



Persistent Storage with Kubernetes in Production

Which solution and why?

Kubernetes London, 30 Nov 2017

Cheryl Hung, Product Manager



Cheryl
Product manager at
StorageOS
[@oicheryl](https://twitter.com/oicheryl)



- Why state is tricky in Kubernetes
- How do I evaluate my storage needs?
- How do I choose a storage solution for Kubernetes?



- Why is state tricky in Kubernetes?
- How do I evaluate my storage needs?
- How do I choose a storage solution for Kubernetes?

Anti-objective:

- Should I use a database, message queue, NoSQL db... for my app?



Why is state so tricky?



Why do I need storage?





Problem 1: No pet storage

@oicheryl



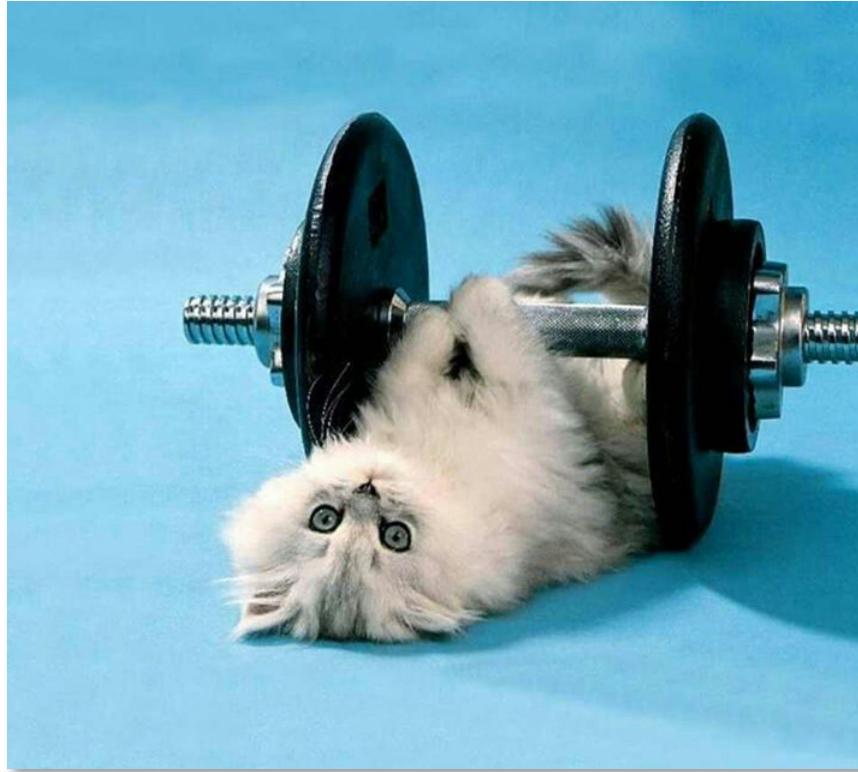
Problem 2: Data needs to follow

@oicheryl



Problem 3: Gravity can be a problem

@oicheryl



Problem 4: Storage isn't just storage

@oicheryl



App Binaries



App data



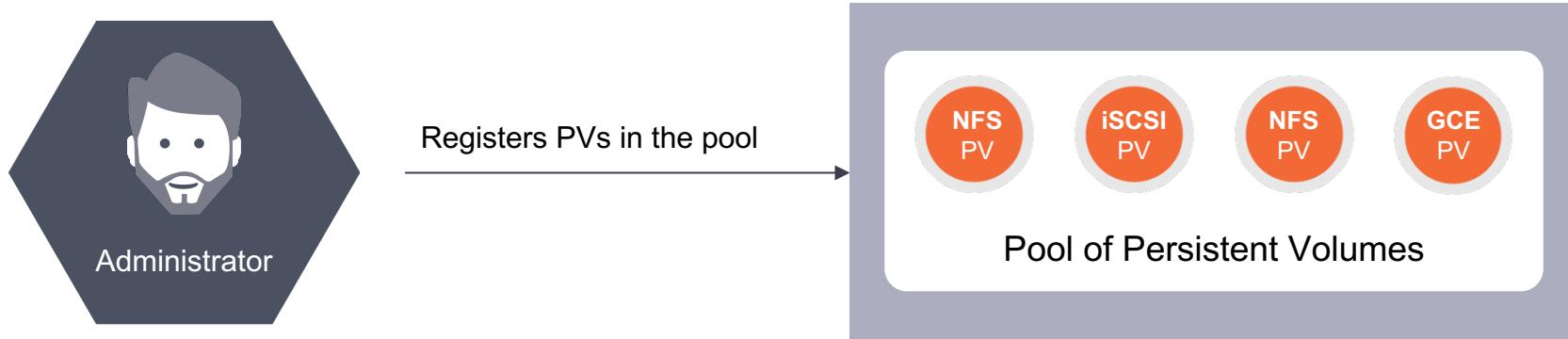
Config



Backup

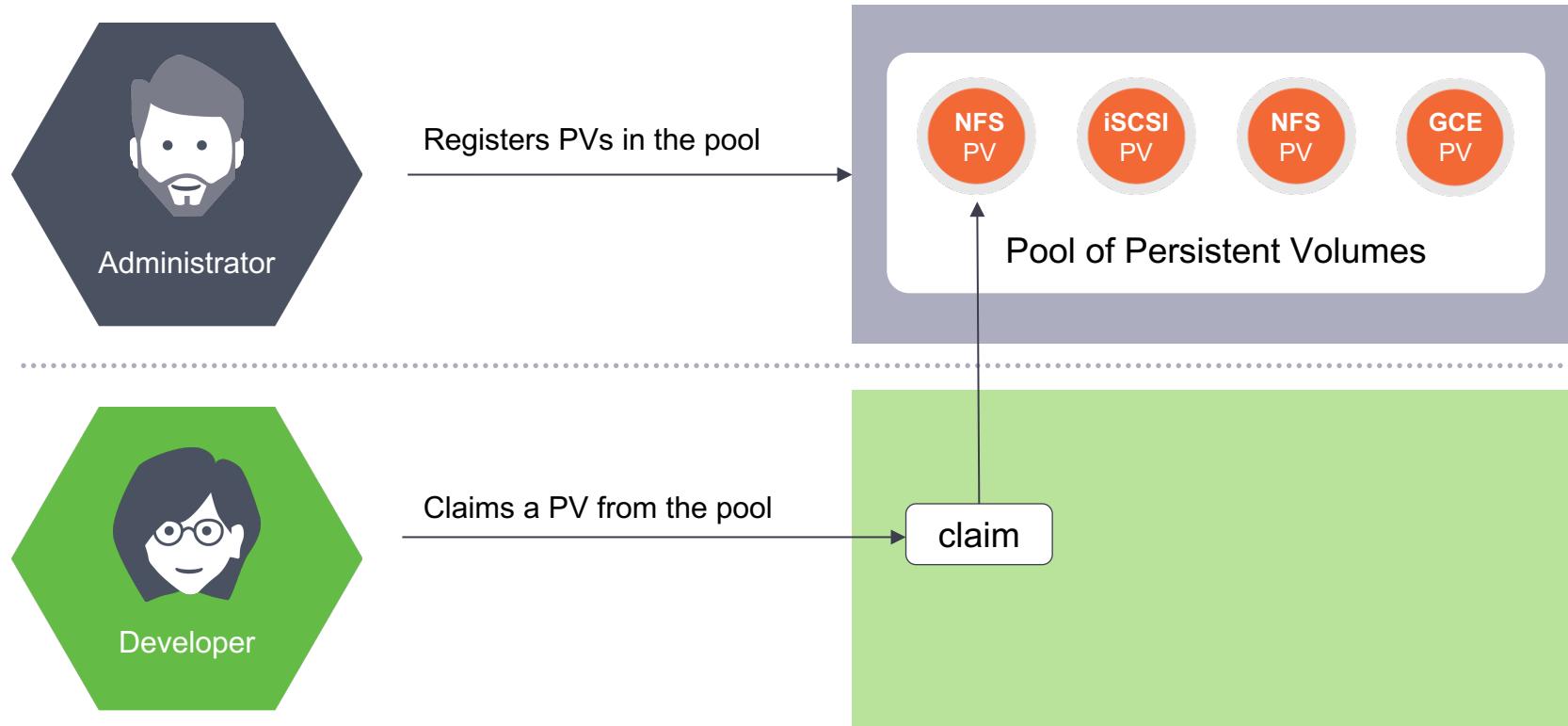
Kubernetes Storage Model: Persistent Volumes and Claims

@oicheryl



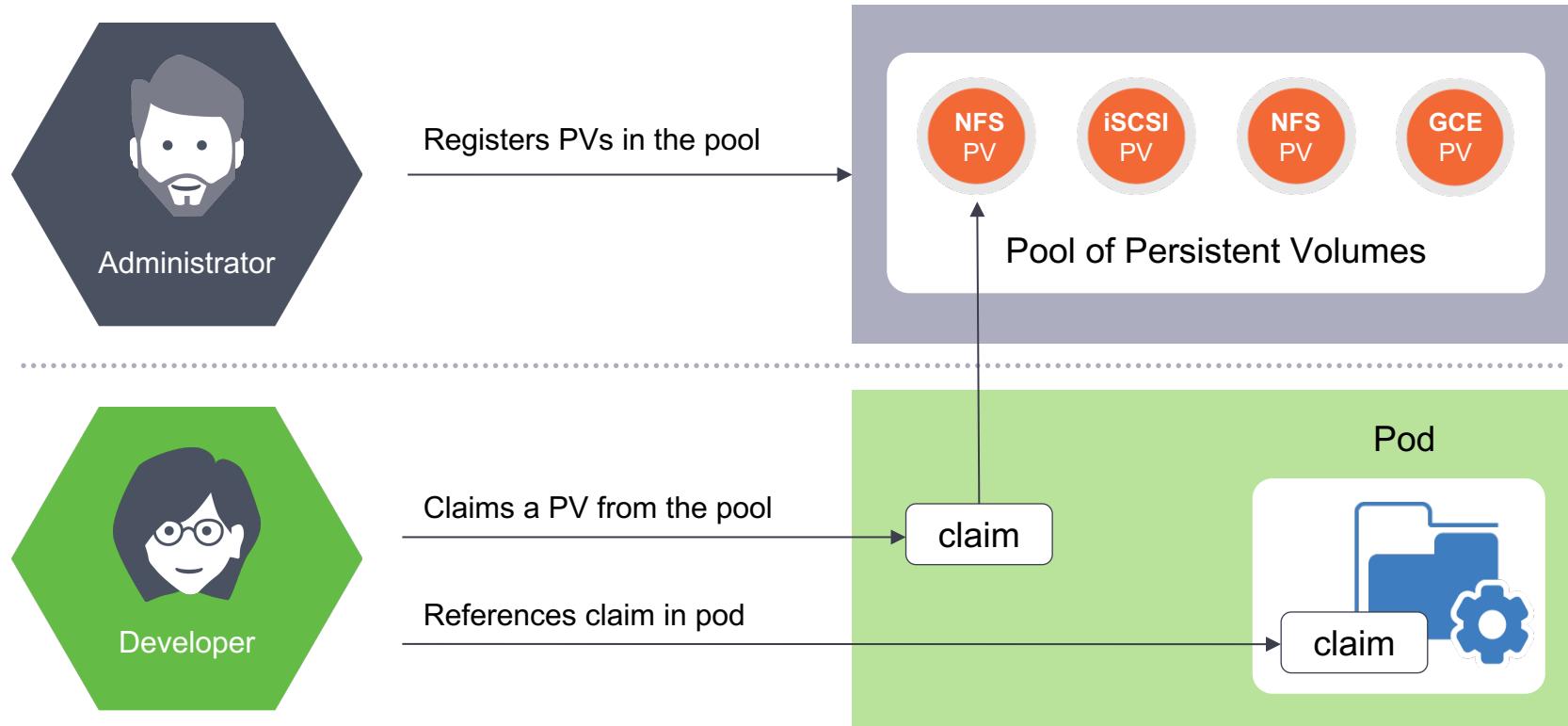
Kubernetes Storage Model: Persistent Volumes and Claims

