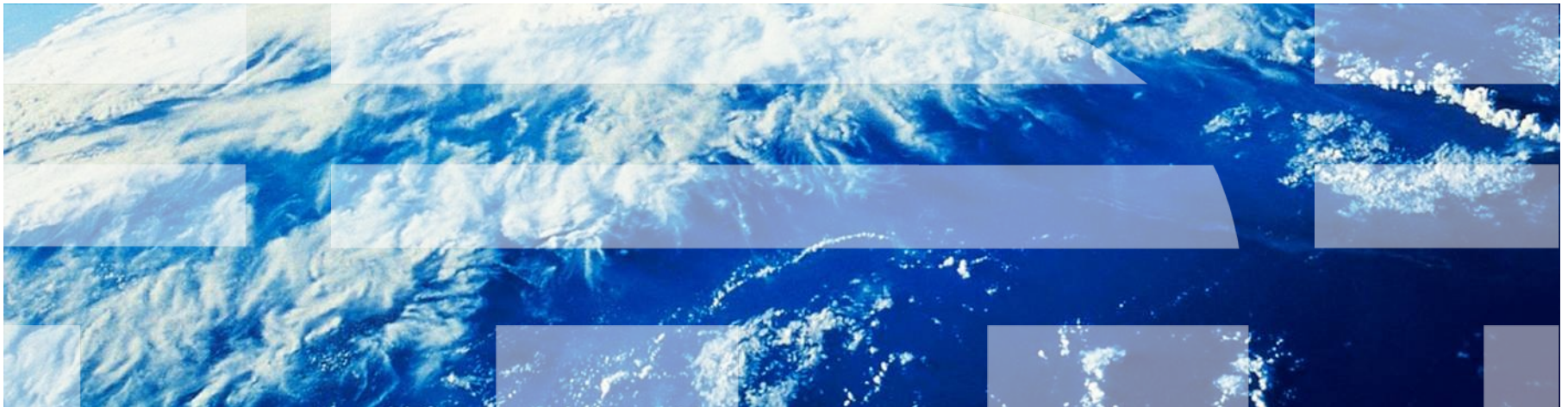




Overview of Openstack





Story Time



openstack™



What it is?

SaaS

Applications

Data

Users are just running the software packages being offered

PaaS

Runtime

Middleware

OS

Deliver a computing platform to the user
User need to install/deploy App & manage Data

IaaS

Virtualization

Servers

Storage

Networking

Delivering the machine resources to the user
User has to manage the compute resource

OpenStack's goal is “essential infrastructure, support platform”

“ Our goal is to produce the ubiquitous Open Source cloud computing platform that will meet the needs of public and private cloud providers regardless of size, by being simple to implement and massively scalable. “

- Open source (Apache 2.0 licensed)
- “Linux of the data center” eliminate vendor lock-in, maintain workload portability
- Build a great engine, packagers make a great car (think Linux to RHEL/SUSE)
- Design Tenets
 - scalability and elasticity are our main goals
 - share nothing, distribute everything (must be asynchronous and horizontally scalable)



Public Cloud

Public Cloud

- Hosted at a Service Provider Site
- Supports multiple customers
- Often utilizes shared infrastructure
- Supports connectivity over the internet
- Suited for information that is not sensitive
- Can be cheaper than private cloud



Private Cloud

Private Cloud

- Hosted at an Enterprise or a Service Provider site
- Supports one customer
- Does not utilize shared infrastructure
- Connectivity over private network/ fiber or the internet
- Suited for information that needs a high level of security

Virtualization Managers and Cloud Infrastructure Services are purpose optimized Infrastructure Management solutions

Virtualization Manager

- Purpose optimized for longer-lived virtual machines managed by Server Administrator
- Centralizes enterprise server virtualization administration tasks
- High degree of flexibility designed to accommodate virtualization all workloads
- Significant focus on managing availability and QoS for long-lived workloads with level of isolation

▪ **Primary attributes derived from enterprise class hardware**

VMControl

VMware vCenter

Cloud Infrastructure Services

- Purpose optimized for shorter-term virtual machines managed via end-user or automated process
- Decentralized control, embraces eventual consistency, focus on making "good enough" decisions
- High degree of standardization and automation
- Significant focus on ensuring availability of control plane

