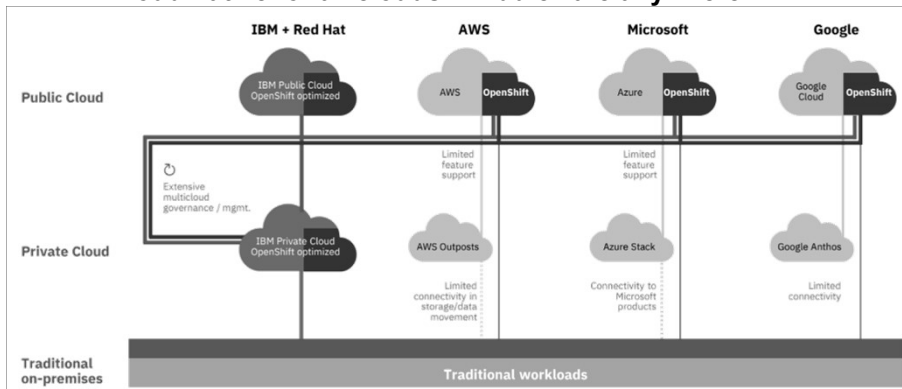
<https://www.metricstream.com/technology/grc-cloud.htm>

james@jstlab.kr

## 3장. 클라우드 서비스 (Cloud Services)

### □ IBM Cloud Pak

- Hybrid, Multicloud platform
- Cloud Pak on OpenShift
- Cloud native for all clouds - middleware anywhere



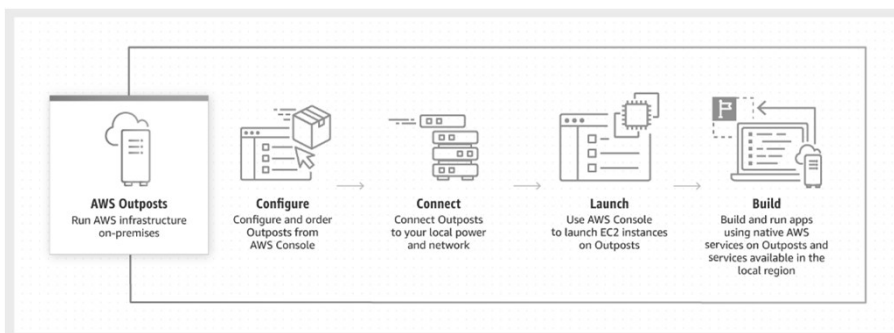
<https://www.metricstream.com/technology/grc-cloud.htm>

JS Lab

## 3장. 클라우드 서비스 (Cloud Services)

### □ AWS Outpost

- AWS 온프레미스 확장
- 관리 플레인 선택: AWS 네이티브 변형 or VMware Cloud on AWS
- 완전관리형



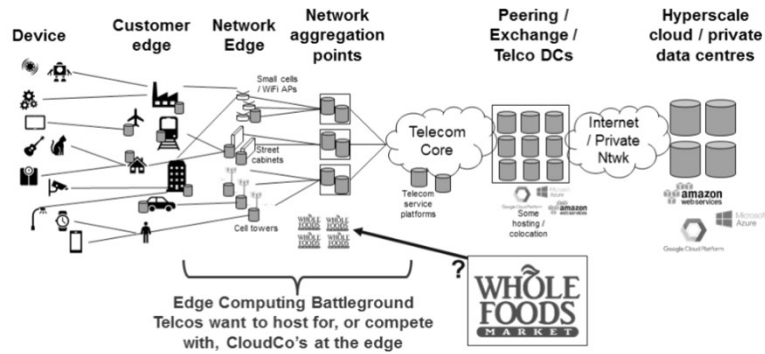
<https://aws.amazon.com/ko/outposts/>

JS Lab

### 3장. 클라우드 서비스 (Cloud Services)

□ 클라우드 환경의 저지연 서비스 요구 증가

- 클라우드 서비스 업체는 에지에서 Telco와 협력 또는 경쟁 (예: AWS, MS Azure, Tencent Cloud 등)






