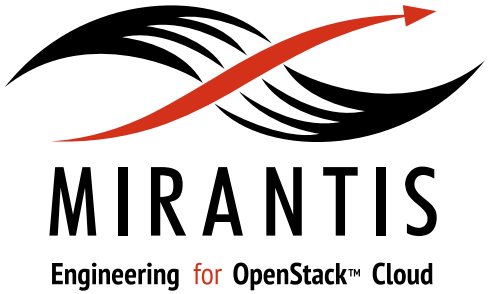
Bootcamp for OpenStack™



PREVIEW

Mirantis

www.mirantis.com/training

2-Day Agenda

MIRANTIS
Engineering for OpenStack™ Cloud

	TOPIC	LECTURE, DEMOS AND GROUP EXERCISES
DAY 1	**OpenStack Overview & Architecture**	Project goals and use cases, basic operating and deployment principles *(EXCERPTED IN THIS PREVIEW)*
	Cloud Usage Patterns	OpenStack codebase overview; creating networks, tenants, roles, troubleshooting; using Nexenta Volume Driver
DAY 2	**In Production**	Deploying OpenStack for real-world use, and practice of OpenStack operation on multiple nodes
	Swift Object Storage	Use cases, architecture, capabilities, configuration, security and deployment
	Advanced Topics	Software Defined Networking, deployment and issues workshop, VMWare/OpenStack comparison

BUILT FOR openstack CLOUD SOFTWARE

Goals

MIRANTIS
Engineering for OpenStack™ Cloud

- Understand OpenStack purpose and use cases
- Understand OpenStack ecosystem
  - history
  - projects
- Understand OpenStack architecture
  - logical architecture
  - components
  - request flow
- Get enough theory for hands-on lab

BUILT FOR openstack CLOUD SOFTWARE

What is OpenStack?

MIRANTIS
Engineering for OpenStack™ Cloud

"Open source software for building private and public clouds"

BUILT FOR
openstack™
CLOUD SOFTWARE

OpenStack capabilities

MIRANTIS
Engineering for OpenStack™ Cloud

- VMs on demand
  - provisioning
  - snapshotting
- Volumes
- Multi-tenancy
  - quotas for different users
  - user can be associated with multiple tenants
- Object storage for VM images and arbitrary files

BUILT FOR openstack CLOUD SOFTWARE

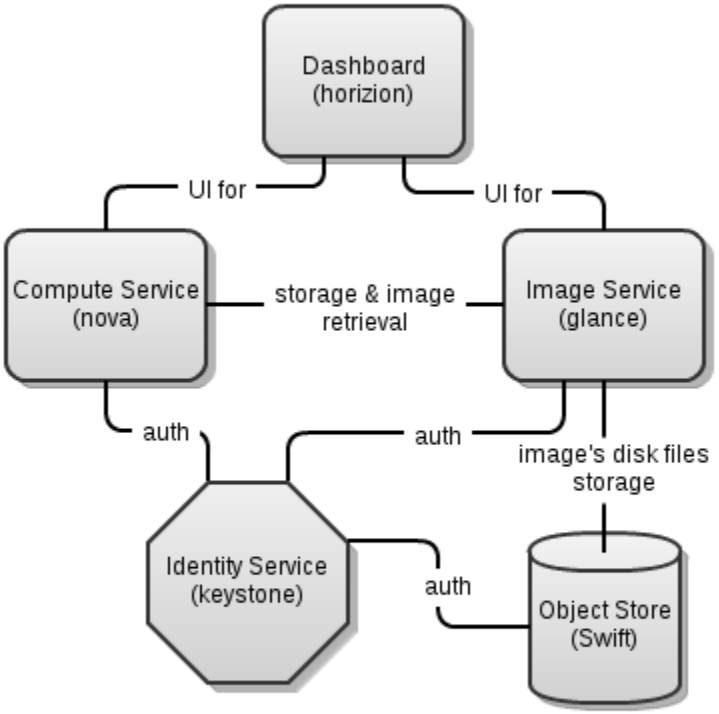
OpenStack History

- July 2010 - Initial announcement
- October 2010 - Austin Release
- April 2011 - Cactus Release
- October 2011 - Diablo Release
- April 2012 - Essex Release
- October 2012 - Folsom Release

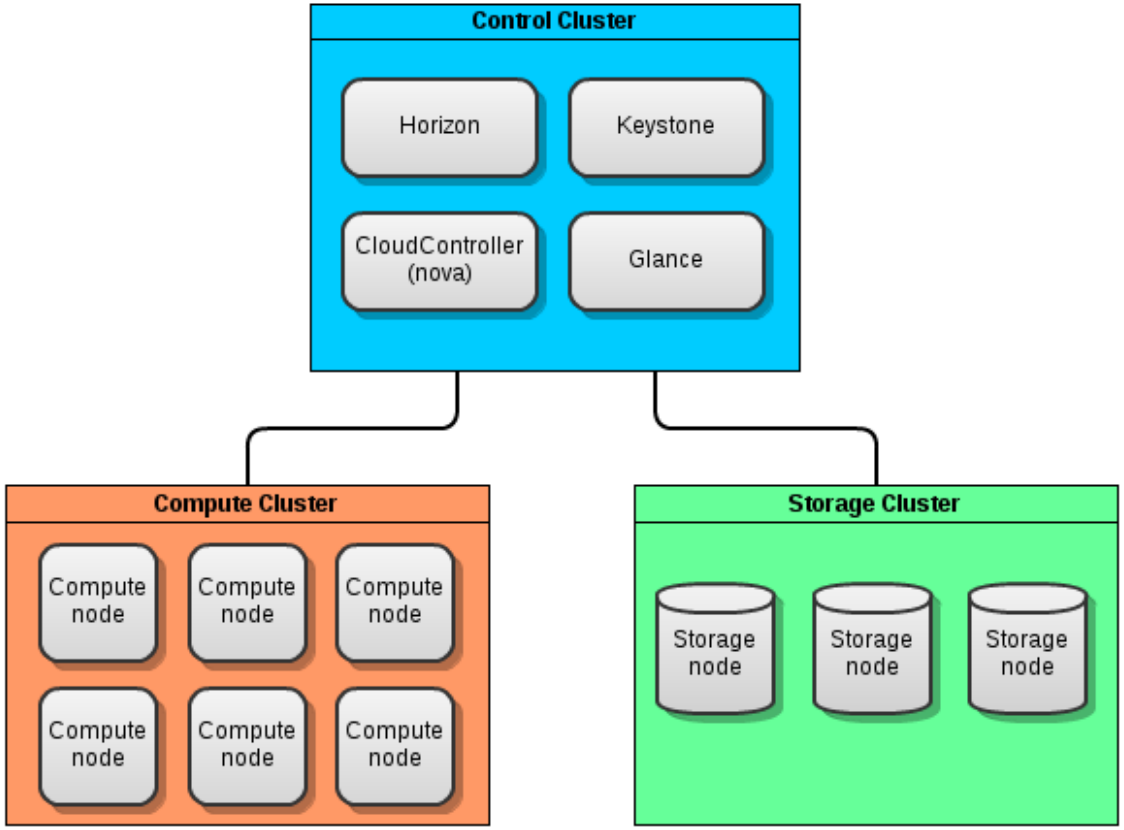
OpenStack Projects

- Nova (Compute)
- Glance (Image Service)
- Swift (Object Store)
- Keystone (auth)
- Horizon (Dashboard)

