EOS + Ceph integration with K8S

Federico Fornari - INFN CNAF

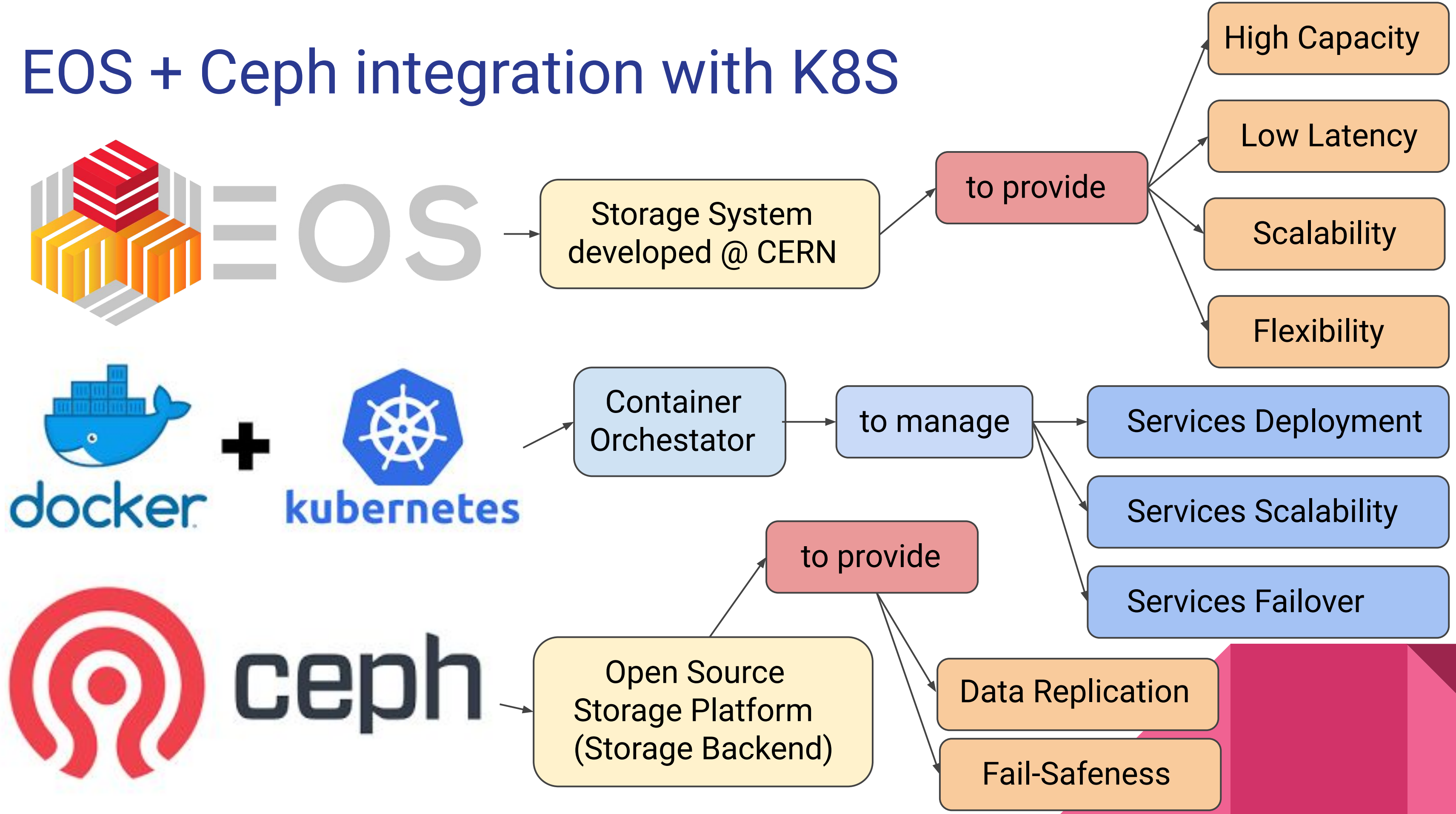
A. Costantini*, A. Cavalli*, D. Cesini*, D. C. Duma*, A. Falabella*, E. Fattibene*, L. Mascetti**, L. Morganti*, A.J. Peters**, A. Prosperini*, V. Sapunenko*

