

Stack Attack

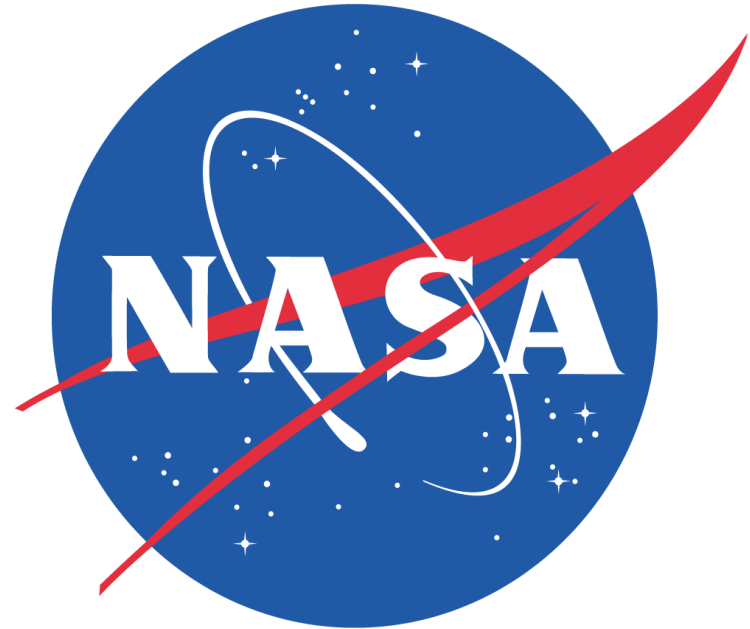
an Openstack primer



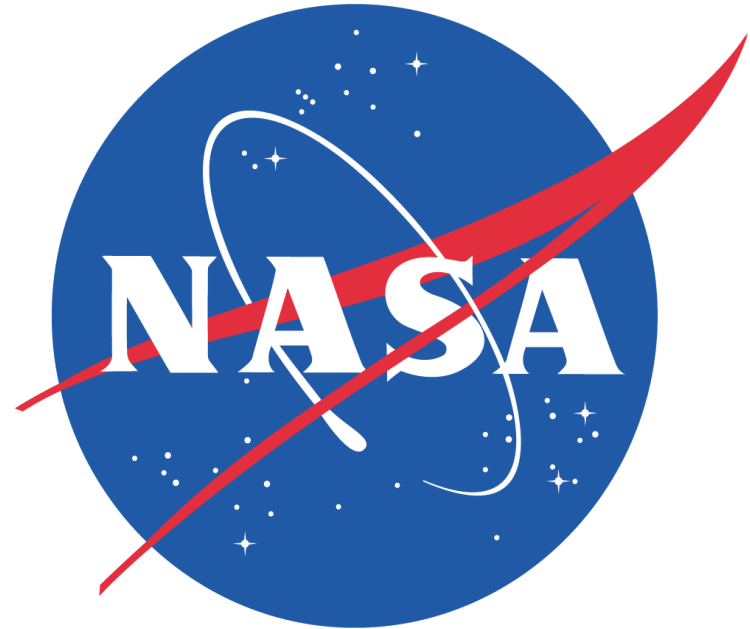
openstack™
CLOUD SOFTWARE

Jacob Melton
VTLUUG Treasurer

Project origins



Project origins



Feb 2010, First release of
Openstack, Codename
Austin

Over 100 developers

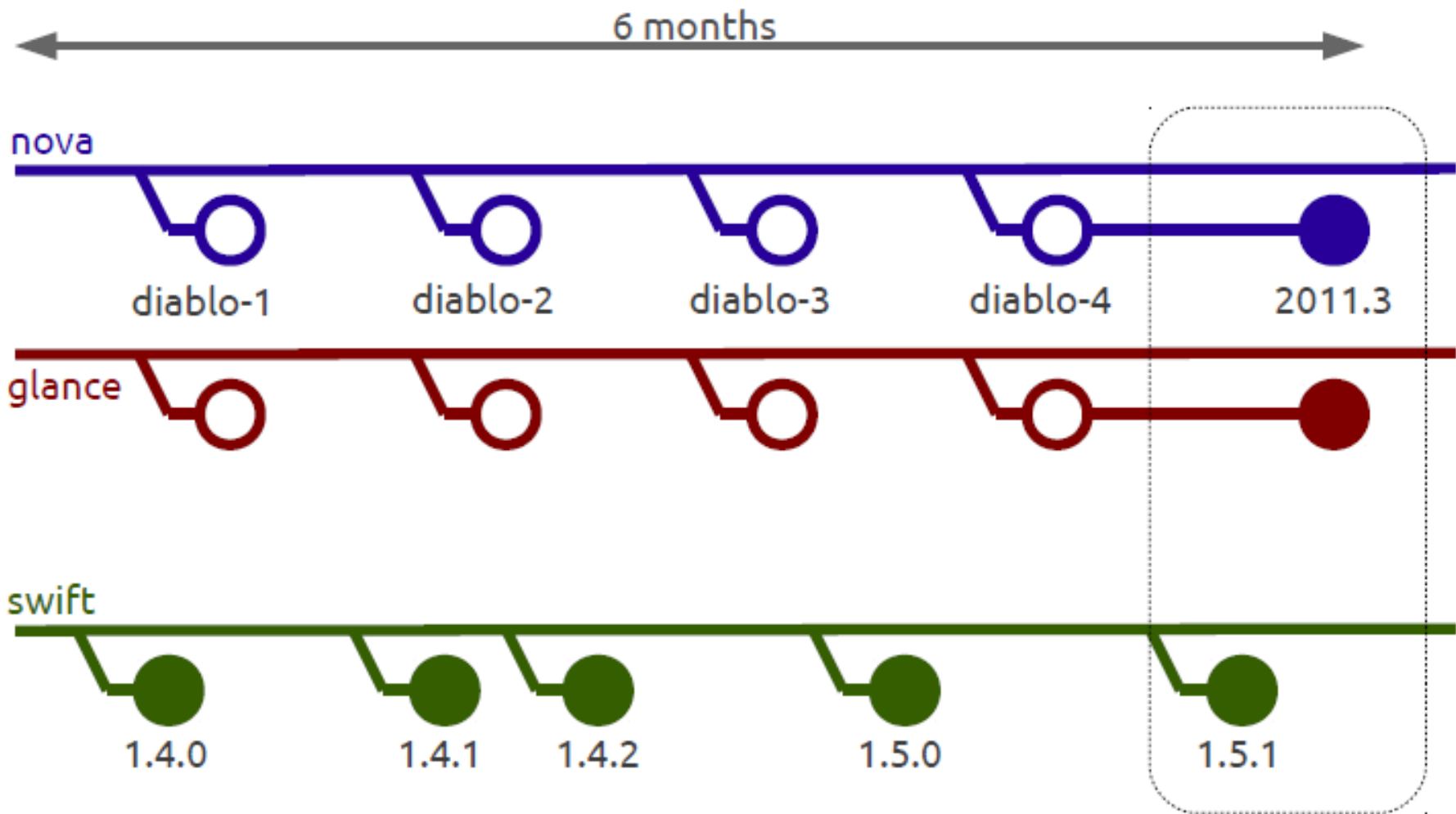




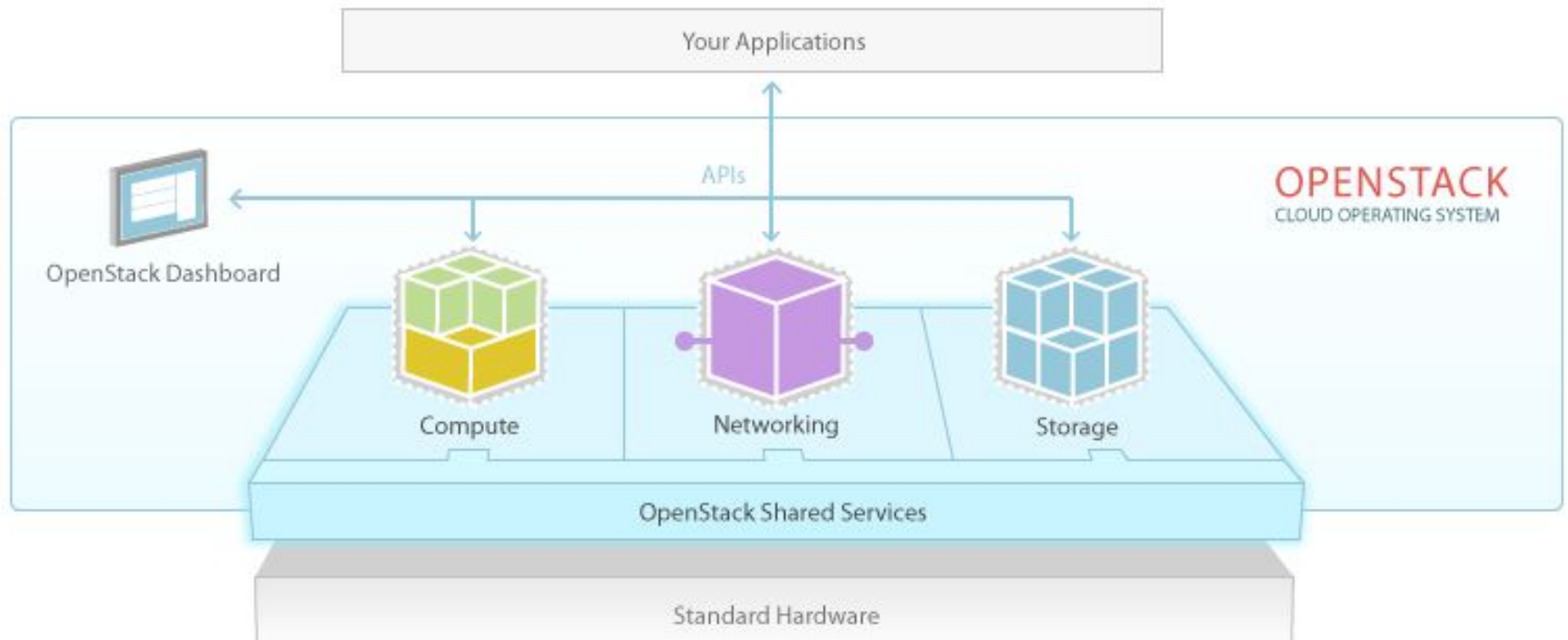
Design and Coding Standards

- ◆ Scalability and elasticity
- ◆ Asynchronous
- ◆ Horizontally scalable
- ◆ Share-nothing (or sharding)
- ◆ Distribute everything
- ◆ Accept eventual consistency
- ◆ Test everything
- ◆ Enforce PEP-8

Release Structure



Project Structure



Dashboard (Horizon)

- ◆ Django based web frontend for the rest of the Openstack components
- ◆ Utilizes the API interfaces and presents them in a simple web panel.
- ◆ Out-of-the-box support for all core OpenStack projects.

<https://www.youtube.com/watch?v=p4eW78gHfCg>



Shared Services

- ◆ Identity Service (Keystone)
- ◆ Image Service (Glance)
- ◆ Telemetry Service
(Ceilometer)
- ◆ Orchestration Service (Heat)



Identity Service (Keystone)

- ◆ Provides Identity, Token, Catalog and Policy services
- ◆ Provides user management, keeps track of permissions and the like
- ◆ Provides Service catalog
 - ◆ What services are available
 - ◆ Where are their API endpoints
 - ◆ Who can access the service



Keystone Architecture

Why make a slide when there is a wonderful webpage already in existence!

<http://docs.openstack.org/developer/keystone/architecture.html>

Keystone Workflow

http://docs.openstack.org/havana/install-guide/install/apt/content/figures/5/figures/SCH_5002_V00_NUAC-Keystone.png



