OpenStack/Magnum and the CERN container service

Spyros Trigazis @strigazi

OpenStack Magnum

What is Magnum?

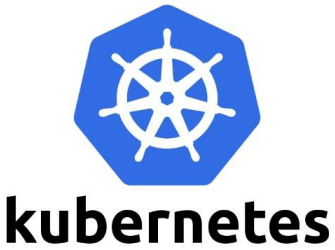
An OpenStack API service that allows creation of container clusters.

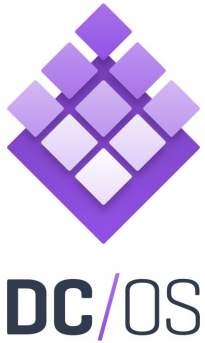
- Use your keystone credentials
- You choose your cluster type
- Multi-Tenancy
- Quickly create new clusters with advanced features such as multi-matser

Terminology (1/3): COE

Container Orchestration Engine

Examples: Docker Swarm (Mode), Kubernetes, Mesos, DC/OS





Terminology (2/3): Magnum Cluster

A Magnum cluster is composed of:

- compute instances (virtual or physical)
- neutron networks
- security groups
- cinder volumes
- other resources (eg Load Balancer)

using OpenStack Heat

- Where your containers run
- Lifecycle operations
  - Scale up/down
  - Upgrade
  - Node heal/replace
- Self contained cluster with each own monitoring, data store, additional resources

Terminology (3/3): Native Client

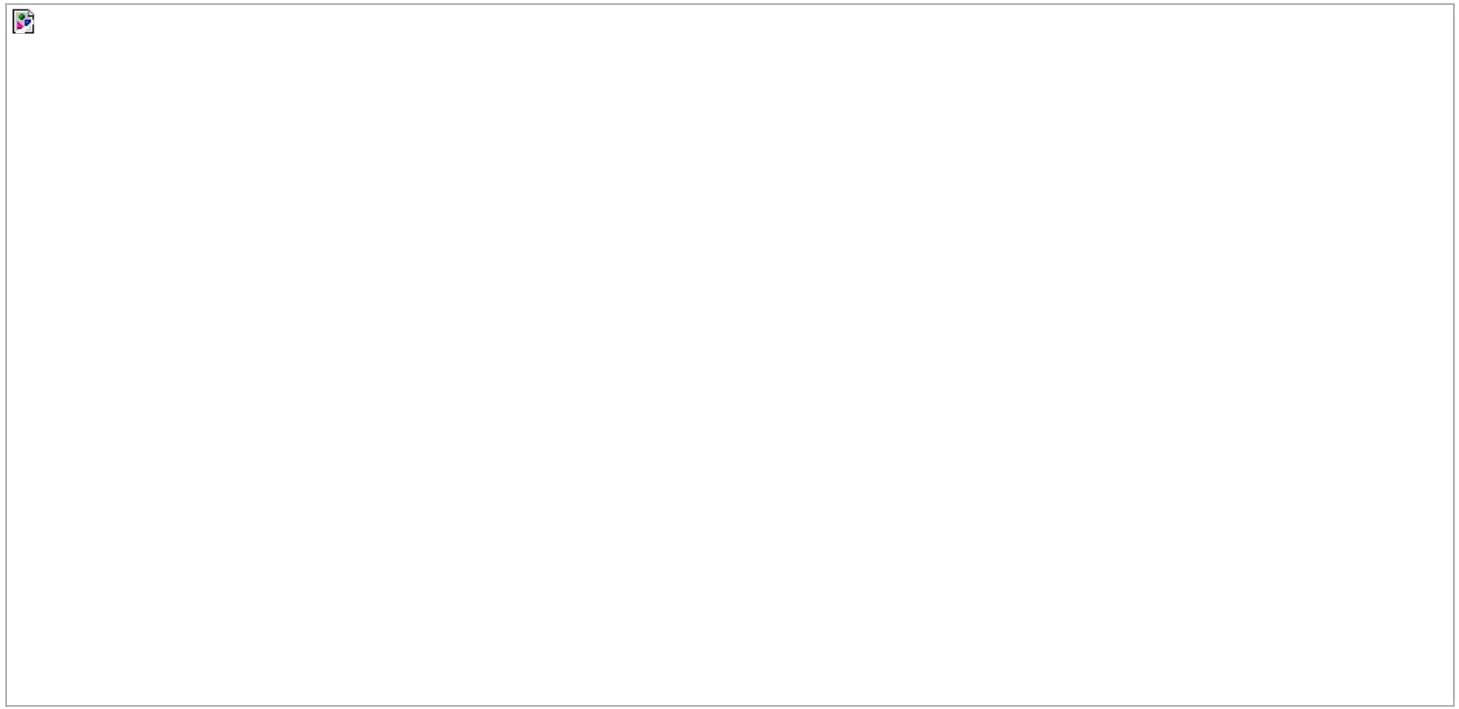
Magnum does offer a container API, but it allows you to use the COE native client or API to contact your cluster securely over TLS.