▪ **Primary attributes derived by software architecture**

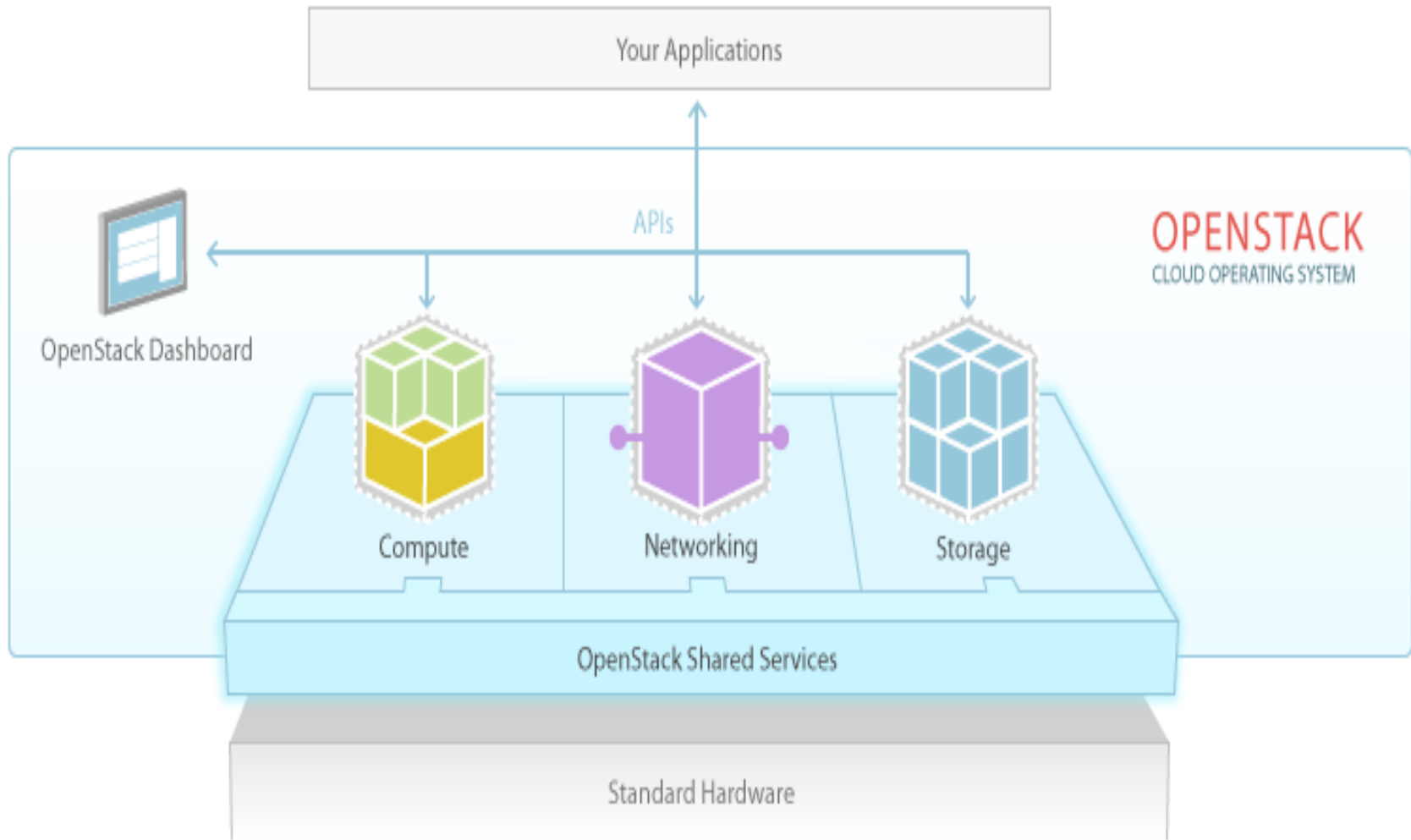
HSLT

OpenStack

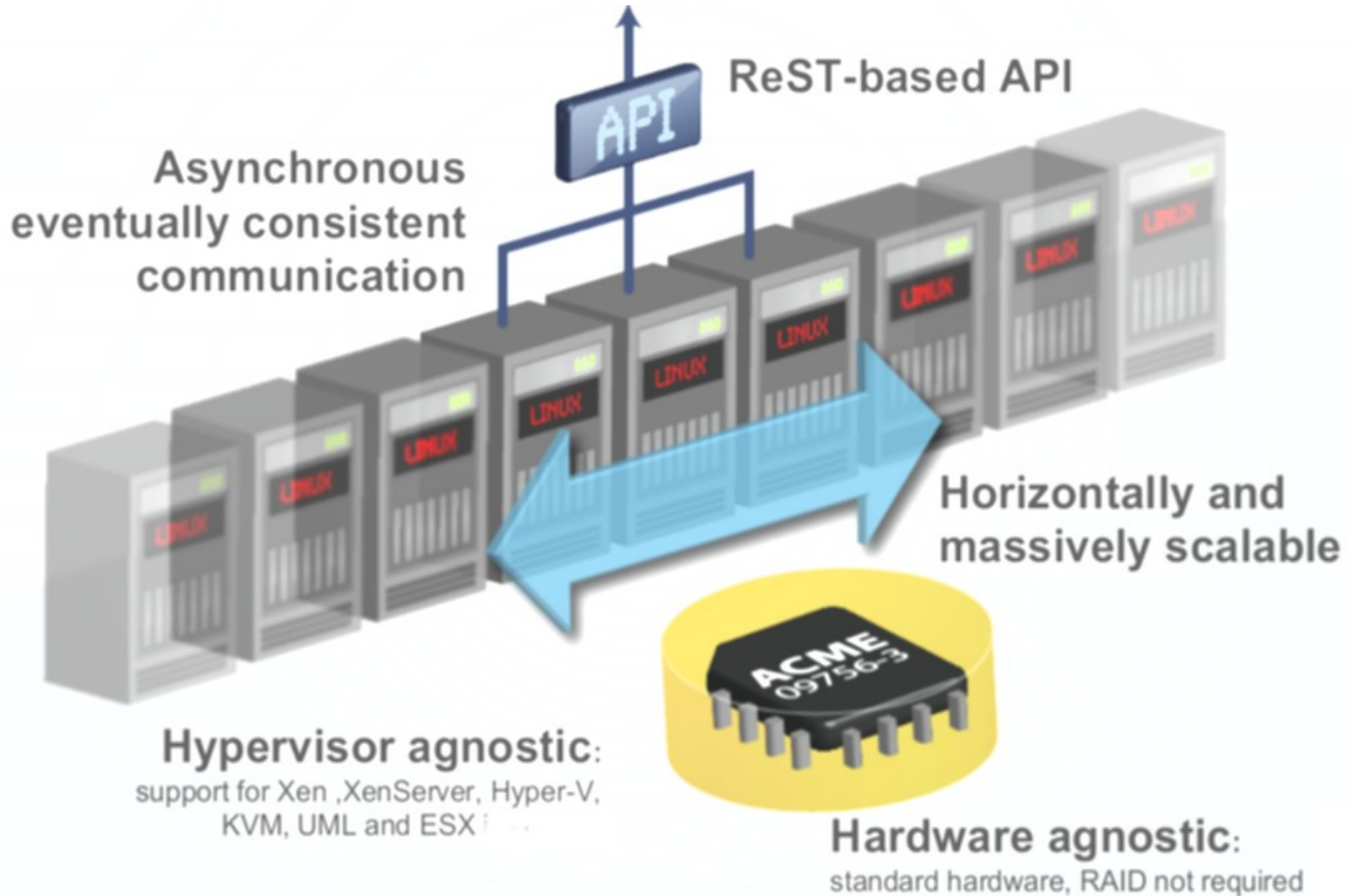
Complementary



Architecture

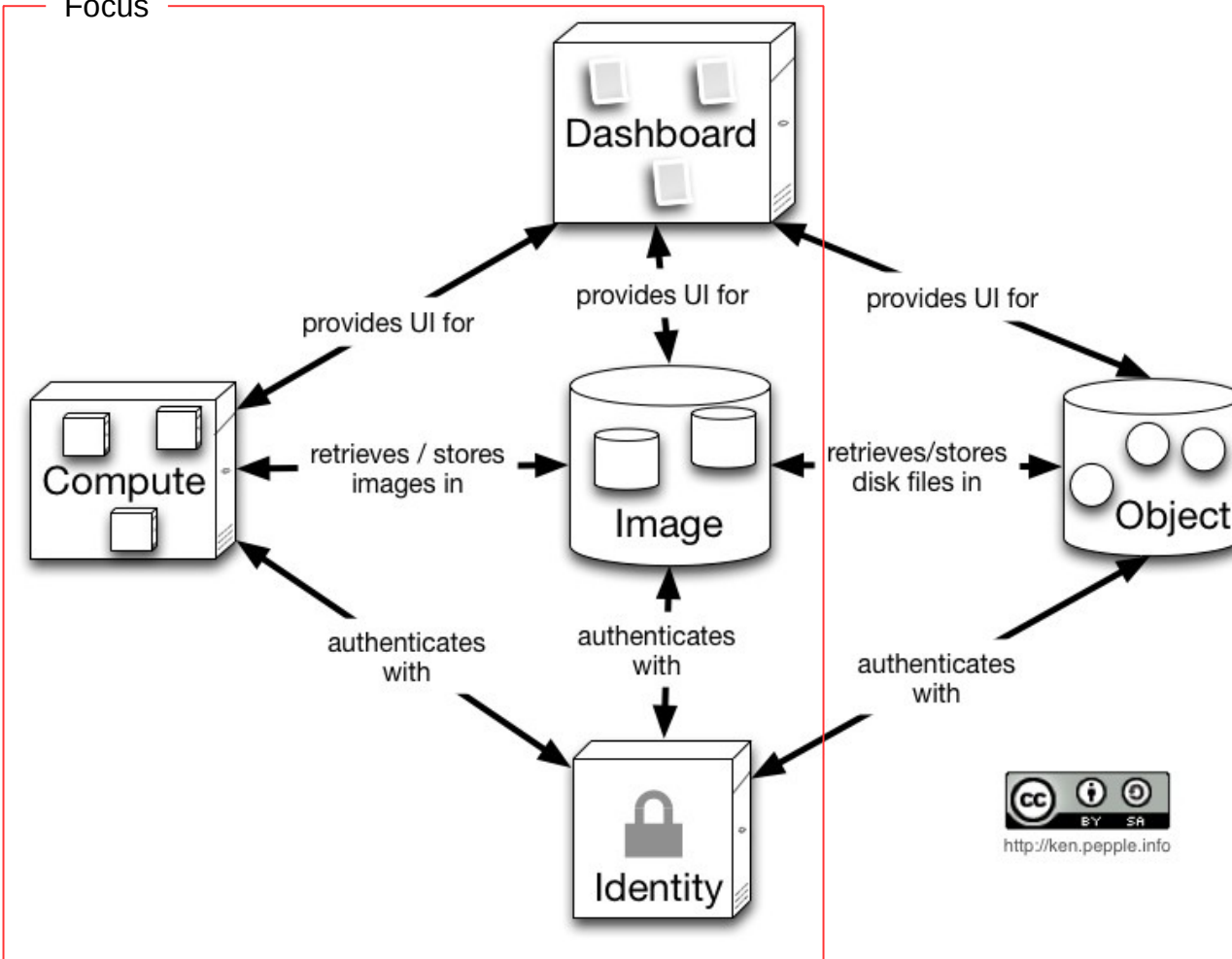