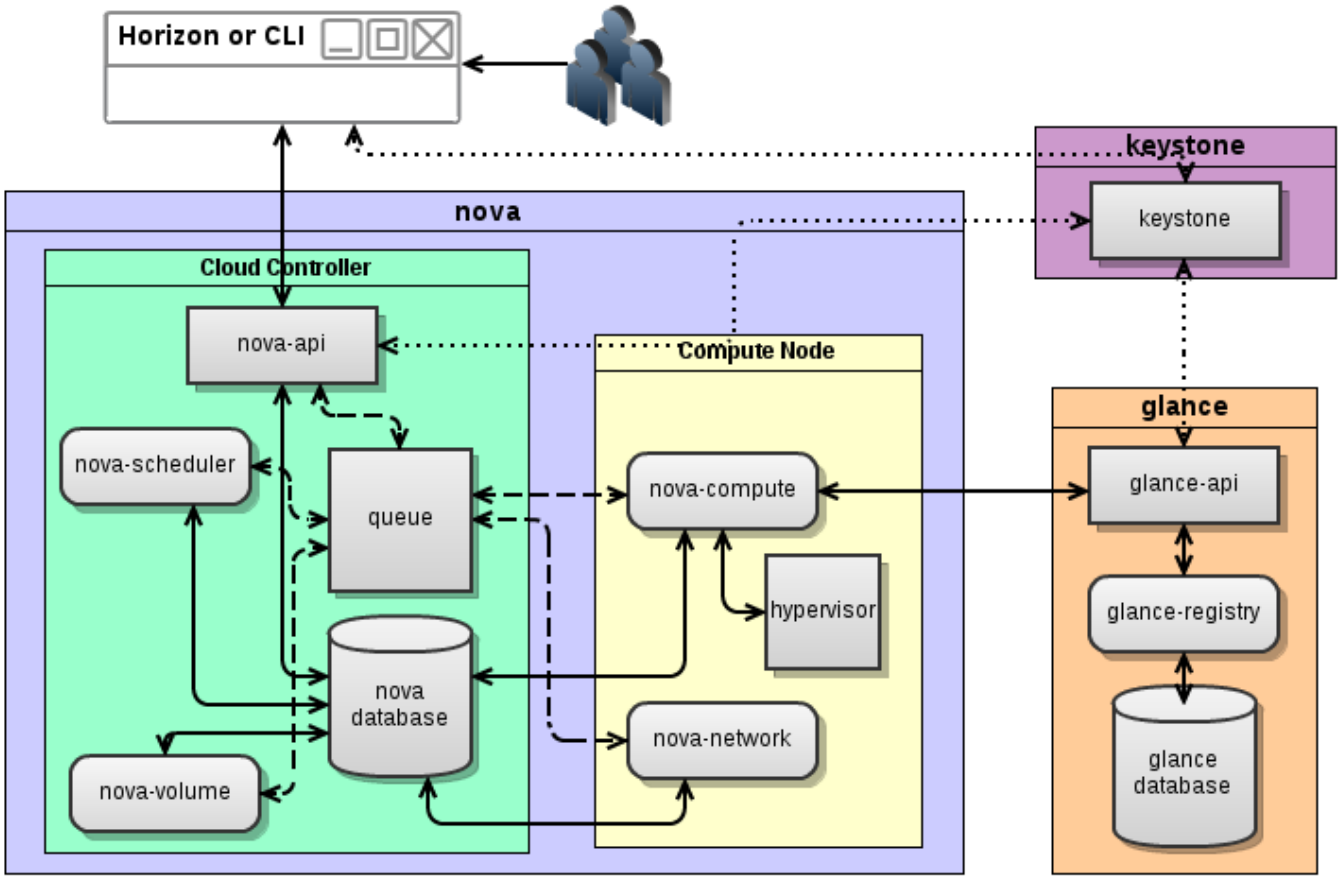
OpenStack Projects: Relationship



OpenStack: Deployment Topology



OpenStack Projects: Detailed View



Horizon

MIRANTIS
Engineering for OpenStack™ Cloud

"The OpenStack Dashboard (Horizon) provides a baseline user interface for managing OpenStack services."

Horizon notes

MIRANTIS
Engineering for OpenStack™ Cloud

- "Stateless"
- Error handling is delegated to back-end
- Doesn't support all API functions
- Can use memcached or database to store sessions
- Gets updated via nova-api polling

BUILT FOR openstack CLOUD SOFTWARE

Horizon internals

MIRANTIS
Engineering for OpenStack™ Cloud

- 2 subprojects
  - horizon - generic Django libraries and components to work with REST-based back-end
  - openstack-dashboard - web app itself, with styles, locale, etc.

- Dashboard for each entity (like `instances` or `images`) - nested Django app

BUILT FOR openstack CLOUD SOFTWARE

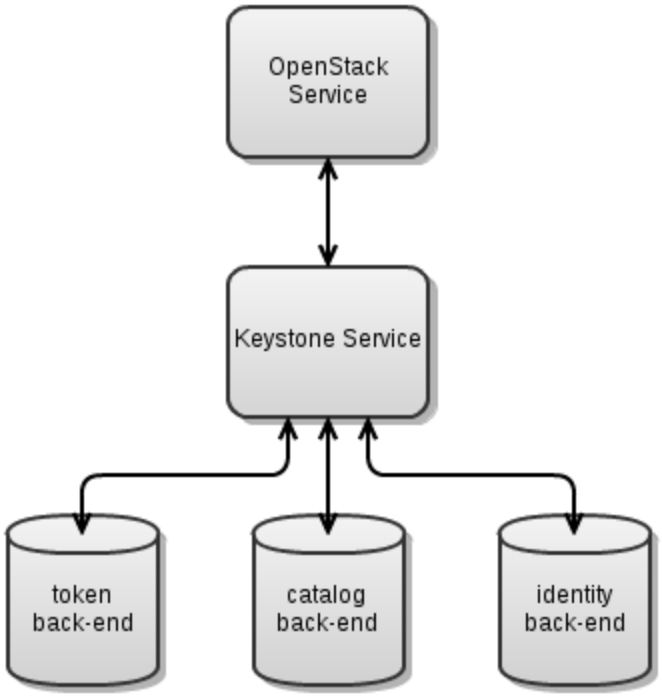
Keystone

MIRANTIS
Engineering for OpenStack™ Cloud

"Keystone is an OpenStack project that provides Identity, Token, Catalog and Policy services for use specifically by projects in the OpenStack family."

