OpenStack Summit Barcelona - Oct 27,2016

Your dudes

- Sean Cohen - Product Manager OpenStack
- Sébastien Han - Principal Software Engineer, Storage Architect
- Federico Lucifredi - Product Management Director, Red Hat Ceph Storage

Realm of containers

Drives

Why are we moving toward containers?

- Packaging format and runtime
  - Dependency isolation
- Upgrade/downgrade flexibility
  - OpenStack services have independent lifecycles making it difficult to perform rolling upgrades and downgrades.
- Deployment flexibility
- Scalability (scale up services individually)
- Immutable infrastructure
- Resource constraints (cpu, memory, block IO)
- Speed

How to use them efficiently?

Entering Kubernetes with OpenStack

Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.

Container management platform:

- Self-healing
- Load balancing
- Automated rollouts and rollbacks
- Pluggable architecture:
  - Storage
  - Network
  - Containers runtime
  - Scheduling

OpenStack in Containers

Why OpenStack on Kubernetes?

Day 1: ease deployment

- OpenStack itself turned into microservice oriented architecture with its services
- Kubernetes manages applications
- Kubernetes load, driven by application needs, can vary dramatically over wide time scales (hours, days, weeks)
- Leverage the ease, scale and power of the Containers

Day 2:

- Share scheduling functions between Nova and Kubernetes

Kolla

Some background..

Let’s go back in time again:

- Started in September 2014 by Red Hat
- Announced and advertised in November 2014 during Paris OpenStack Summit
- Kolla != Magnum
- Deployment methods
  - Mesos, started in October 2015 and abandoned as an orchestration scheduler by Kolla in April 2016
  - Ansible is the de facto tool
  - Kubernetes with the kolla-kubernetes project (started in May 2016)

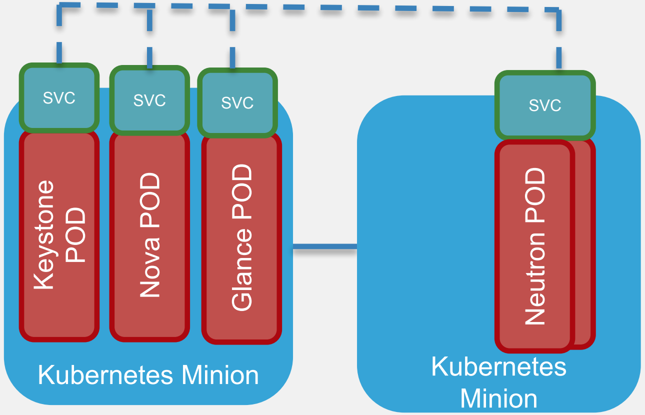
Kolla

Installation of OpenStack services as set of containers

- Solves a **manageability** and **availability** problem with the current state of the art deployment systems in OpenStack

- Containerizing OpenStack is meant to **optimize image-based management** of OpenStack
- Uses Heat Templates and YAML to define services and pods

Pods communicate using Services



IP subnets per POD

IP subnets per POD

- Deploy OpenStack using container technology **for atomic** upgrades in seconds

https://launchpad.net/kolla

Kuryr and Fuxi

Exposing the power of the cloud to containers

- Taking advantage of the pluggable nature of Kubernetes
- Network and storage “interfaces” to interact with your Cloud
- Enhance the capabilities of Kubernetes by exposing more:
    - Network drivers from Neutron - Kuryr
    - Storage drivers from Cinder - Fuxi

Ceph in Containers

Deploying Storage services with Kolla

Kolla support the containerization of the following storage services already today:

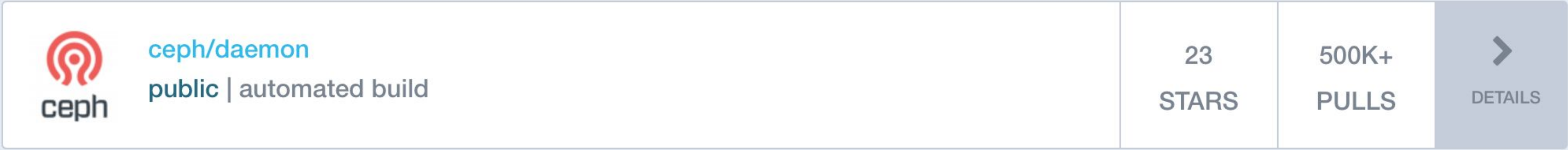
- Glance
- Cinder
- Swift
  - Kolla can deploy a full working Swift setup in either a **all-in-one** or **multinode** setup.
- Manila
  - Currently deploys manila-api, manila-scheduler and Manila-share
- Ceph
  - The out-of-the-box Ceph deployment requires 3 hosts with at least one block device each that can be dedicated for sole use by Ceph
  - Kolla also support external Ceph clusters by simply disabling Ceph deployment in /etc/kolla/global.yml
  - With tweaks to the Ceph cluster you can deploy a functional cluster with a single host and a single block device (this configuration provides no data resiliency).