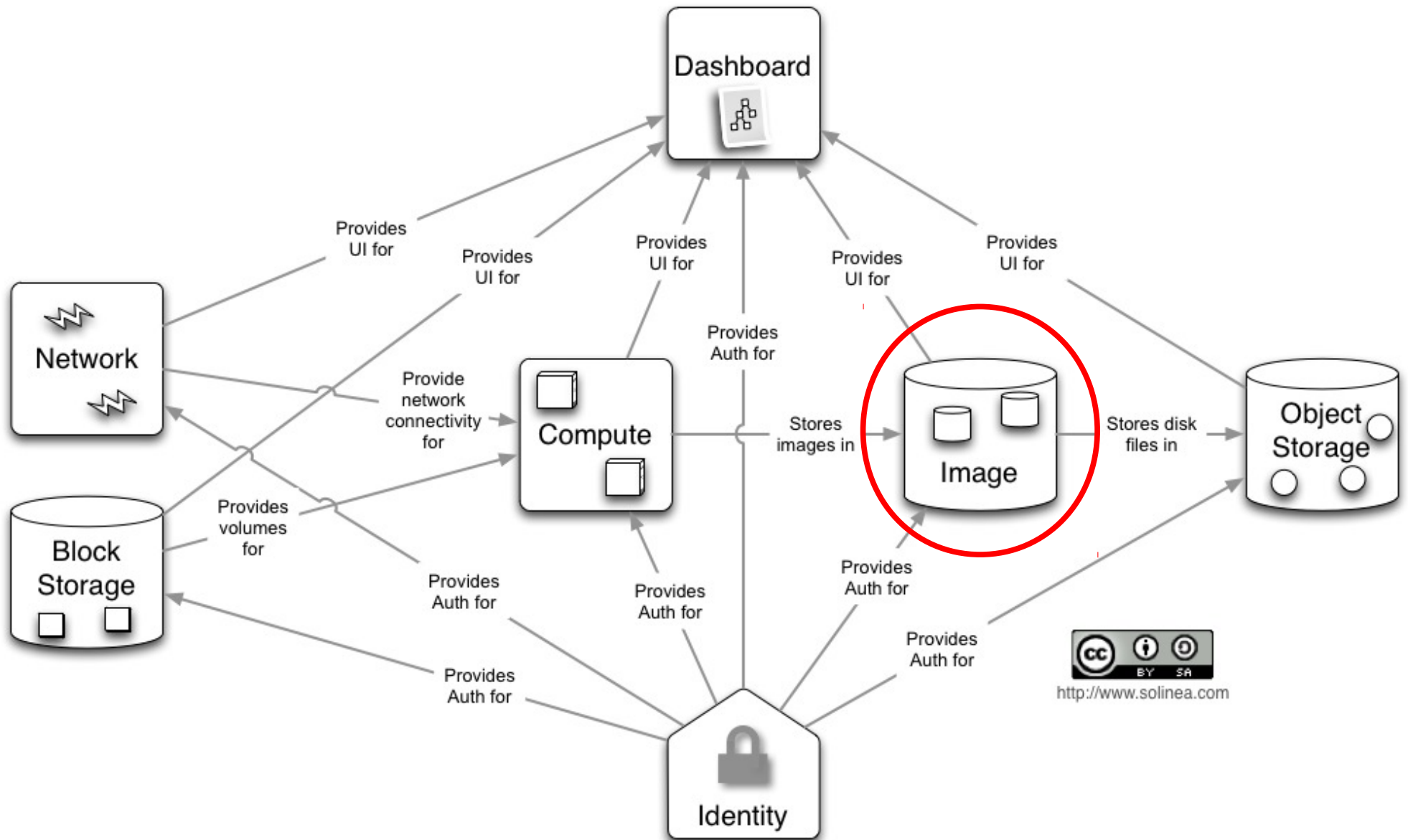
Shared Services

- ◆ ~~Identity Service (Keystone)~~
- ◆ Image Service (Glance)
- ◆ Telemetry Service
(Ceilometer)
- ◆ Orchestration Service (Heat)

Image Service (Glance)

- ◆ Provides discovery, registration and delivery services for disk and server images
- ◆ Administrators can create base templates from which their users can start new compute instances
- ◆ Users can choose from available images, or create their own from existing servers
- ◆ Snapshots can also be stored in the Image Service so that virtual machines can be backed up quickly
- ◆ Supports Raw, Machine, VHD, VDI, qcow2, VMDK, and OVF images

Image Service (Glance)





Shared Services

- ◆ ~~Identity Service (Keystone)~~
- ◆ ~~Image Service (Glance)~~
- ◆ Telemetry Service
(Ceilometer)
- ◆ Orchestration Service (Heat)

Telemetry Service (Ceilometer)

- ◆ The OpenStack Telemetry service aggregates usage and performance data across the services deployed in an OpenStack cloud
- ◆ Provide efficient collection of metering data, in terms of CPU and network costs.
- ◆ Allow deployers to integrate with the metering system directly or by replacing components.
- ◆ Data may be collected by monitoring notifications sent from existing services or by polling the infrastructure.
- ◆ Allow deployers to configure the type of data collected to meet their operating requirements.
- ◆ The data collected by the metering system is made visible to some users through a REST API.
- ◆ The metering messages are signed and non-repudiable.



Ceilometer Architecture

Woop woop for documentation.