Magnum creates a CA for each cluster and stores it in Barbican (recommended but optional). You can store certificates locally or in magnum's DB.

As soon as your cluster is running, you don't have to use the magnum to run containers or even create cinder volumes or Load Balancers. You can use:

- docker
- kubectl
- dcos
- marathon API

OpenStack Magnum Architecture



Containers and the CERN Cloud

What is CERN?

European Organization for Nuclear Research
(Organisation européenne pour la recherche nucléaire)

Founded in 1954

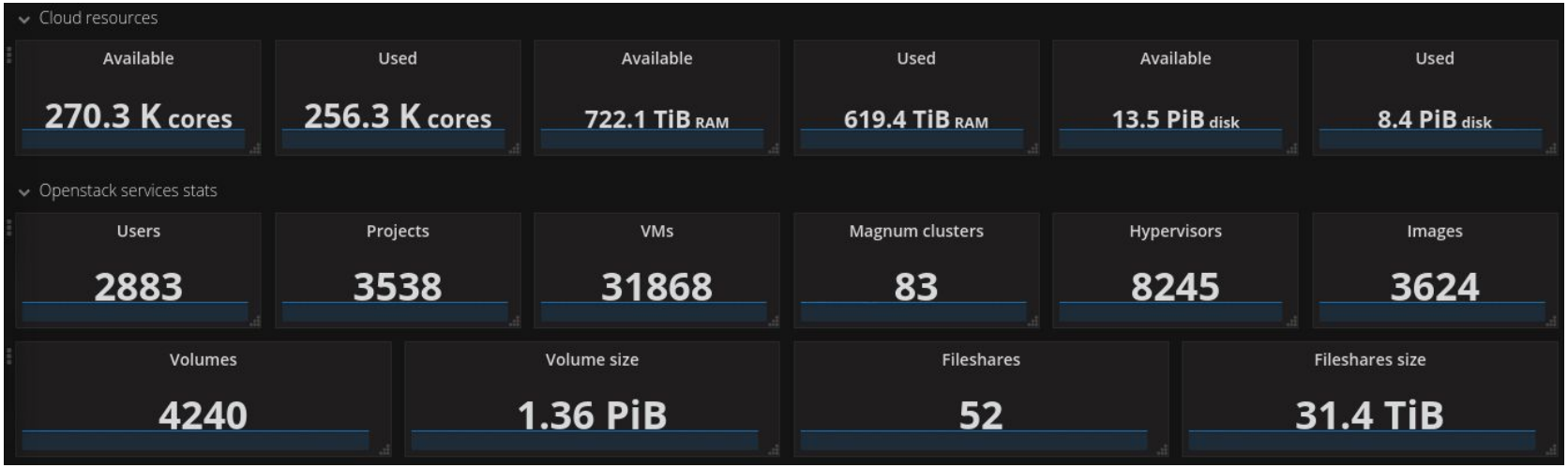
22 member states
With many others contributing to experiments

