



Automated and sustainable IT



Automated



Swedish
operations
and
support



No lock-in



Security

Joakim Öhman



- 20 years of datacenter
 - Started with Unix
 - Storage and HA
 - Built two hosting platforms from scratch
 - Automation
 - Open source
 - IaaS
 - PaaS
 - CaaS





We believe automation and
open source makes IT awesome

Public Cloud made in Sweden



cloudops professional services
Automates from metal to code



IT operation services



Jelastic PaaS



Openstack IaaS



Other public
or private
IaaS



Automated



Swedish
operations
and
support



No lock-in



Security

Why Openstack



1. Large community
2. Fastest growing infrastructure project
3. Opensource
4. Very modular
5. We like Python
6. Built for commodity HW



Our Openstack journey



- First deploy Havana
 - January 2014
- Upgrade to Icehouse
 - June 2014
- Production
 - August 2014

The logo for the Havana OpenStack release. It features the word 'HAVANA' in a large, grey, sans-serif font. Below it, the text 'THE EIGHTH OPENSTACK RELEASE' is written in a smaller, grey, sans-serif font. The word 'EIGHTH' is highlighted in yellow. Four small yellow arrows point to the right, positioned between the letters of 'HAVANA'.

Deploy



Home brew Ansible deploy

- Custom for our needs
 - Forced us to learn
 - Did not take to long
- Very easy to manage
- A lot of work upgrading
 - Revue all release notes
 - Rebase all config files
- Lag 6 months after release



Platform



- Single datacenter
- Nova
 - White box servers
- Neutron
 - Native Neutron network
 - LBaaS v1 (no HA)
 - FWaaS
 - Hot standby



Cinder



- Centralized storage is scary
 - Did not have the resources to assure stability
- Wanted to avoid spinning disk
- Wanted a cloud design
 - Horizontal scaling etc.
- Went for Solidfire
 - All SSD
 - Compression and de-dup
 - Based on commodity HW
 - Horizontal scaling
 - PTL of Cinder



Ephemeral / boot disk



- Based on Solidfire volumes
 - /var/lib/nova/instances
- Possible to fail over compute nodes
- SAN storage performance



Swift

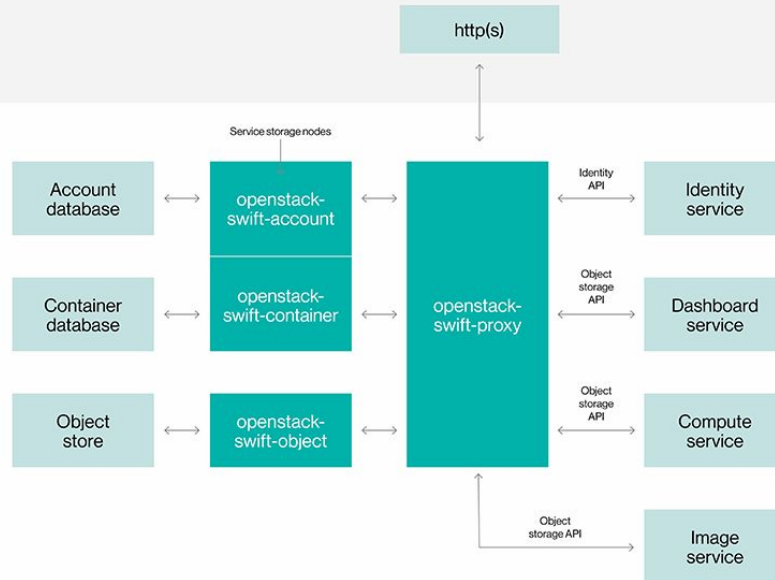


- Native Swift
- 3 copies (started with 2, bad idea)
- Proven to be very robust
 - Storage nodes are completely independent from each other
- Performance issues with many objects
 - Recommended not to have more than 1 million object per container
 - We had containers with 900 million objects
 - SSD for accounts and container DB
 - SSD cached object disks

Swift under the hood



OpenStack Object Storage Service



SOURCE: REDHAT.ORG

©2017 REDHAT. ALL RIGHTS RESERVED. Red Hat

The ring



swift-ring-builder

- Created and modifies the ring

```
# swift-ring-builder object.builder add --region 1 --zone 1 --ip 1.2.3.4 --port 6000 --replication-ip  
1.2.3.4 --replication-port 6000 --device d2 --weight 12000
```

```
# swift-ring-builder object.builder set_weight z1-1.2.3.4:6002/d12 0 (before removing device)
```

```
# swift-ring-builder object.builder rebalance (compiles ring files)
```

Hash keys in /etc/swift/swift.conf (don't lose)

Find objects



Find account DB

```
# swift-get-nodes ./account.ring.gz AUTH_c645cb4a8e994e4a91b06dd67d71be5c
```

Find container DB

```
# swift-get-nodes ./container.ring.gz AUTH_c645cb4a8e994e4a91b06dd67d71be5c test
```

