

Déployer son propre cloud avec OpenStack

GULL

18.11.2014

François Deppierraz

francois.deppierraz@nimag.net



Who Am I ?

- System and Network Engineer
- Stuck in the Linux world for almost 2 decades
- Sysadmin who doesn't like to type the same command twice !
- <https://github.com/ctrlaltdel>
- My company provides consulting services

Outline

- Cloud Computing
- What is OpenStack ?
- How do you use an OpenStack cloud ?
- How do you deploy your very own cloud ?



Cloud Computing

- Features
 - On demand, self service
 - Network access
 - Resource pooling, multi-tenancy
 - Elasticity
 - Metered service

Cloud Computing

- Service Models
 - SaaS – Software as a Service
 - Gmail, wordpress.com, wiki du GULL ;)
 - PaaS – Platform as a Service
 - Google App Engine, Heroku, Salesforce
 - IaaS – Infrastructure as a Service
 - Amazon Web Services, Google Compute Engine



18.11.2014

François Deppierraz
Source: <http://www.dopenstack.com/>

What the Cloud is OpenStack ?

- Infrastructure as a Service (IaaS) Platform
- A “clone” of Amazon Web Services (AWS)
- Apache 2.0 License
- Written mostly in Python
- Project started around 2010
- `<buzzword>The Linux of the Cloud</buzzword>`

Foundation Platinum Members



AT&T



Canonical



HP



IBM



Nebula



Rackspace



Red Hat, Inc.



SUSE

Foundation Gold Members



Aptira



CCAT



Cisco



Cloudscaling



Dell



DreamHost



Ericsson



Hitachi



Huawei



Intel



Juniper Networks



Mirantis



NEC



NetApp



Piston Cloud Computing, Inc.



VMware



Yahoo!