CERN's mission is fundamental research

CERN OpenStack Infrastructure

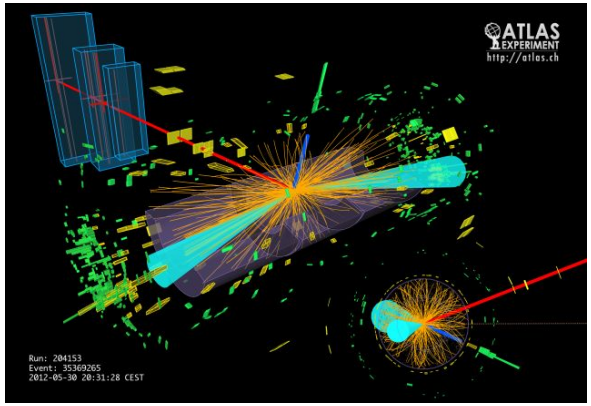
Production since 2013

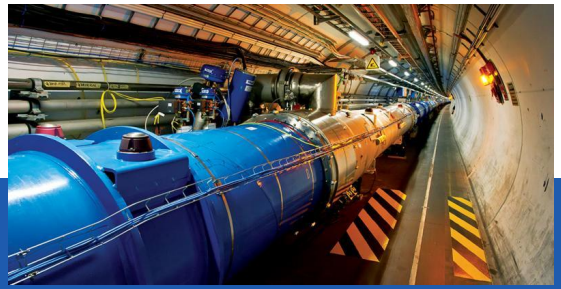
~ 270,000 cores ~5 million vms created ~200 vms per hour ~32,000 vm running

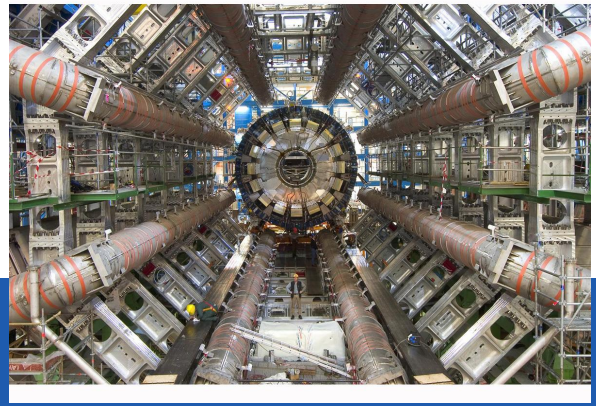


CERN Container Use Cases

- Batch Processing
- End user analysis / Jupyter Notebooks
- Machine Learning / TensorFlow / Keras
- Infrastructure Management
  - Data Movement, Web servers, PaaS …
- Continuous Integration / Deployment
- Run OpenStack :-)
- And many others

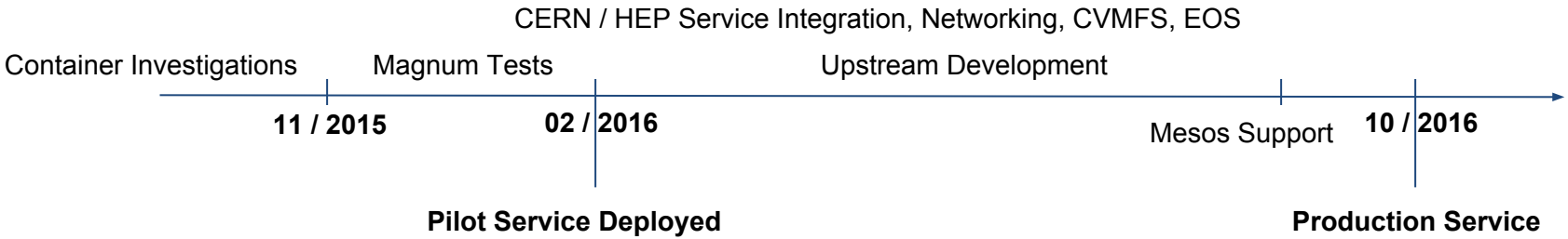






CERN Magnum Deployment

- Integrate containers in the CERN cloud
  - Shared identity, networking integration, storage access, …
- Add CERN services in *system* containers
- **Fast, Easy to use**



CERN Magnum Deployment

- Clusters are described by *cluster templates*
- Shared/public templates for most common setups, customizable by users

```
$ magnum cluster-template-list
+------+--------------------------+
| uuid | name                     |
+------+--------------------------+
| .... | swarm                    |
| .... | swarm-ha                 |
| .... | kubernetes               |
| .... | kubernetes-ha            |
| .... | mesos                    |
| .... | mesos-ha                 |
| .... | dcos                     |
+------+--------------------------+
```

