



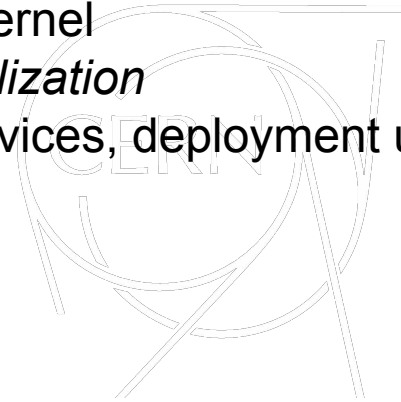
CERN Cloud Containers

Ricardo Rocha

(on behalf of the CERN Cloud team)

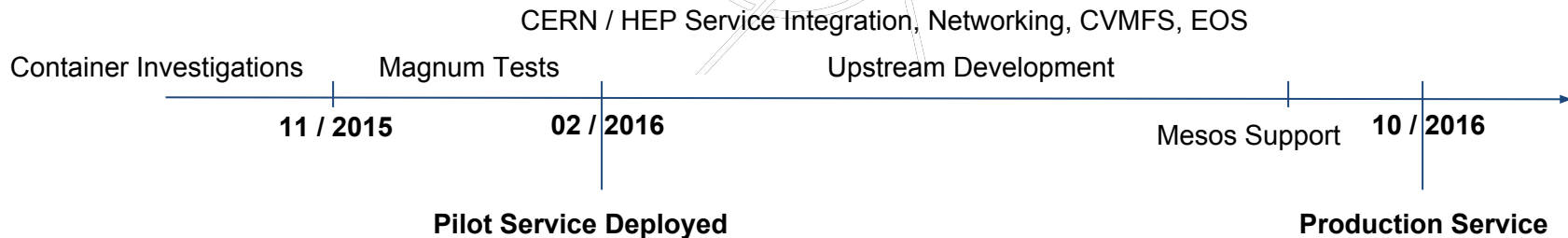
Why containers?

- *Isolation*, via kernel namespaces and cgroups
- *Performance*, same kernel
- *Improved resource utilization*
- *Ease of use*, microservices, deployment units, image repositories



Goals and Timeline

- Integrate containers in the CERN cloud
 - Shared identity, networking integration, storage access, ...
- Agnostic to container orchestration engines
 - Docker Swarm, Kubernetes, Mesos
- Fast, easy to use



OpenStack Magnum Overview

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

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

```
$ magnum cluster-template-show kubernetes
...
| coe | kubernetes |
| master_flavor_id | m1.medium |
| flavor_id | m1.medium |
| server_type | vm |
| image_id | fedora-atomic-25 |
| labels | {} |
| network_driver | flannel |
```

OpenStack Magnum Overview (2)

- Create a cluster in a single command (no matter what size)

```
$ magnum cluster-create --name mykubcluster --cluster-template kubernetes --node-count 100

$ magnum cluster-list
+-----+-----+-----+-----+-----+
| uuid | name           | node_count | master_count | status           |
+-----+-----+-----+-----+-----+
| .... | mykubcluster  | 100        | 1             | CREATE_COMPLETE |
+-----+-----+-----+-----+-----+

$ $(magnum cluster-config mykubcluster --dir magnum/mykubcluster)

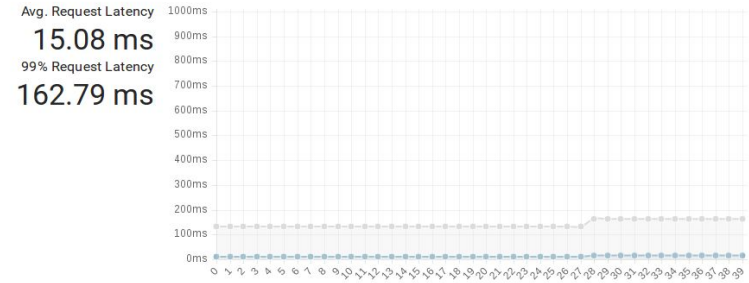
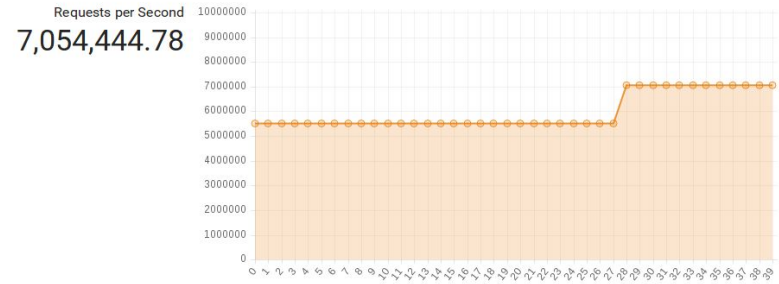
$ kubectl get pod
```