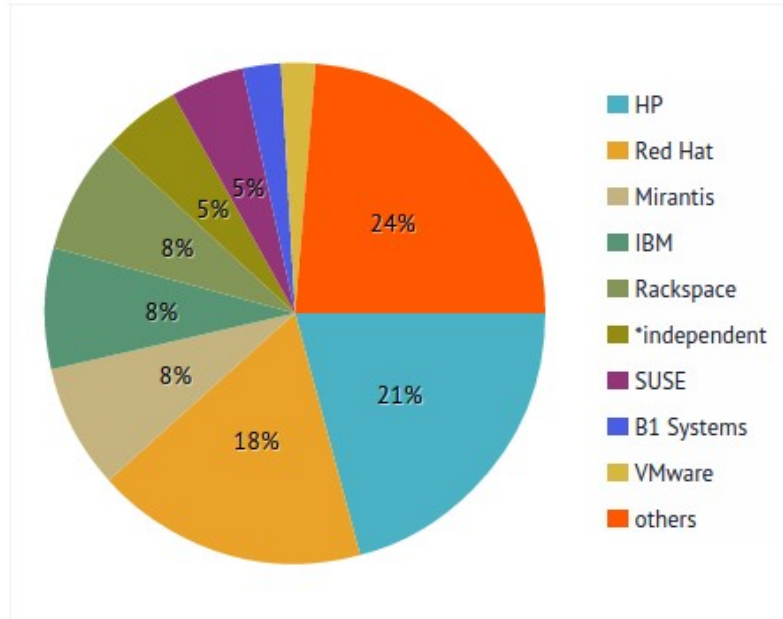
18.11.2014

Current release

JUN10
THE TENTH OPENSTACK RELEASE

OpenStack Contributors

Contribution by companies



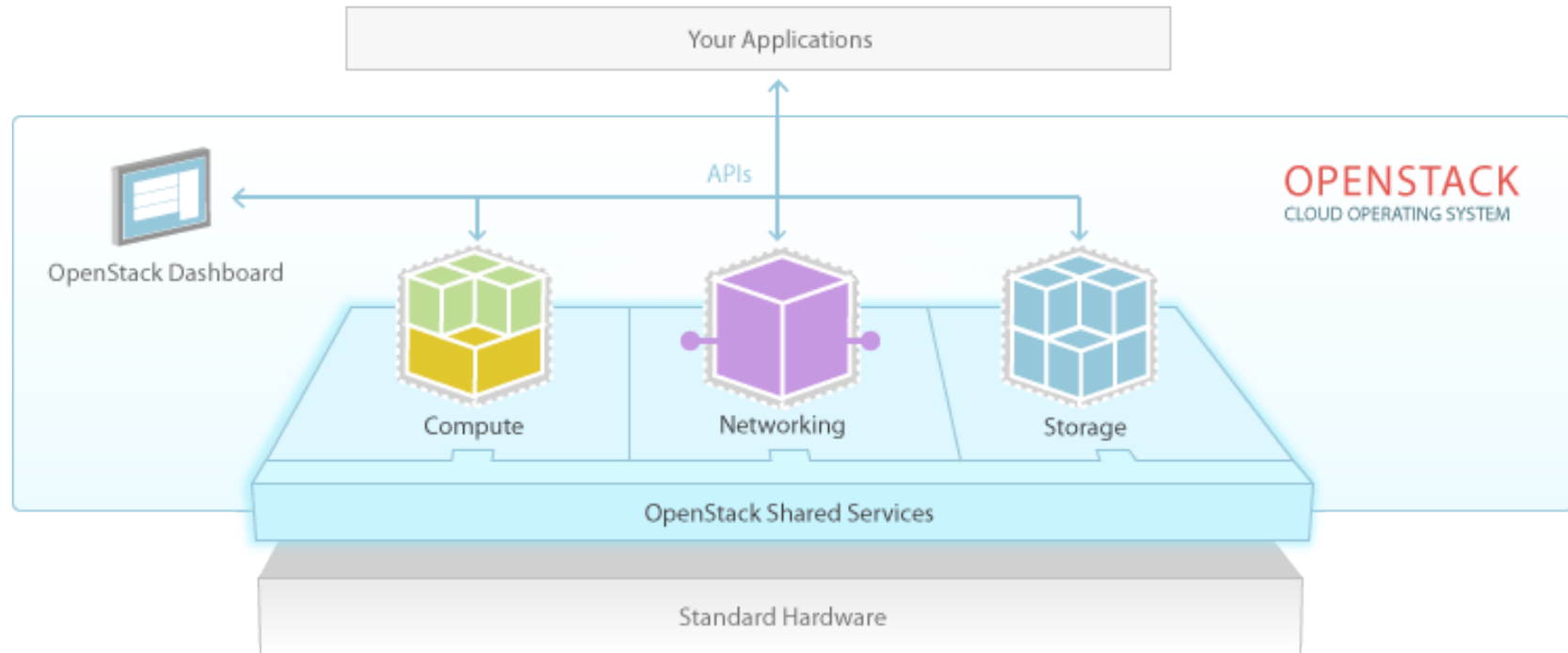
- Juno release
- Chart by commits
- ~19k commits total
- 6 month timeframe
- <http://stackalytics.com/>