CERN Magnum Deployment

- Clusters are described by *cluster templates*
- Shared/public templates for most common setups, customizable by users

```
$ magnum cluster-create --name myswarmcluster --cluster-template swarm --node-count 100
     ~ 5 mins later
$ magnum cluster-list
+------+----------------+------------+--------------+----------------------------+
| uuid | name           | node_count | master_count | keypair   | status         |
+------+----------------+------------+--------------+----------------------------+
| .... | myswarmcluster | 100        | 1            | mysshkey | CREATE_COMPLETE |
+------+----------------+------------+--------------+----------------------------+
$ $(magnum cluster-config myswarmcluster --dir magnum/myswarmcluster)
$ docker info / ps / ...
$ docker service create --mount
'type=volume,volume-driver=cvmfs,source=cms.cern.ch@trunk-previous,destination=/cvmfs/cms.cer
n.ch' busybox sleep 10000
```

Magnum Benchmarks

Rally Benchmarks and resource scalability

- Benchmark the Magnum service
  - How fast can I get my container cluster?
  - Use Rally to measure to performance like any other OpenStack service
- Benchmark the resources
  - Ok, it was reasonably fast, what can I do with it?
  - Use a demo provided by Google to measure the performance of the cluster
    - Rally tests for container are under development and near completion

Deployment Setup at CERN and CNCF

CERN

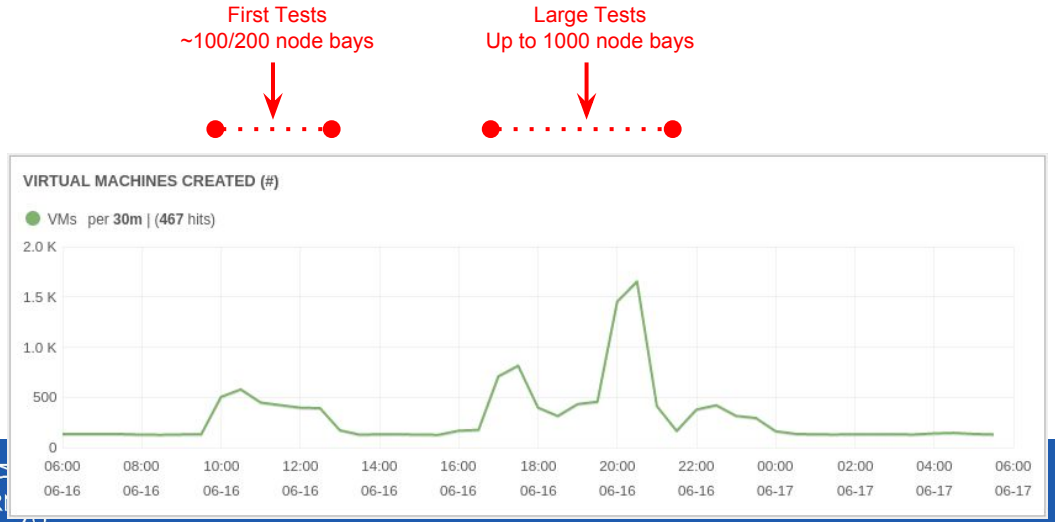
- 240 hypervisors
  - 32 cores, 64 GB RAM, 10Gb inks
- Container storage in our CEPH cluster
- Magnum / Heat setup
  - Dedicated 3 node controllers, dedicated 3 node RabbitMQ cluster
- Flat Network for vms