ceph-docker

Containerizing Ceph daemons

The project:

- Launched on Jan 18, 2015
- Upstream project: https://github.com/ceph/ceph-docker
- Support from Hammer to the latest version of Ceph (currently Jewel)
- Wide range of distros: Ubuntu (14.04 and 16.04), Fedora (24), CentOS (7)
- Automated builds on the Docker Hub
- More than 500K+ pulls!



ceph/daemon

public | automated build

23 STARS

500K+ PULLS

Single daemon image

A single **ceph/daemon** container image is used to bootstrap a Ceph cluster with all its daemons running.

- To deploy MON, OSD, MDS, RGW, rbd-mirror and RGW-NFS the type/name of the daemon needs to be passed as an argument to ceph/daemon process
- Large variety of OSD scenarios:
  - Journal collocation
  - Dedicated journal device
  - Dmcrypt
  - Bluestore
  - Directory
- Deployment methods:
  - Ceph-ansible: containers are managed by systemd
  - Kubernetes (experimental)

Deployment tools

TripleO

OpenStack standard with Heat

- Use Heat hook that allow Heat to orchestrate container deployment
- Result:
    - You can create a containerized OpenStack solution by using TripleO
    - Using the containers from the Kolla project, we mirrored the TripleO workflow by using the undercloud (management cloud) to deploy most of the core services in the overcloud (user cloud), but now those services are containerized.

Ansible

The de-facto standard

Main purpose:

- Friendly learning curve
- “Flat” deployment, no lifecycle
- No placement intelligence or orchestration
- More traditional users, not kubernetes-ready
- Let systemd manage your containers (start/stop/restart/watchdog)

Ansible is available and ready today!

Eventually, Ansible will be used to install Kubernetes only.

Kubernetes

The future

For thugs:

- Not fully compliant yet (see next slide)
- Are you ready for containers?
- It’s all about the network

Challenges - Gap analysis

There ain’t no challenges for me bro!

Component	Requirement	Met
Network	Network Isolation	No
Network	Ability to disable network overlays and use host's network	Yes
Network	IPv6	WIP
Network	SSL terminaison	Yes
Storage	Data persistency (outside of containers)	Yes
HA	Cluster bootstrapping (boot order, node replicas, etc)	Yes
HA	Pods Monitoring	Yes
HA	Load balancing strategies	Yes
HA	Pod Fencing	No

Address the networking problems

I want my networks back bro!

- CNIs (Container Network Interface) are probably the best way to implement this
- Chicken and en egg problem
  - Sequencing issue
  - Ephemeral container to configure a Neutron?

Yo!

What about **migrating** to and from a **containerized** platform

Potential ways to address migration

From non-containerized to containerized

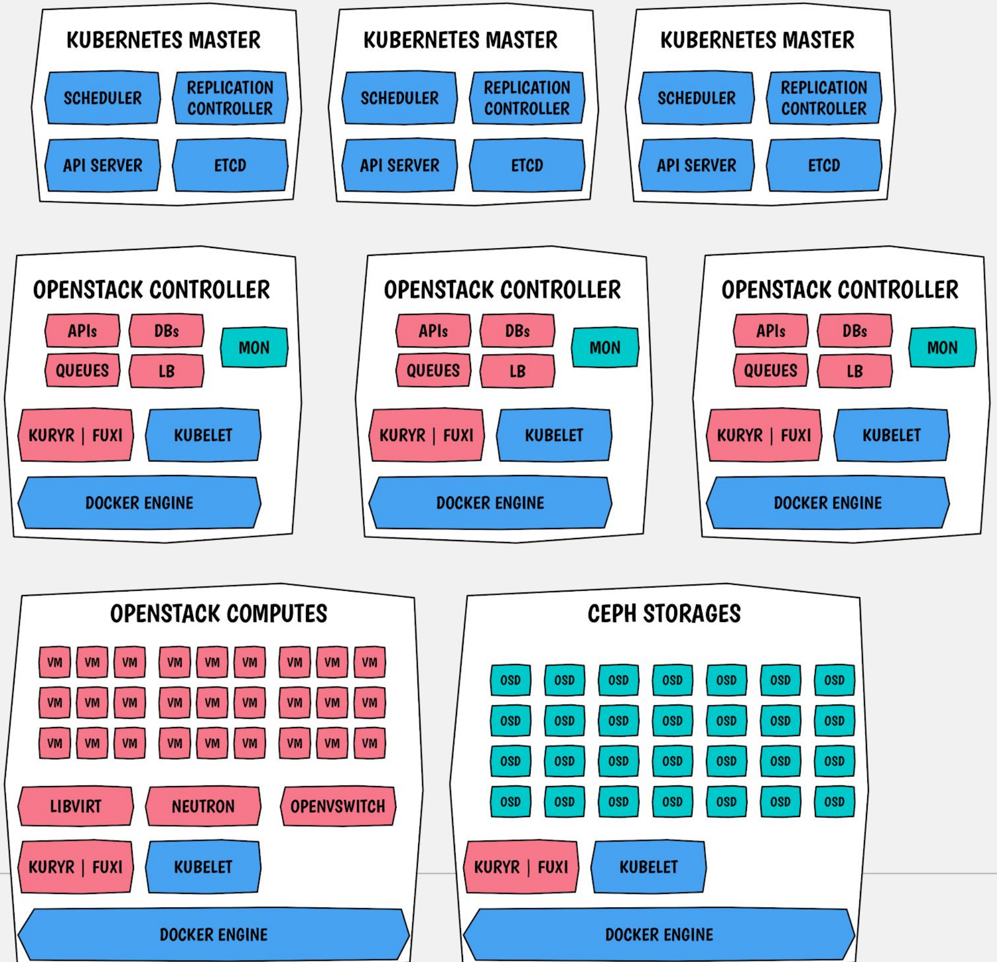
Kubernetes or not?