@oicheryl



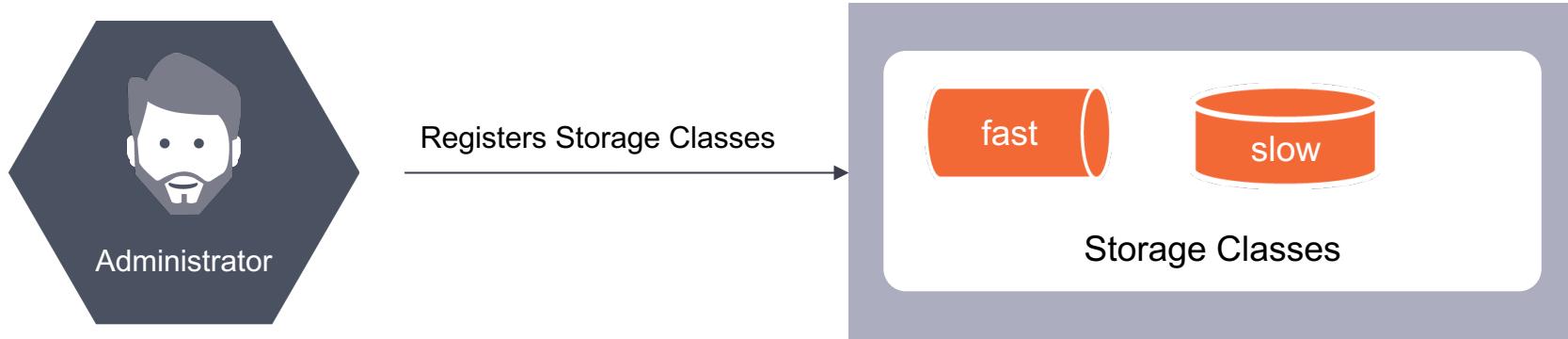
Kubernetes Storage Model: Persistent Volumes and Claims

@oicheryl



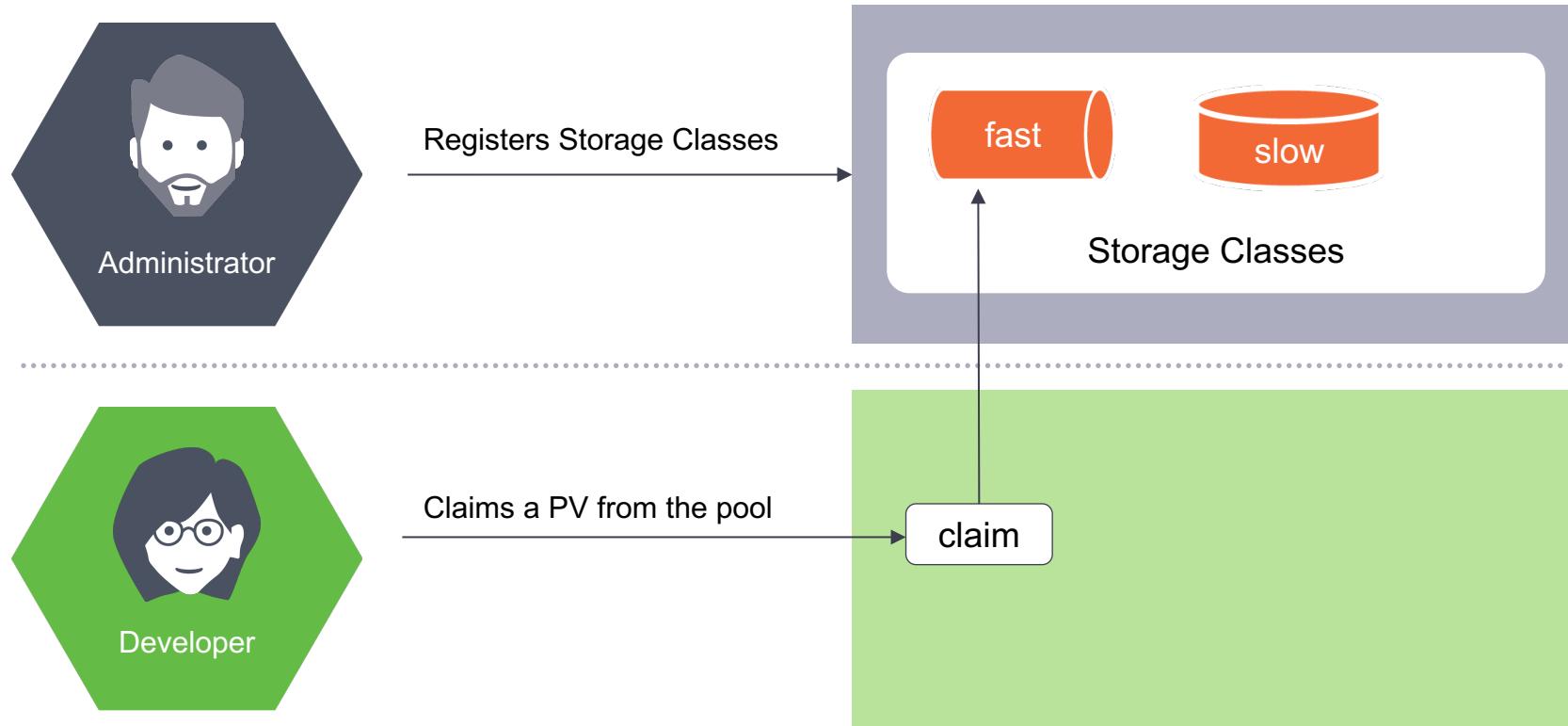
Dynamic provisioning with Storage Classes