Releases

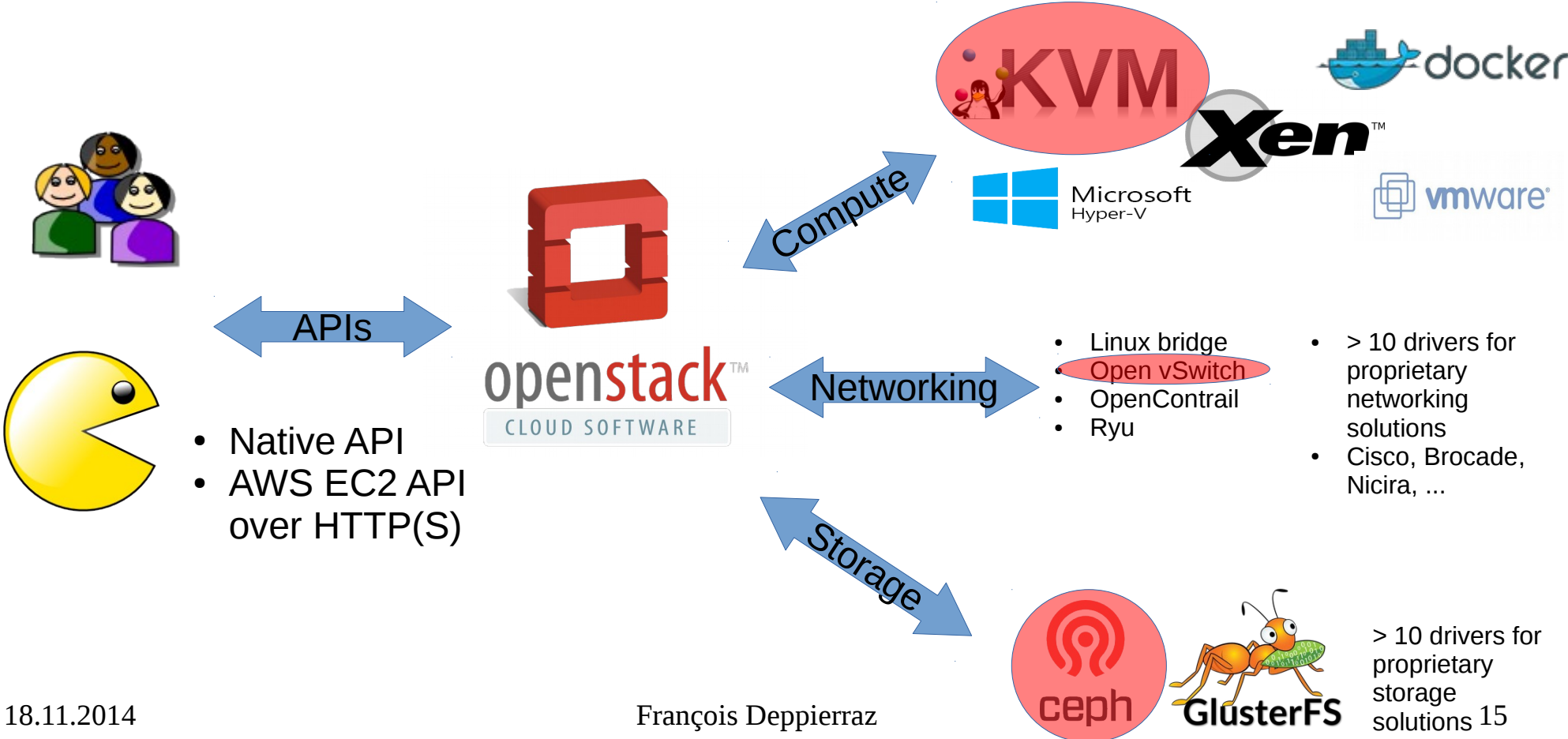
OpenStack Releases

Release Name	Release Date	Included Components
Austin	21 October 2010	Nova, Swift
Bexar	3 February 2011	Nova, Glance, Swift
Cactus	15 April 2011	Nova, Glance, Swift
Diablo	22 September 2011	Nova, Glance, Swift
Essex	5 April 2012	Nova, Glance, Swift, Horizon, Keystone
Folsom	27 September 2012	Nova, Glance, Swift, Horizon, Keystone, Quantum, Cinder
Grizzly	4 April 2013	Nova, Glance, Swift, Horizon, Keystone, Quantum, Cinder
Havana	17 October 2013	Nova, Glance, Swift, Horizon, Keystone, Neutron, Cinder, Ceilometer, Heat
Icehouse	17 April 2014	Nova, Glance, Swift, Horizon, Keystone, Neutron, Cinder, Ceilometer, Heat, Trove
Juno	October 2014	Nova, Glance, Swift, Horizon, Keystone, Neutron, Cinder, Ceilometer, Heat, Trove, Sahara
Kilo	April 2015	Nova, Glance, Swift, Horizon, Keystone, Neutron, Cinder, Ceilometer, Heat, Trove, Sahara, Ironic