(*) - INFN CNAF
(**)- CERN

Introduction

- Collaboration between INFN (Italian Institute for Nuclear Physics) center dedicated to Research and Development on Information and Communication Technologies (CNAF) and CERN.
- Different technologies tested and evaluated for next-generation storage challenges at CNAF:
    - EOS: open-source storage software for multi-PetaByte storage management at CERN LHC.
    - Ceph: open-source platform to expose data through object, block and posix-compliant storage.
    - Kubernetes: open-source container-orchestration system for automating computer application deployment, scaling and management.
- Results obtained by measuring performances of the different combined technologies, comparing for instance block device and file system as backend options provided by a Ceph cluster deployed on physical machines, are shown and discussed hereafter.

EOS + Ceph integration with K8S

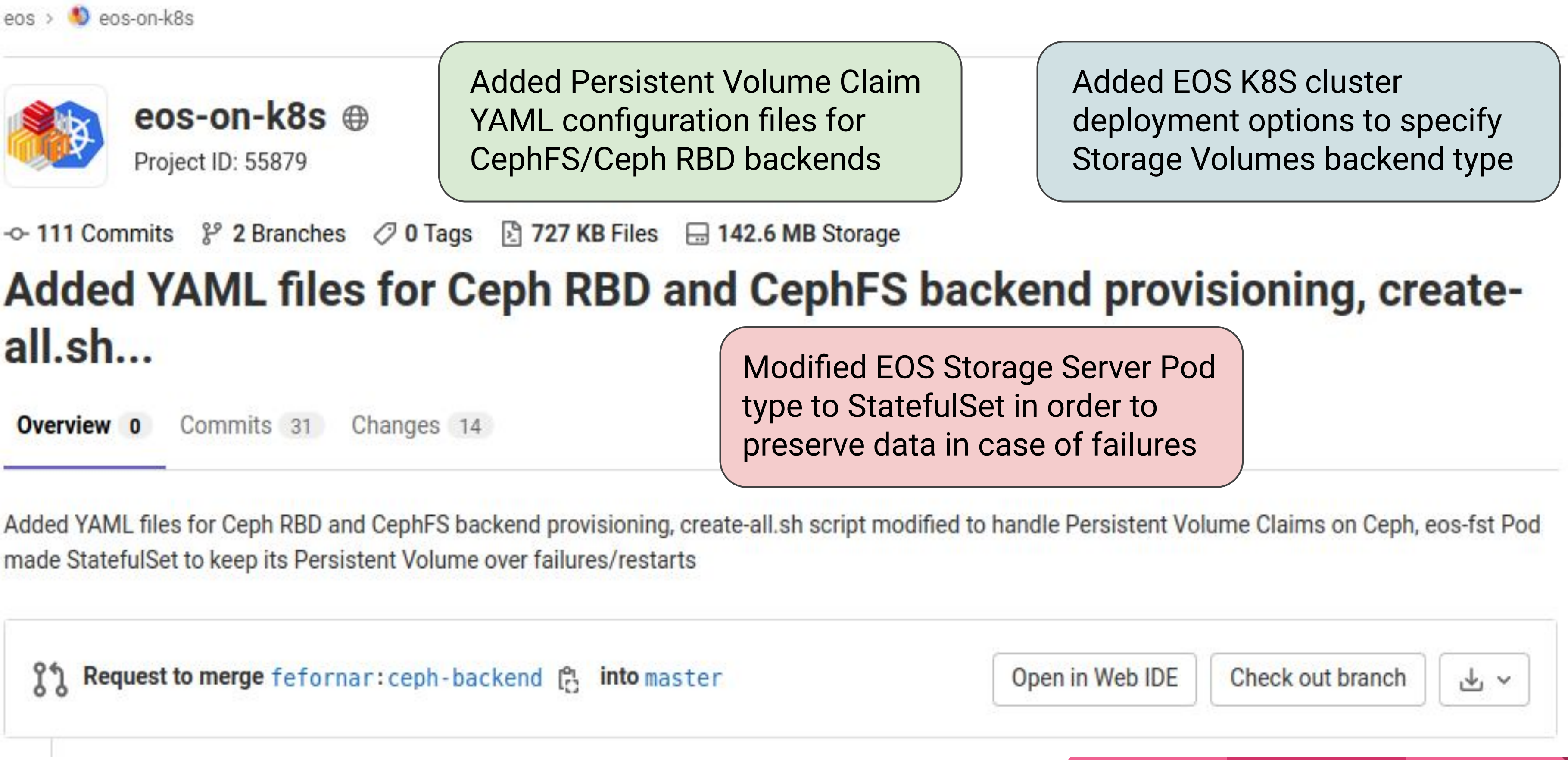


Storage System developed @ CERN → to provide → High Capacity, Low Latency, Scalability, Flexibility

Container Orchestator → to manage → Services Deployment, Services Scalability, Services Failover

Open Source Storage Platform (Storage Backend) → to provide → Data Replication, Fail-Safeness

EOS on K8S Project @ CERN - Personal Contributions



Added Persistent Volume Claim YAML configuration files for CephFS/Ceph RBD backends

Added EOS K8S cluster deployment options to specify Storage Volumes backend type

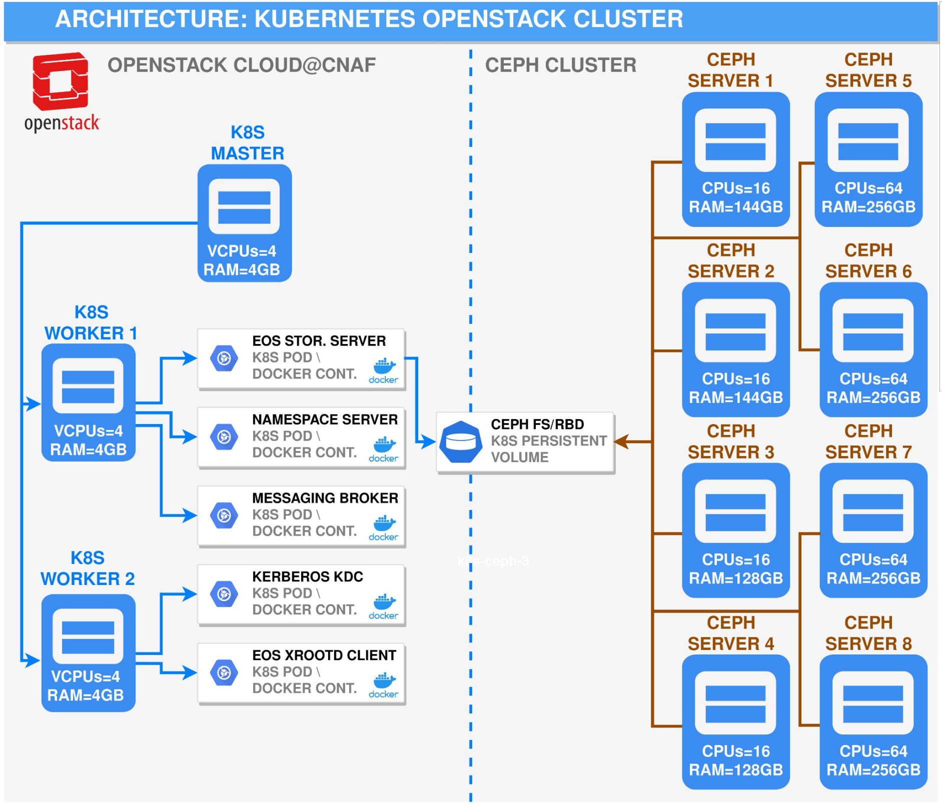
Added YAML files for Ceph RBD and CephFS backend provisioning, create-all.sh...