<http://docs.openstack.org/developer/ceilometer/architecture.html>



Shared Services

- ◆ ~~Identity Service (Keystone)~~
- ◆ ~~Image Service (Glance)~~
- ◆ ~~Telemetry Service
(Ceilometer)~~
- ◆ Orchestration Service (Heat)

Orchestration Service (Heat)

- ◆ Heat is a service to orchestrate multiple composite cloud applications using the AWS CloudFormation template format, through both an OpenStack-native REST API and a CloudFormation-compatible Query API.
- ◆ Heat primarily manages infrastructure, but the templates integrate well with software configuration management tools such as Puppet and Chef.
- ◆ Multiple components
 - ◆ heat – CLI tool for communicating to heat-api
 - ◆ heat-api – OpenStack-native REST API that processes API requests by sending them to the heat-engine over RPC.
 - ◆ heat-engine – orchestrate the launching of templates and provide events back to the API consumer.



Heat example template

https://github.com/openstack/heat-templates/blob/master/cfn/F19/WordPress_NoKey.yaml

Shared Services

- ◆ ~~Identity Service (Keystone)~~
- ◆ ~~Image Service (Glance)~~
- ◆ ~~Telemetry Service
(Ceilometer)~~
- ◆ ~~Orchestration Service (Heat)~~



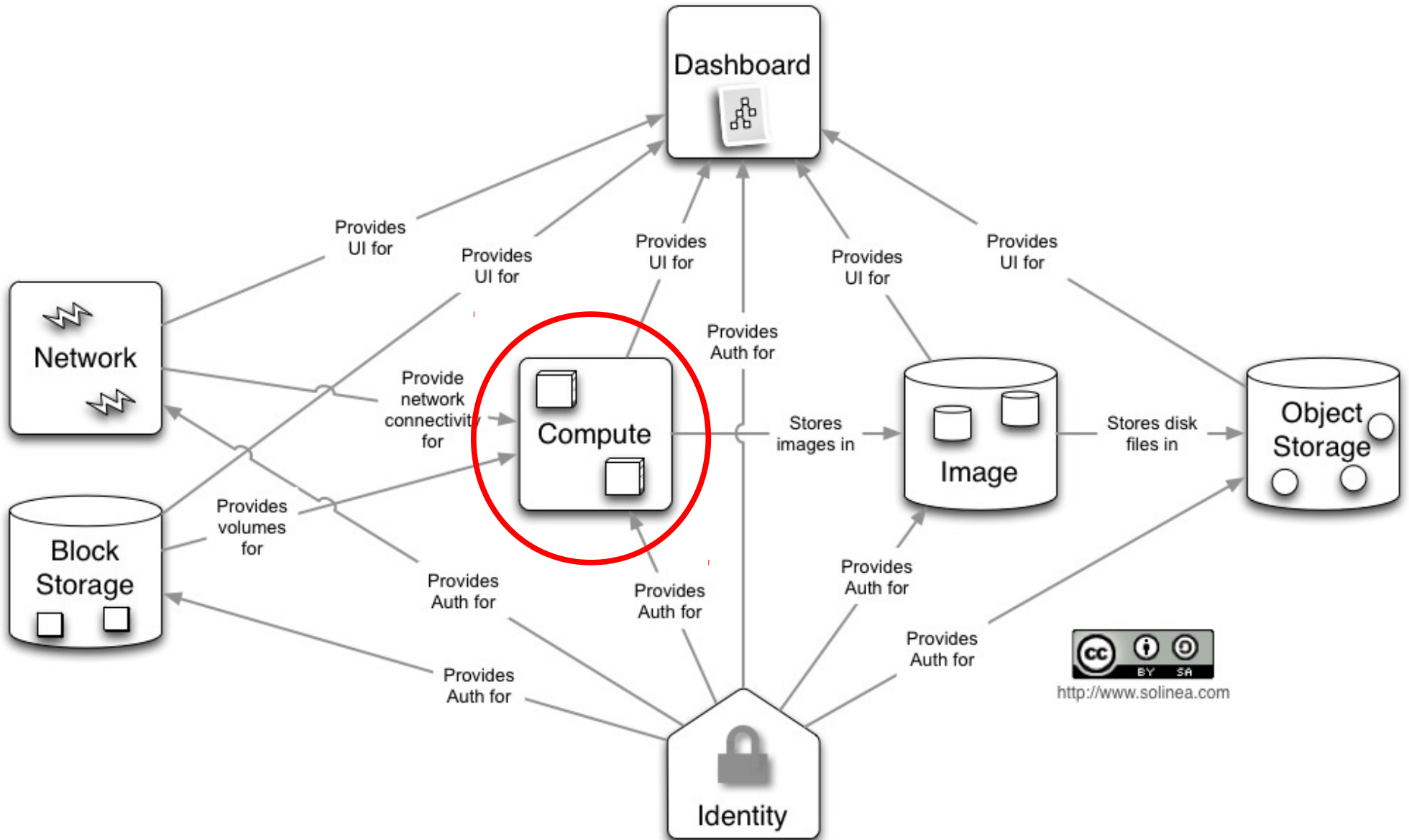
All those other services

- ◆ Compute Service (Nova)
- ◆ Networking Service (Neutron)
- ◆ Block Storage (Cinder)
- ◆ Object Storage (Swift)

