

# OpenStack

Public과 Private 클라우드를 구축하기 위한 오픈 소스 software



# OpenStack 이란?



Public과 Private 클라우드를 구축하기 위한  
오픈 소스 software를 만들어내는 커뮤니티





OpenStack **Compute**

▶ Massive scale의 스탠다드 하드웨어상에서 가상머신들을 **provision** 하기 위한 소프트웨어

Public과 Private 클라우드를 구축하기 위한  
오픈 소스 software를 만들어내는 커뮤니티



OpenStack  
**Object Storage**

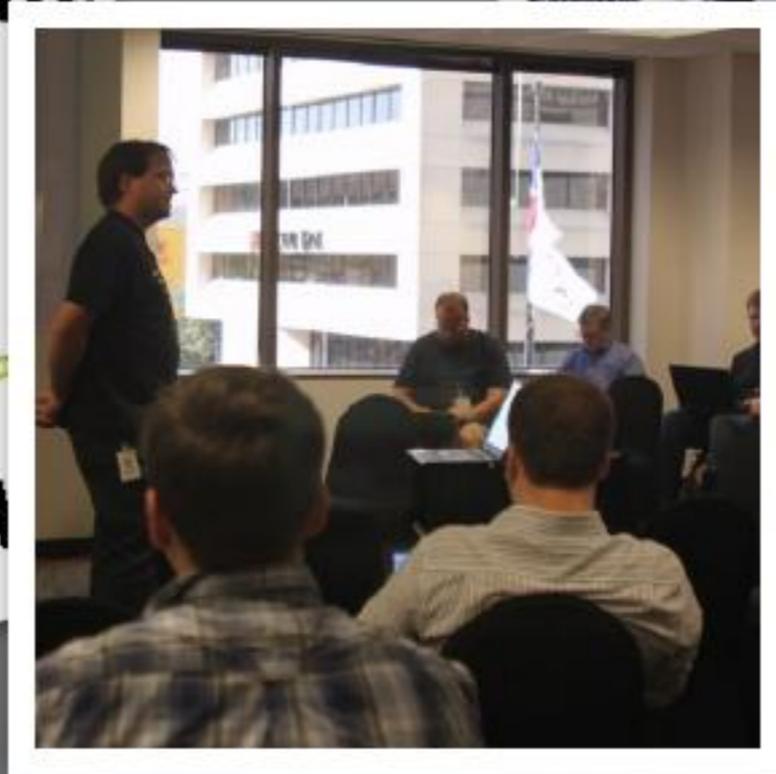
▶ Software to reliably **store billions of objects** distributed across standard hardware



# OpenStack Community Snapshot

50+ Participating Companies

Enterprise & Service Provider Users



Open Source 개발자들

# OpenStack Mission

“규모와 상관없이 Public & Private 클라우드  
제공자들의 요구사항들을 만족시키고,  
implement 하기 간단하며  
동시에 massively scalable 한  
유비쿼터스 오픈소스 클라우드 플랫폼을  
제공한다.”



# Why is OpenStack important?

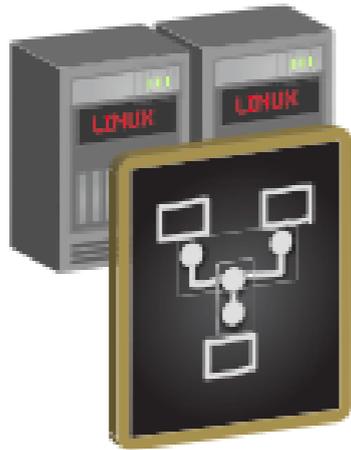
- ▶ **Open** eliminates vendor lock-in
- ▶ Working **together**, we all go faster
- ▶ **Freedom** to federate, or move between clouds



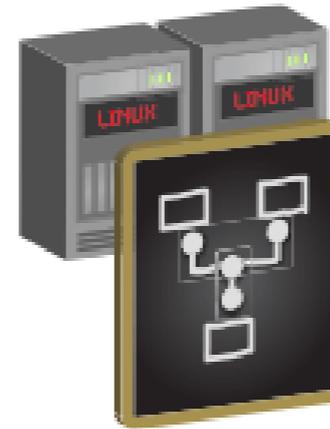
# OpenStack Founding Principles

- ▶ Apache 2.0 license (OSI), **NO** 유료 ‘엔터프라이즈’ 버전
- ▶ Open design process, 수차례의 공개 Design Summits
- ▶ 언제나 접근 가능한 소스코드 repository
- ▶ 모든 커뮤니티 프로세스는 문서화되고 투명하게 운영
- ▶ Open Standard들을 항상 추구하고 받아들임.
- ▶ 유연한 deployment를 위한 Modular 디자인 (API)





Architect for in-house



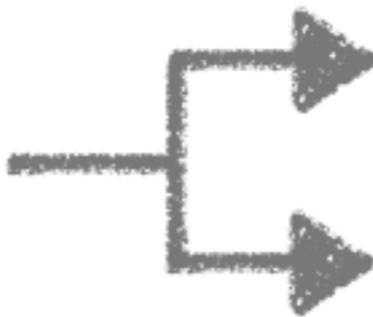
Re-Architect for service provider

Today's Reality

**Future with OpenStack**



Architect once



Deploy anywhere



# OpenStack 진행과정

2010

3월

Rackspace가  
자사의 클라우드  
스토리지  
소프트웨어를  
오픈소스화 하도록  
결정

5월

NASA 가 Nebula  
플랫폼을  
오픈소스화

6월

Rackspace 와  
NASA가  
OpenStack을 설립

7월

Design Summit  
(Austin, TX)

- \* Rackspace: IaaS분야에서의 2위 사업자
- \* NASA Nebula: NASA의 서버가상화 플랫폼



# OpenStack 진행과정



7월

25개의 파트너와 함께 OpenStack 커뮤니티 Launch

10월

첫번째 공식 버전 릴리즈 (Austin Release) - 35 파트너

11월

첫번째 Public Design Summit (San Antonio, TX)

2011

2월

두번째 Bexar 버전 릴리즈



7월

OpenStack Follow-Up 시작

9월

테스트 베드 구성

11월-12월

첫번째 Public Design Summit 참여

2월

OpenStack 한국 커뮤니티 구축

OpenStack 공식 파트너 등록

# OpenStack History



April

July

세번째 'Cactus' 코드 릴리즈 계획

Design Summit  
Santa Clara, CA

4번째 'Diablo' 코드 릴리즈 계획

kt 3 ~ 6월

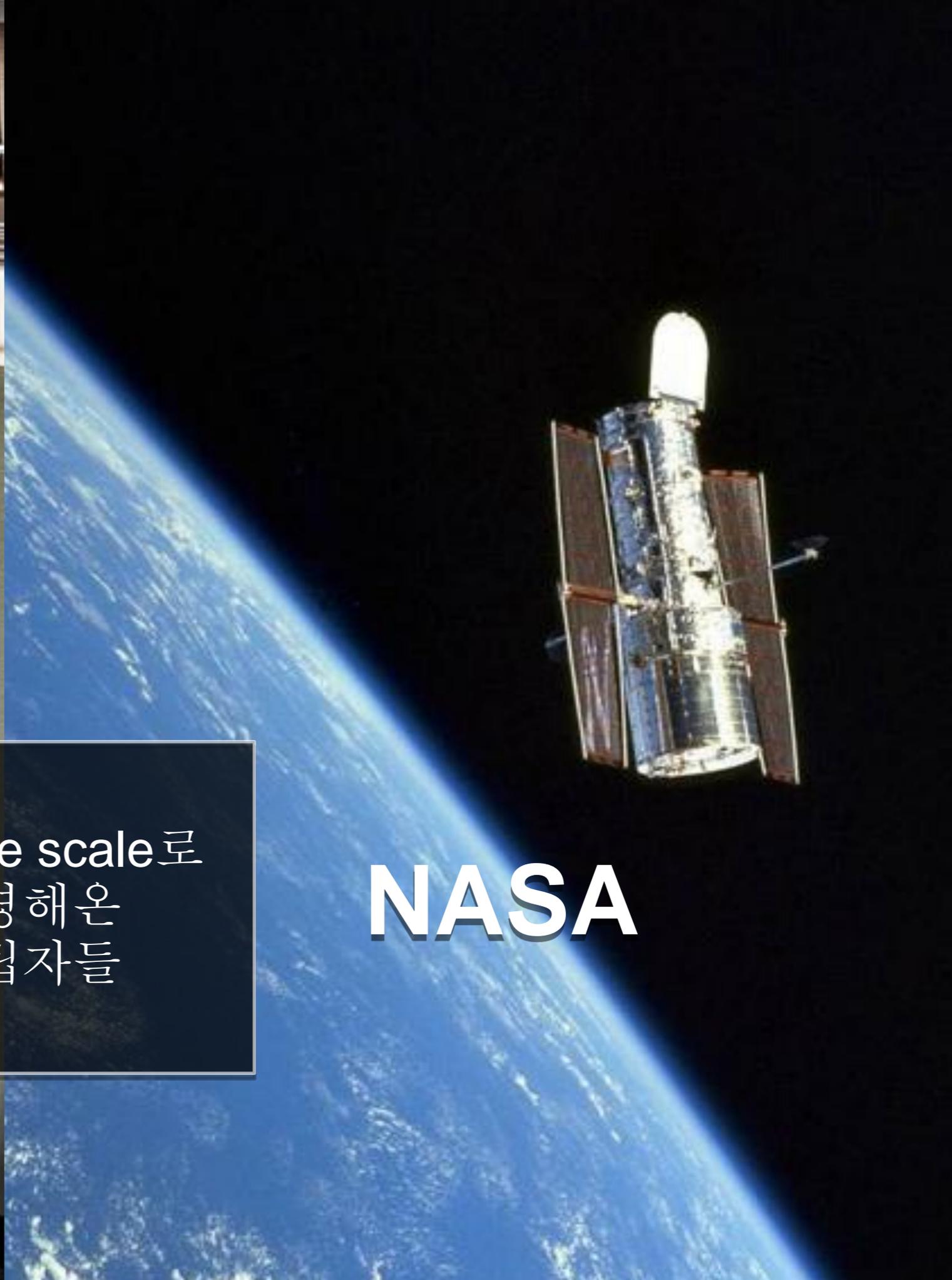
OpenStack Object Storage 기반 클라우드 스토리지 서비스 상용화 (ucloud SS)