Key Features



OpenStack is comprised of five core projects that form a complete IaaS solution

Focus



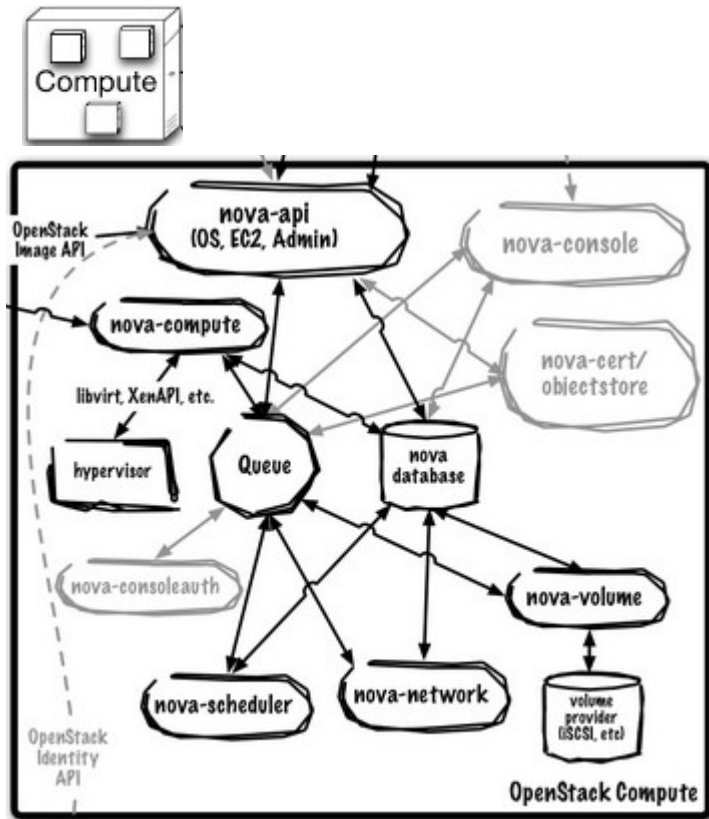
Focus

Compute (Nova)
Storage
 Block – Cinder
 Object – Swift
Network - Quantum
Dashboard (Horizon)
Image (Glance)
Identity (Keystone)

Sources:

<http://ken.pepple.info/openstack/2012/02/21/revisit-openstack-architecture-diablo/>

Nova (Compute) delivers a fully featured, redundant, and scalable cloud computing platform