Compute Service (Nova)

- ◆ Component based architecture: Quickly add new behaviors
- ◆ Highly available: Scale to very serious workloads
- ◆ Fault-Tolerant: Isolated processes avoid cascading failures
- ◆ Recoverable: Failures should be easy to diagnose, debug, and rectify
- ◆ Open Standards: Be a reference implementation for a community-driven api
- ◆ API Compatibility: Nova strives to provide API-compatible with popular systems like Amazon EC2

Compute Service (Nova)





Nova Architecture

<http://docs.openstack.org/developer/nova/devref/architecture.html>

Virtual Machine States and Transitions

<http://docs.openstack.org/developer/nova/devref/vmstates.html>



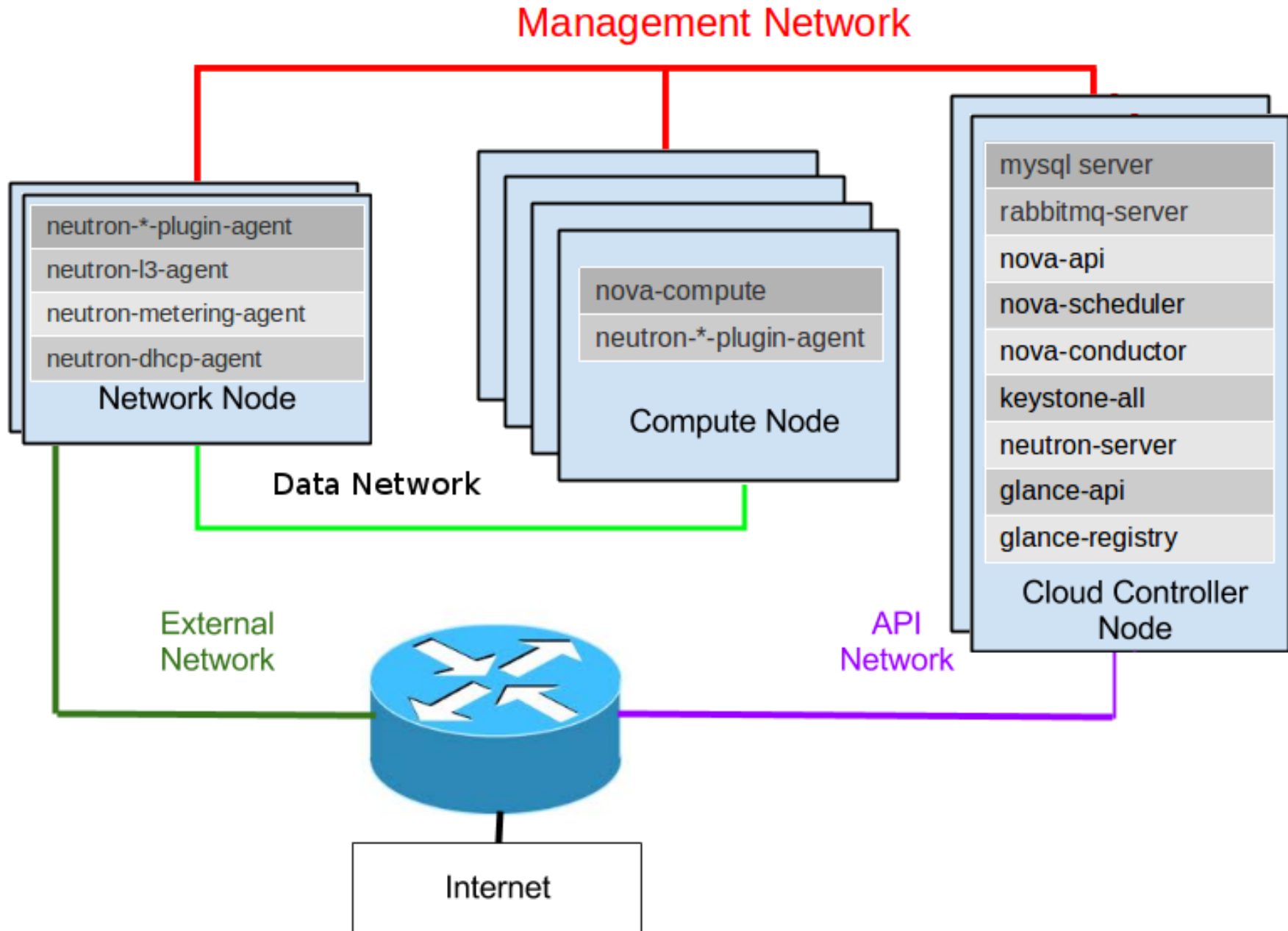
All those other services

- ◆ ~~Compute Service (Nova)~~
- ◆ Networking Service (Neutron)
- ◆ Block Storage (Cinder)
- ◆ Object Storage (Swift)

Networking Service (Neutron)

- Provide “network connectivity as a service” between interface devices (e.g., vNICs) managed by other Openstack services (e.g., nova)
- Neutron is relatively new to the project
- Nova used to provide simple networking and still can, Neutron is meant to replace that functionality
- Allows for more complex network topologies
- https://wiki.openstack.org/wiki/Neutron#Using_Neutron

Neutron Architecture



All those other services

- ◆ ~~Compute Service (Nova)~~
- ◆ ~~Networking Service (Neutron)~~
- ◆ Block Storage (Cinder)
- ◆ Object Storage (Swift)

Block Storage (Cinder)

- ◆ Provides an infrastructure for managing volumes in OpenStack
- ◆ Time to steal some slides
- ◆ <https://wiki.openstack.org/w/images/3/3b/Cinder-grizzly-deep-dive-pub.pdf>

All those other services

- ◆ ~~Compute Service (Nova)~~
- ◆ ~~Networking Service (Neutron)~~
- ◆ ~~Block Storage (Cinder)~~
- ◆ Object Storage (Swift)

Object Storage (Swift)

- ◆ Swift is not a traditional file system, but rather a distributed storage system for static data such as virtual machine images, photo storage, email storage, backups and archives
- ◆ Objects and files are written to multiple disk drives spread throughout servers in the data center, with the OpenStack software responsible for ensuring data replication and integrity across the cluster
- ◆ Storage clusters scale horizontally simply by adding new servers
- ◆ <http://docs.openstack.org/training-guides/content/module003-ch003-obj-store-capabilities.html>

Object Storage (Swift)