- OpenStack 저변확대를 위한 지속적인 커뮤니티 지원, 기술세미나, 컨퍼런스 활동
- OpenStack 기반의 서비스 확장
- 



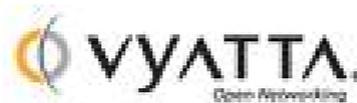
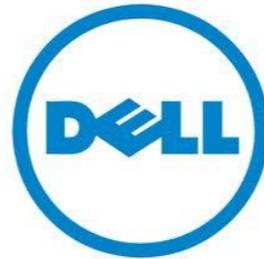
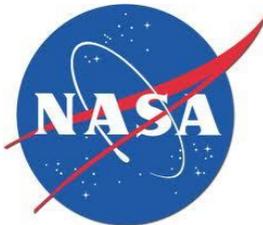


massive scale로  
운영해온  
창립자들



NASA

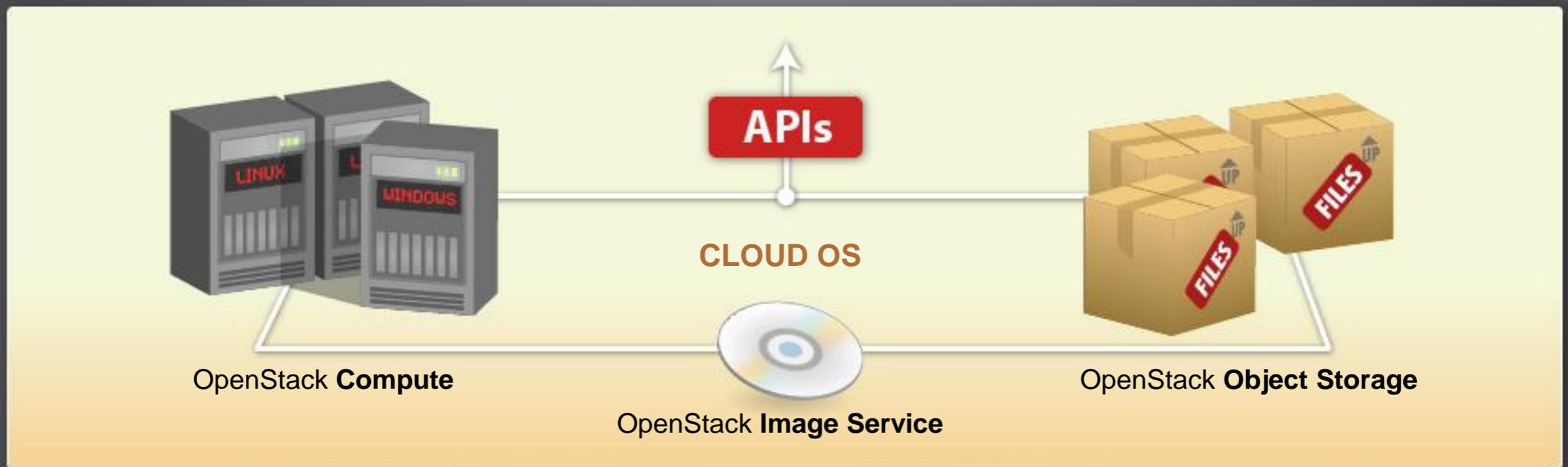
# OpenStack Community Today

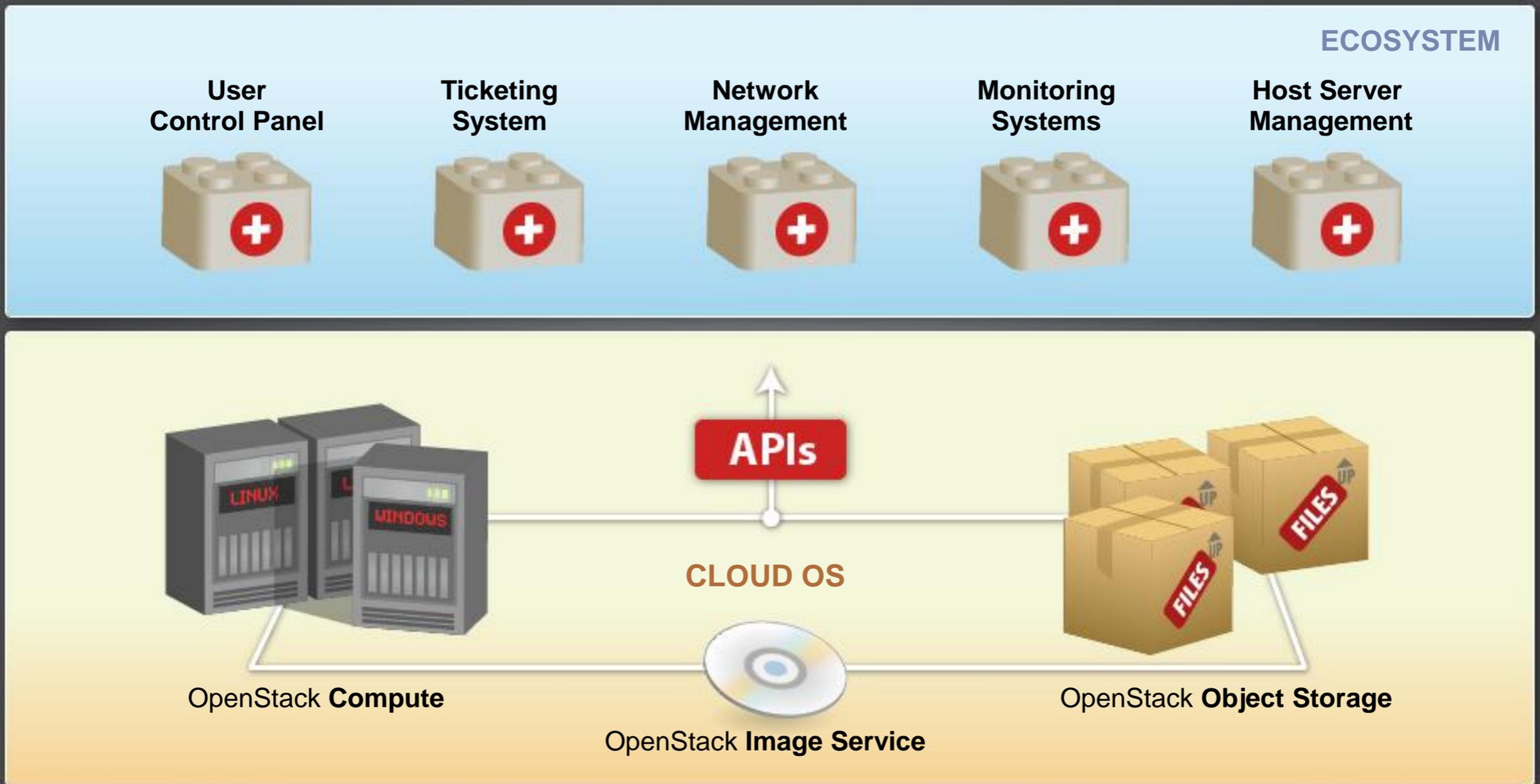


A photograph of a server room with rows of server racks on both sides of a central aisle. A person is standing in the aisle, looking at a laptop. The room is filled with cables and server equipment. The lighting is bright, coming from overhead fixtures.

