There’s no such thing as a “stateless” architecture

Cheryl Hung @oicheryl

Cheryl
@oicheryl

Why do I need storage?

Why do I need storage?

Why do I need storage?

App binaries

App data

Config

Backup

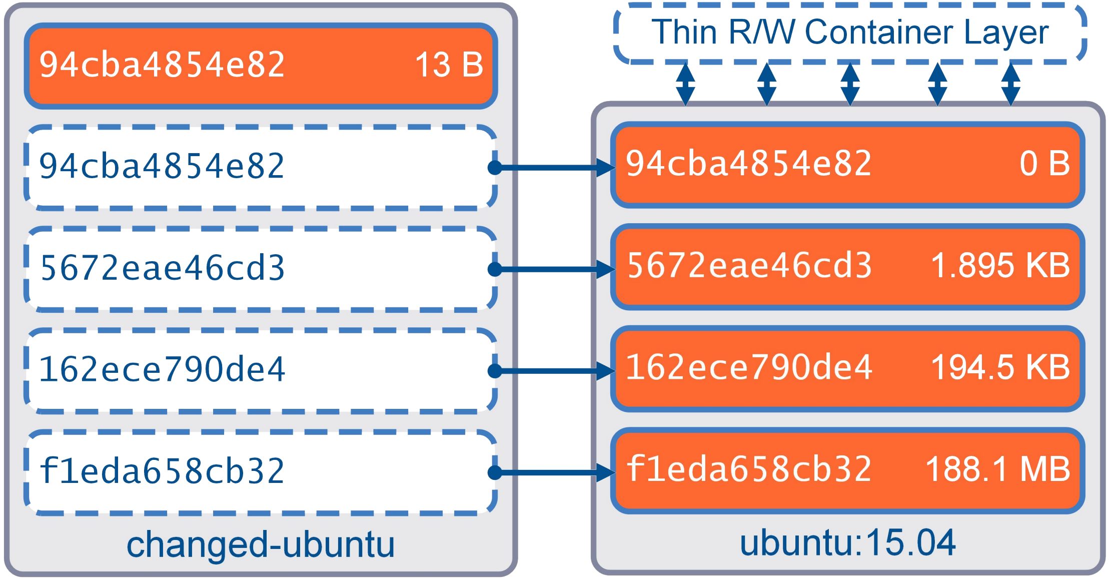
Why is this tricky with containers?

No storage pets

Data follows

Humans are fallible

Docker container layers

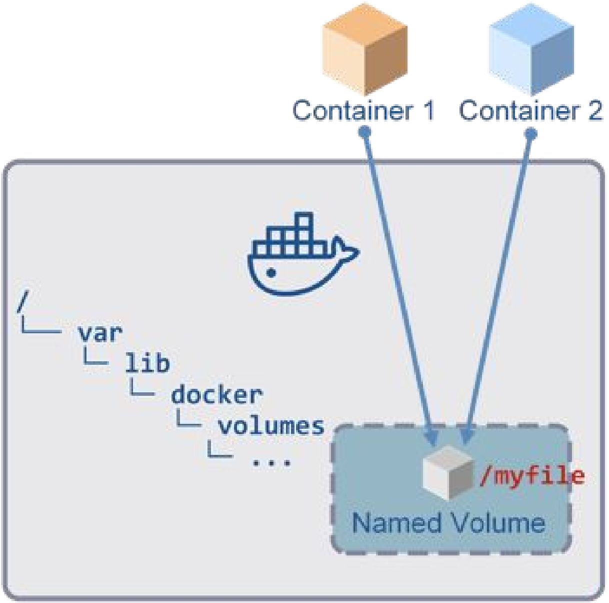


Docker local volumes

```
$ docker volume create --name mydata
```

```
$ docker run --rm -v mydata:/data:rw alpine ash -c \
  "echo hello world > /data/myfile"
```

```
$ sudo cat /var/lib/docker/volumes/mydata/_data/myfile
hello world
```



Eight principles of Cloud Native Storage

What is Cloud Native?

Horizontally scalable

No single point of failure

Resilient and self healing

Minimal operator overhead

Decoupled from the underlying platform

Jane

DevOps eng in a bank
How do I migrate my Postgres database to containers?