BUILT FOR openstack CLOUD SOFTWARE

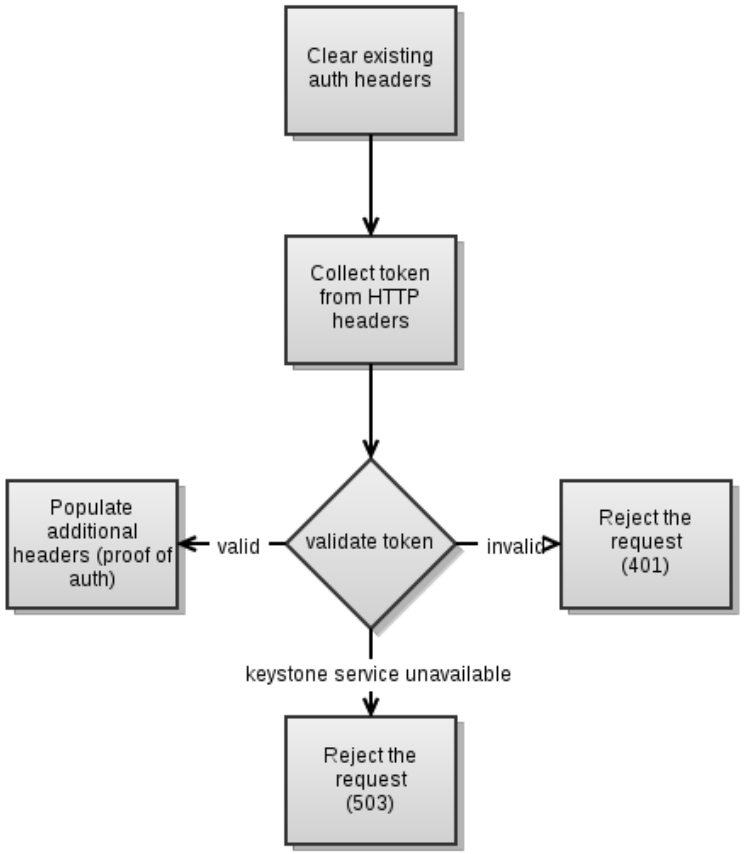
Keystone Architecture



Keystone data model

- **User**: has account credentials, is associated with one or more tenants
- **Tenant**: unit of ownership in openstack, contains one or more users
- **Role**: a first-class piece of metadata associated with many user-tenant pairs.
- **Token**: identifying credential associated with a user or user and tenant
- **Extras**: bucket of key-value metadata associated with a user-tenant pair.
- **Rule**: describes a set of requirements for performing an action.

Keystone: auth flow



Keystone: populating auth data

- Add tenants
- Add users
- Add roles
- Grant roles to users
- Add endpoint templates
- Map endpoint templates to zones

nova-api

MIRANTIS
Engineering for OpenStack™ Cloud

"nova-api is a RESTful API web service which is used to interact with nova"

BUILT FOR openstack CLOUD SOFTWARE

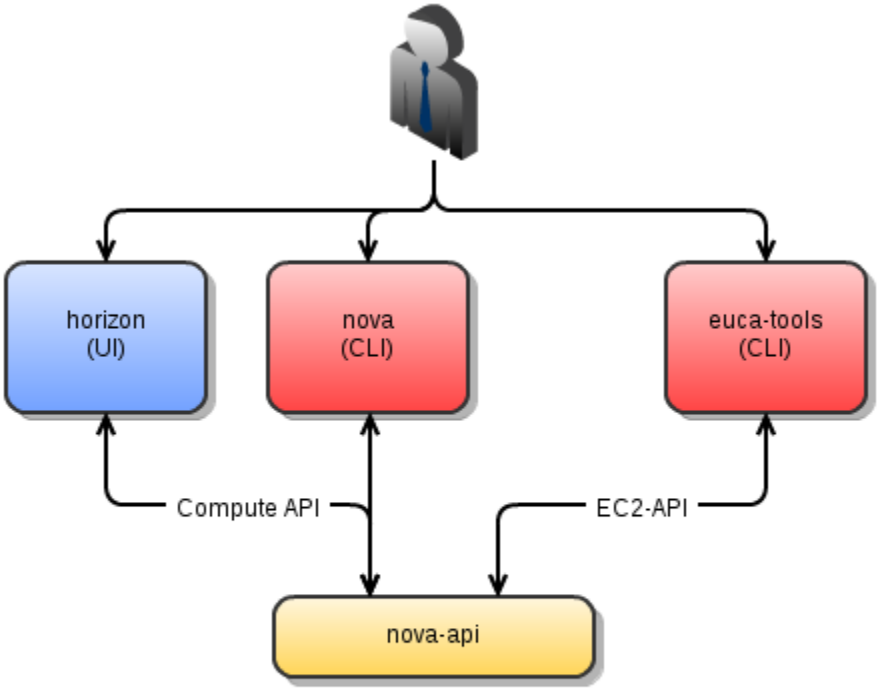
nova-api characteristics

MIRANTIS
Engineering for OpenStack™ Cloud

- Exposes REST API
- Provides system for managing multiple APIs on different sub-domains
  - EC2-compatible - will be deprecated
  - OpenStack Compute API - all innovation happens here
- The only "allowed" way to interact with nova
- Stateless - HA-ready

BUILT FOR openstack CLOUD SOFTWARE

nova-api clients



nova database

MIRANTIS
Engineering for OpenStack™ Cloud

"nova database stores current state of all objects in compute cluster."

BUILT FOR openstack CLOUD SOFTWARE

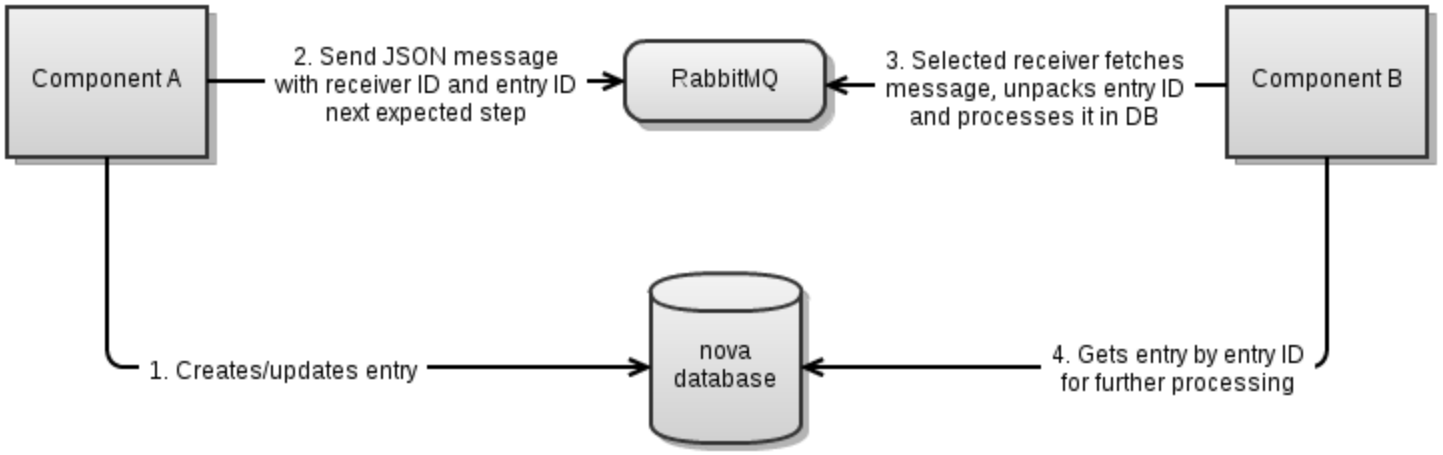
nova database

- Can be any relational database
- nova-api talks to DB via SQLAlchemy (python ORM)
- Most of the deployments are done with MySQL or PostreSQL
- DB HA should be done via external tools (like MMM for MySQL)

Message queue

"Message queue is a unified way for collaboration between nova components."