출처: <https://disruptivewireless.blogspot.com/2017/06/does-amazons-purchase-of-whole-foods.html>


JS Lab

### 3장. 클라우드 서비스 (Cloud Services)

□ MS Azure Stack

- Azure Stack Edge: ML, RAN, IoT
- Azure Stack HCI
- Azure Stack Hub

 <p><b>Azure Stack Edge</b> Cloud-managed appliance</p> <p>Machine learning at the edge Edge compute and IoT solutions Network data transfer to cloud</p>	 <p><b>Azure Stack HCI</b> Hyperconverged solution</p> <p>Scalable virtualization and storage Remote branch office High-performance workloads</p>	 <p><b>Azure Stack Hub</b> Cloud-native integrated system</p> <p>Disconnected scenarios Data sovereignty Application modernization</p>
--	--	---

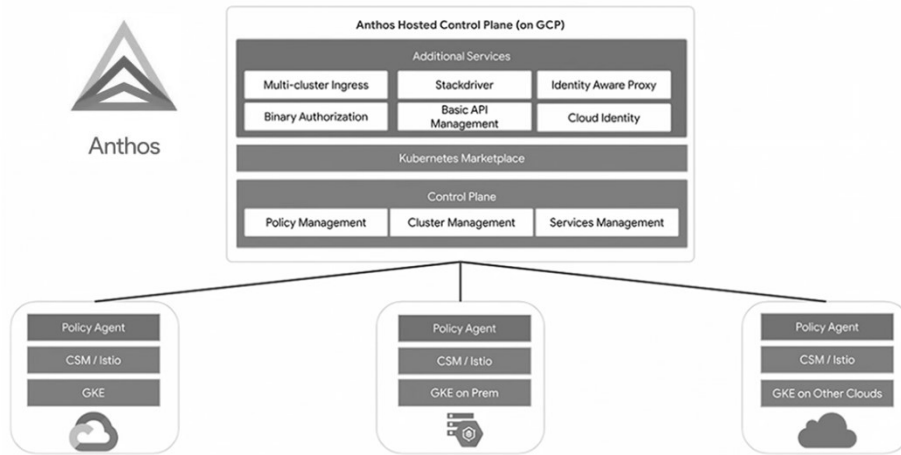


<https://azure.microsoft.com/ko-kr/overview/azure-stack/>

JS Lab

### 3장. 클라우드 서비스 (Cloud Services)

#### □ Google Anthos



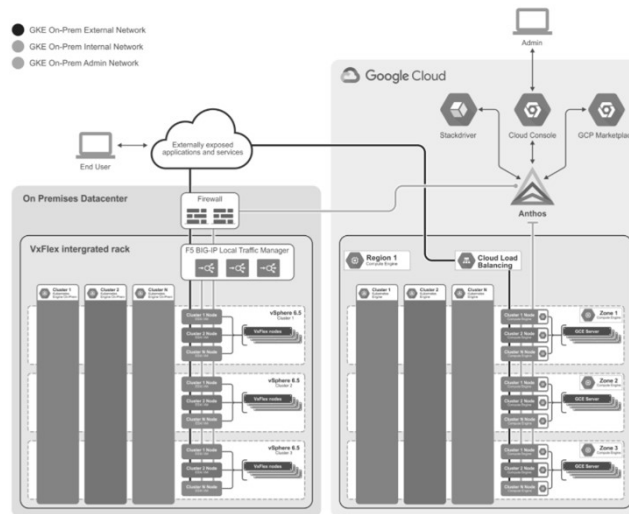
<https://www.bespinglobal.com/google-anthos-multi-cloud/>