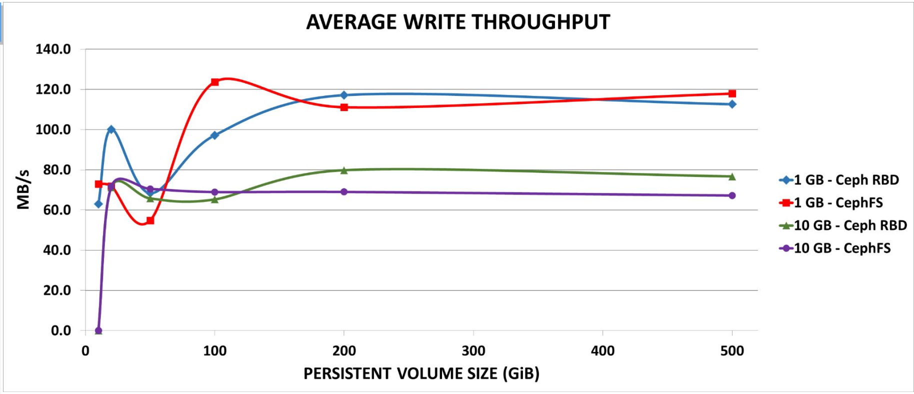
Modified EOS Storage Server Pod type to StatefulSet in order to preserve data in case of failures

Added YAML files for Ceph RBD and CephFS backend provisioning, create-all.sh script modified to handle Persistent Volume Claims on Ceph, eos-fst Pod made StatefulSet to keep its Persistent Volume over failures/restarts

Request to merge `fefornar:ceph-backend` into `master`

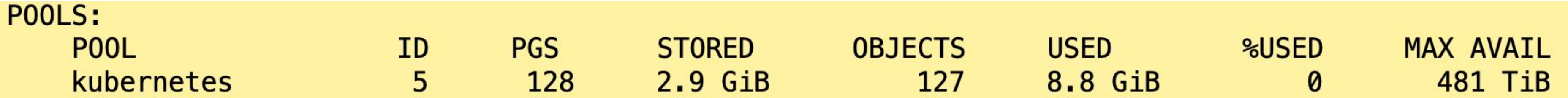
Functionality tests on Openstack Cloud @ CNAF



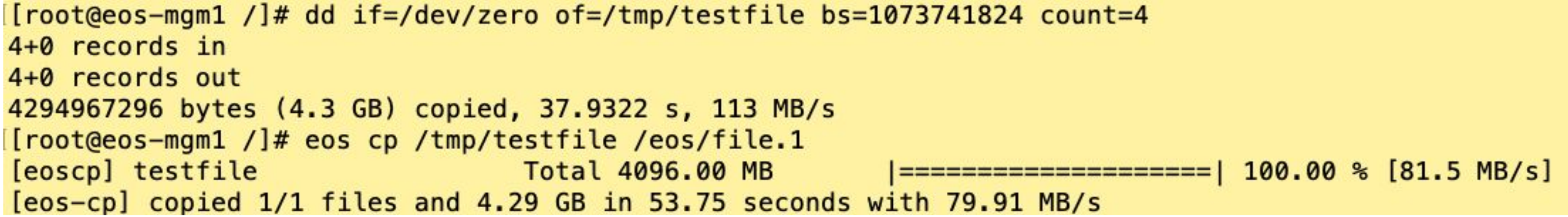


Preliminary tests on Openstack cluster showed network bandwidth saturation (1 Gbit/s) and setup stability, leading to tests on bare-metal hardware

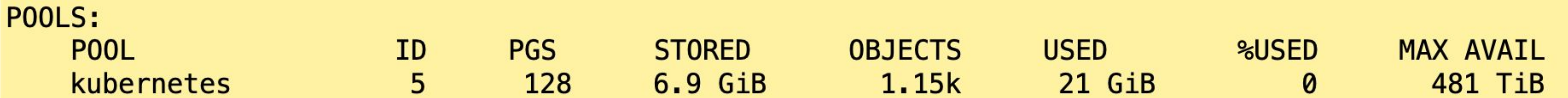
Functionality tests on Openstack Cloud @ CNAF



```
POOLS:
    POOL                ID     PGS     STORED     OBJECTS     USED       %USED     MAX AVAIL
    kubernetes           5     128     2.9 GiB        127     8.8 GiB        0       481 TiB
```