**HOW TO:** OpenStack을  
사용하여 Commodity  
하드웨어를 클라우드로 전환.

# Open, scalable 플랫폼으로 시작





필요한 3<sup>rd</sup> party 툴과 솔루션들을 에코 시스템으로 부터 수용

**ECOSYSTEM**

**User Control Panel**



**Ticketing System**



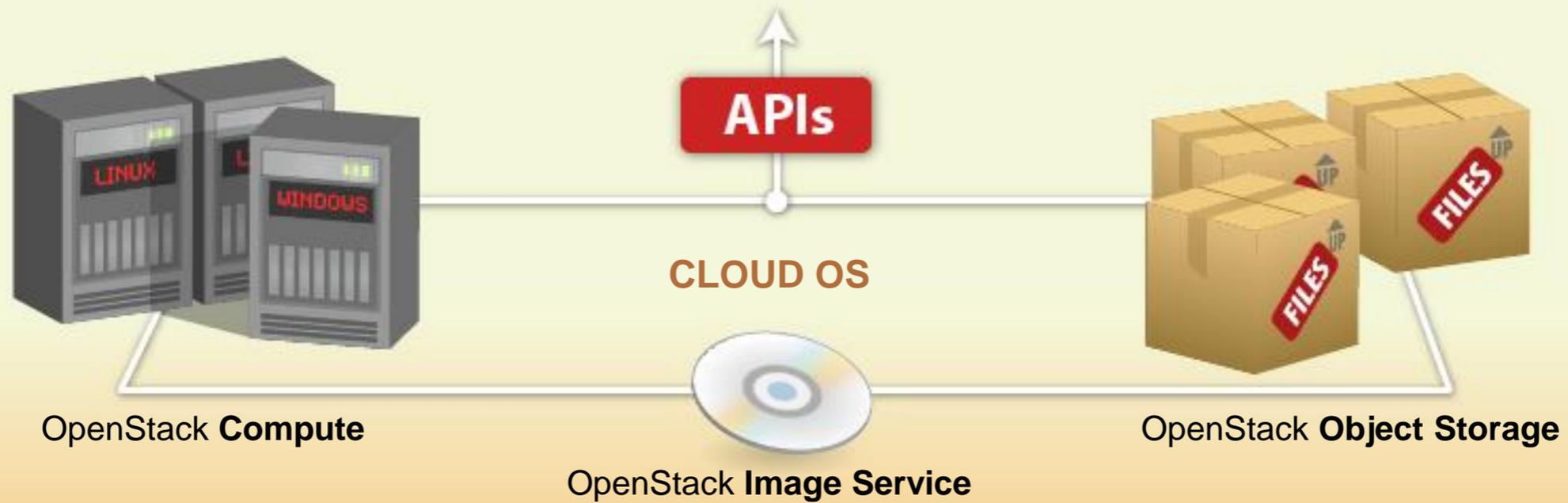
**Network Management**



**Monitoring Systems**



**Host Server Management**



**Account Billing**



**Admin CLI Tools**



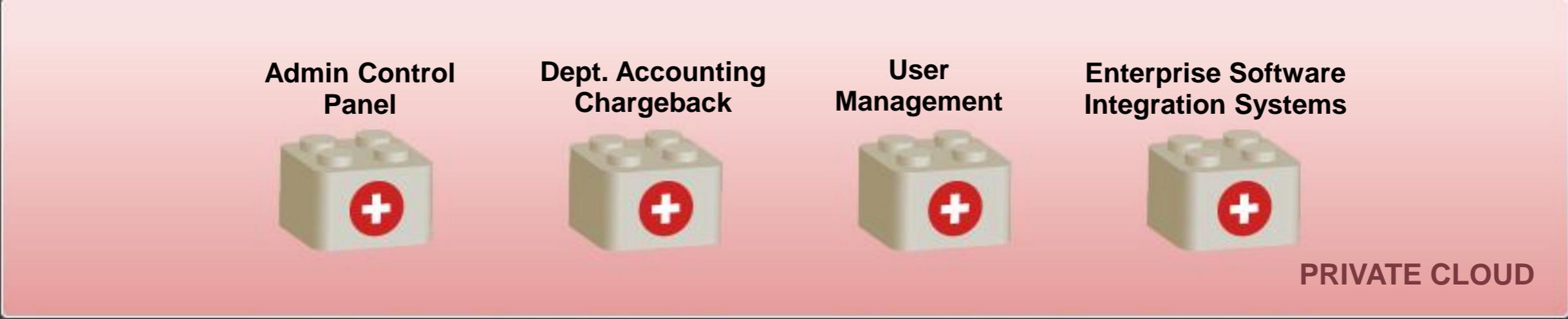
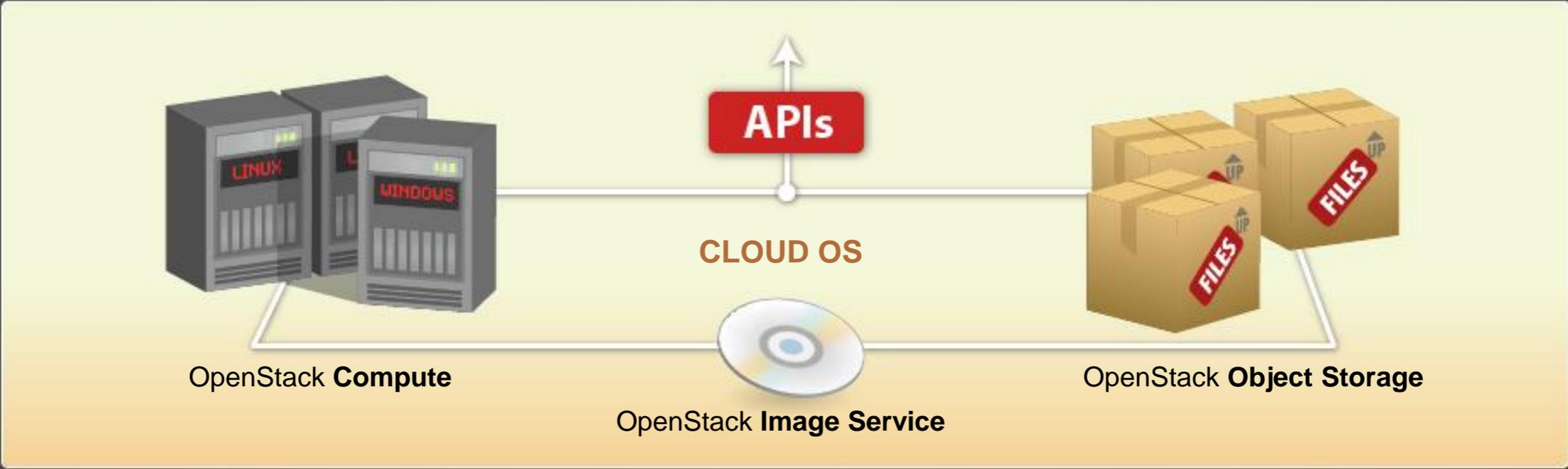
**Live Chat Support**



**Account Management**



**PUBLIC CLOUD**



Integrate with existing enterprise systems

# OpenStack Compute 상세내용

Massive Scale, Commodity 하드웨어상의 소프트웨어 가상머신들을  
Provision 하기위한 소프트웨어



# 컴퓨팅 자원 가상화가 필요한 이유 ?

- 가상화를 사용하는 이유

- 필요에 따른 셀프서비스

클라우드는 셀프서비스를 제공 가능. 사용자는 서버/네트워크/스토리지 등의 자원을 필요에 따라 자동화된 툴 이용 셀프 프로비저닝

- Network 액세스

가상화된 자원은 대부분 네트워크를 통해 표준화된 메커니즘으로 제공됨.

- 리소스 pooling

클라우드는 이용자들에게 제공되는 가상화 리소스를 pooling 함으로써 효율적인 자원 공유 및 절감이 가능함

- Elasticity

가상화 자원은 필요에 따라 신속하게 scale out 처리가 될 수 있는 것은 클라우드의 최대 장점.

- 사용한 만큼 지불

공유된 가상화 자원은 필요한 만큼 사용하고 이용한 만큼 지불.



# 컴퓨팅 가상화 분야의 경쟁자들은?

- 비 OpenSource 분야

- 가상화 관리 소프트웨어 벤더

VMWare : 가장 강력하고 안정적인 가상화 SW 제공. 관련 사업을 리딩.

- 가상화 서비스 제공 분야

Amazon EC2 / S3 : IaaS 서비스 분야의 독보적인 1위. 3<sup>rd</sup> party 들에 의한 ecosystem 이 자연적으로 구축됨.

- OpenSource 분야

- 가상화 관리 소프트웨어

Eucalyptus, Open Nebula, ... 등

다수의 오픈소스 기반의 VM 관리 솔루션들 존재

Amazon EC2/S3 와의 호환성 제공

- OpenStack 은 Open Source 를 지향

- 클라우드 산업에 참여하는 리딩 컴퍼니들의 커뮤니티

- 이제까지 이렇게 많은 업체의 후원과 참여를 보인 클라우드 솔루션은 없었다!



# OpenStack Compute ?

- Compute 자원 가상화를 위한 OpenStack 의 Subproject
  - Codename : NOVA
  - Cloud fabric controller
    - 사용자에게 가상 컴퓨터, 가상 네트워크, 가상 스토리지 등의 리소스를 제공하고 관리하는 기능을 처리함
    - Infrastructure as a Service(IaaS) 클라우드 컴퓨팅 플랫폼을 제어하기 위한 운영체제
  - Amazon EC2, Rackspace Cloud Servers, Eucalyptus와 같은 범주
- 특징
  - REST-based API 제공 : 외부 연동 및 관리의 편의성 제공
  - 하드웨어에 제한이 없음 : Commodity 하드웨어를 이용하여 저렴한 구축 비용
  - 지원되는 Hypervisor : KVM, Xen, XenServer, UML, Hyper-V
  - Asynchronous eventually consistent communication
  - Horizontally and Massively Scalable
  - Amazon EC2 호환 API 및 OpenStack 자체 API 제공

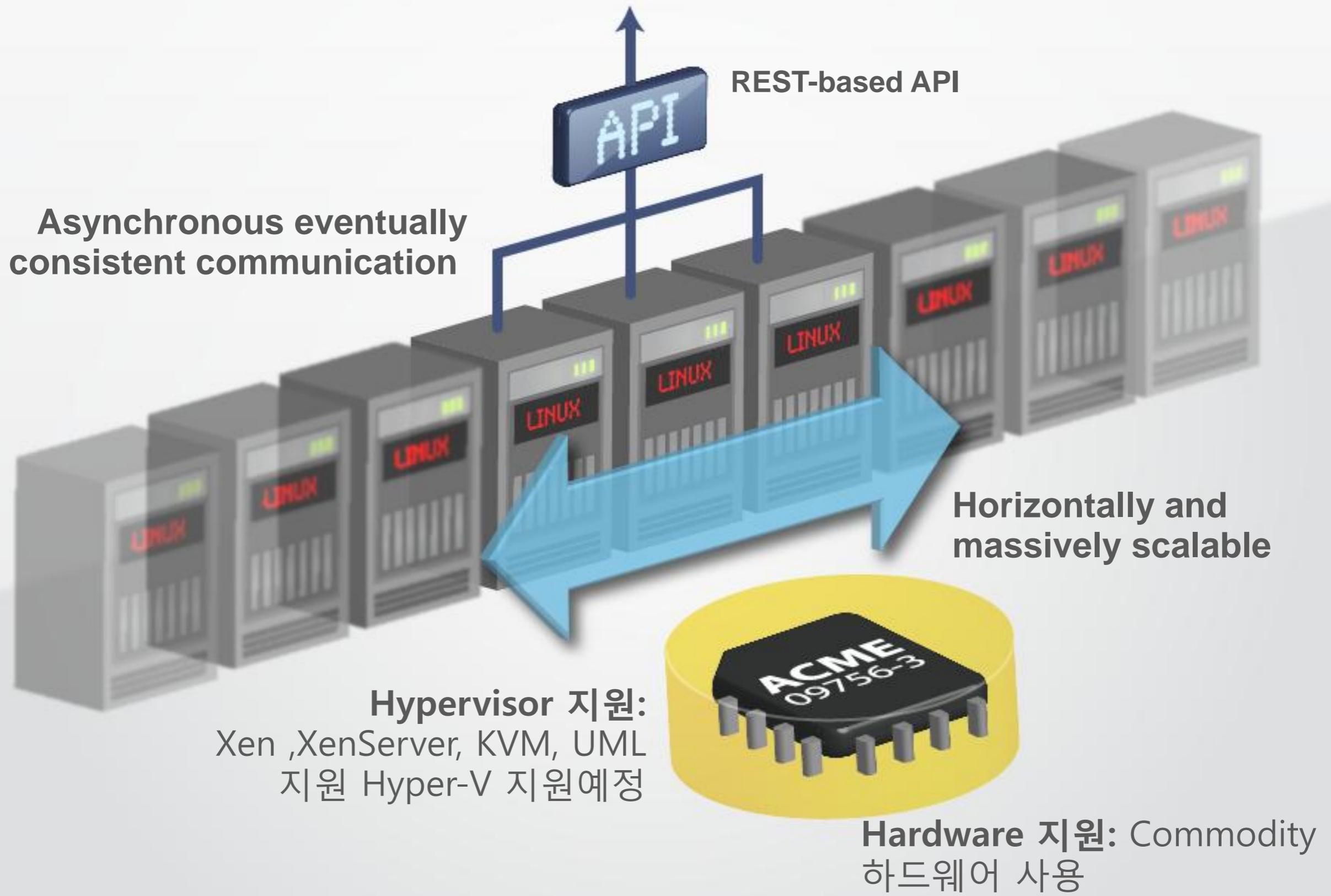


# OpenStack Nova 개발 환경

- 개발 언어 : Python 2.6 기반
- 개발 및 초기 시험 환경 : Ubuntu 10.04 기반
  - ✓ (CentOS 및 Mac 에도 포팅)
- Standard AMQP 메시지 전달 방식 이용
  - ✓ 실제 구현에서는 RabbitMQ 를 채용하여 개발하였음
- Data Base : Open Source 기반의 DBMS 를 이용하여 정보 관리
  - ✓ 실제 구현에서는 MySQL 를 이용 개발되었으며, PostgreSQL, Sqlite 등도 pluggable
- 개발 소스 및 리소스 제공
  - ✓ Launchpad 를 통해 소스 공유 및 버그리포트, QnA 제공
  - ✓ <https://launchpad.net/nova>
  - ✓ 누구나 가입하여 참여가능, 코드 기여(contribution 가능)