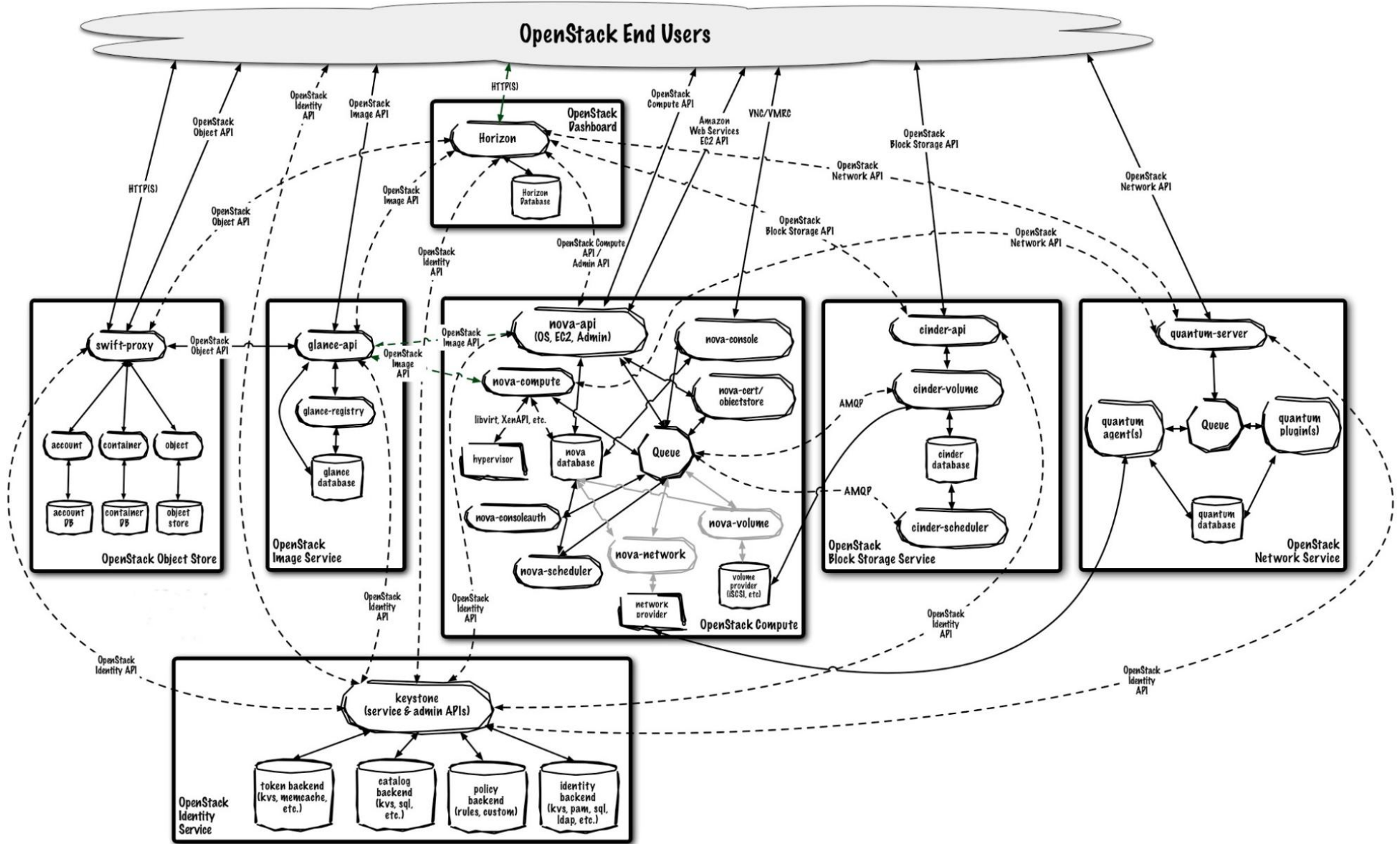
- ◆ I dont totally understand Swift yet.
- ◆ So youtube to the rescue, if you are interested in the how and what of Swift whatch the video below.
- ◆ <https://www.youtube.com/watch?v=G0zz6ahZjnE>



All those other services

- ◆ ~~Compute Service (Nova)~~
- ◆ ~~Networking Service (Neutron)~~
- ◆ ~~Block Storage (Cinder)~~
- ◆ ~~Object Storage (Swift)~~

The big picture





What the future holds

- ◆ Database Service (Trove), coming out with Icehouse release.
- ◆ Bare Metal (Ironic), under development
- ◆ Queue Service (Marcconi), ^
- ◆ Data Processing (Savannah), ^