1. Easier if you are not ready for Kubernetes
   - Stop/disable daemon service, start the container with a systemd unit file
   - Rolling scenario
   - Ansible playbook ready for ceph-docker

2. I’m ready for Kubernetes! (and Kubernetes is ready too!)
   - Decommission non-containerized nodes
   - Add previously decommissioned nodes to Kubernetes
   - Rolling scenario, update your Kubernetes template to add more nodes

Architecture

CONTAINERIZED OPENSTACK CLOUD



🟥 OPENSTACK COMPONENT

🟦 KUBERNETES COMPONENT

🟩 CEPH COMPONENT

KUBERNETES MASTER
SCHEDULER | REPLICATION CONTROLLER
API SERVER | ETCD

KUBERNETES MASTER
SCHEDULER | REPLICATION CONTROLLER
API SERVER | ETCD

KUBERNETES MASTER
SCHEDULER | REPLICATION CONTROLLER
API SERVER | ETCD

OPENSTACK CONTROLLER
APIs | DBs | MON
QUEUES | LB
KURYR | FUXI | KUBELET
DOCKER ENGINE

OPENSTACK CONTROLLER
APIs | DBs | MON
QUEUES | LB
KURYR | FUXI | KUBELET
DOCKER ENGINE

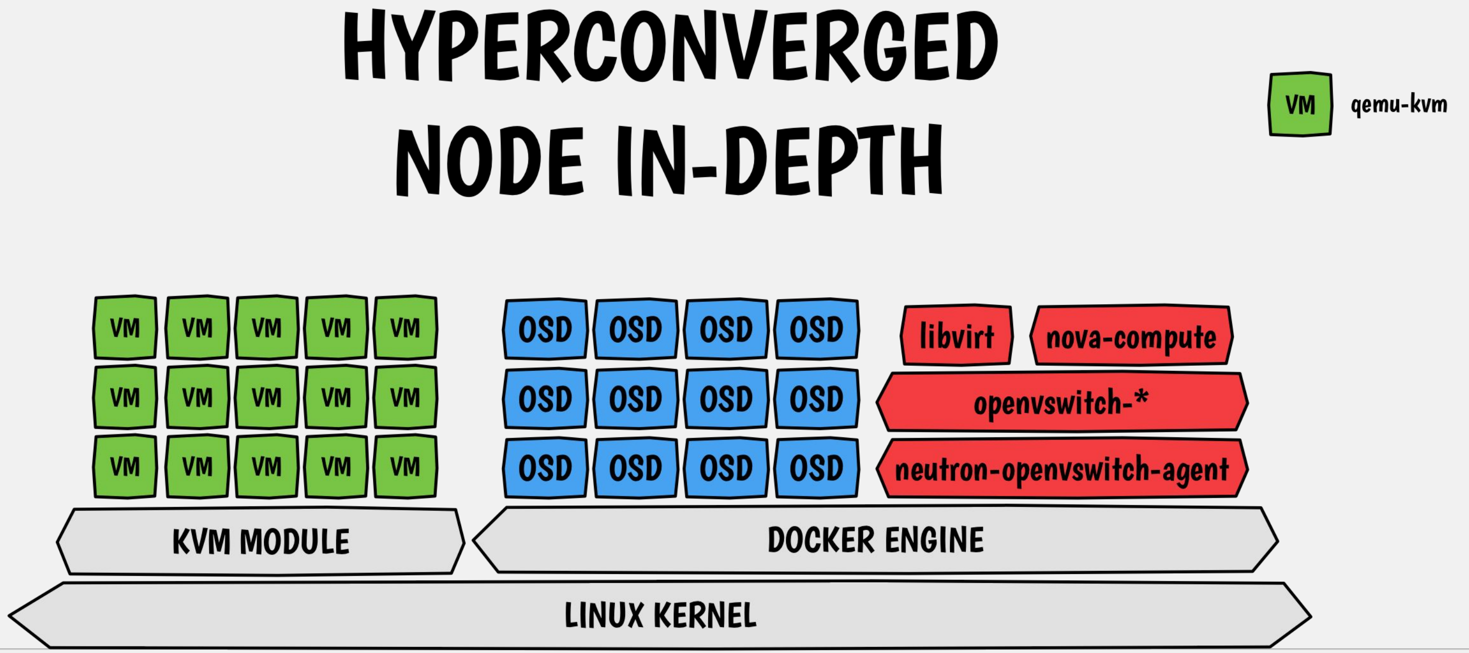
OPENSTACK CONTROLLER
APIs | DBs | MON
QUEUES | LB
KURYR | FUXI | KUBELET
DOCKER ENGINE