@oicheryl



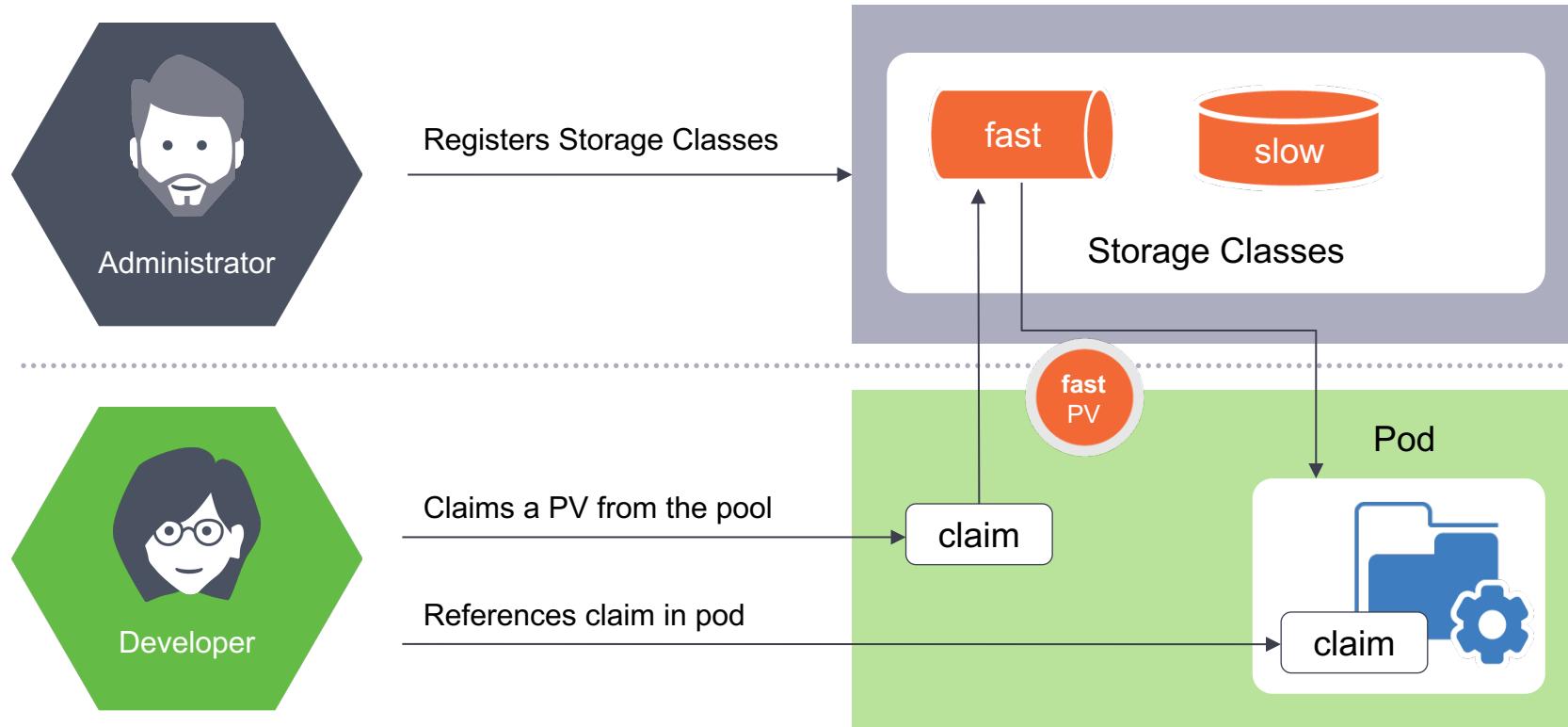
Dynamic provisioning with Storage Classes

@oicheryl



Dynamic provisioning with StorageClass

@oicheryl



Eight Principles of Cloud Native Storage



Horizontally scalable

No single point of failure

Resilient and self healing

Minimal operator overhead

Decoupled from the underlying platform



1

Storage should be presented to and consumed by applications, not by operating systems or hypervisors.

Storage needs to be able to follow an application as it scales, grows, and moves between platforms and clouds.





The storage platform should be able to run anywhere.
Upgrades and scaling is non-disruptive.



3

Storage resources should be declared and composed just like all other resources required by applications and services.





Storage resources and services should be easy to be provisioned, consumed, moved and managed via an API.



Storage services should integrate and inline security features such as encryption and RBAC.





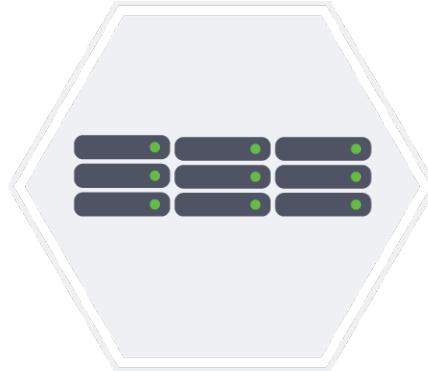
The platform should be able to move application data between locations, dynamically resize volumes for growth, take point in time copies of data for data retention or to facilitate rapid recovery of data.





The storage platform should be able to offer deterministic performance in complex distributed environments.





Block Storage

Data stored in a 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

8. Consistently available

@oicheryl

8

The storage platform should manage data distribution with a predictable, proven data model to ensure high availability, durability, consistency of application data.