The Big Picture

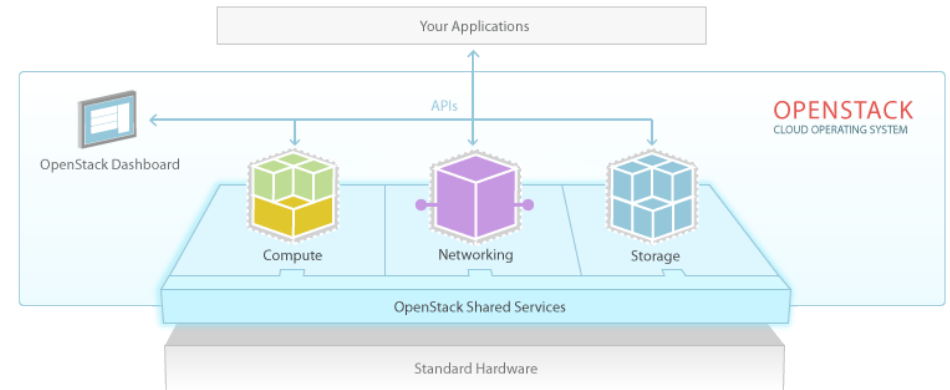


OpenStack is like glue

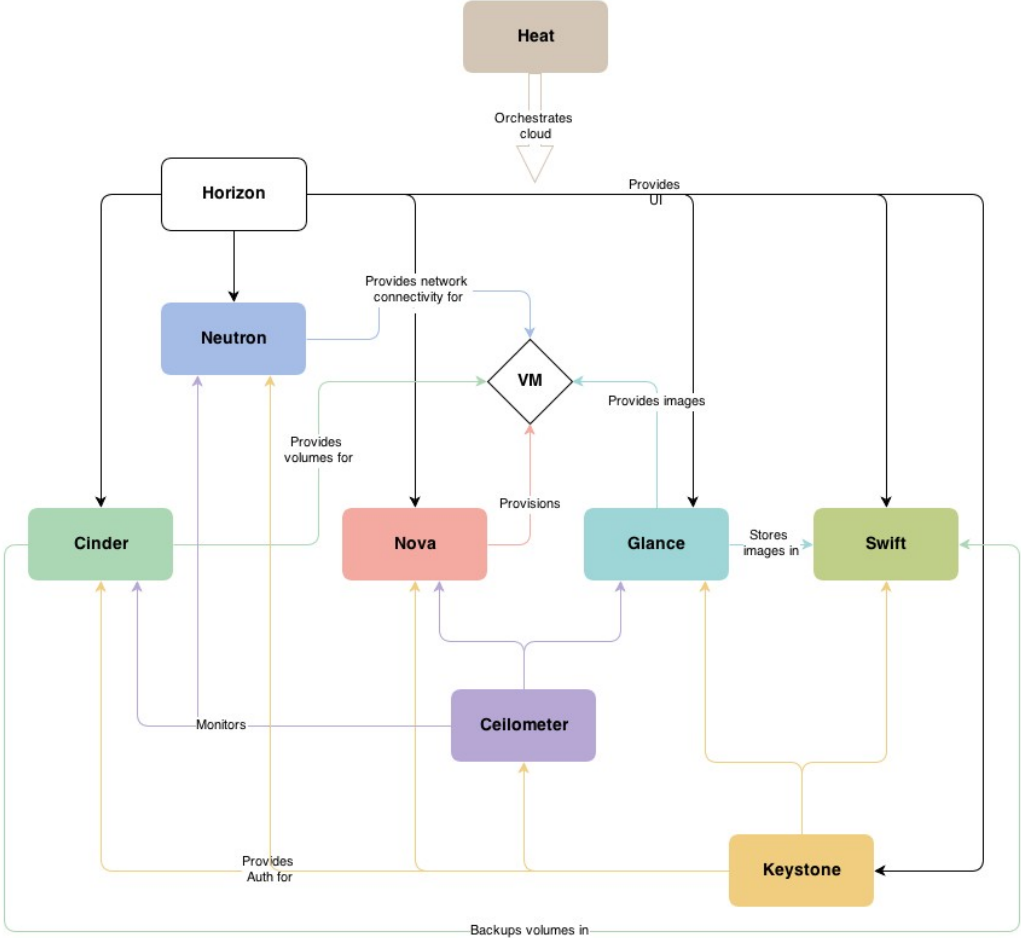


OpenStack Components

- Compute
 - Servers (Nova)
- Networking (Neutron)
- Storage
 - Object Storage (Swift)
 - Block Storage (Cinder)
- Shared Services
 - Dashboard (Horizon)
 - Identity Service (Keystone)
 - Image Service (Glance)
 - Orchestration Service (Heat)
 - Telemetry Service (Ceilometer)
 - Database Service (Trove)
 - Hadoop (Sahara)
 - Baremetal (Ironic)