Find object

```
# swift-get-nodes ./object.ring.gz AUTH_c645cb4a8e994e4a91b06dd67d71be5c test  
login-background.png
```

Objects



swift-object-info

- See object info about a file
- Uses file extended attributes and the ring file
- Normal copy do not copy xattr

Etag

- MD5 checksum of file

Object hash file

- `/srv/node/d8/objects/182891/ hashes.pkl`

Issues we have seen

- Performance limitations in early versions of Openvswitch
 - Network node
- Qemu Suspend/Resume issues
 - The root cause was a bug in openvswitch that was triggered by multiple network port removals. The fault made a number of important processes to end up in state D and the only way to recover was a reboot.
- Ceilometer performance issues
 - First version with MongoDB backend
 - Disabled ceilometer for swift



Platform evolves



Openstack upgrades

- Juno
- Kilo
- Liberty

Second datacenter GA

- Q1 2017

Planning to deploy third datacenter and Mitaka



Dead end



- FWaaS deprecated
 - Not to bad
- No upgrade path to LBaaS v2
 - A bigger issue
- Moving to fully distributed SDN
 - Mission impossible

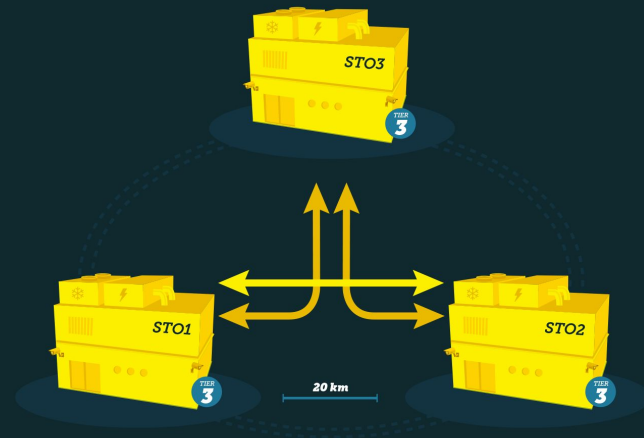


Greenfield deploy



Build for enterprise requirements

- Resilient
 - 1 Region, 3 AZ's
 - Three DC's
- Secure
 - Best conditions for secure operations
- Scalable
 - Scalable design
- AWS pricing level
- Openstack Pike



Deploy



Openstack Ansible (OSA)

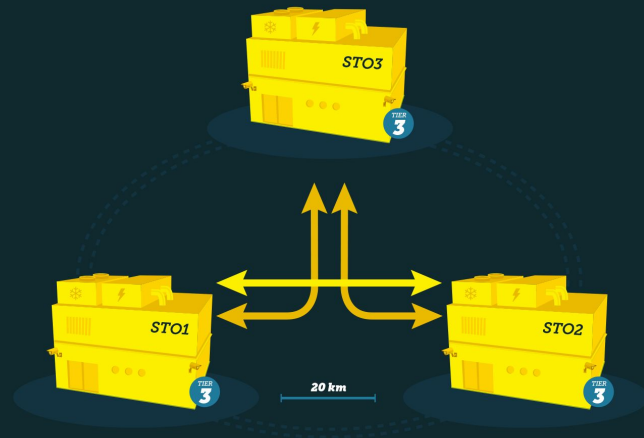
- No more rebase of config files :)
- A large community
- More complicated
- Works out of the box for vanilla deploy
- We have spent a lot of time on missing projects and integrations
- Lag 12 at least months after release



Networking is key



- Distributed network services
 - Anycast routing
 - Distributed (on compute nodes)
 - DHCP
 - Metadata
 - NAT
- Dataplane not dependent of management
- As much in ASICs as possible
- 100Gb Network fabric (SDN)
 - Spine leaf design
 - Cisco ACI



Storage



- Swift
 - One copy in each DC
- Cinder
 - One storage cluster per DC
 - No cross attach
- Ephemeral
 - Local SSD
 - Different SLA