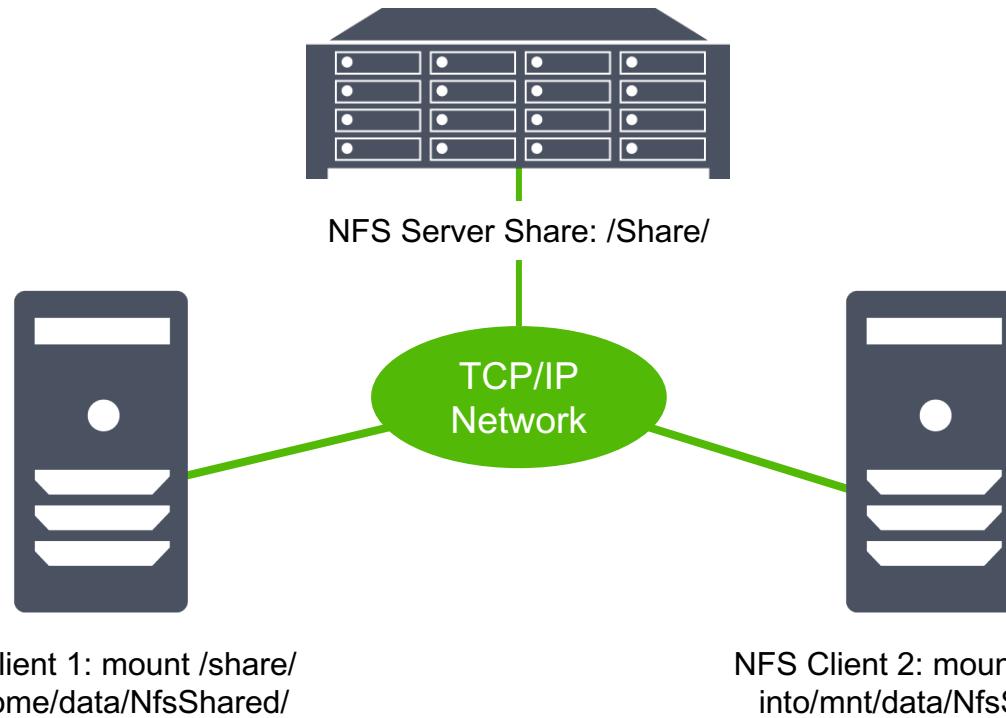
- A DevOps engineer in a bank
- She wants to migrate a Postgres database to containers

Storage Landscape



Centralized file system

@oicheryl



Single point of failure
Hard to scale horizontally
No native integration

0

Proprietary storage array

@oicheryl

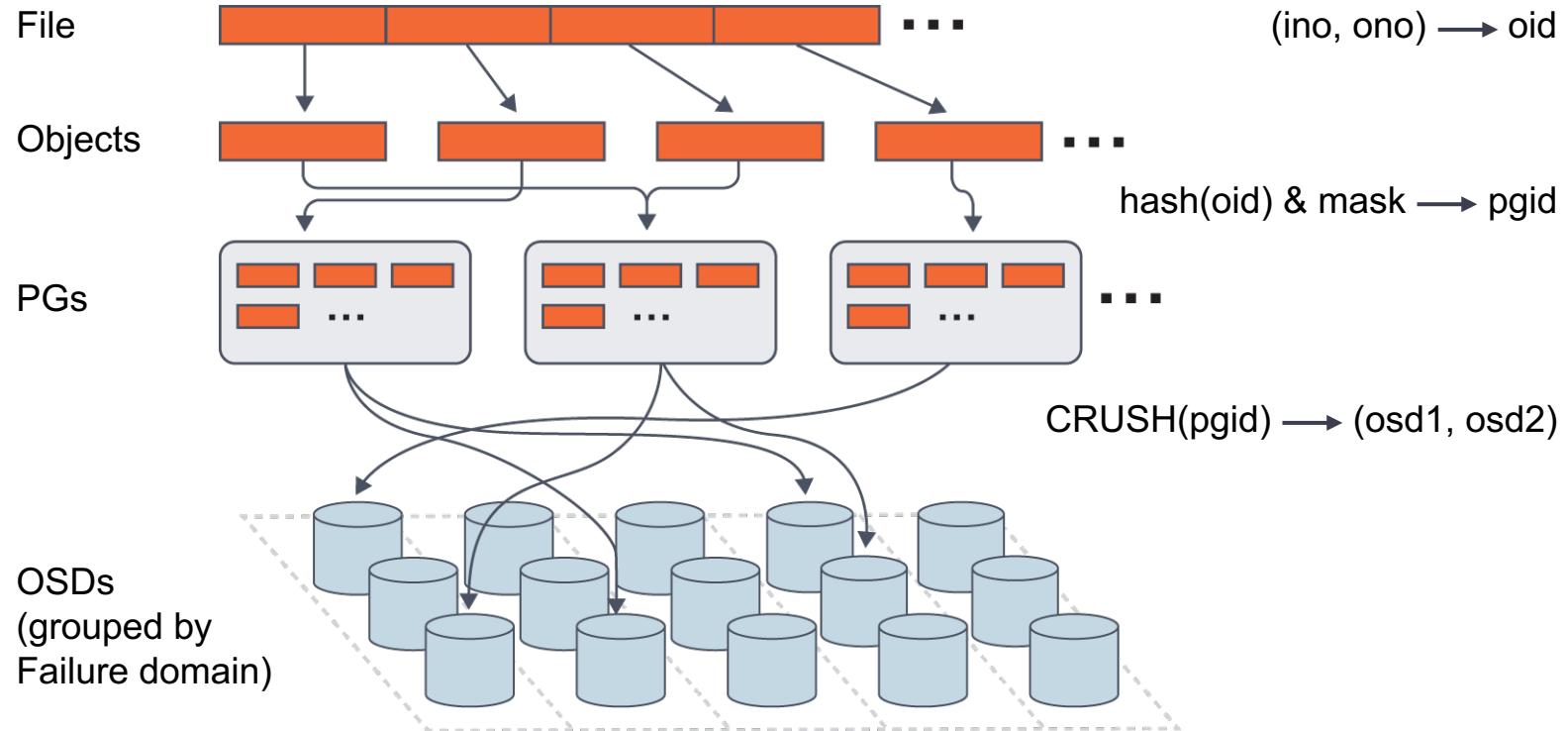


Deterministic performance
Vendor lock in
No thin provisioning
Hard to scale horizontally
Expensive and long lead times