OPENSTACK COMPUTES
VM VM VM VM VM VM VM VM VM
VM VM VM VM VM VM VM VM VM
VM VM VM VM VM VM VM VM VM
LIBVIRT | NEUTRON | OPENVSWITCH
KURYR | FUXI | KUBELET
DOCKER ENGINE

CEPH STORAGES
OSD OSD OSD OSD OSD OSD OSD
OSD OSD OSD OSD OSD OSD OSD
OSD OSD OSD OSD OSD OSD OSD
OSD OSD OSD OSD OSD OSD OSD
KURYR | FUXI | KUBELET
DOCKER ENGINE

Hyperconverged

Hypercon... what?



What’s next?

Ceph container roadmap

The road ahead

Done:

- Ceph-docker - container images available for a year and a half
- RHCS 1.3.2: Containerized Ceph in Tech Preview since 2/29

Next:

- CI to continuously test containers
- Remove the need to have a privileged container
- QA the Kubernetes prototype

Summary

Takeaways:

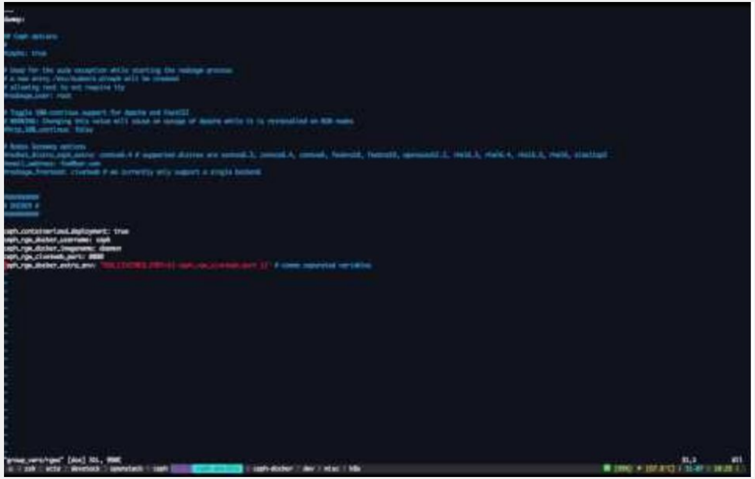
- Ansible + Kolla are excellent candidates to deploy containerized infrastructures
- Support for Ceph in containers is here
- Kubernetes is the future but not there yet
- Getting the right networking model/setup is difficult but it’s critical

Wanna hear more?

Resources

Resources:

- Ceph Docker on Docker Hub
- Ceph Docker on GitHub
- Ceph Ansible
- Ceph in Kolla
- Kolla: External Ceph
- Manila in Kolla
- Swift in Kolla
- Cinder in Kolla
- Containerized Ceph from Red hat and QCT



```
dummy:

# Ceph options
#
cephx: true

[illegible]

ceph_containerized_deployment: true
ceph_rgw_docker_username: ceph
ceph_rgw_docker_imagename: daemon
ceph_rgw_civetweb_port: 8080
ceph_rgw_docker_extra_env: [illegible] # comma separated variables
```

Video: manually
Video: with Ansible

@sebastien_han
@SeanCohen_RH
@0xF2

redhat.

THANK YOU

plus.google.com/+RedHat	facebook.com/redhatinc
linkedin.com/company/red-hat	twitter.com/RedHatNews
youtube.com/user/RedHatVideos	