OpenStack messaging



2 modes:

- rpc.cast - don't wait for result
- rpc.call - wait for result (when there is something to return)

Messagings notes

- OpenStack uses multiple queues within single RabbitMQ instance
- OpenStack messages traffic is not intensive
- OpenStack doesn't send broadcast messages
- HA for MQ should be configured separately

nova-scheduler

"nova-scheduler is a daemon, which determines, on which compute host the request should run."

nova-scheduler: users' demand

- provision VM to particular host
- provision VMs of the particular tenant to isolated hosts
- provision all VMs on different hosts
- provision VMs to "higher density" hosts

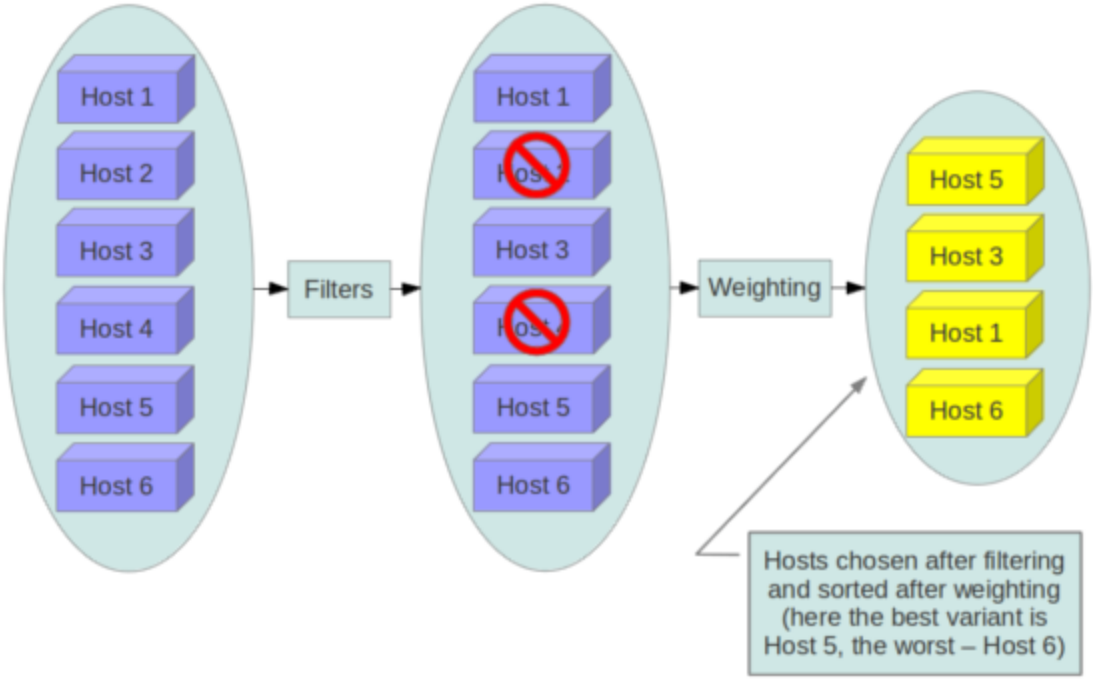
nova-scheduler: available schedulers

MIRANTIS
Engineering for OpenStack™ Cloud

Scheduler	Description
Chance	Picks a host that is up at random
Simple	Picks a host that is up and has the fewest running instances
Filter	Picks the best-suited host which satisfies selected filter
Multi	A scheduler that holds multiple sub-schedulers

BUILT FOR openstack CLOUD SOFTWARE

nova-scheduler: filtering



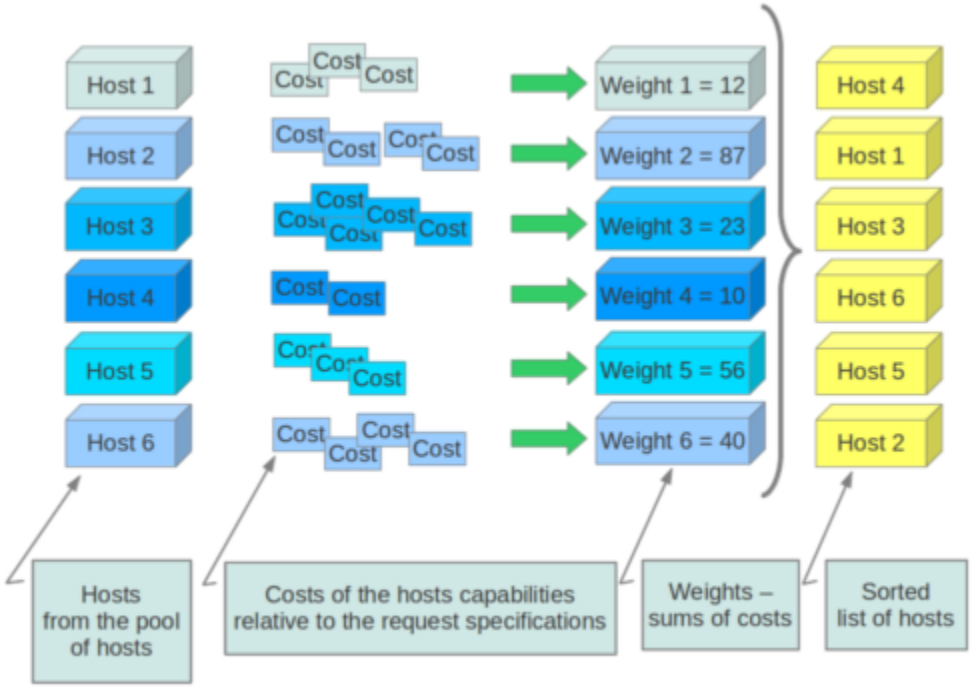
nova-scheduler: filters

Filter	Description
affinity	Same host or different host
availability zone	Least cost inside selected availability zone
core	Least CPU core utilization
ram	Only return hosts with sufficient RAM
json	Allows simple JSON based grammar. Can be used to build custom schedulers.

nova-scheduler: filters

- Filters are statically configured in nova.conf
- Multiple filters can be specified
- It is possible to create custom filter
  - Inherit from BaseHostFilter class
  - override host_passes(self, host_state, filter_properties)

nova-scheduler: weights and costs



nova-scheduler: weights and costs

- Cost - integer value
- Every compute host can have several cost functions associated with it
- If no cost functions associated - use default from nova.conf
- weight = sum($cost_i$ + $weigth\_fn_i$)

nova-scheduler: summary

- Allow to tweak provisioning by adjusting filters, cost and weights
- Still doesn't cover all customer demands - exposes framework for building custom schedulers instead

Questions

- How does OpenStack understand that specific request can be executed by the user?
- How to get a status for a requested server? Where it will come from?
- What is the difference between rpc.call vs rpc.cast?
- How to create a filter, which will determine servers with 8GB to 16GB RAM available?