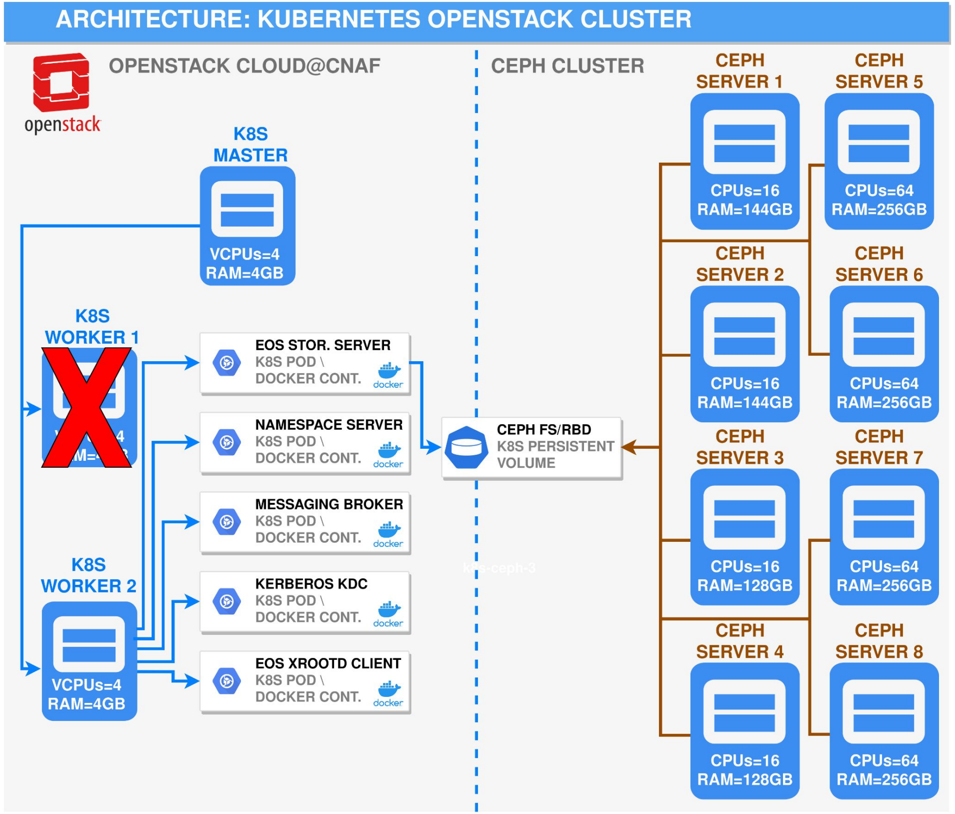
```
[root@eos-mgm1 /]# dd if=/dev/zero of=/tmp/testfile bs=1073741824 count=4
4+0 records in
4+0 records out
4294967296 bytes (4.3 GB) copied, 37.9322 s, 113 MB/s
[root@eos-mgm1 /]# eos cp /tmp/testfile /eos/file.1
[eoscp] testfile                  Total 4096.00 MB       |====================| 100.00 % [81.5 MB/s]
[eos-cp] copied 1/1 files and 4.29 GB in 53.75 seconds with 79.91 MB/s
```



```
POOLS:
    POOL                ID     PGS     STORED     OBJECTS     USED       %USED     MAX AVAIL
    kubernetes           5     128     6.9 GiB      1.15k      21 GiB        0       481 TiB
```

Writing a 4 GB file on an EOS partition leaning on a Ceph RBD created from a Replica 3 RBD pool makes the used space on the related pool increasing from 9 to 21 GB (i.e., 21 GB - 9 GB = 12 GB = 4 GB x 3, q.e.d.).