Core Use Cases:

Manage virtualized server resources (CPU, memory, disk, network)

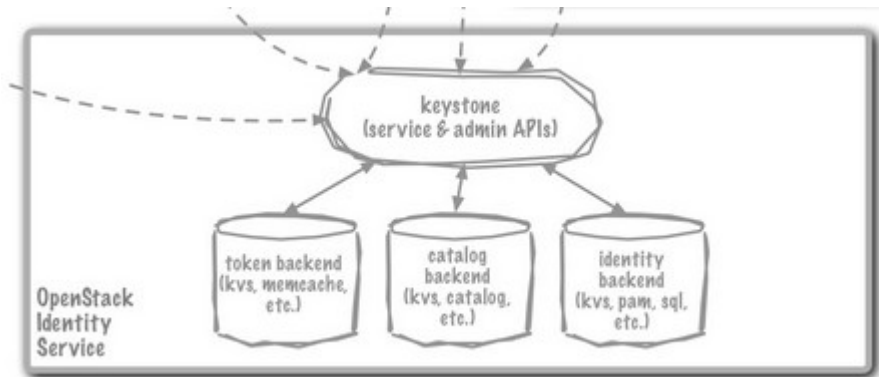
Key Capabilities:

- REST-based APIs with rate limiting and authentication
- Manage Local Area Networks (LAN)
 - Programmatically allocate IPs and VLANs
- Massively scalable and highly available system
 - Distributed and asynchronous architecture
- Live migration of guests
- VM management (Instance)
 - Run, reboot, suspend, resize, terminate instances
- Floating IP addresses
- Security Groups
- VNC Proxy through web browser

Sources:

<http://ken.pepple.info/openstack/2012/02/21/revisit-openstack-architecture-diablo/>
<http://openstack.org/projects/compute/>

Keystone (Identity Service) offers project-wide identity, token, service catalog, and policy service designed for integrate with existing systems



Core Use Cases:

- Authenticate user / password requests against multiple backends (SQL, LDAP, etc) (Identity Service)
- Validates / manages tokens used after initial username/password verification (Token Service)
- Endpoint registry of available services (Service Catalog)
- Authorize API requests (Policy Service)

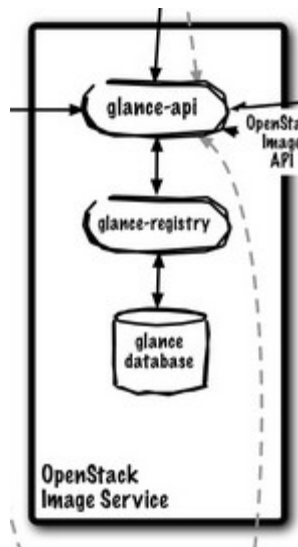
Compare To

- very basic version of TFIM
- pair with LDAP for enterprise-wide repository

Key Capabilities:

- User / Tenant model with Role-Based Access Control
- Policy service provides a rule-based authorization engine and the associated rule management interface.
- Each service configured to serve data from pluggable backend (Key-Value, SQL, PAM, LDAP, Templates)
- REST-based APIs

Glance (Image Service) provides basic discovery, registration, and delivery services for virtual disk images



Core Use Cases:

- Administrator registers available guest images
- End-user discovers available guest images
- Deliver image to compute node on provisioning

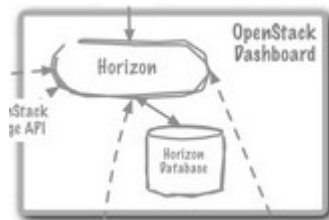
Compare To

- pair with ICON for image building

Key Capabilities:

- think Image Registry, not Image Repository
- REST-based APIs
- Query for information on public and private disk images
- Register new disk images
- Disk images can be stored in and delivered from a variety of stores
- Supported formats: Raw, Machine (a.k.a. AMI), VHD (Hyper-V), VDI (VirtualBox), qcow2 (Qemu/KVM), VMDK (VMWare), OVF (VMWare, others)

Horizon (Dashboard) enables administrators and users to access and provision cloud-based resources through a self-service portal



Core Use Cases:

- End-user self service portal for compute, object storage
- Cloud administration (users/projects, quotas, etc.)