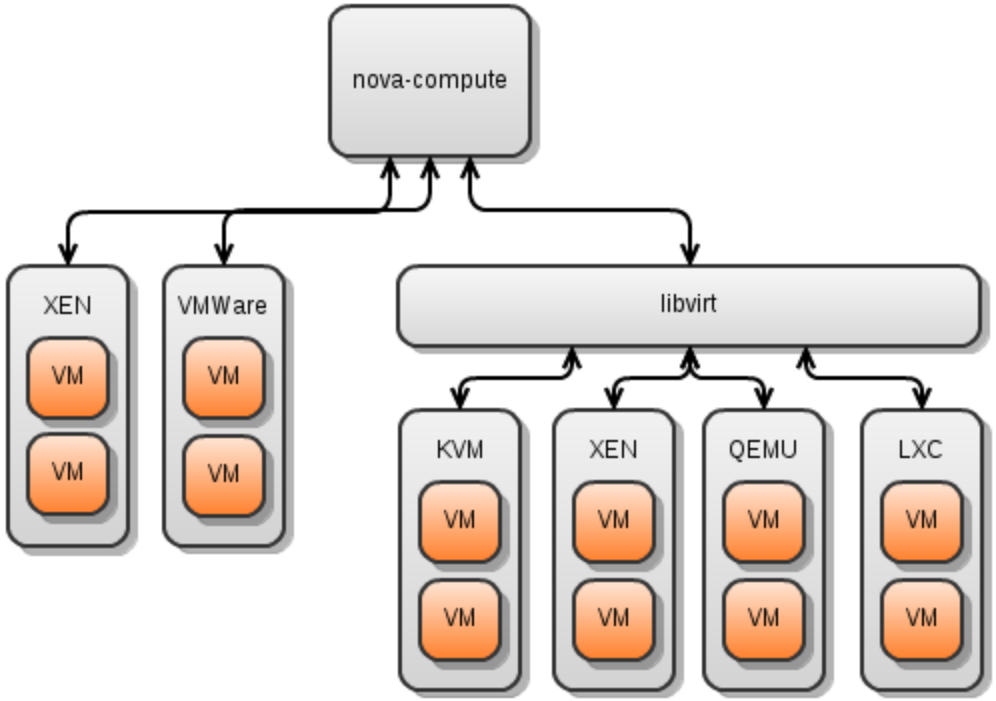
nova-compute

MIRANTIS
Engineering for OpenStack™ Cloud

"nova-compute is a worker daemon, which primarily creates and terminates VMs via hypervisor API."

BUILT FOR
openstack™
CLOUD SOFTWARE

nova-compute



nova-compute: drivers

- Functionality is not 100% similar
- Exact "run_instance" flow depends on driver implementation
- Most of the features are tested on KVM

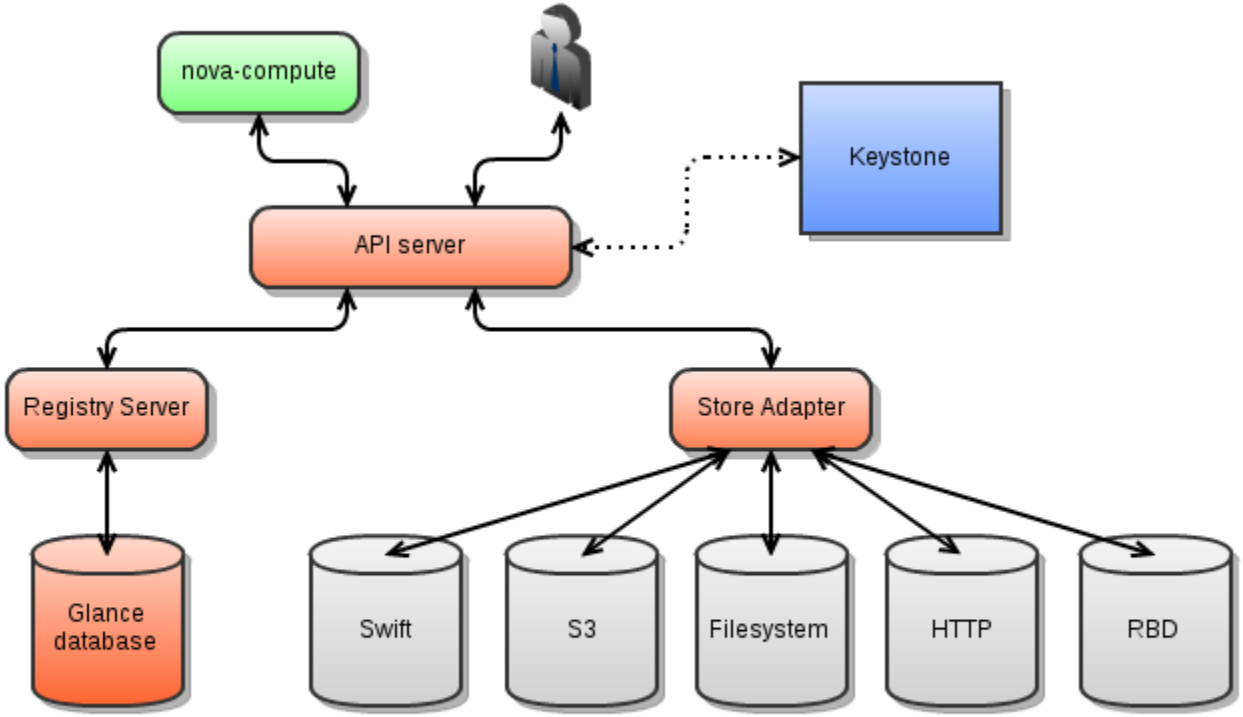
Glance

"The Glance project provides services for discovering, registering, and retrieving virtual machine images."

Glance summary

- Image-as-a-service
- Can use multiple back-ends for image storage
- Supports multiple image formats

Glance architecture



Glance capabilities

- CRUD images
- Search images via filters
  - name
  - container format
  - disk format
  - size_min, size_max
  - status
- Caches images
  - uses SQLite or FS that supports xattrs for caching
  - queues images for prefetching
  - prefetches images
  - prunes images
  - cleans invalid cache entries

Glance image formats

Disk Format	Description
raw	This is an unstructured disk image format
vhd	This is the VHD disk format, a common disk format used by virtual machine monitors from VMWare, Xen, Microsoft, VirtualBox, and others
vmdk	Another common disk format supported by many common virtual machine monitors
vdi	A disk format supported by VirtualBox virtual machine monitor and the QEMU emulator
iso	An archive format for the data contents of an optical disc (e.g. CDROM).
qcow2	A disk format supported by the QEMU emulator that can expand dynamically and supports Copy on Write
aki	This indicates what is stored in Glance is an Amazon kernel image
ari	This indicates what is stored in Glance is an Amazon ramdisk image
ami	This indicates what is stored in Glance is an Amazon machine image

Fetching image from glance

1. GET http://<glance-url>/images/<ID>
2. If image can be found, API returns image-uri
3. nova-compute passes image-uri to hypervisor driver
4. hypervisor driver fetches image directly from glance back-end store using image-uri

Custom image creation

MIRANTIS
Engineering for OpenStack™ Cloud

1. Get installation ISO
2. Create VM (qemu-img create)
3. Start VM and connect to it via VNC console
   a. Install image without LVM
   b. Create default iptables rules
   c. Install and configure cloud-init
   d. With cloud-init configure image
4. Prepare image for OpenStack
   a. Extract root partition, kernel and ramdisk
   b. cleanup
   c. package