# OpenStack Compute 주요 특징



REST-based API

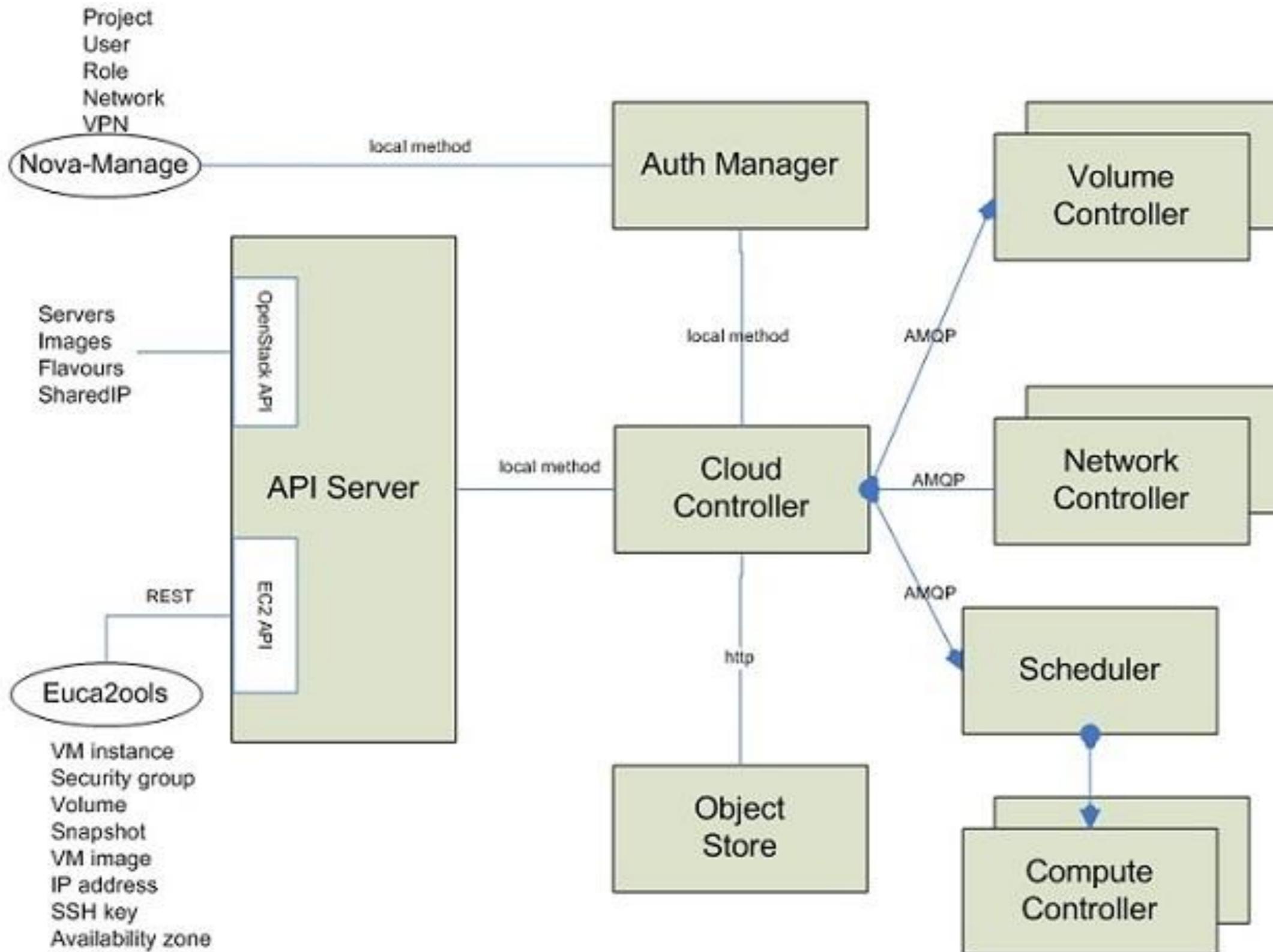
Asynchronous eventually consistent communication

Horizontally and massively scalable

Hypervisor 지원:  
Xen, XenServer, KVM, UML  
지원 Hyper-V 지원예정

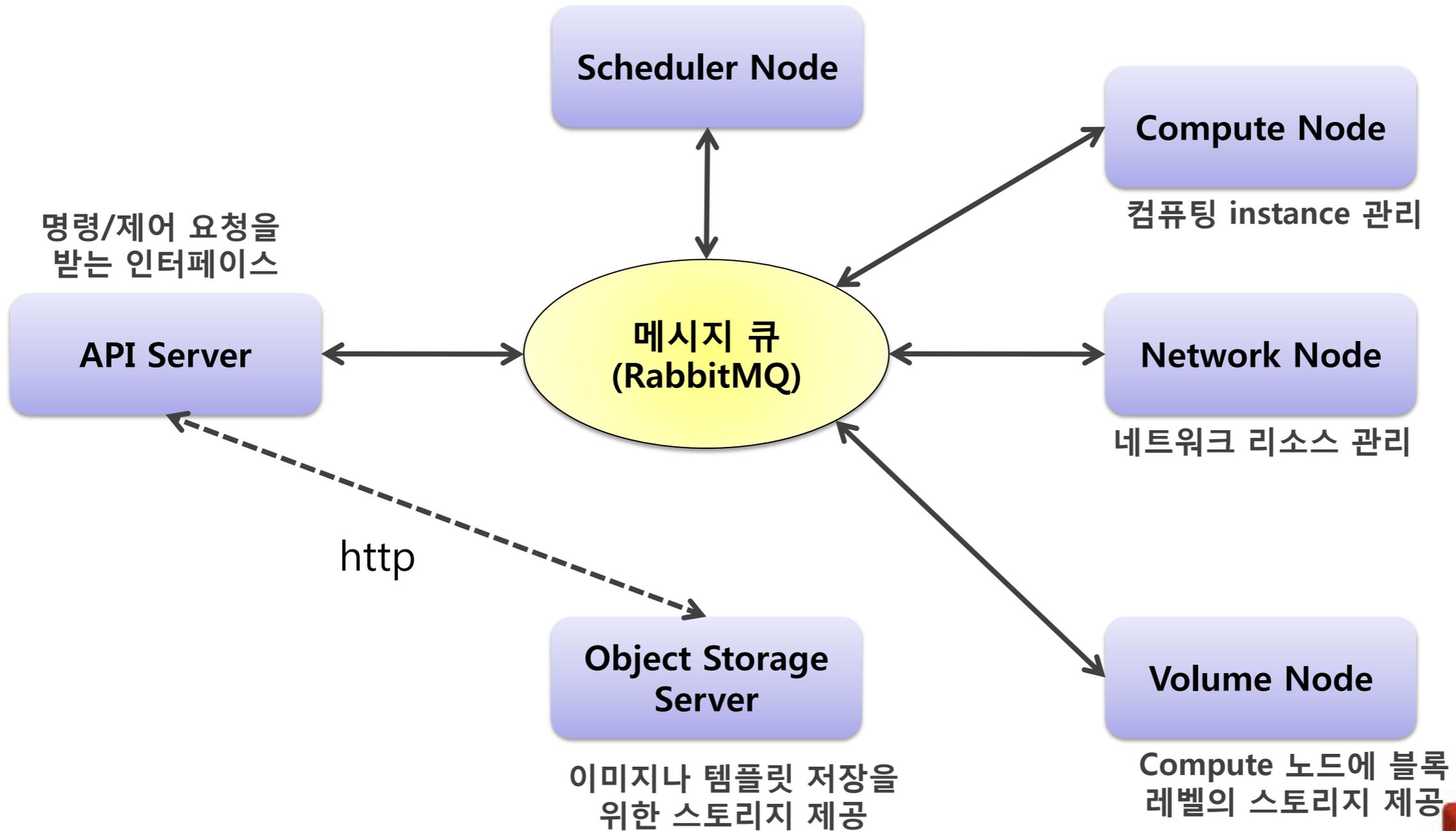
Hardware 지원: Commodity  
하드웨어 사용

# NOVA 소프트웨어 구조



# NOVA 프로세스 구조

요청된 리소스를 제공할  
노드 선택 및 VM 관리 지시



# NOVA 프로세스 구조

## ▶ 메시지 큐

- ▶ Nova 내의 구성 요소들 상호간의 통신을 담당
- ▶ AMQP 방식의 메시지 교환, 현재는 RabbitMQ로 구현되었음

## ▶ API Server

- ▶ 사용자가 명령/제어를 받기 위한 기본적인 HTTP 웹 서비스 형태의 인터페이스 제공
- ▶ 다양한 API를 수용할 수 있도록 모듈 형태로 구현됨
- ▶ 현재 Amazon EC2 호환 API 와 OpenStack API (Rackspace 와 유사) 지원

## ▶ Scheduler

- ▶ 요청된 컴퓨팅 리소스를 실제로 제공할 노드를 선택
- ▶ 현재 Round-Robin, Random 두 가지 방식 지원



# NOVA 구조

- ▶ **Compute Node**

- ▶ Computing instance들을 관리함
- ▶ VM run, terminate, reboot / Volume attach, detach / Console output

- ▶ **Network Node**

- ▶ 네트워크 리소스를 관리
- ▶ 현재 Flat, Flat DHCP, VLAN 모드 지원

# NOVA 의 네트워크 제공 구조

## ▶ Flat 모드

- ▶ 가장 간단한 네트워킹 구조
- ▶ 각 VM 인스턴스는 IP Pool 에서 고정 IP 를 할당받는 방식
- ▶ VM 이 부팅될때 네트워킹 구성정보가 VM 인스턴스로 삽입(injecting) 됨