Functionality tests on Openstack Cloud @ CNAF

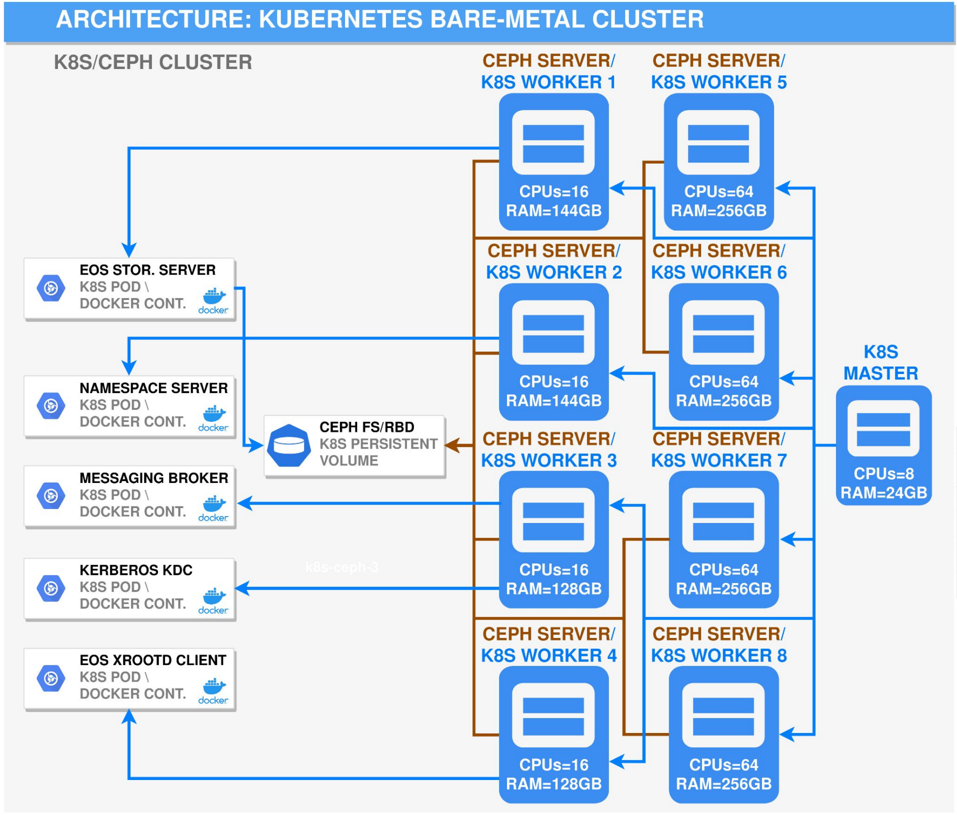


If a K8S worker node is shut down, simulating a failure, and then removed from the cluster, EOS Pods automatically migrate

Since EOS Storage Servers are StatefulSet, associated Volumes are Persistent across Pods redeployment, preserving original data

EOS Services Failover successfully provided by K8S

Performance tests on bare-metal cluster @ CNAF

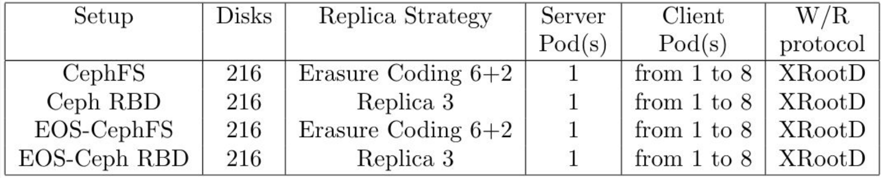

ARCHITECTURE: KUBERNETES BARE-METAL CLUSTER
K8S/CEPH CLUSTER
CEPH SERVER/ K8S WORKER 1
CPUs=16 RAM=144GB
CEPH SERVER/ K8S WORKER 5
CPUs=64 RAM=256GB
CEPH SERVER/ K8S WORKER 2
CPUs=16 RAM=144GB
CEPH SERVER/ K8S WORKER 6
CPUs=64 RAM=256GB
CEPH SERVER/ K8S WORKER 3
CPUs=16 RAM=128GB
CEPH SERVER/ K8S WORKER 7
CPUs=64 RAM=256GB
CEPH SERVER/ K8S WORKER 4
CPUs=16 RAM=128GB
CEPH SERVER/ K8S WORKER 8
CPUs=64 RAM=256GB
K8S MASTER
CPUs=8 RAM=24GB
EOS STOR. SERVER K8S POD \ DOCKER CONT.
NAMESPACE SERVER K8S POD \ DOCKER CONT.
MESSAGING BROKER K8S POD \ DOCKER CONT.
KERBEROS KDC K8S POD \ DOCKER CONT.
EOS XROOTD CLIENT K8S POD \ DOCKER CONT.
CEPH FS/RBD K8S PERSISTENT VOLUME
k8s-ceph-3
docker