BUILT FOR openstack CLOUD SOFTWARE

Network configuration flow

1. Allocate MAC addresses
2. Allocate IPs (for each network)
3. Associate IPs with VMs (DB)
4. Setup network on host
   a. Update DHCP config
   b. Initialize gateway
   c. VPN configuration (optional)
5. Update networking info in DB

nova-network

MIRANTIS
Engineering for OpenStack™ Cloud

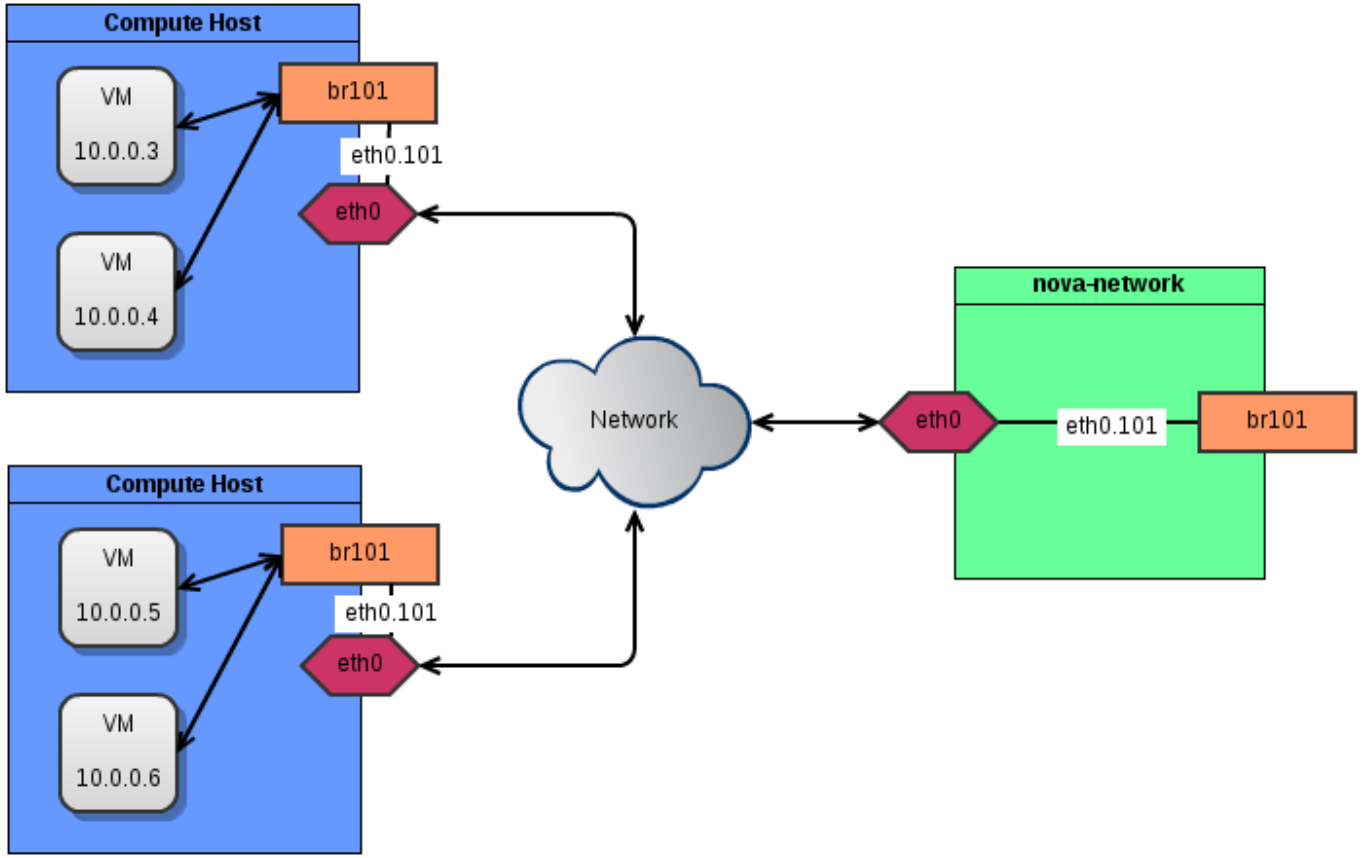
"nova-network is a worker daemon which performs tasks to manipulate network via external commands."

BUILT FOR
openstack
CLOUD SOFTWARE

nova-network responsibilities

- Allocate and configure network via network manager
  - FlatManager
  - FlatDHCPManager
  - VlanManager
- Manage Floating IPs
- Manage Security groups

FlatManager



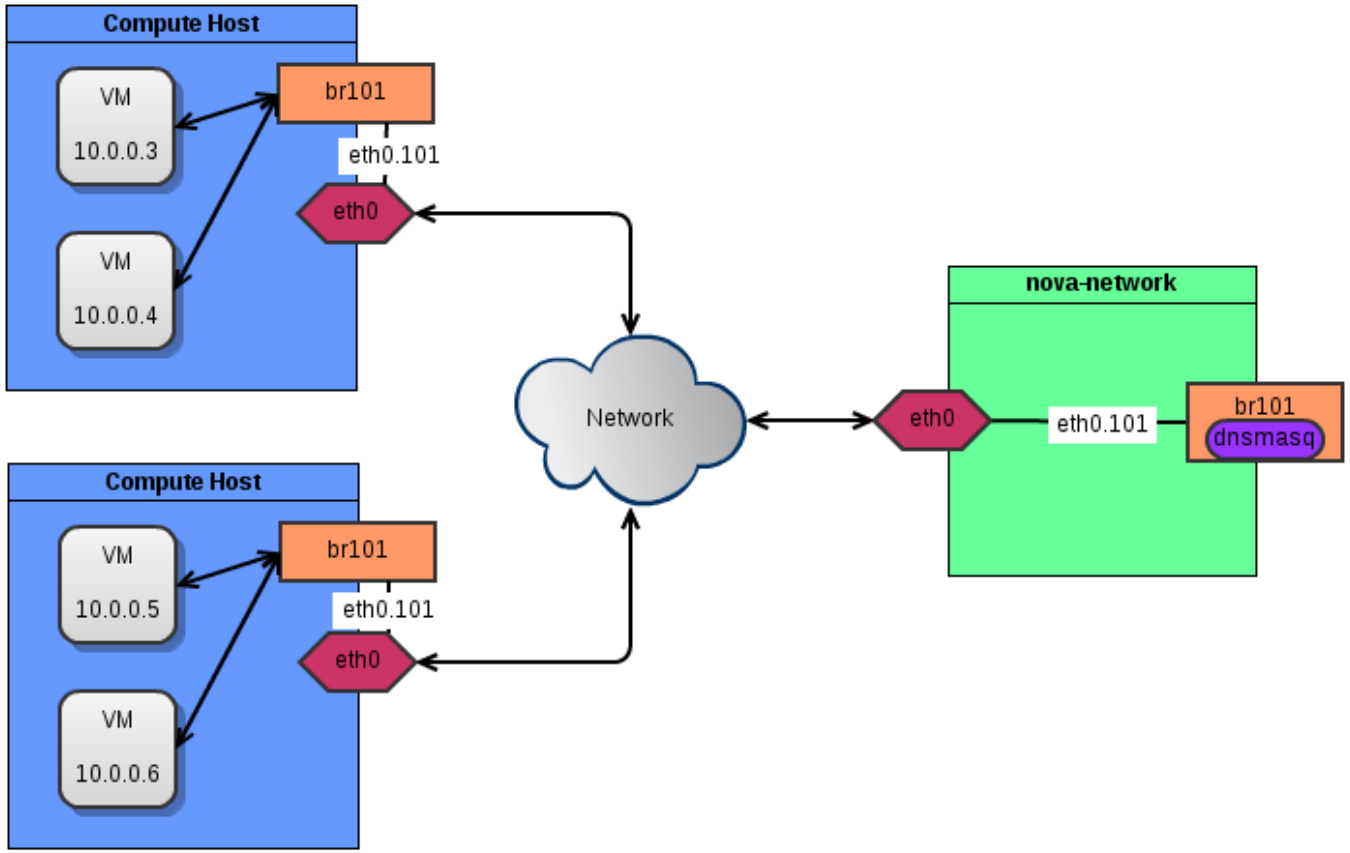
FlatManager

MIRANTIS
Engineering for OpenStack™ Cloud

- Supports only single network
- Doesn't do any bridge/vlan creation
- The bridge needs to be manually created on all hosts
- Compute host attempts to inject network settings into /etc/network/interfaces