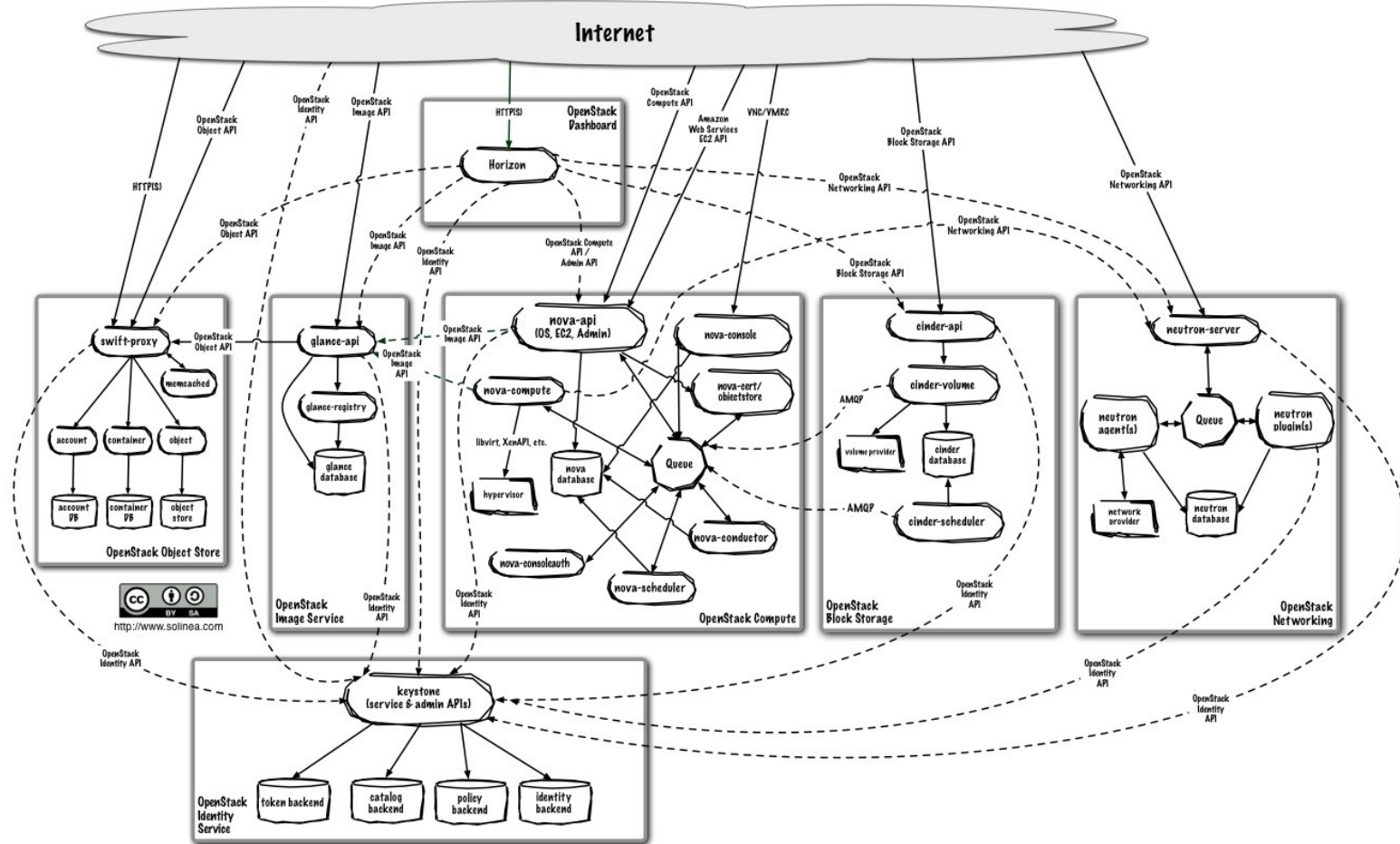


Conceptual Architecture





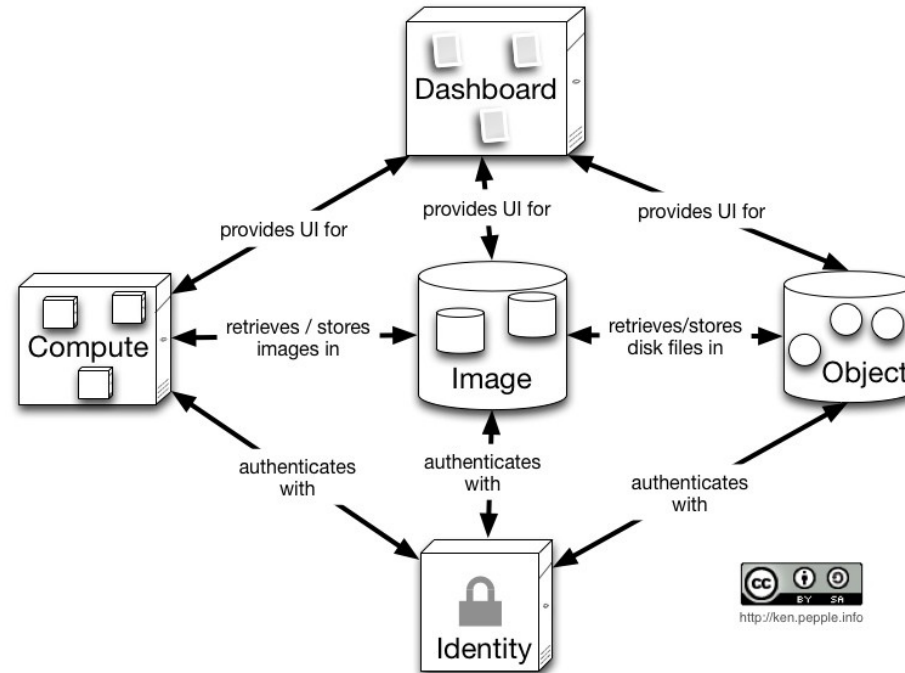
- Command-line interfaces (nova, neutron, swift, and so on)
- Cloud Management Tools (Rightscale, Enstratus, and so on.)
- GUI tools (Dashboard, Cyberduck, iPhone client, and so on.)



“Core”^{*} components

* The actual definition of what is OpenStack Core is currently a hot debate

Components (Essex release)



Identity (Keystone)

- Identity
 - Projects (sometimes called tenants)
 - Users
 - Quota
- Token
 - Single Sign-on across OpenStack components
- Catalog
 - Lists components endpoints
- Policy
 - Authorization

Compute (Nova)

- Provision and manage Virtual Machines
- Multiple hypervisor support
 - KVM
 - Xen/XenServer
 - VMWare
 - Hyper-V
 - LXC
 - Docker
 - Baremetal (Ironic)
- *Amazon EC2*

Image (Glance)

- Manages virtual machine images and snapshots
- Supports multiple formats
 - QCOW2
 - Raw
 - ISO
 - VDI (VirtualBox)
 - VHD (Hyper-V)
 - VMDK (VMWare)
 - AMI (Amazon)

Networking (Neutron)

- Virtual layer 2 networks
- IP address management
- Floating IP support (NAT)
- Firewall
 - Security groups
 - FWaaS
- Virtual routers
- Load-balancing
- VPN as a Service
- *Amazon VPC*

Object Storage (Swift)

- Objets are replicated between servers
- Eventual consistency
- Usage
 - VM images and snapshots (glance)
 - Backups
 - Application files (pictures, music, e-mails, ...)
 - Archives
- *Amazon S3*

Additional Components

Block Storage (Cinder)

- Block level storage (ie. disks)
- Backends
 - iSCSI
 - Ceph
 - GlusterFS
- Snapshots
- *Amazon EBS*

Dashboard (Horizon)

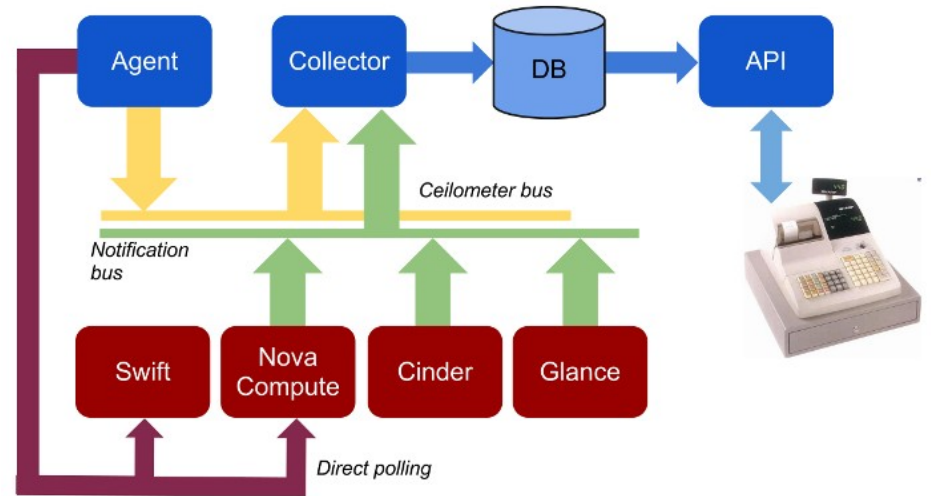
- Web interface
- Available to users and operators
- Django application