## ▶ Flat DHCP 모드

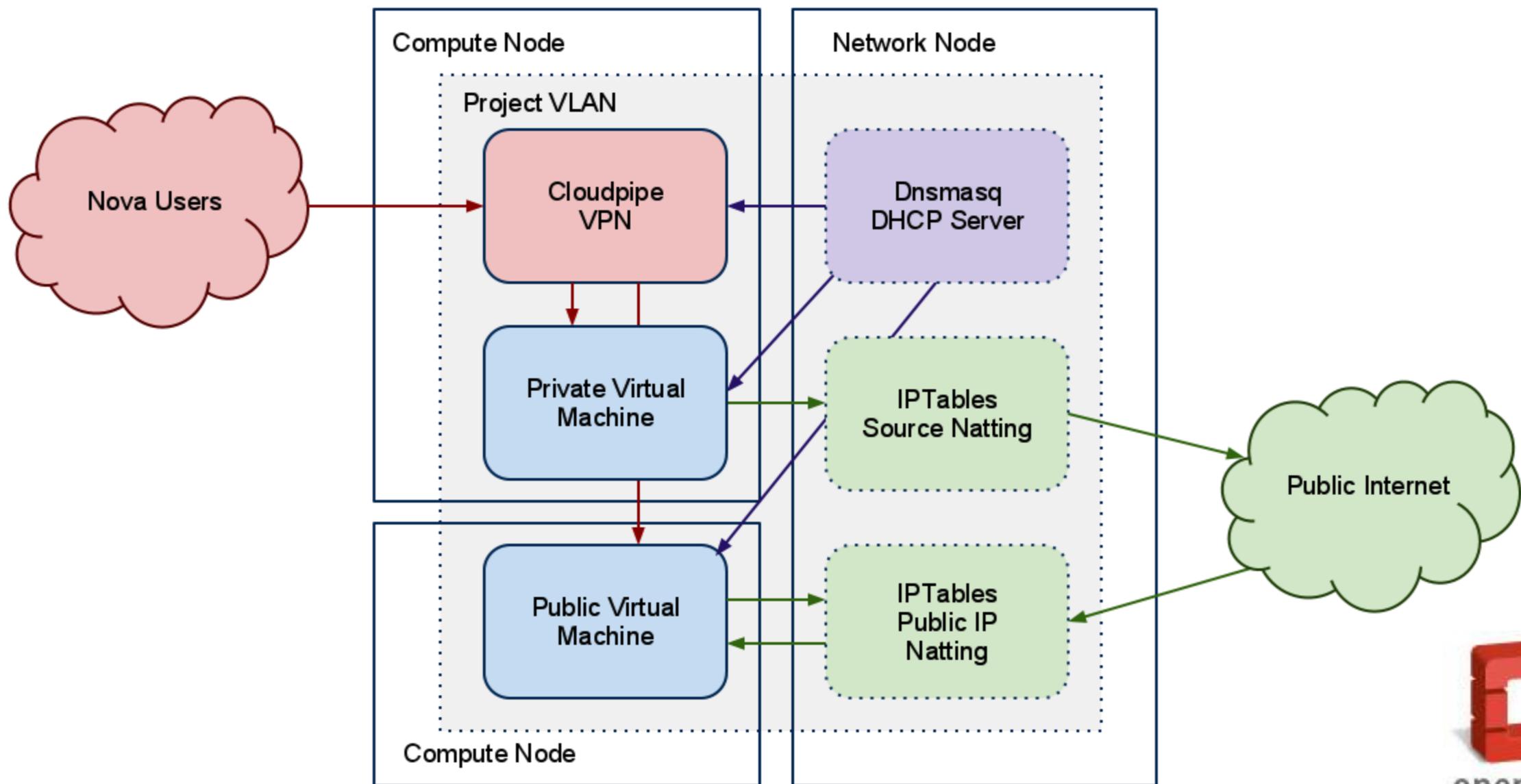
- ▶ Flat 모드와 유사하나, VM 이 할당 받는 IP 는 DHCP 처리

## ▶ VLAN DHCP 모드

- ▶ Default networking 모드이며, Nova가 제공 모든 네트워킹 기능 사용 가능
- ▶ 각 프로젝트 마다 별도의 VLAN 을 생성하여 제공
- ▶ 각 프로젝트 마다 VM 에 접근가능한 VPN 제공. OpenVPN 이용.

# NOVA 의 네트워크 제공 구조

## ▶ VLAN DHCP 모드의 네트워크 구조



# OpenStack Object Storage 상세내용

Software to reliably store billions of objects distributed across standard hardware



# OpenStack Storage 주요 특징

REST-based API



데이터는 시스템들상에 분산됨.

수억개의 Object,  
Petabyte까지 확장등의  
Scalability



Account/Container/Object 구조  
(파일시스템이 아님)

Replication (Account, Container,  
Object들에 대한 N개의 copy)



No central  
database



Hardware agnostic: commodity  
hardware, RAID not required

# OpenStack Object Storage 개요

## ▶ OpenStack Object Storage (Swift)

- ▶ Object Store: Petabyte 단위의 storage에 대한 clustering and management를 위한 소프트웨어와 아키텍처

### ▶ Function:

- ▶ Reliable, redundant, and large-scale storage of static object
- ▶ Ensuring data replication and integrity across the cluster
- ▶ Scaling horizontally by adding new nodes automatically configured
- ▶ Node fail over

## ▶ OpenStack Object Storage (Swift)

- ▶ commodity server들을 모아서 안정적, 중복성, 대규모 스토리지 서비스 제공
- ▶ 객체들이 여러 클러스터 내 하드웨어 장치에 분산 중복 저장되고 이들간에 무결성을 제공.
- ▶ 새로운 노드가 추가되면 자동적으로 구성되고, 오류 발생에 대비하여 타 액티브 노드에 중복
- ▶ OpenStack은 소프트웨어 기반으로 데이터 중복 및 장치간 분산을 구현하므로, 저렴한 서버 및 하드 드라이브로 실현 가능하고 값비싼 장비가 필요 없음.



# OpenStack Object Storage 개요

## ▶ Object Storage

- ▶ Highly available, distributed, eventually consistent<sup>1</sup> object/blob<sup>2</sup> store
- ▶ Storage-as-a-Service로서 분산된 스토리지 자원들을 이용하여 사용자 (사람, 응용)에게 파일 업로드, 다운로드 등 스토리지 서비스 제공
  - ▶ GET, PUT, DELETE, COPY 등 simple storage service를 제공
- ▶ Open source로서 제공되며 모든 API는 REST로 구현되어 있어 사용자 및 개발자들이 쉽고 편리하게 접근

\* Eventually consistent: 분산 시스템에서, 중복 데이터 중 하나를 수정시, 즉시 반영하지 않고 차후에 처리하여, 결국은 일관성을 유지하도록 하는 효율화 개념

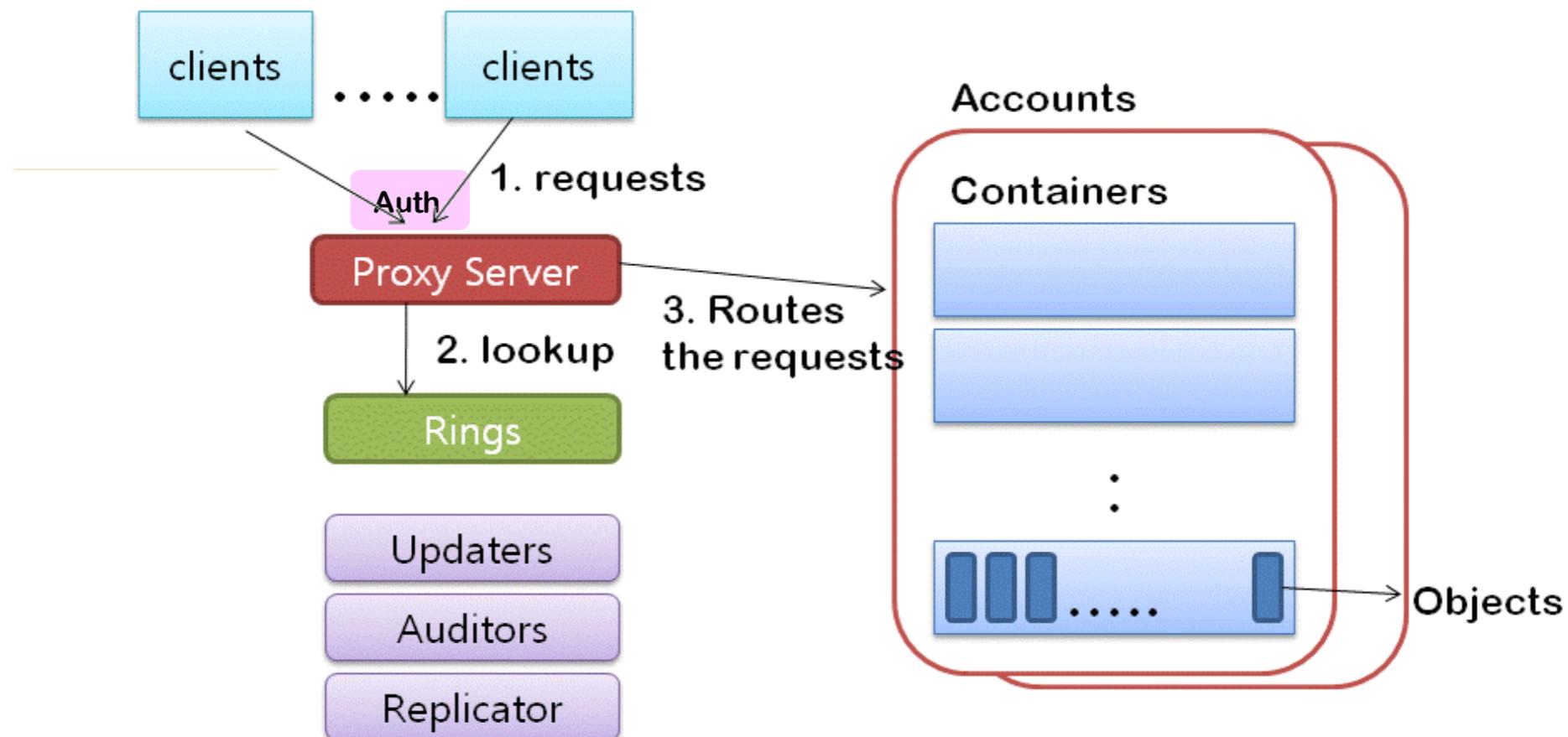
\* Blob (binary large object): 이미지와 같은 디지털 데이터로서 비 정형적, 큰 크기의 객체