<https://blog.aquasec.com/hybrid-cloud-security-google-anthos>

JS Lab

### 3장. 클라우드 서비스 (Cloud Services)

#### □ VxFlex integrated rack for Google Cloud's Anthos



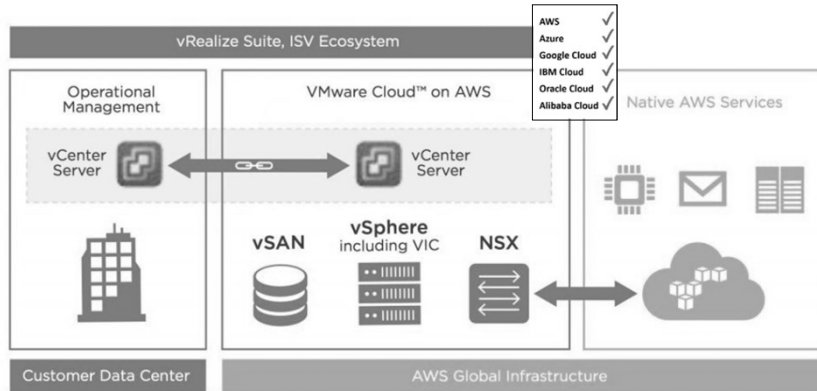
JS Lab



### 3장. 클라우드 서비스 (Cloud Services)

#### □ VMware Cloud

- VMware Cloud on AWS
- 지원 확장: Azure, Google, IBM, Oracle, Alibaba and KT Cloud



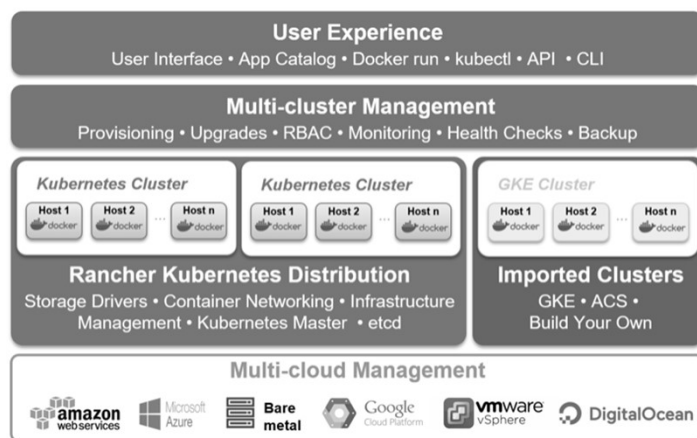
<https://www.actualtech.io/getting-hands-dirty-installing-vmware-cloud-aws/>

JS Lab

### 3장. 클라우드 서비스 (Cloud Services)

#### □ Rancher Labs

- Rancher 2.x



<https://rancher.com/announcing-rancher-2-0/>

JS Lab

### 3장. 클라우드 서비스 (Cloud Services)

□ Rancher Labs

■ Rancher 2.x



<https://rancher.com/announcing-rancher-2-0/>

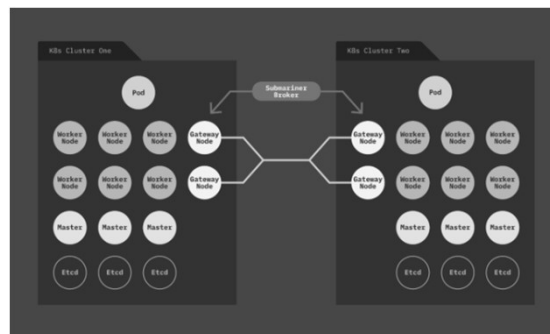
JS Lab

### 3장. 클라우드 서비스 (Cloud Services)

□ K8s Cluster간 네트워크 연결 요구

□ Open source project 'Submariner' (예)

- 카산드라(Cassandra)와 같은 HA 데이터베이스의 지역 분산
- 분산화 트레이스 (Distributed Tracing)
- Service Mesh의 클러스터들 간에 확대