Performance Evaluation

- August 2016
 - Kubernetes stress test
 - Managed **7 million requests / sec**
- And an analysis of cluster deployments

Cluster Size (Nodes)	Deployment Time (min)
2	2.5
32	4
128	5.5
512	14
1000	23

Kubernetes 1M 10M Reqs/Second



Server Availability 100% # Servers 500 # Loadbots 9,449

Integration Work

- OpenStack Cinder
- CVMFS
 - Swarm / Mesos: implemented as a docker volume plugin
 - Kubernetes: implemented as a FlexVolume wrapper
- EOS
 - Same plugin structure

<https://gitlab.cern.ch/cloud-infrastructure/docker-volume-cvmfs>

<https://gitlab.cern.ch/cloud-infrastructure/docker-volume-eos>

Integration Work

```
$ docker volume create -d cvmfs --name lhcb.cern.ch

$ docker run -it --rm -v lhcb.cern.ch:/cvmfs/lhcb.cern.ch centos:7 /bin/bash
[root@874cbf8199d0 /]# ls /cvmfs/lhcb.cern.ch/
etc  group_login.csh  group_login.sh  lib
```



Integration Work

```
$ cat nginx-cvmfs.yaml
...
spec:
  containers:
  - name: nginx-cvmfs
    image: nginx
    ports:
    - containerPort: 80
    volumeMounts:
    - name: lhcb
      mountPath: /cvmfs/lhcb.cern.ch
  volumes:
  - name: lhcb
    flexVolume:
      driver: "cern/cvmfs"
      options:
        repository: "lhcb.cern.ch"

$ kubectl create -f nginx-cvmfs.yaml
```

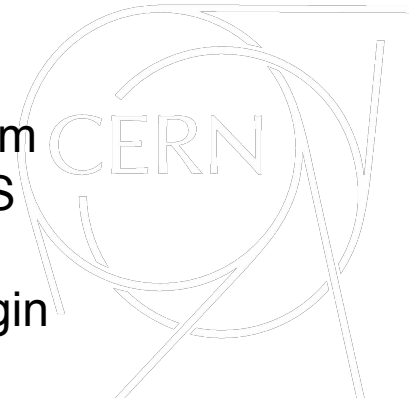
Some Use Cases

- Infrastructure Services
 - Ex: FTS, OpenStack Horizon, ...
 - In most cases currently using Kubernetes
- Custom GitLab CI
 - Docker API compatible gitlab-ci-multi-runner
 - In most cases using Docker Swarm
- Analysis / Jupyter Notebooks
- ...

<http://clouddocs.web.cern.ch/clouddocs/containers/tutorials/index.html>

Conclusion & Future Work

- Production service since end of last year
- ~80 clusters deployed
 - ~50 Kubernetes
 - ~20 Docker Swarm
 - ~6 Mesos / DCOS
- Future Work
 - Finalize EOS plugin
 - Cluster upgrades
 - Availability zones



<http://clouddocs.web.cern.ch/clouddocs/containers/index.html>

Questions?