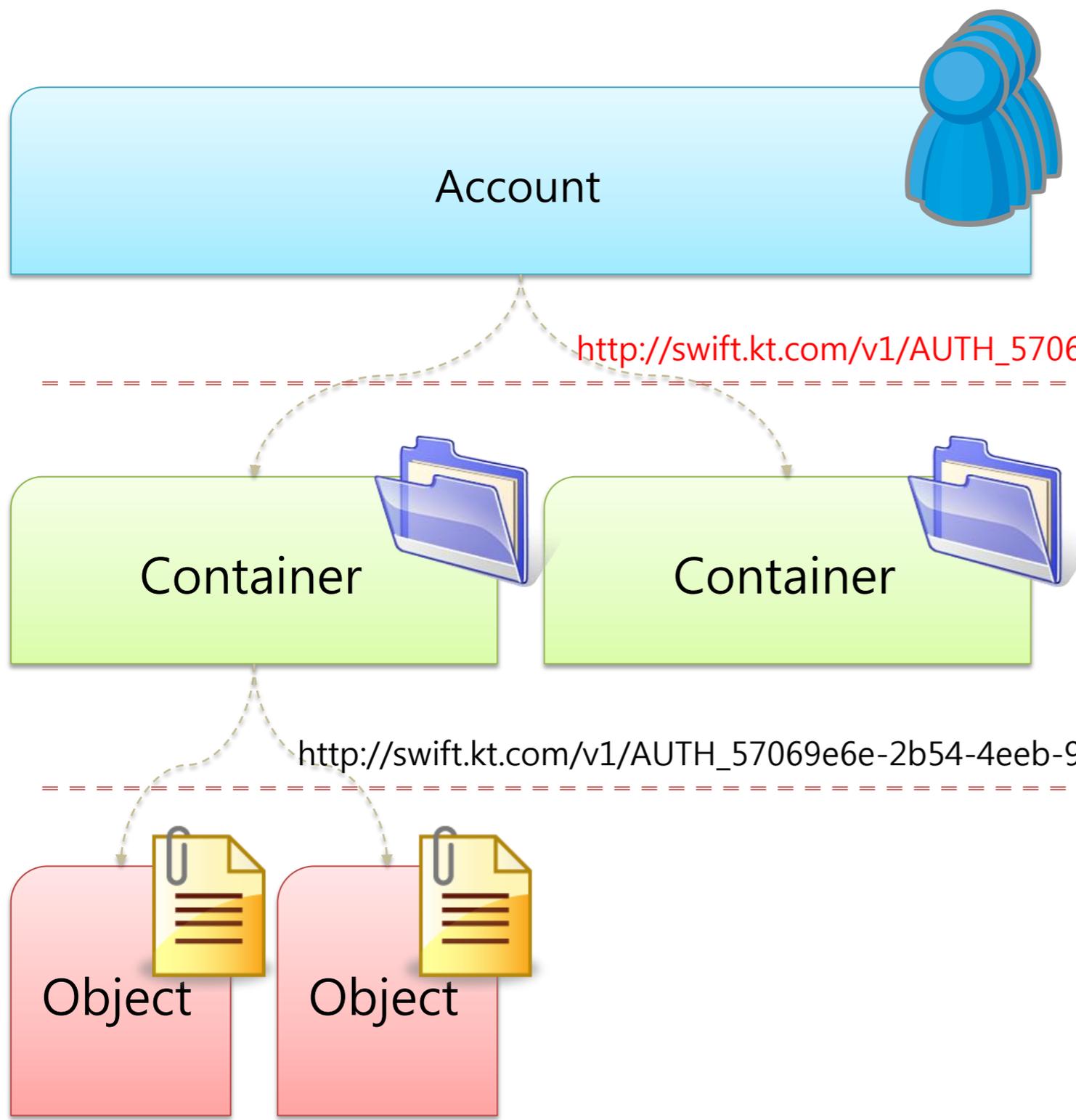
# OpenStack Object Storage - 구조 및 기능

## Swift 구조 및 기능

- ▶ **Auth middleware:** 사용자 생성 및 로그인, 인증 토큰을 생성하고 관리
- ▶ **Proxy Server:** 사용자의 요청에 따라 알맞은 서버에 연결하여 서비스를 제공
- ▶ **Account Server:** 사용자 계정관리 및 계정 별 컨테이너 조회
- ▶ **Container Server:** 사용자 계정의 컨테이너를 관리하는 서버. 계정하의 컨테이너들을 관리하며 컨테이너가 가지고 있는 오브젝트들을 조회
- ▶ **Object Server:** 컨테이너 내의 오브젝트들을 관리하는 서버. 각각의 오브젝트를 실질적으로 저장하며 조회



# Account, Container, Object



- 사용자를 대표하는 URL을 할당 받는 user 그룹
- Container들을 관리하는 객체
- Account안에는 여러 user가 존재가능하며 사용자는 Account:user를 아이디로서 암호를 갖고 token으로서 구분 가능

[http://swift.kt.com/v1/AUTH\\_57069e6e-2b54-4eeb-95a9-1eff53637978](http://swift.kt.com/v1/AUTH_57069e6e-2b54-4eeb-95a9-1eff53637978)

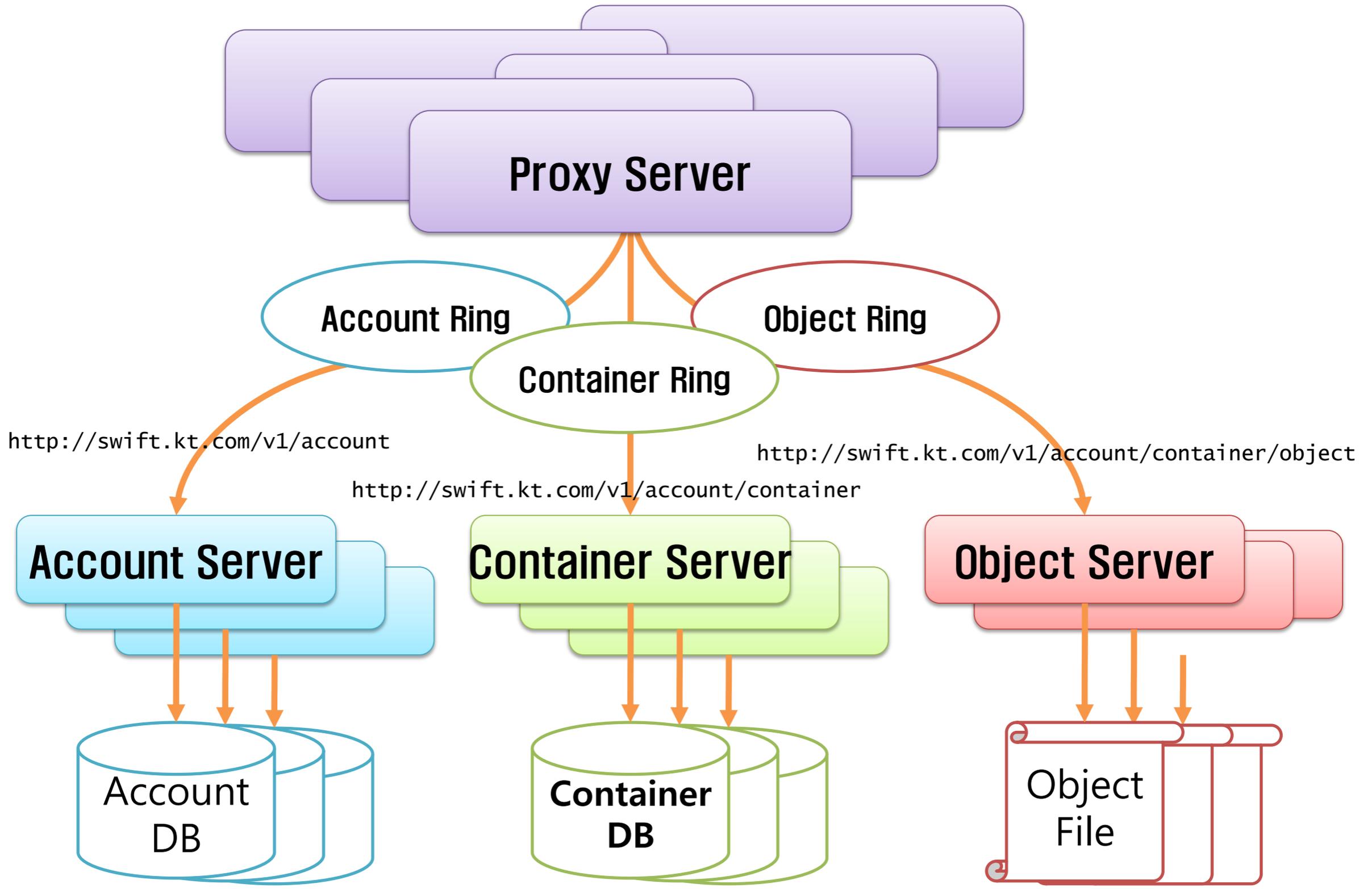
- 폴더와 같이 Object를 관리하는 객체 (ex. Amazon S3의 Bucket)
- user들은 Access Control List(ACL)를 통해 Container의 읽기 쓰기 권한을 관리

[http://swift.kt.com/v1/AUTH\\_57069e6e-2b54-4eeb-95a9-1eff53637978/TESTCONTAINER](http://swift.kt.com/v1/AUTH_57069e6e-2b54-4eeb-95a9-1eff53637978/TESTCONTAINER)

- 실질적으로 저장되는 파일 오브젝트
- Container의 ACL에 적용되어 관리
- Object는 파일 뿐만이 아닌 폴더형태로도 저장가능 (즉, 폴더 구조 가능)