Security



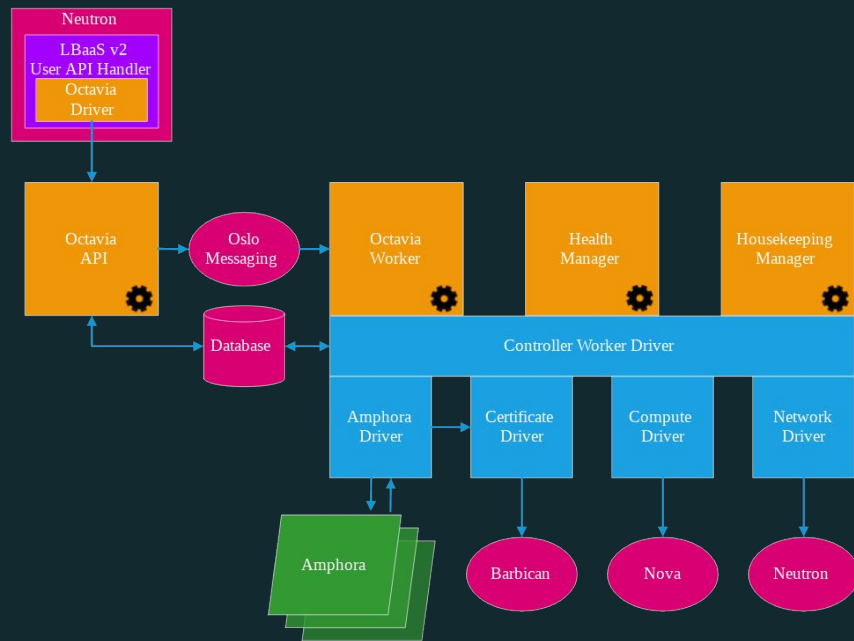
- Barbican HSM service
 - Encrypted Cinder volumes
 - Certificates in Octavia
- Dedicated Compute nodes
 - “Meltdown and Specter” protection
 - Local dedicated disk
 - Kubernetes
- Inline DDoS protection
- Threat protection
- WAF



Other



- Introduced Octavia (LBaaS v2)
 - Runs on instances (Amphora)
 - VRRP for HA (Not perfect)
 - Anycast redundancy is the goal
- No FWaaS
- No Designate
 - Anyone wants to contribute?
- Cloudkitty
- Gnocchi



Issues seen in the new platform



- High load on
 - Ceilometer
 - Gnocchi
 - Cloudkitty
 - Redis
- Neutron and CDN integration complex
 - Distributed
 - Many moving parts

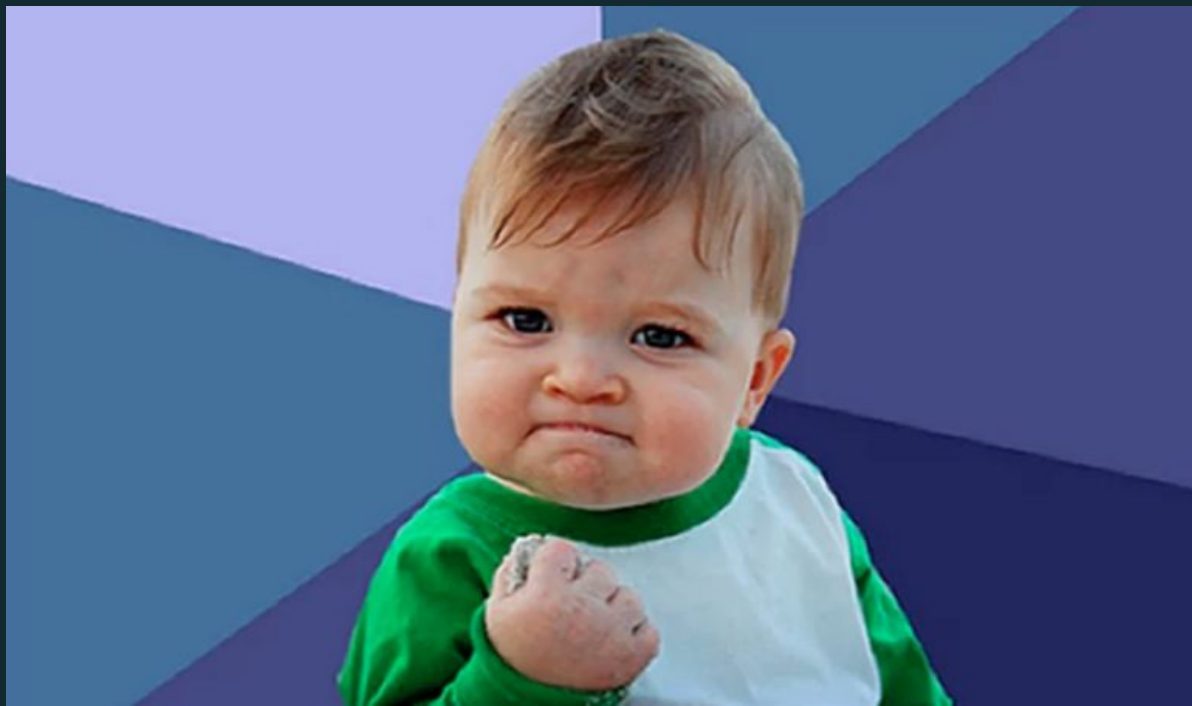
Rocky solves a lot of issues



- Improved Octavia functionality (not fully supported in Pike)
- Support for more than 26 volumes per instance
- Improved dependency handling in Openstack-Ansible



Verdict



Scalable



- Traditional operations, SMB
20:1
- Average, Enterprise
100:1
(Usually between 70:1 to 140:1)
- Best practice, Cloud platforms
2000:1 and above

Stability



A totally new platform but everything automated gave the following result:

- No configuration variations and mistakes
- Better stability from day 1
- Not a single outage caused by Openstack





Thanks for
listening