Eight principles of Cloud Native Storage

1. **API driven**

Eight principles of Cloud Native Storage

1. API driven
2. **Declarative and composable**

Eight principles of Cloud Native Storage

1. API driven
2. Declarative and composable
3. **Application centric**

Eight principles of Cloud Native Storage

1. API driven
2. Declarative and composable
3. Application centric
4. **Agile**

Eight principles of Cloud Native Storage

1. API driven
2. Declarative and composable
3. Application centric
4. Agile
5. **Performant**

Block storage

Data stored in fixed-size 'blocks' in a rigid arrangement—ideal for enterprise databases

File storage

Data stored as 'files' in hierarchically nested 'folders'—ideal for active documents

Object storage

Data stored as 'objects' in scalable 'buckets'—ideal for unstructured big data, analytics and archiving

Eight principles of Cloud Native Storage

1. API driven
2. Declarative and composable
3. Application centric
4. Agile
5. Performant
6. **Natively secure**

Eight principles of Cloud Native Storage

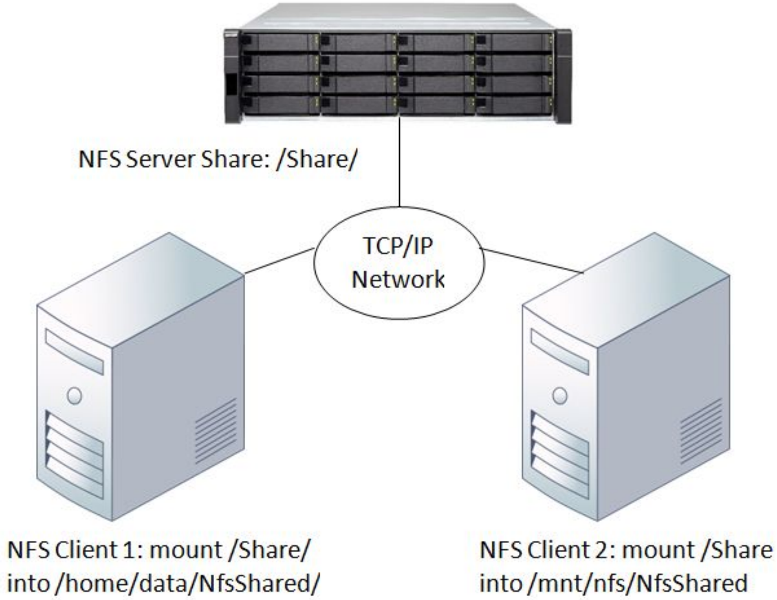
1. API driven
2. Declarative and composable
3. Application centric
4. Agile
5. Performant
6. Natively secure
7. **Consistently available**

Eight principles of Cloud Native Storage

1. API driven
2. Declarative and composable
3. Application centric
4. Agile
5. Performant
6. Natively secure
7. Consistently available
8. **Platform agnostic**

Storage landscape

Centralised file system: NFS


NFS Server Share: /Share/
TCP/IP Network
NFS Client 1: mount /Share/ into /home/data/NfsShared/
NFS Client 2: mount /Share into /mnt/nfs/NfsShared