Orchestration (Heat)

- Template based resource provisioning
 - AWS Cloud Formation
 - HOT
- Autoscaling

Telemetry (Ceilometer)

- Collects metering data (CPU, Bandwidth, I/O, ...)
- And stores them (SQL DB, MongoDB, Hbase)
- Alarming



Source : <https://julien.danjou.info/blog/2012/openstack-metering-ceilometer>

Database (Trove)

- Supports both relational and noSQL databases
- Replication
- Sharding
- Backup and restore
- MySQL, Postgresql, Apache Cassandra, MongoDB, Couchbase, Redis

Data Processing (Sahara)

- Big Data processing
 - Map Reduce
- Hadoop, Apache Spark
- *Amazon Elastic MapReduce*

Baremetal (Ironic)

- PXE provisioning
- IPMI support
- Deploy physical servers just like VMs
- Used by TripleO (Openstack on Openstack)

How do you use an OpenStack cloud ?



From the cloud user point of view

- Horizon web GUI
- REST API
- Software which supports OpenStack APIs
 - Ansible
 - Vagrant
 - Salt
 - ...

Horizon Demo

OK, enough GUI for today !

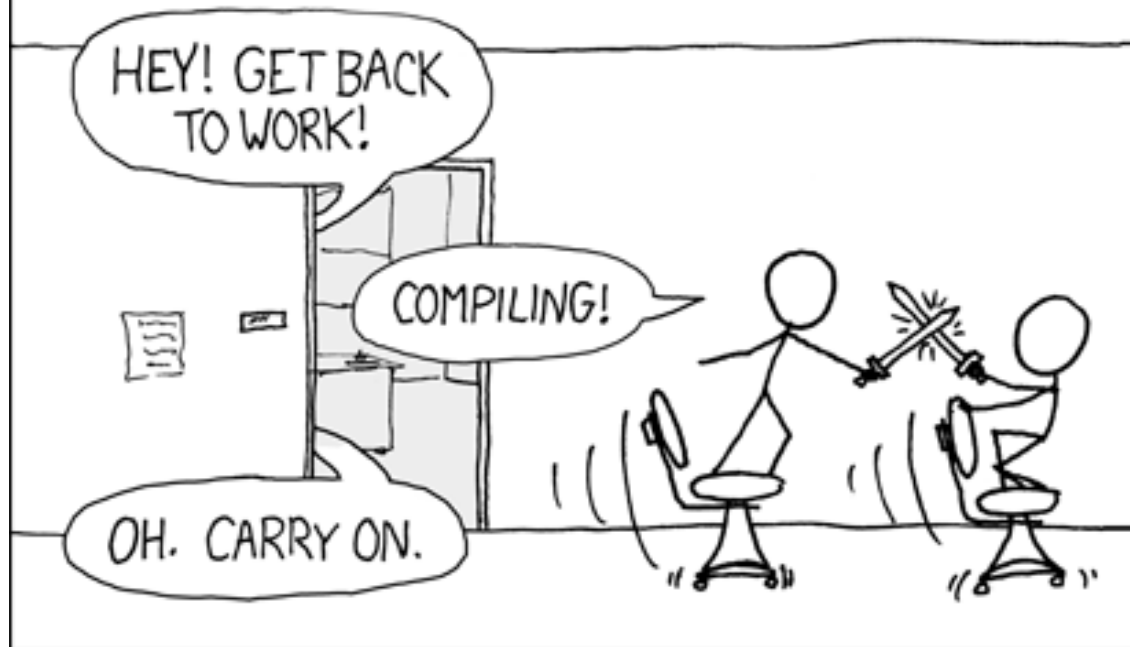


OpenStack CLI

- Packages python-*client
- Unified CLI in progress (python-openstackclient)
 - Doesn't yet support Neutron...
- Make use of OS_* environment variables