Ceph cluster nodes used to host a K8S cluster

CephFS vs. Ceph RBD as EOS backends

EOS + Ceph vs. stand-alone Ceph comparison

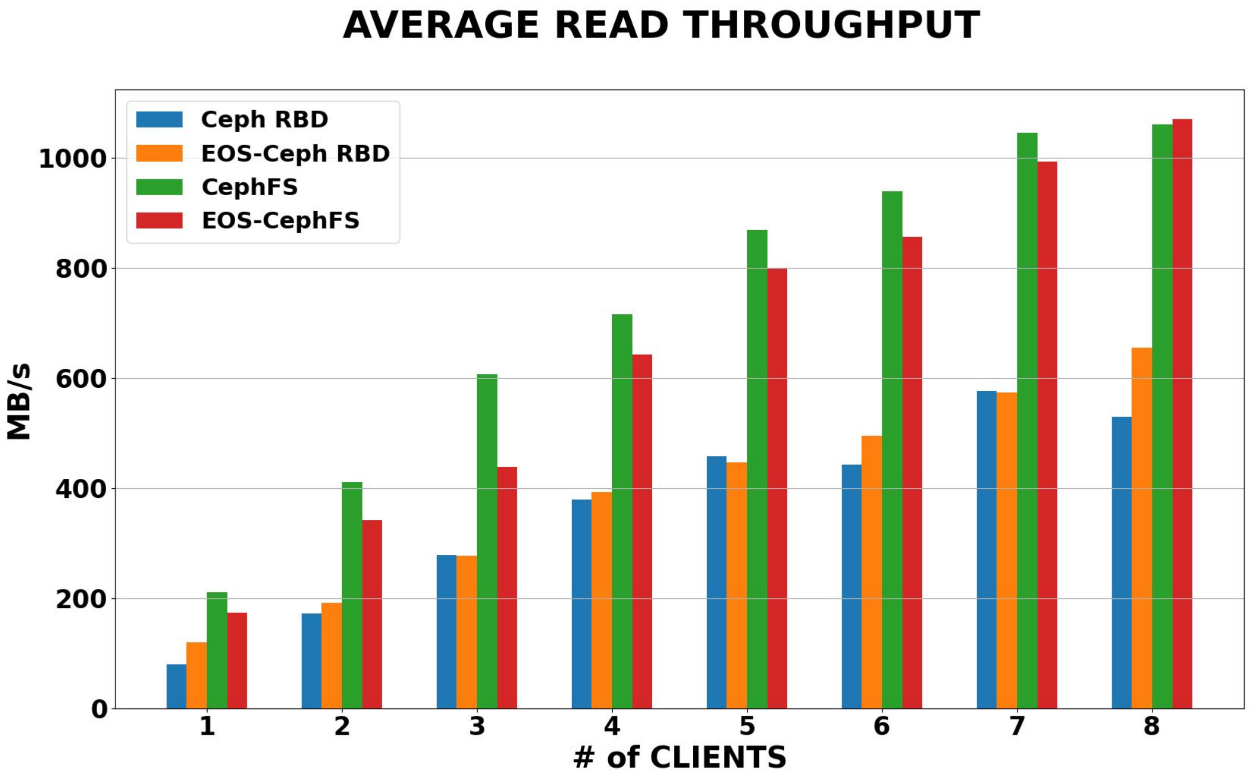
Tests done using XRootD file transfer protocol



Setup	Disks	Replica Strategy	Server Pod(s)	Client Pod(s)	W/R protocol
CephFS	216	Erasure Coding 6+2	1	from 1 to 8	XRootD
Ceph RBD	216	Replica 3	1	from 1 to 8	XRootD
EOS-CephFS	216	Erasure Coding 6+2	1	from 1 to 8	XRootD
EOS-Ceph RBD	216	Replica 3	1	from 1 to 8	XRootD