BUILT FOR openstack™ CLOUD SOFTWARE

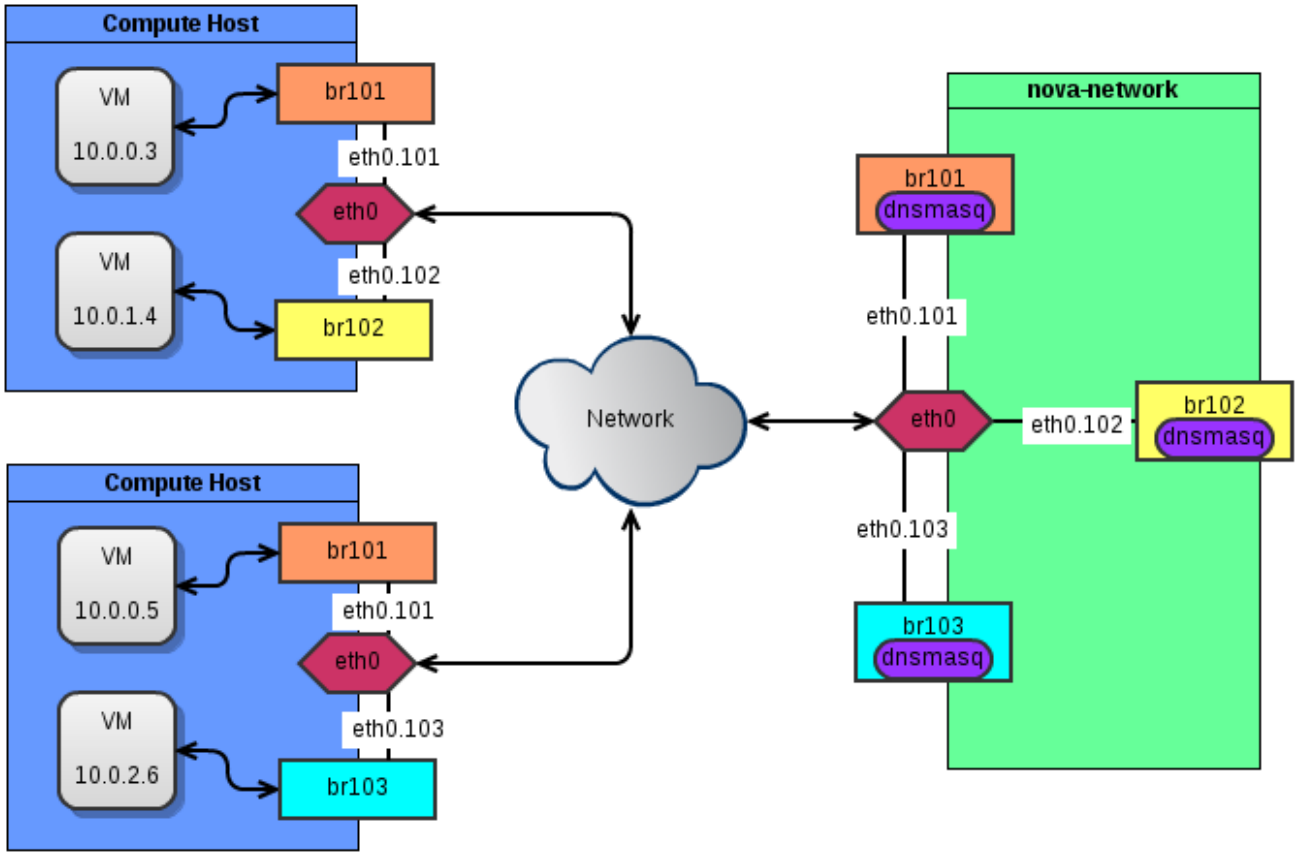
FlatDHCPManager



FlatDHCPManager

- Improvement of FlatManager
- Stars up 1 DHCP server to give out addresses
- Never injects network settings into guest
- Manages bridges