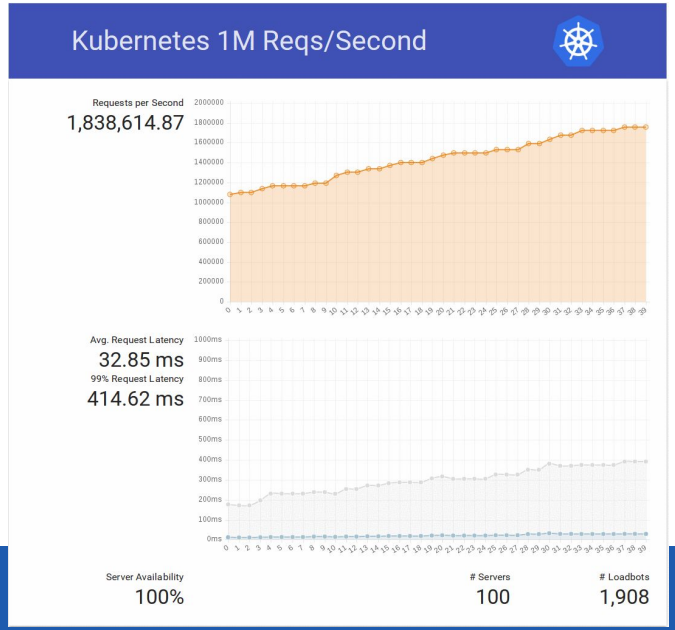
CNCF

- 100 hypervisors
  - 24 cores, 128 GB RAM
- Container storage in local disk
- Magnum / Heat setup
  - Shared 3 node controllers, shared 5 node RabbitMQ cluster
- Private networks with linux bridge

CERN Results

- Several iterations before arriving at a reliable setup
- First run: 2 million requests / s
  - Bay of 200 nodes (400 cores, 800 GB Ram)





Cluster Creation benchmark

CERN cloud

Cluster Size (Nodes)	Concurrency	Deployment Time (min)
2	50	2.5
16	10	4
32	10	4
128	5	5.5
512	1	14
1000	1	23

CNCF testing cloud

Cluster Size (Nodes)	Concurrency	Number of Clusters	Deployment Time (min)
2	10	100	3.02
2	10	1000	Able to create 219 clusters
32	5	100	Able to create 28 clusters

Cluster Upgrades

Phase 1

- Add new a new actions API to implement upgrades
- Upgrade clusters by doing server rebuild
  - All components in the operating system image will be safely upgraded
  - Hostnames and IPs won't change
- The only new data passed in the compute instances will be the COE version (e.g. Kubernetes version)

Phase 1 con’d

Add an additional volume to the master to store cluster’s state

- Etcd in kubernetes and legacy swarm
- Embedded swarm-mode data store in /var/lib/docker/swarm
- Zookeeper in Mesos and DC/OS

Making an upgrade available to the users

The operator should only upload to the Image Service the new OS image and notify users that an upgrade is available.

For minor versions, only the image will change.

For major versions, further changes might be required.

Upgrading a cluster

1. The users is notified by the operations team that an upgrade is available for their clusters

2. Master nodes need to be upgraded first:

```
$ magnum cluster-upgrade --masters image=<new image> mycluster
```

3. Worker nodes can be upgraded later on:

```
$ magnum cluster-upgrade --nodes image=<new image> mycluster
```

Notes on Operations

- Moving from puppet workflows to containerized application is totally different mindset
- How to monitor the software for security?
  - https://developers.redhat.com/blog/2016/05/02/introducing-atomic-scan-container-vulnerability-detection/
  - https://github.com/coreos/clair
  - All images based on a golden image approach. What about “FROM alpine/scratch” images?

What do you need for Magnum?

For existing OpenStack deployments:

- If you already have Heat, just go ahead
- If you don’t, Heat is a stable enough service and can be managed easily and scales easily for HA

New OpenStack deployments:

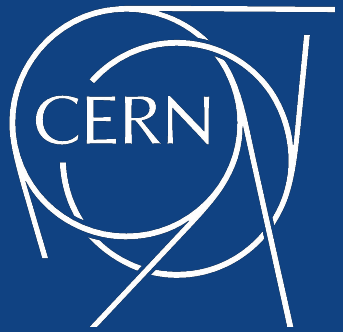
- You need, Keystone, Glance, Nova, Neutron, Cinder and Heat

To whom Magnum is target

- Magnum deploys containers ON OpenStack
- Teams that want to manage a few clusters
- Private or Public clouds that want to offer clusters in their organization and have central monitoring, central logging and central management

Who should not use Magnum:

- You need a 4 node kubernetes cluster and you don’t have OpenStack
- You need one small cluster and you have one user


CERN