Compare To

- similar to SmartCloud Provisioning UI
- upgrade to SmartCloud Entry for billing, approvals

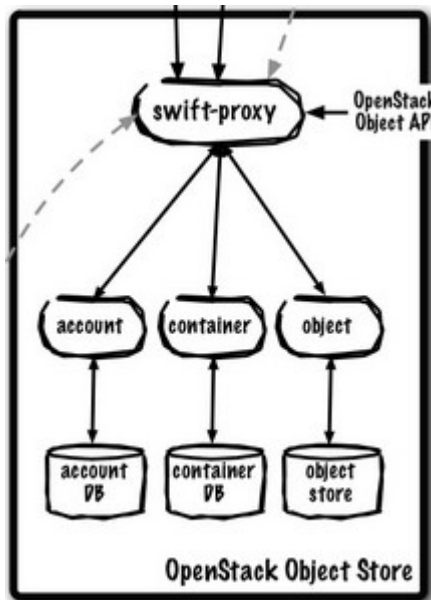
Key Capabilities:

- Thin wrapper over APIs, no local state
- Registration pattern for applications to hook into
- Out-of-the-box support for all core OpenStack projects.
- Anyone can add a new component as a “first-class citizen”.
- Visual and interaction paradigms are maintained throughout.

References

<http://horizon.openstack.org/intro.html>

Swift (Object Storage) provides redundant, scalable storage using clusters of standardized servers to store petabytes of data.



Core Use Cases:

- Content storage / distribution (e.g. documents, web images, thumbnails)
- Backup / archiving (e.g. log files, database dumps)

Solution Components

- similar to Amazon S3

Key Capabilities:

- REST-based APIs, store and manage objects programmatically
- Public and private containers
- Leverages commodity hardware
- HDD/node failure agnostic (self-healing)
- Built-in replication
- Easily add capacity (elasticity)
- No central database
- RAID *not* required (or recommended)



- Database
 - Backup
 - Web content
- Block Storage
 - Object Storage
 - Object Storage