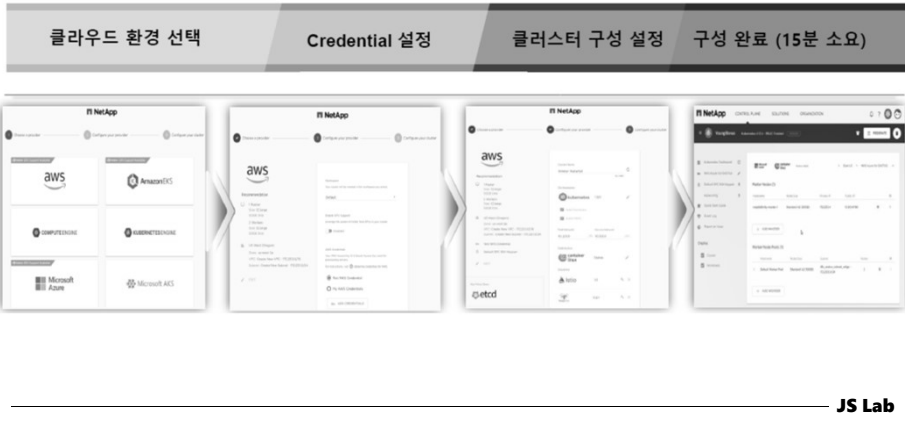
<https://submariner.io/>

JS Lab

### 3장. 클라우드 서비스 (Cloud Services)

#### □ NKS (NetApp Kubernetes Service)

- 자동화 기반의 배포 및 관리

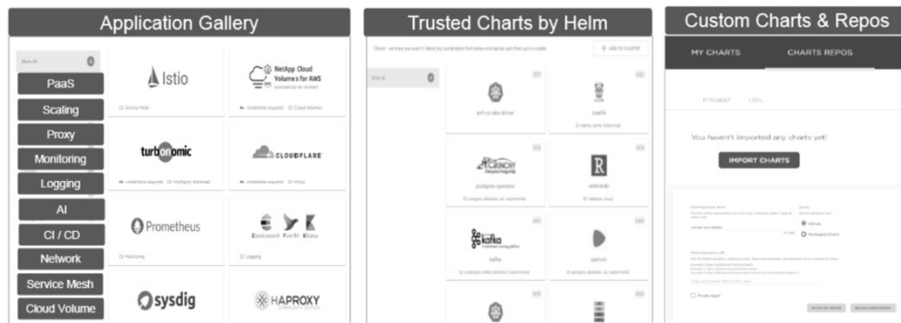


JS Lab

### 3장. 클라우드 서비스 (Cloud Services)

#### □ NKS (NetApp Kubernetes Service)

- 관리 및 배포를 위한 Cloud Native Solutions Marketplace

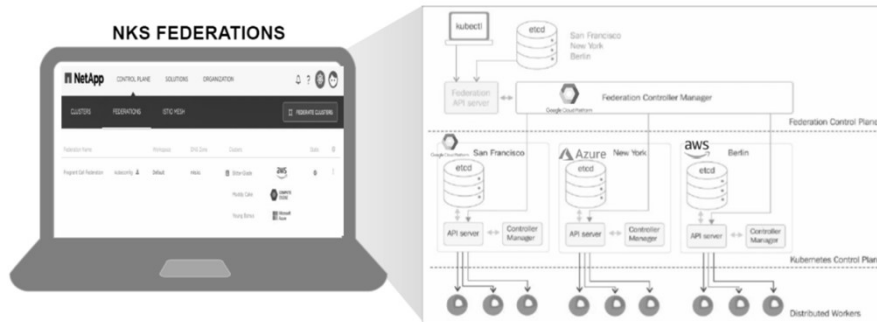


JS Lab

### 3장. 클라우드 서비스 (Cloud Services)

□ NKS (NetApp Kubernetes Service)