Centralised file system: NFS

0

Single point of failure
Hard to scale horizontally
No native integration

Storage array: Dell EMC

Storage array: Dell EMC

2

Deterministic performance

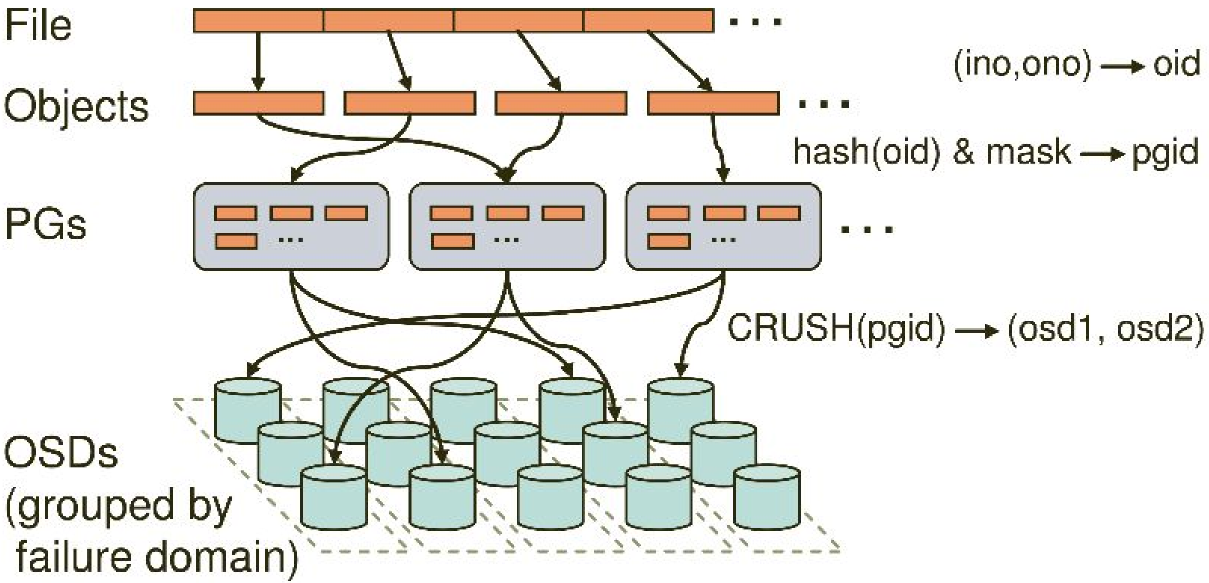
Vendor lock in

No thin provisioning

Hard to scale horizontally

Expensive and long lead times

Distributed: Ceph



Distributed: Ceph

4

Horizontally scalable

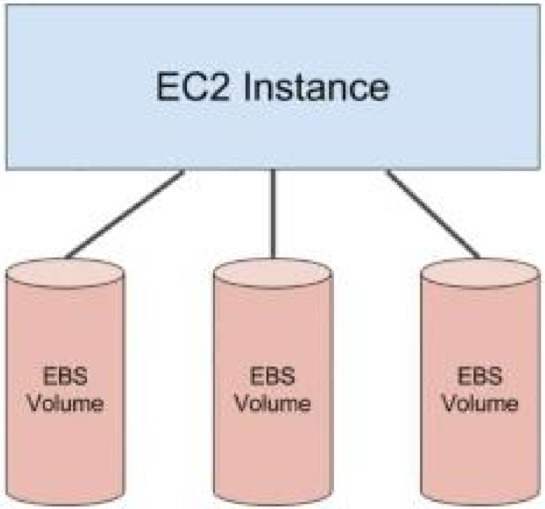
Hardware agnostic

Complicated to set up (see: Rook)