2

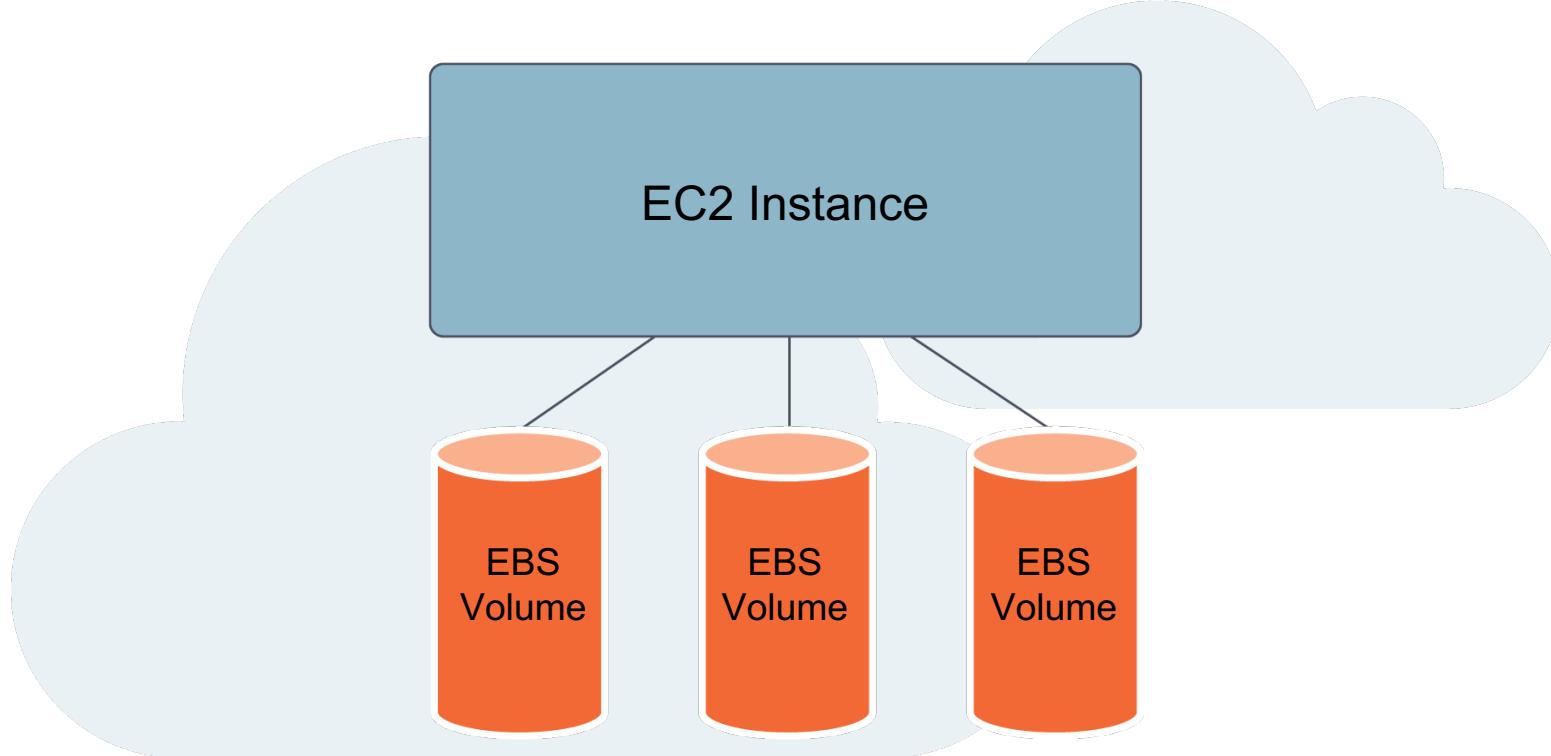
Distributed storage

@oicheryl



Horizontally scalable
Hardware agnostic
Write amplification
Failures are expensive
Complicated to set up