- Multi Kubernetes Cluster Multi Federation

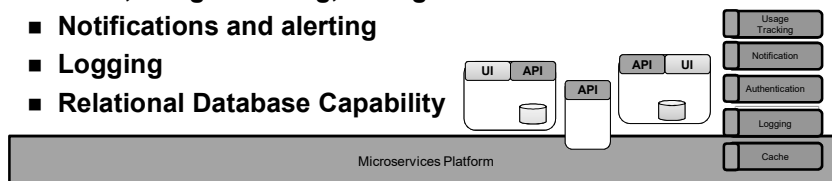


JS Lab

### 3장. 클라우드 서비스 (Cloud Services)

□ Microservices cross platform Capabilities

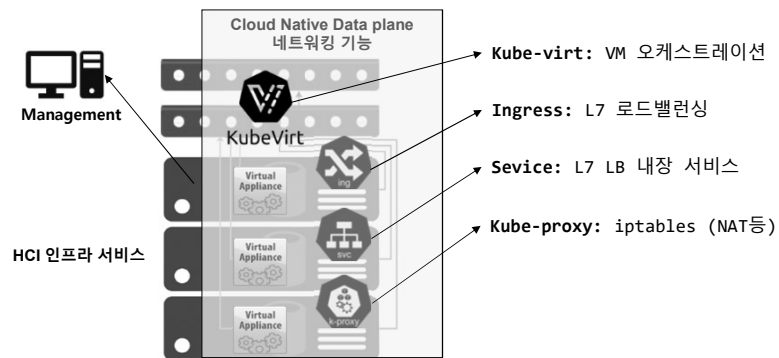
- Authentication
- Persistent Storage
- Cache
- Load balancing/API Gateway
- Discovery/Lookup
- Monitoring
  - ✓ Functional/Business KPIs
  - ✓ Non Functional Platform/Container & Infra
- Audit, Usage tracking, Billing
- Notifications and alerting
- Logging
- Relational Database Capability



JS Lab

## 3장. 클라우드 서비스 (Cloud Services)

- SDDC 기반 클라우드 네이티브 네트워킹
- 멀티 클라우드화 분산 네트워크 기능의 중앙 관리 (SDN)



JS Lab

### 1장. 개요

- 2장. 가상화 (Virtualization)
- 3장. 클라우드 서비스 (Cloud Services)
- 4장. 컨테이너 (Containers)
- 5장. DevOps와 CI/CD
- 6장. 도구(Tools)
- 7장. 서비스 메시 (Service Mesh)
- 8장. 서버리스 (Serverless Computing)
- 9장. 관리 (Management)
- 10장. 보안 (Security)
- 11장. 디자인 패턴 (Design Pattern)
- 12장. Use Case

JS Lab

james@jlab.kr

---

### 4장. 컨테이너 (Containers)

1. 개요
2. Micro OS
3. Orchestration (오케스트레이션)
4. Unikernels
5. Microservices (마이크로서비스)
6. SDN(소프트웨어 정의 네트워크)과 컨테이너
7. SDS(소프트웨어 정의 스토리지)와 컨테이너

JS Lab

james@jlab.kr

---

### 5장. DevOps와 CI/CD

- DevOps
- CI/CD

JS Lab

james@jlab.kr

---

**6장. 도구(Tools)**

- 1. Configuration Management**
- 2. Build & Release**
- 3. Key-Value Pair Store**
- 4. Image Building**
- 5. Debugging, Logging, and Monitoring for Containerized Applications**
- 6. Immutable Infrastructure**

JS Lab

james@jlab.kr

---

**7장. 서비스 메시 (Service Mesh)**

- Control Plane**
- Data Plane**
- Istio**

JS Lab

james@jlab.kr

---

## 8장. 서버리스 (Serverless Computing)

- Serverless
- FaaS

JS Lab

james@jlab.kr

---

## 9장. 관리 (Management)

- Trace
- Log
- Monitor

JS Lab



james@jlab.kr

---

**10장. 보안 (Security)**

- 가상환경 보안
- 컨테이너 환경 보안
- DevSecOps

JS Lab

james@jlab.kr

---

**11장. 디자인 패턴 (Design Pattern)**

- 구성
- Design Pattern References

JS Lab

---

james@jlab.kr

**12장. Use Case**

- 5G Core Infrastructure
- AI Infrastructure
- IoT

JS Lab

---

james@jlab.kr



JS Lab