CLI Demo

THE #1 PROGRAMMER EXCUSE
FOR LEGITIMATELY SLACKING OFF:
"MY CODE'S COMPILING."




```
#!/usr/bin/env python

import os

# Nova client setup
from novaclient.v1_1 import client as nova_client
nova = nova_client.Client(os.environ['OS_USERNAME'], os.environ['OS_PASSWORD'], os.environ['OS_TENANT_NAME'],
    os.environ['OS_AUTH_URL'], service_type='compute')

# Display currently running instances
print nova.servers.list()

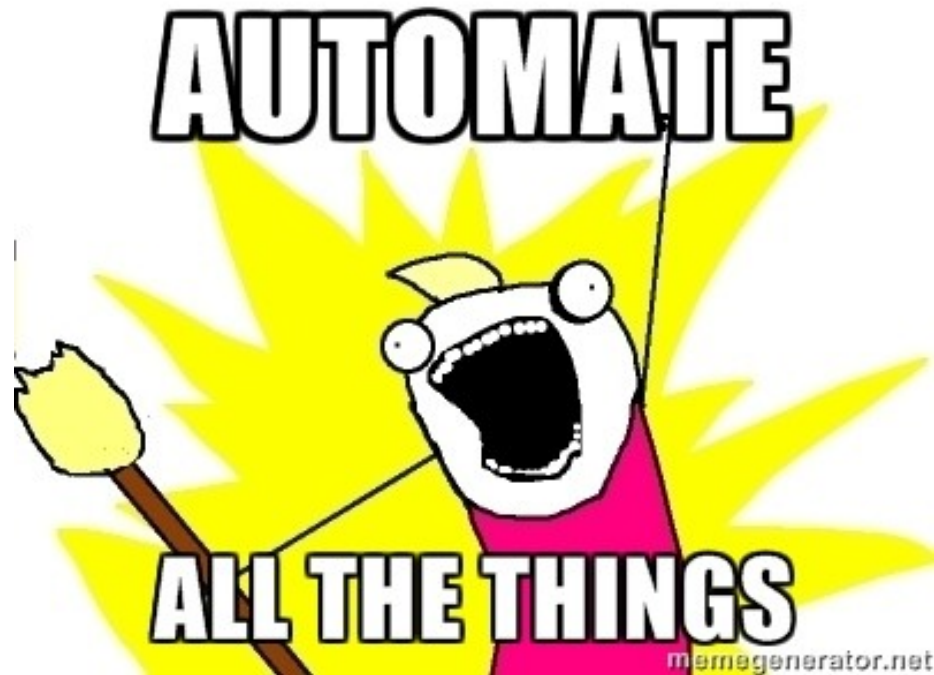
# Create a new instance
instance = nova.servers.create(
    name = 'fourth_server',
    image = '390ebdbe-6473-4166-b3da-4ef753f09eef',
    flavor = 3, # m1.medium
    key_name = 'francois',
    nics = [{'net-id': 'e4e7cc83-c41a-4b76-9b99-804063ef7170'}])

# Wait during instance boot
while True:
    instance = nova.servers.get(instance)
    if instance.status == 'ACTIVE':
        break

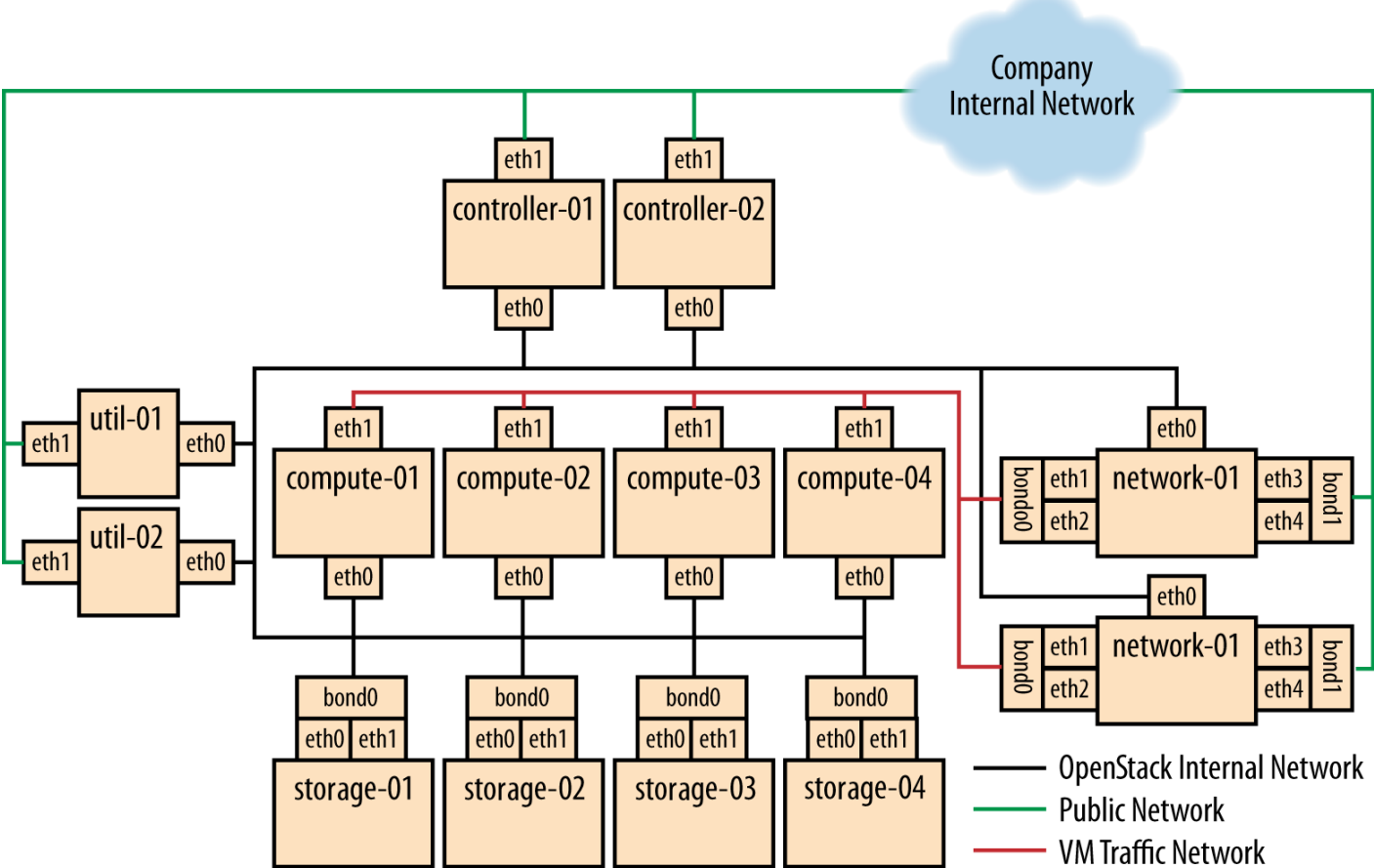
# Display the serial console output
print instance.get_console_output()

# Get rid of it, it's just a demo after all!
instance.delete()
```