Writes fan out 13-40 times

Failures are expensive

Public cloud: AWS EBS



Public cloud: AWS EBS

6

Horizontally scalable

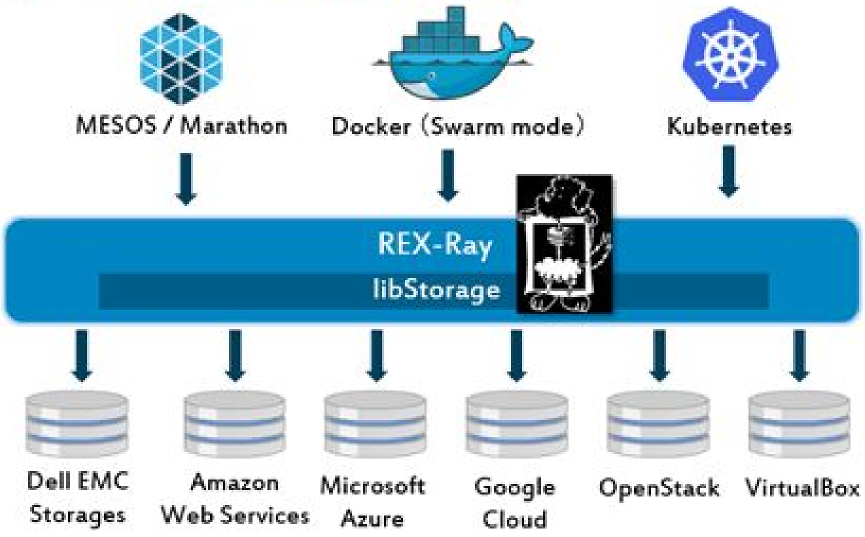
Consistent and performant

40 EBS instances per EC2 instance

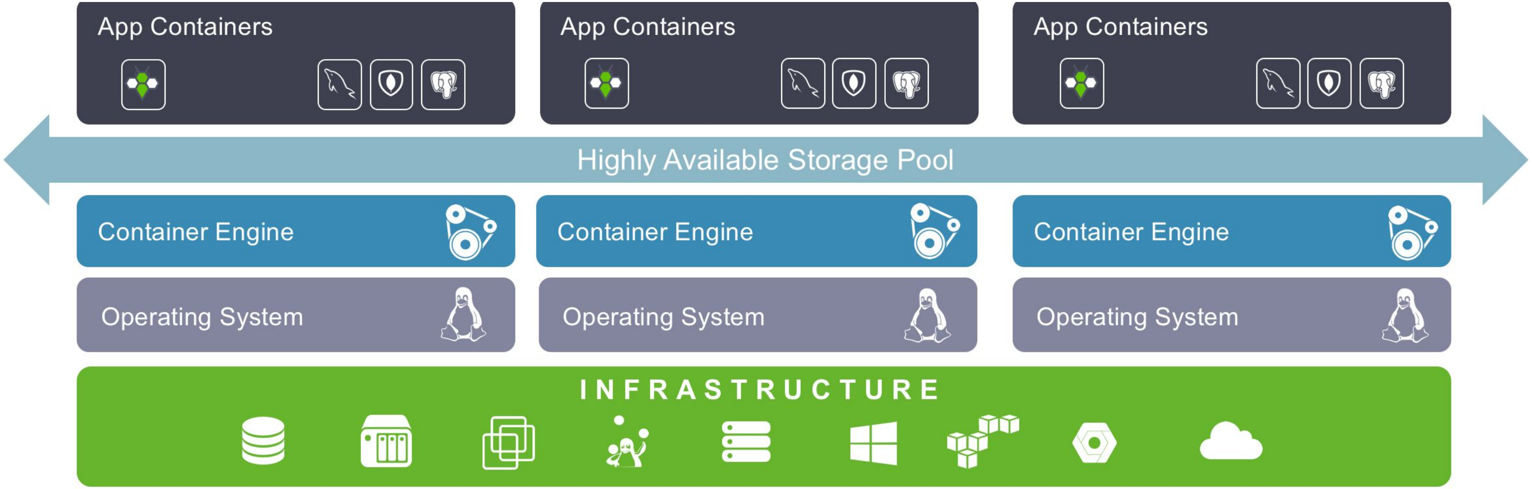
Mounting physical block devices is slow

Expensive, vendor lock in, privacy issues

Plugin framework: REX-Ray

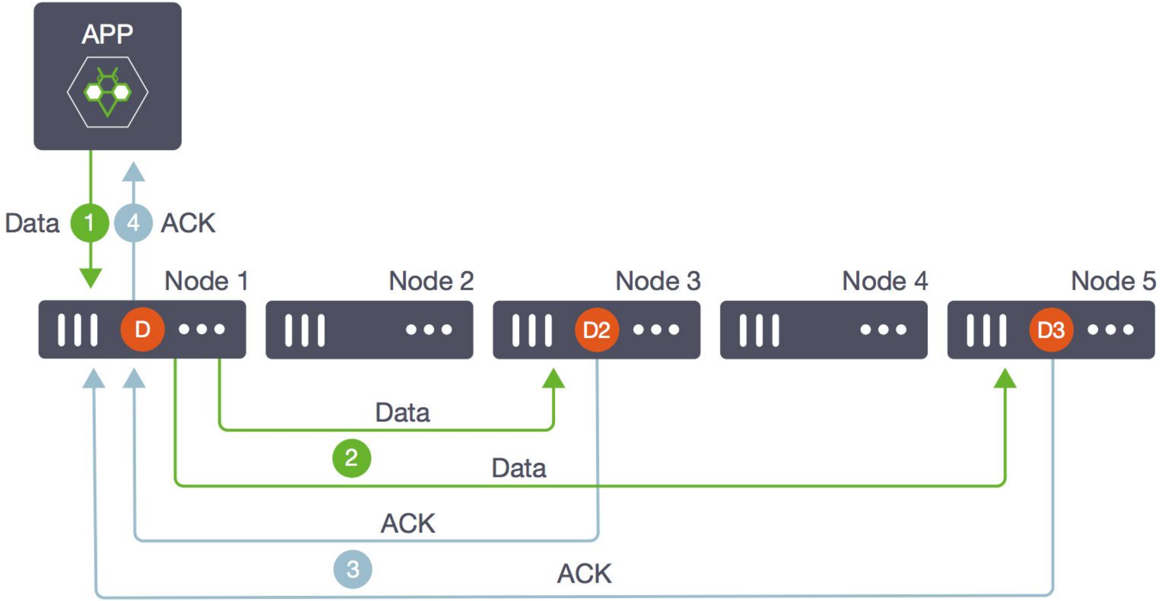


Volume plugin: StorageOS



Volume plugin: StorageOS

High availability with StorageOS



Volume plugin: StorageOS

8

Horizontally scalable

Consistent and performant

Vendor agnostic

Synchronous replication

Volume is limited to the size of one node

Conclusion

K8S Storage SIG & CNCF Storage WG: https://github.com/cncf/wg-storage

Objective is to define an industry standard “Container Storage Interface” (CSI) that will enable storage vendors to develop a plugin once and have it work across a number of container orchestration systems.

Cloud Native London meetup

- Join us next Tuesday
- Speakers from Monzo, Attest, Government Digital Service
- meetup.com/Cloud-Native-London

Join StorageOS!

C, Go, DevOps, pre-sales eng

STORAGEOS

Thanks

Slides at oicheryl.com