1 test -> 100 1GB files read and written

Performance tests on bare-metal cluster @ CNAF



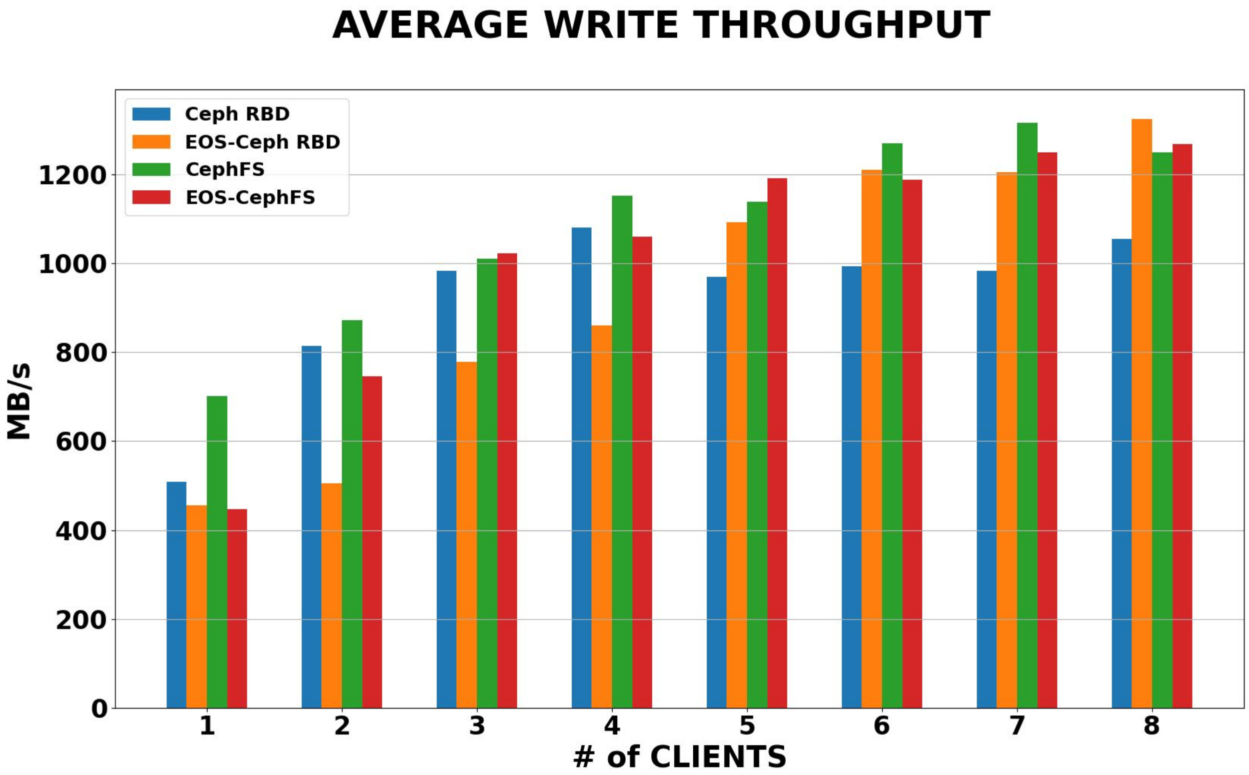
CephFS shows better read scores if compared with Ceph RBD

This can be explained by different access patterns to the disks

CephFS = FS shared over the network, so different machines can access it all at the same time

Ceph RBD uses images shared over the network

Performance tests on bare-metal cluster @ CNAF



EOS+Ceph has a better throughput than Ceph, as the number of clients increases

This can be due to cache effects among EOS and Ceph that becomes evident by increasing the clients

The same cache effects can also explain some of the EOS+Ceph read performance results shown before

Conclusions

- Integration between EOS and Ceph using Kubernetes gave good results in terms of scalability and stability (given mainly by EOS services), reliability and redundancy (provided by Ceph), integration and management (provided by Kubernetes) and overall performances.
- Testing different scenarios allowed to deal with different problems for which proper solutions have been developed, bringing also important improvements in the integration of such services.
- New advancements are planned for the next future, such as analyses implying setups with higher number of servers and parallel clients.

Thank you for your attention!