Contents

Reference Architecture Target Requirements

Highly Available access to Openstack services (Nova, Glance, Keystone, Quantum, SWIFT, Cinder)
Support for micro, small, medium, large, x-large, and xx-large instances (up to 32GB memory and 8 cores per VM)
Support for local instance persistence
Ability to support future VM migration and restart
Support for single interface, or multi-interface networks
Support for bonded phsyical network interface resilience
Support OpenStack Essex and Folsom releases against Ubuntu 12.04 LTS and Centos 6

Physical Infrastructure Model

To allow for simplified operational management, there are only two types of systems included in the model:

Compute centric systems and Storage centric systems.

Compute Model

Element	Type	Quantity	Description
CPU	Intel E5-2665	2	Mid-tier high core count CPU for a balance of power and VM scale
Memory	1600MHz 16GB dual rank DIMM	16	Supports up to 4 xx-large instances per physical system
NIC	Cisco VIC	1	Provides dual port 10G interfaces for resiliency and the ability to supprt VM-FEX for hypervisor bypass (on supported Hypervisors)
Disk Controller	Mega-RAID 9266i	1	Provide memory backed Raid operation across the local disks, improve performance of lower cost/denser disk options
Disk Drives	600GB 10Krpm SAS	8	Provide a large possible footprint for local VM instances with reasonable performance

Storage Model

Element	Type	Quantity	Description
CPU	Intel E5-2609	2	Lower core count CPU for a reduced computationally non-complex workload
Memory	1600MHz 8GB dual rank DIMM	4	Provides working memory for in system disk cache
NIC	Cisco VIC	1	Provides dual port 10G interfaces for bonded NIC resiliency
Disk Controller	Mega-RAID 9266i	1	Provide memory backed Raid operation across the local disks for non SWIFT based storage
Disk Drives	1TB 7.2Krpm SATA-6	24	Disk for SWIFT or block or NAS depending on useage model

Network Model

The upstream network is based on the Nexus 5500 series switch enabling the use of VM-FEX when supported by the operating system, Fabric path and L3 services north bound as necessitated by the rest of the network model (either VMDC or MSDC scale systems), or as a L3 termination for a dedicated private cloud. The network also can provide a Virtual Port Channel capability when combined with the default Bonded-NIC host attachment model.

Software architecture

The system software architecture is based on a highly available software architecture, with the goal to support an active management stack on every compute instance in the environment. There are two elements that may limit scale in the current software control plane that we have yet to test at scale, namely AMQP via RabbitMQ and MySQL as implemented under galera (providing write-anywhere consistency across a mysql cluster). To that end, we are intentionally limiting the control plane to the first 3 compute nodes in the environment, and therefore making 3 the smallest compute environment supported by our systems model.

The rest of the architecture follows a fairly default OpenStack system:

API level load balancing via HAproxy on the 3 control nodes, using either keepalived to pick a single active haproxy front end, or using DNSRR or other GSS toolset to select across all three instances.
Nova compute and network along with APIs on every node
Glance, Keystone, and other entities (such as Quantum and eventually Cinder) on the first three compute nodes
Swift on it's own storage instances
Swift proxy on its own compute instances with it's own set of haproxy instances frontending the requests, also either DNSRR fronted or with keepalived pointing to an active proxy front end.

Rack scale example system

Why the system can be deployed in a large number of configurations, from the simplest single compute

"all-in-one" model to a multi-rack scale out environment, a more common model is as follows:

Rack Location	Principal Function	Element
R41	Network	5548-UP
R40	Network	5548-UP
R39		Expansion
R38		Expansion
R37	Control/Compute	Compute
R36	Control/Compute	Compute
R35	Control/Compute	Compute
R34	Compute	Compute
R33	Compute	Compute
R32	Compute	Compute
R31	Compute	Compute
R30	Compute	Compute
R29	Compute	Compute
R28	Compute	Compute
R27	Compute	Compute
R26	Compute	Compute
R25	Compute	Compute
R24	Compute	Compute
R23	Compute	Compute
R22	Compute	Compute
R21	Compute	Compute
R20	Compute	Compute
R19	Storage Proxy	Compute
R18	Storage Proxy	Compute
R17	Storage Proxy	Compute
R15	Cinder - Block	Storage
R13	Cinder - Block	Storage
R11	Cinder - Block	Storage
R9	SWIFT	Storage
R7	SWIFT	Storage
R5	SWIFT	Storage
R3	SWIFT	Storage
R1	SWIFT	Storage