- ◆ See <https://www.openstack.org/software/roadmap/>
- ◆ Status on next release
<http://status.openstack.org/release/>

What they are using



What they are using



Eventlet

Paste.deploy

webob



routes

What they are using



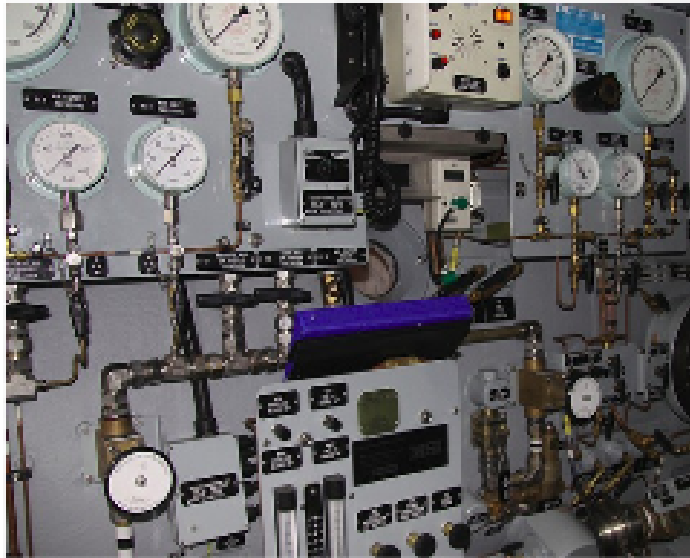
Carrot / AMQPlib

(Kombu ?)

SQLAlchemy

& sqlalchemy-migrate

What they are using



boto

M2Crypto

gflags

Installing Openstack

- ◆ Lots of documentation, lots of guides
- ◆ Devstack
- ◆ Links
 - ◆ <http://docs.openstack.org/havana/install-guide/install/yum/content/>
 - ◆ <http://devstack.org/>



Questions/Comments