OpenStack Swift

is Object Storage for OpenStack



- Highly Scalable
- Extremely Durable
- High Concurrency

Swift Use Cases

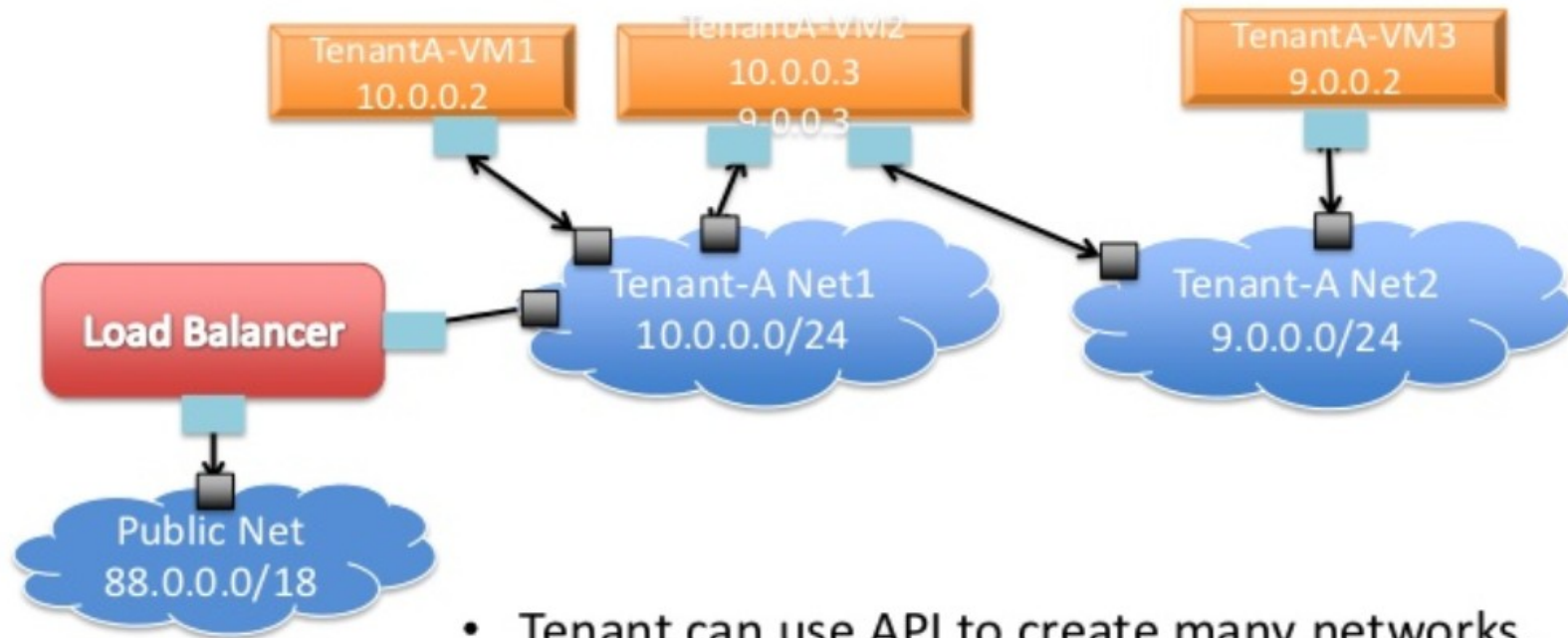
Web/Mobile Applications

Private Filesharing Applications

Data Analytics

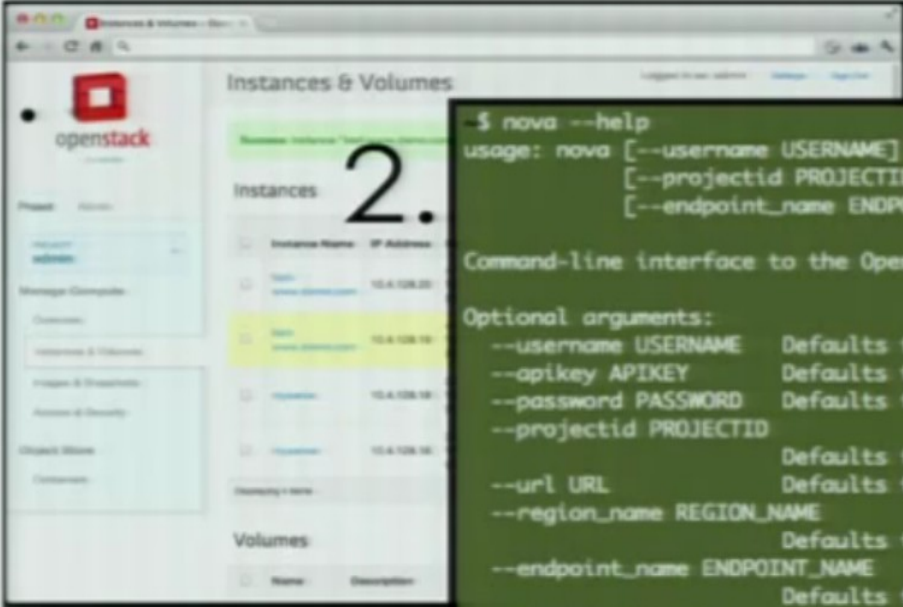
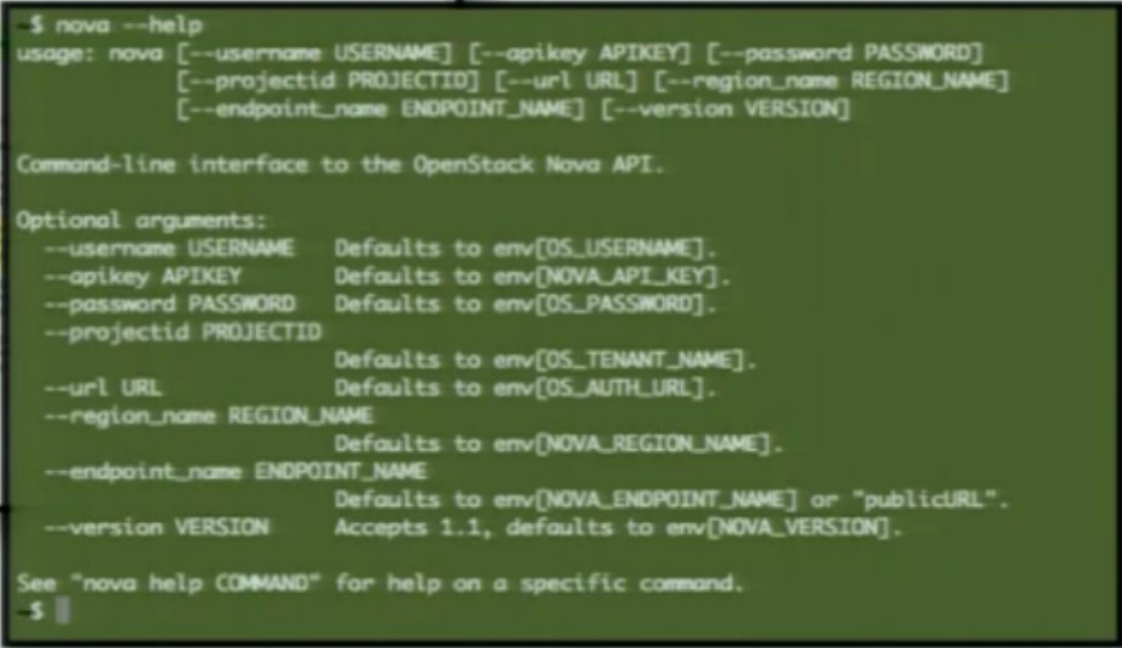
Infrastructure as-a-service

Quantum Model: Dynamic Network Creation + Association



- Tenant can use API to create many networks.
- When booting a VM, define which network(s) it should connect to.
- Can even plug-in instances from other services (e.g., a load-balancing service).