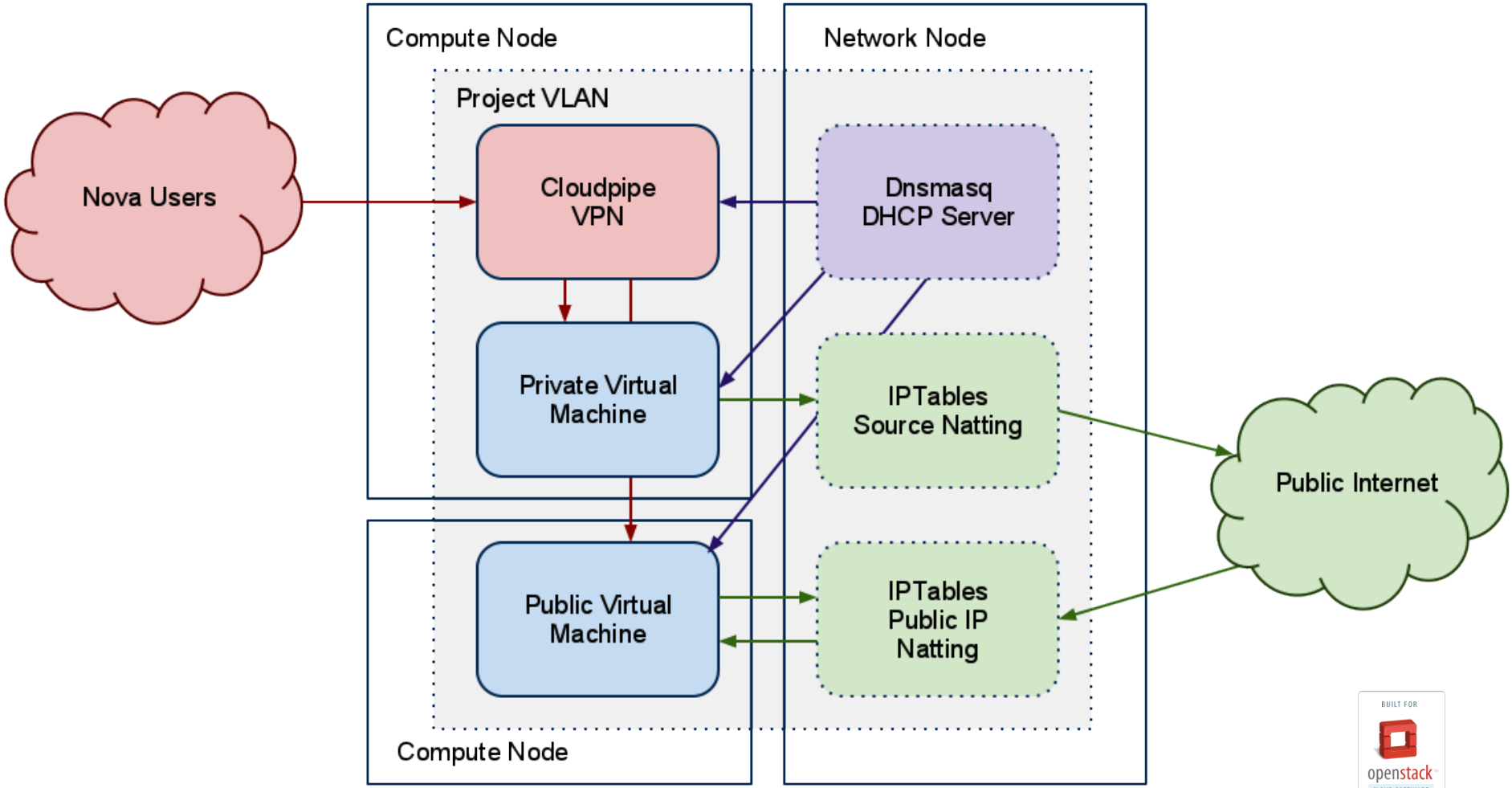
VlanManager



VlanManager features

- Creates host-managed VLAN for each project
- Requires switch that supports VLAN tagging (IEEE802.1Q)
- Each project gets own subnet (VPN is required to access VMs via private IPs)
- DHCP server is running for each subnet
- All instances belonging to one project are bridged into the same VLAN for that project

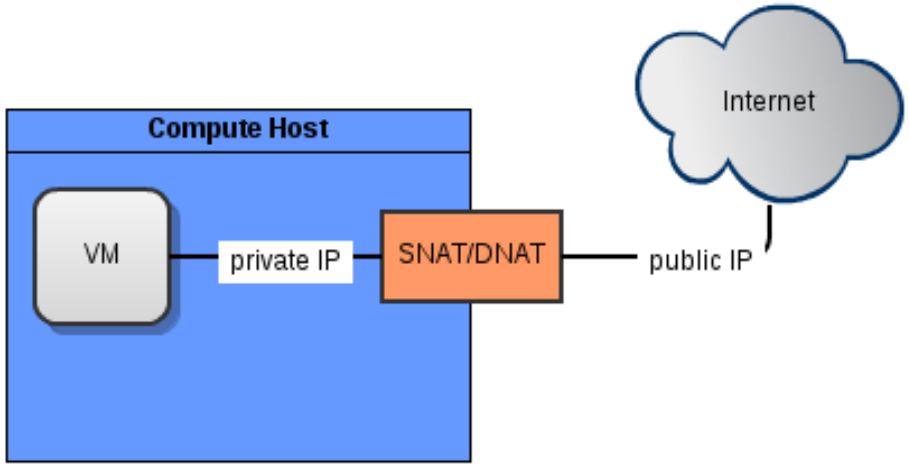
CloudPipe


Compute Node
Network Node
Project VLAN
Nova Users
Cloudpipe VPN
Dnsmasq DHCP Server
Private Virtual Machine
IPTables Source Natting
Public Internet
Public Virtual Machine
IPTables Public IP Natting
Compute Node

Floating IPs

- Shared pool of public IP addresses
- Each user gets a quota of how many IPs to use
- Managed by admin

Floating IPs traffic



Assigning Floating IPs

MIRANTIS
Engineering for OpenStack™ Cloud

OpenStack Admin

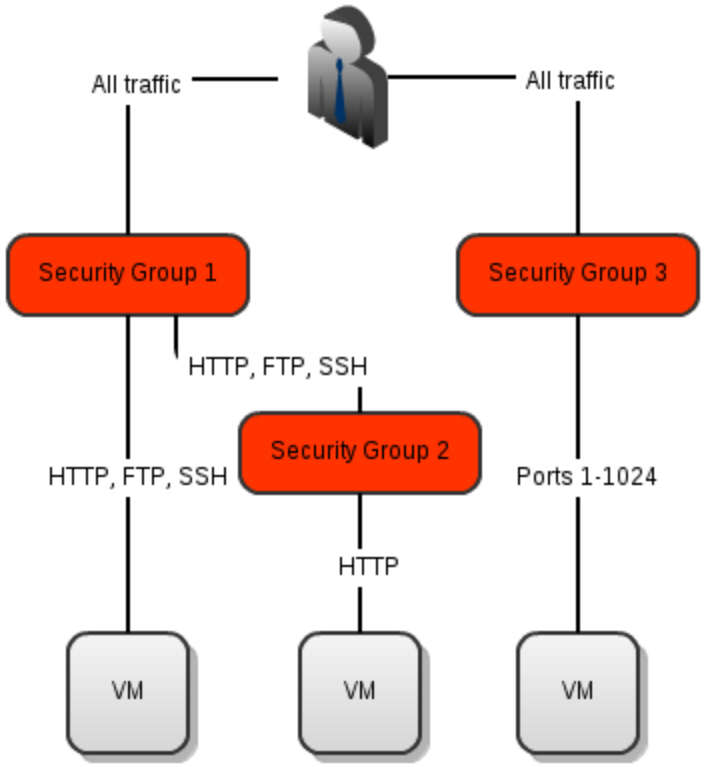
- Dedicate floating IPs to cluster

OpenStack User

1. Allocate public IP for tenant within given quota
2. Associate public IP with VM
   a. Find host
   b. Add IP address to public network interface of the host
   c. NATting all network traffic via associated floating IP

BUILT FOR openstack CLOUD SOFTWARE

Security Groups



Security Groups

- Security group is a named collection of network access rules
- User can select multiple security groups during VM creation
- If no security groups specified - default is selected
- Security groups are applied on the host node

nova-volume

MIRANTIS
Engineering for OpenStack™ Cloud

"nova-volume manages the creation, attaching and deattaching of persistent volumes to compute instances"

BUILT FOR
openstack™
CLOUD SOFTWARE

nova-volume summary

- Optional
- iSCSI solution which uses LVM
- Volume can be attached only to 1 instance at a time
- Persistent volumes keep their state independent of instances
- Within single OpenStack deployment different storage providers cannot be used

nova-volume drivers

- iSCSI
- Xen Storage Manager
- Nexenta
- NetApp
- SAN

2-Day Agenda

MIRANTIS
Engineering for OpenStack™ Cloud

	TOPIC	LECTURE, DEMOS AND GROUP EXERCISES
DAY 1	**OpenStack Overview & Architecture**	Project goals and use cases, basic operating and deployment principles *(EXCERPTED IN THIS PREVIEW)*
	Cloud Usage Patterns	OpenStack codebase overview; creating networks, tenants, roles, troubleshooting; using Nexenta Volume Driver
DAY 2	**In Production**	Deploying OpenStack for real-world use, and practice of OpenStack operation on multiple nodes
	Swift Object Storage	Use cases, architecture, capabilities, configuration, security and deployment
	Advanced Topics	Software Defined Networking, deployment and issues workshop, VMWare/OpenStack comparison

BUILT FOR openstack CLOUD SOFTWARE

END PREVIEW

Bootcamp for OpenStack

www.mirantis.com/training