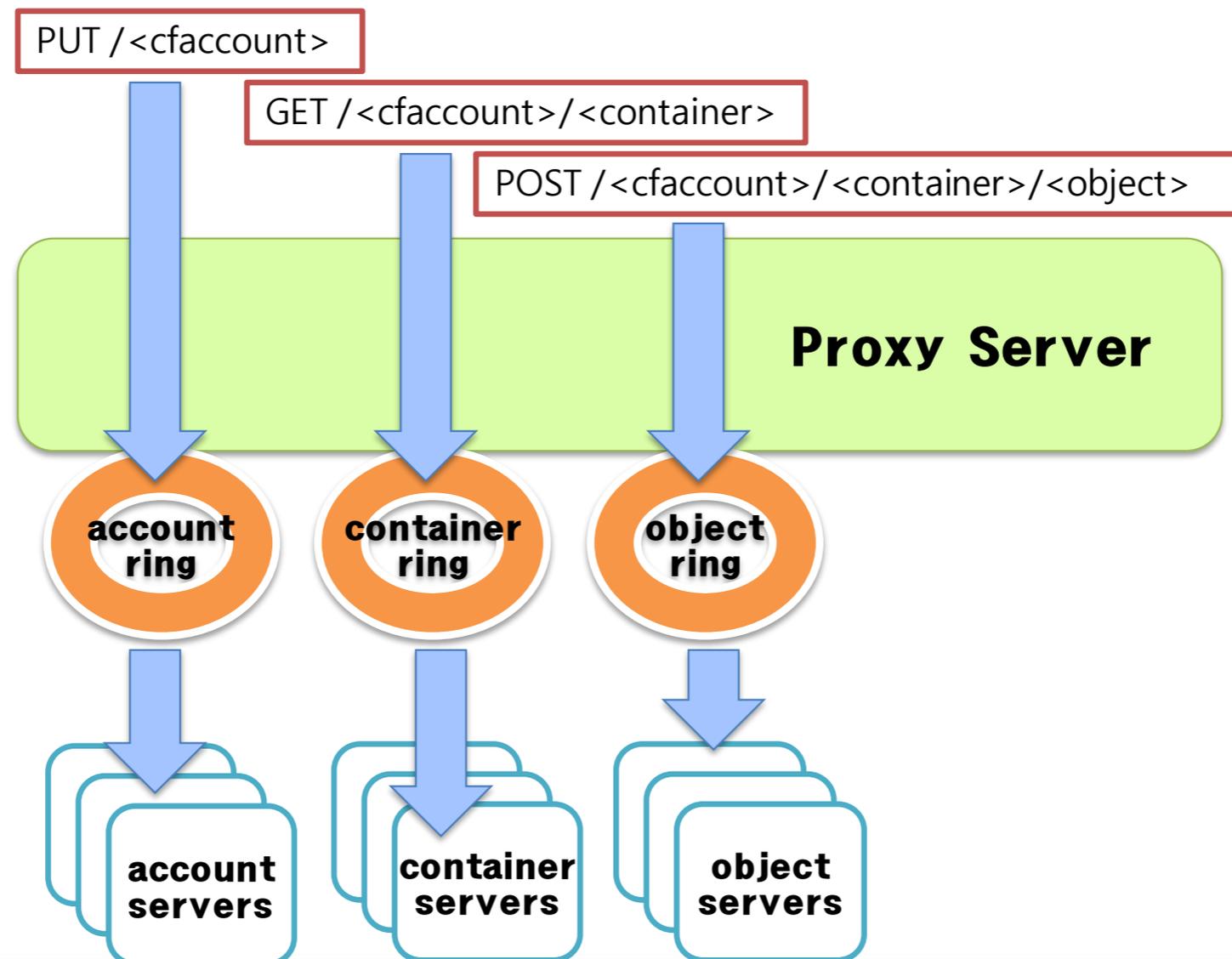
[http://swift.kt.com/v1/AUTH\\_57069e6e-2b54-4eeb-95a9-1eff53637978/TESTCONTAINER/TESTFILE.txt](http://swift.kt.com/v1/AUTH_57069e6e-2b54-4eeb-95a9-1eff53637978/TESTCONTAINER/TESTFILE.txt)

# 구성도



# RING

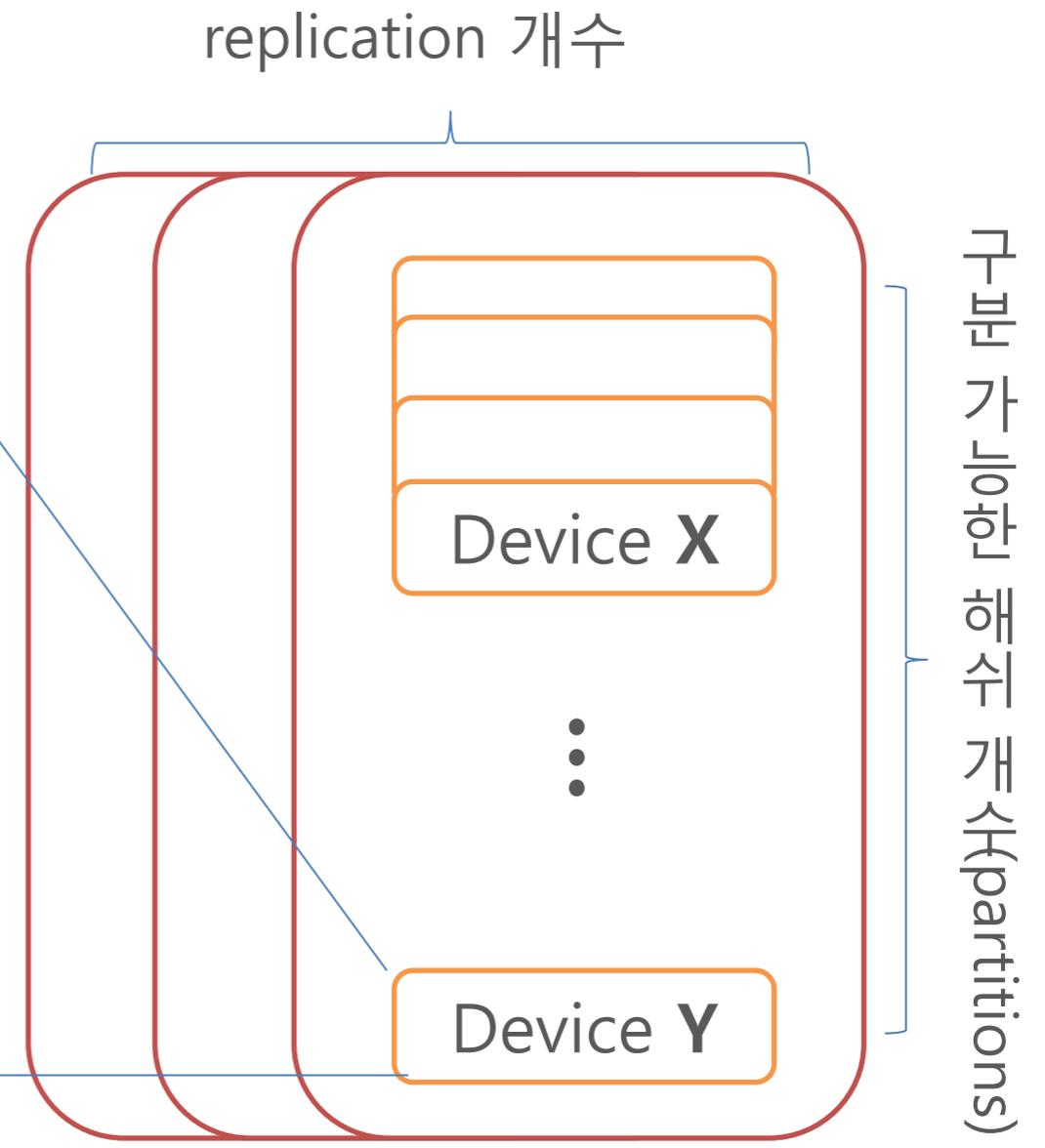
- › 데이터의 논리적 위치와 물리적 저장 위치간의 매핑 제공
- › 경로 값의 MD-5 hash bits를 사용해서 정보저장 위치 계산
- › Account, Container, Objects Ring 존재 (동일 구조)
- › proxy 서버에서만 접근. ring의 생성은 관리자가 직접 수행
- › Fail 시 handoff device 를 결정



# RING 구조

**Device Y**  
(A,C,O 서버에 존재하는 실제 폴더로서 분산의 단위)

- **ID** : 각 디바이스를 구분하기 위한 번호
- **Zone number** : 해당하는 zone
- **IP** : 머신의 IP 주소
- **Port** : 머신의 PORT 주소
- **Device Name** : 머신의 접근할 폴더 이름
- **Weight** : 디바이스의 비중
- **Meta** : 디바이스에 추가적으로 입력할 내용



# Replication, Updater, Auditor

## ▶ Replication

- ▶ Network 파티션 또는 드라이버 failure와 같은 일시적 error 발생 시 시스템의 안정성 유지
- ▶ Local data와 remote copy의 비교를 통해 push형태로 이루어짐
- ▶ Object replication: hash list
- ▶ Container / Account replication: combination 방식(hash list & high water marks)
- ▶ Tombstone(object, container, account의 최신 버전 관리)을 통한 데이터 삭제관리

## ▶ Updaters

- ▶ High load시 발생할 수 있는 Update 실패에 대한 대비 프로세스로서 update fail 발생 시 파일시스템에 update를 저장하고 updater가 실패한 updates를 실행함으로써 consistency window를 보증

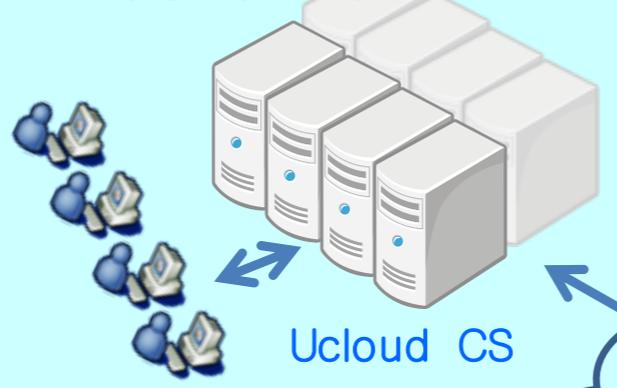
## ▶ Auditors

- ▶ Local server상에서 object, container, account들의 무결성을 검사하며, 이상 발생시 파일을 격리하고 replication으로 대체