3 Interfaces

1.  2. 

3. `> curl http://mycloud/v2.0/glance/image007`

Getting Started

Use it:



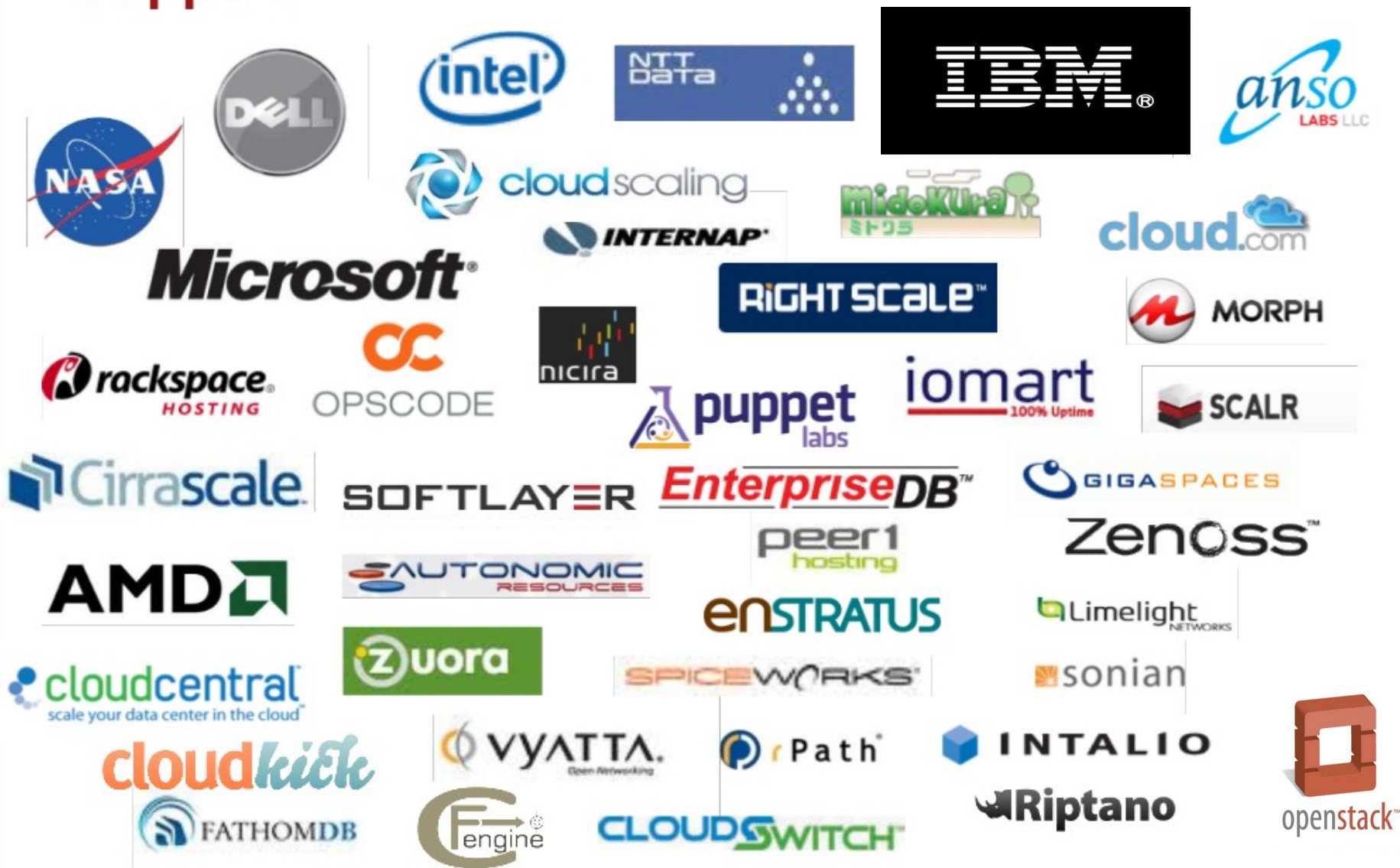
Run it:



Build it:



Community with Broad Commercial Support



Questions?

References:

Openstack Design Summit

Openstack 101 – Joshua McKenty

OpenStack Swift Introduction - Joe Arnold, John Dickinson