4



6

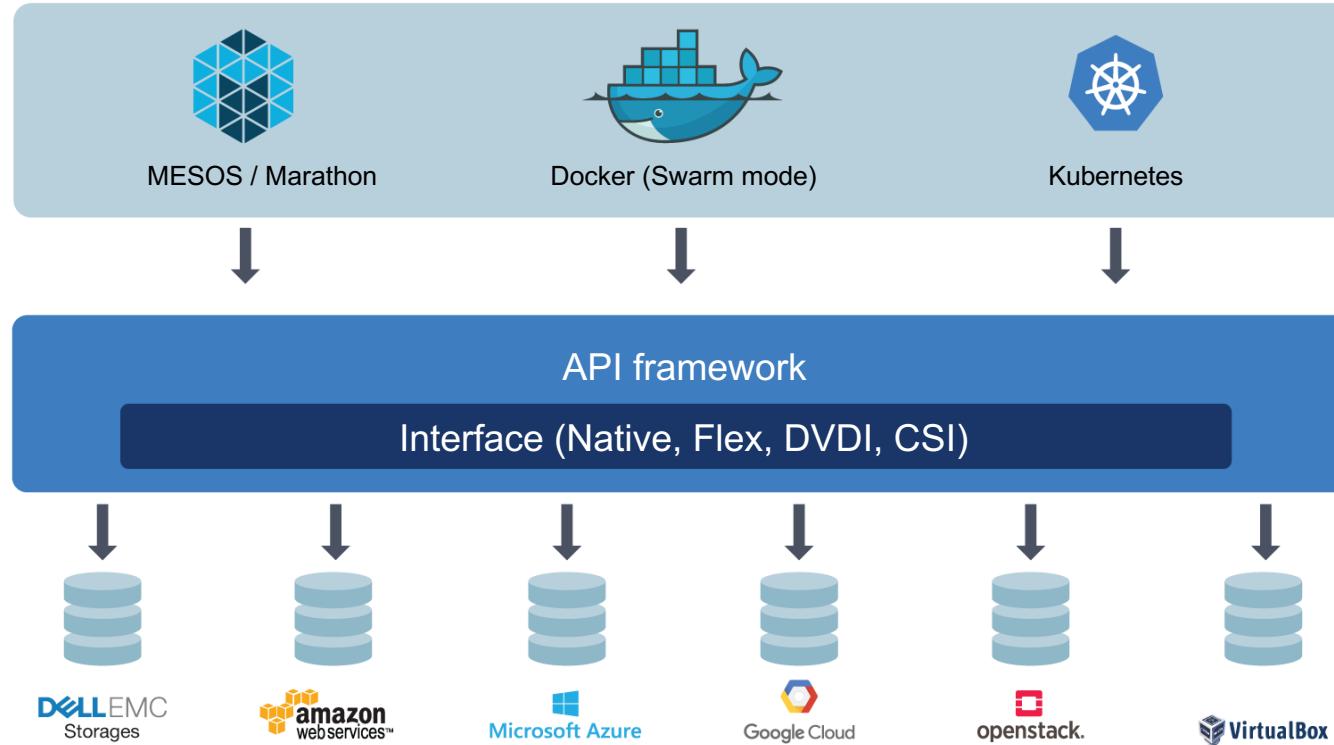
Horizontally scalable

Consistent and performant

40 EBS volumes per EC2 instance

Mounting physical block devices is slow

Expensive, vendor lock in, compliance



Docker/Kubernetes integration
No guarantees for performance,
replication, failover, scalability

5

Orchestrated storage (StorageOS)

@oicheryl



A software-defined, scale-out storage platform for running enterprise containerized applications in production



Platform
agnostic

A software-defined, scale-out storage
platform for running enterprise
containerized applications in production



Platform
agnostic

Horizontally
scalable

A software-defined, scale-out storage
platform for running enterprise
containerized applications in production



Platform
agnostic

Horizontally
scalable

Database
(ie. block)

A software-defined, scale-out storage
platform for running enterprise
containerized applications in production



Platform
agnostic

Horizontally
scalable

Database
(ie. block)

A software-defined, scale-out storage
platform for running enterprise
containerized applications in production

Docker/K8s
integration



Platform
agnostic

Horizontally
scalable

Database
(ie. block)

A software-defined, scale-out storage
platform for running enterprise
containerized applications in production

Docker/K8s
integration

High
availability





**KEEP
CALM
IT IS
DEMO
TIME**

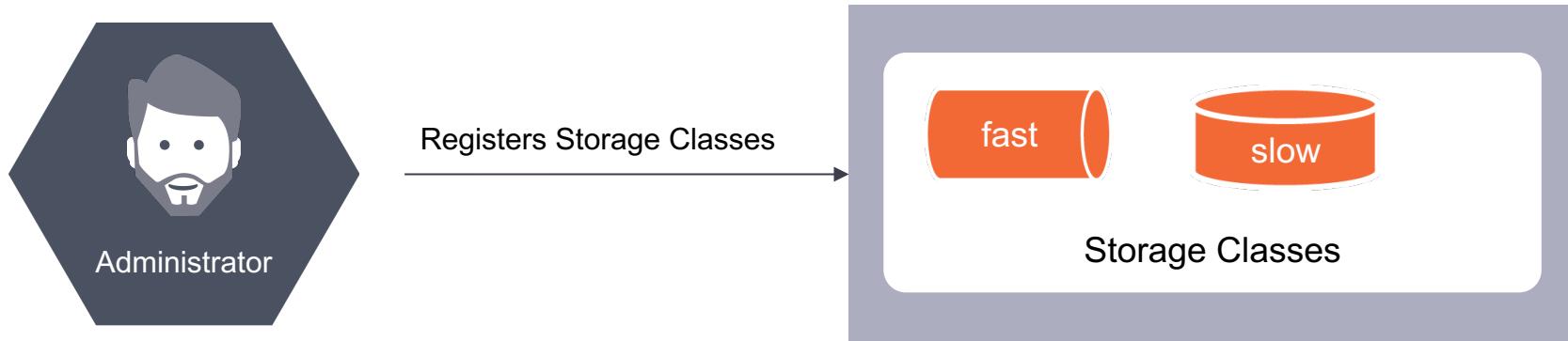
Orchestrated storage (StorageOS)