# 적용 서비스 사례

## 사용2: 컴퓨트 클라우드의 백업 스토리지

- ▶ 가상머신들의 이미지 archive 용으로 사용
- ▶ 가상머신 사용자들의 2nd 스토리지로 사용



## 사용1: 개인 또는 기업 고객이 스토리지로 사용

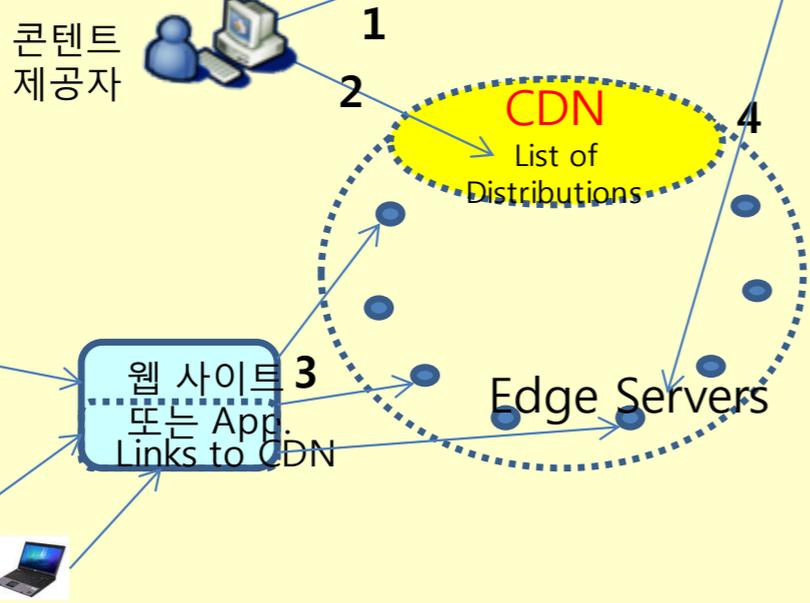
- 1. http를 통한 파일 업로드
- 2. http를 통한 파일 다운로드



고객

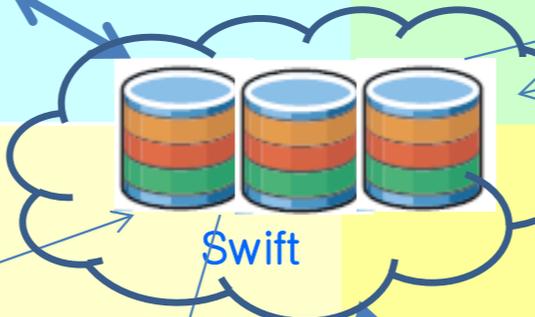
## 사용3: CDN origin server로 사용

1. 콘텐츠 제공자가 SS에 콘텐츠 업로드
2. 콘텐츠 제공자가 distribution list 생성



## 사용4: Open API를 통한 3rd party 솔루션

- ▶ 3rd party 솔루션 개발자들이 Swift API 를 통해 솔루션에 붙여 사용



# OpenStack Object Storage APIs

## Authentication API

- .account 생성
- .account 조회
- .account 삭제
- .user 생성
- .user 조회
- .user 삭제
- .사용자 인증처리 (token, storageUrl)

## Storage Account API

- .Container 리스트, 수량 조회
- .Account 전체 사용량 조회
- .Account 사용자 metadata 추가

## Storage Container API

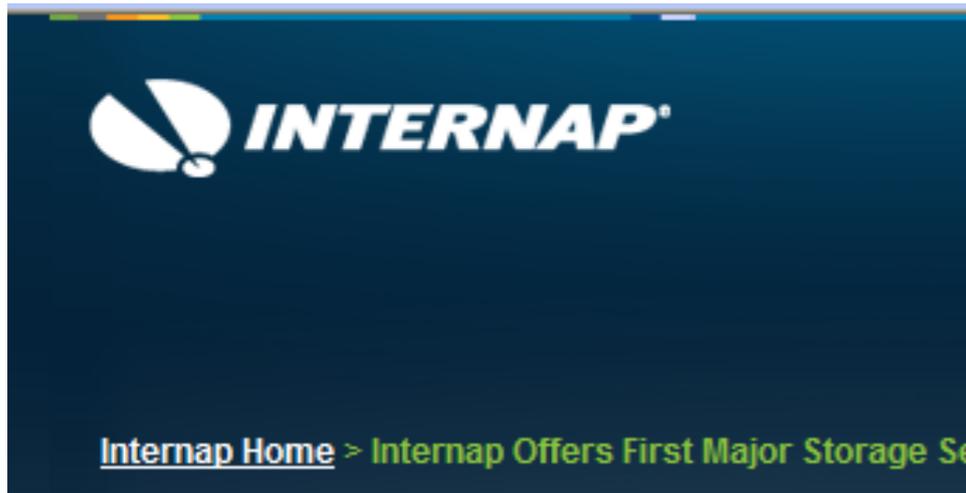
- .저장된 object 리스트, 수량 조회
- .Container 전체 사용량 조회
- .Container 생성 및 삭제
- .Container 사용자 metadata 추가
- .Container ACL 설정

## Storage Object API

- .object 정보조회
- .object 업로드 및 다운로드
- .object 삭제
- .object 사용자 metadata 추가

- ▶ Swift에서 제공하는 API는 ReSTful 기반의 web API로 구현되어 있음
- ▶ 제공되는 API는 Authentication, Account, Container, Object에 대한 다양한 기능 제공
- ▶ CDN 연동을 위한 ACL 기능과 사용자 정의 metadata 추가를 위한 기능이 API로 제공되고 있음

# OpenStack Object Storage 사업화 사례 (Internap, NephoScale )



[Internap Home](#) > [Internap Offers First Major Storage Service](#)

## Internap Offers First Major OpenStack (CIO)

January 18, 2011

Internap is offering a new public cloud storage service as the open-source cloud software project kicked off.

...Users will be able to log into an online portal to scaling the services up and down as needed. The to move content in and out of storage without the said. [Read more of the CIO article.](#)

Share @Internap



## NephoScale Joins OpenStack Community in Support of Open and Interoperable Cloud Services



Buzz up! 0 votes

Share

retweet 3

Email

Print

– Wed Feb 16, 12:15 pm ET

### RELATED QUOTES

^DJI	12,391.25	+73.11
^GSPC	1,343.01	+2.58
^IXIC	2,833.95	+2.37

AdChoices



NephoScale's IaaS storage offering, built upon OpenStack Object Storage architecture, is highly redundant and scalable to petabyte levels.

San Jose, CA (PRWEB) February 16, 2011

Cloud services provider NephoScale is fostering innovation and interoperability in the global cloud community by formally joining OpenStack™, an open source cloud project with more than 50

participating organizations working to drive industry standards. Having recently launched in January, NephoScale's object-based Cloud Storage service is the first generally available offering, outside of Rackspace, to be built using the OpenStack storage system.

"When we initially sat down in mid-2010 to research cloud storage systems for our IaaS platform, we evaluated several options and technology paths, and OpenStack was the logical choice for us for several reasons," said Telemachus Luu, CTO of NephoScale. "It met our requirements for object-based storage, scalability and redundancy, and was hands-down the fastest way for us to develop a core infrastructure that we could build upon."

NephoScale's infrastructure services, which include cloud servers and on-demand dedicated servers, in addition to cloud storage, were designed for true global scalability and fault tolerance. The company used OpenStack Object Storage as the framework for its Cloud Storage service and added value by layering on management functions that were built in-house, including custom authentication, auto-scaling, and storage inventory control. By using clusters of commodity servers, its Cloud Storage can store large amounts of data and replicate objects several times.

# OpenStack Release Process: Four Phases



\*Design phase and Design Summit occur every other release, 2x per year



# OpenStack Releases

Austin:  
2010 10월

- OpenStack Object Storage 상용화 가능 버전
- OpenStack Compute developer preview, 테스트와 PoC (proofs of concept) 가능 버전

Bexar:  
2011 2월

- OpenStack Compute: 엔터프라이즈급 private 클라우드 및 중간 사이즈급의 public 서비스 구축 가능 버전
- 향상된 Documentation
- 용이한 설치 및 구축

Cactus:  
2011 4월

- OpenStack Compute: Large 스케일의 Public 클라우드 서비스 구축 가능 버전



# OpenStack Compute 'Bexar' Release Features

- Object Storage
  - Large objects (greater than 5 GB) ; client-side chunking and segmentation now allows virtually unlimited object sizes, limited only by the size of the cluster it is being stored into
  - Experimental S3 compatibility middleware
  - Swauth authentication and authorization service on top of Object Storage
- Compute
  - Support for raw disk images for hypervisors that are libvirt compatible (e.g. KVM) and XenAPI
  - IPv6 support in all network modules but FlatManager (coming in Cactus)
  - Support for new virtual volume backends to provide highly available block volumes for virtual machines: Sheepdog, CEPH/RADOS, and iSCSI (XenApi only)
  - Microsoft Hyper-V hypervisor is supported
  - Updated OpenStack API – admin features to pause, suspend, lock, and password reset instances
  - New “rescue” mode allows an instance to mount affected disks and fix problems



# Object Storage 'Bexar' Release Features

- ▶ Compute (cont'd)
  - ▶ Web-based serial console to access instances when networking fails is available through the OpenStack API
  - ▶ Database versioning and migration support
  - ▶ Instances now use copy-on write by default for better performance
  - ▶ Support for availability zones through new scheduler: ZoneScheduler
- ▶ Glance (Image Registry)
  - ▶ Registry and Delivery APIs were unified; specific client class created
  - ▶ Support for uploading disk images thru Glance REST-full API
  - ▶ Glance-upload tool can register new AMI-like images or raw disk images
  - ▶ Fetch image data on S3-like backend as well as from Object Storage
  - ▶ Documentation at <http://glance.openstack.org>



# Join Us



[stephen.spector@openstack.org](mailto:stephen.spector@openstack.org)



<http://openstack.org>

<http://www.openstack.or.kr>  
(한국 커뮤니티)



<http://openstack.org/blog>

twitter



<http://twitter.com/openstack>



Freenode :  
#openstack

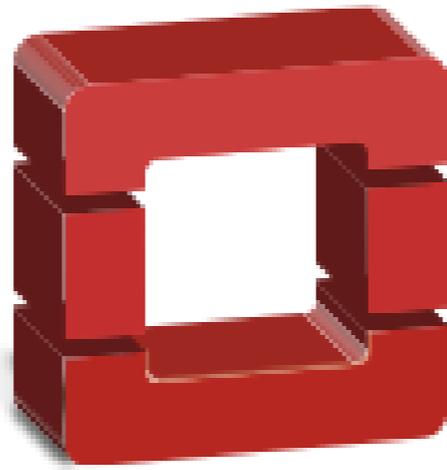


LinkedIn

flickr



openstack  
CLOUD SOFTWARE



openstack™  
CLOUD SOFTWARE