OK, I'm convinced!
So you do I deploy my own OpenStack Cloud ?



Example Architecture



Deploy your own OpenStack cloud

- The OpenStack Installation Guide for Ubuntu 14.04 is 162 pages long !
- For test/dev purposes
 - DevStack is a bunch of well-documented shell scripts
- For production, use your favorite configuration management tool
 - Puppet (module puppetlabs-openstack is a good start)
 - Chef
 - Ansible
 - Juju
- Add bunch of commercial tools exists as well

Devstack Demo

- Install a Linux distribution (Ubuntu 14.04, Fedora 20 or CentOS/RHEL 6.5)

```
git clone https://git.openstack.org/openstack-dev/devstack
cd devstack
vi localrc
./stack.sh
```

Devstack Demo

Puppet Modules

- <https://wiki.openstack.org/wiki/Puppet-openstack>
- Collaborative project
- Supported platforms
 - Fedora 18 / 19 / 20
 - RHEL 6.4 / 6.5 / 7
 - Ubuntu 12.04 (Precise) / 14.04 (Trusty)
 - Debian 7.0 (Wheezy)
- Puppet 2.7 and greater

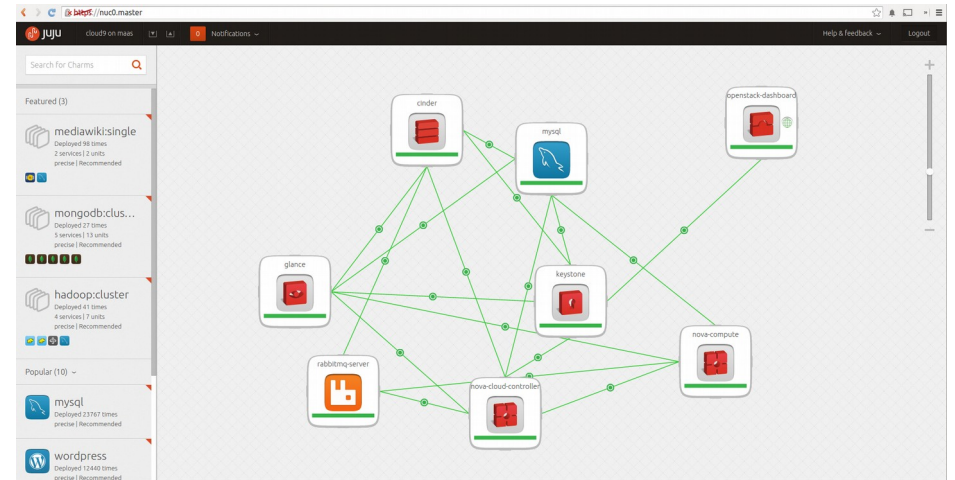
ceilometer ceph cinder
designate glance heat
horizon ironic keystone
manila neutron nova sahara
swift tempest trove tuskar
vswitch

Puppet Composition Layer

- There are (too) many...
- Here's the one from Puppet Labs
- <https://github.com/puppetlabs/puppetlabs-openstack>
- Role classes
 - `openstack::role::controller`
 - `openstack::role::network`
 - `openstack::role::compute`

Juju

- Point & Click web interface
- Uses MaaS
 - Ubuntu's baremetal provisioning tool
- Handle upgrades as well



Takeaways

- Empower your developers with self-service APIs
- OpenStack glues together different pieces of infrastructure
 - Hypervisors
 - Networking
 - Storage
- Automate the cloud installation...
 - ...or you'll get crazy !

Thanks!

- Q&A

francois.deppierraz@nimag.net