@oicheryl



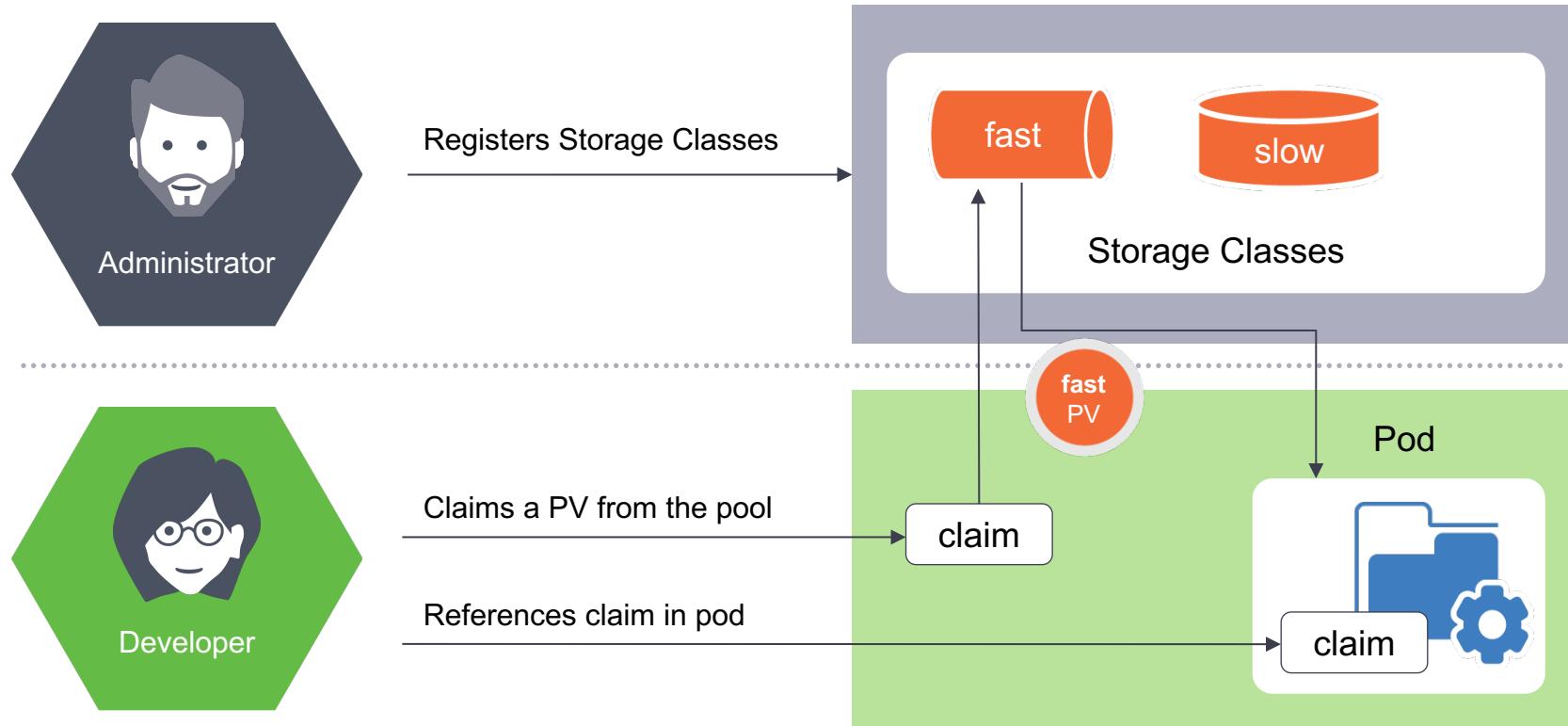
Create a StorageClass

@oicheryl



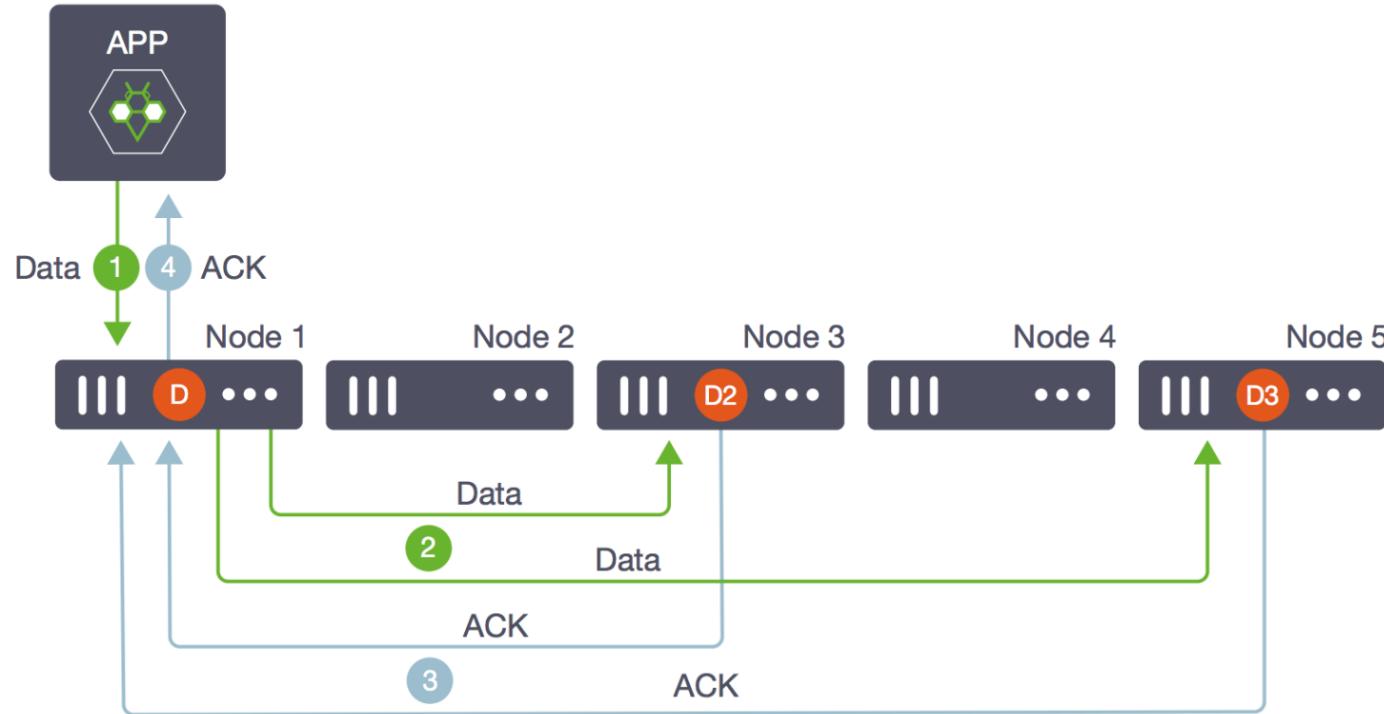
Claim a persistent volume

@oicheryl



High availability with StorageOS

@oicheryl



8

Horizontally scalable
Consistent and performant
Platform agnostic
Synchronous replication
Volume is limited to the size of one node

To Recap...



But, you can compare storage using the eight principles:

1. Application centric

5. Natively secure

2. Platform agnostic

6. Agile

3. Declarative and
composable

7. Performant

4. API driven

8. Consistently available



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.

Browser-based tutorials

- my.storageos.com/main/tutorials

Quickstart

- storageos.com/kubernetes

Slack channel

- slack.storageos.com



We're hiring!





Cloud Native London, Tue 12 Dec

Speakers from UK Gov Digital Service, Bitnami,
RealKinetic

meetup.com/Cloud-Native-London

Questions

Slides at oicheryl.com

