AmazingStuffPro
AMAZE NOT AMUSE

Powered by

Shoot yourself in the foot - Databases on Kubernetes. Fundamentals

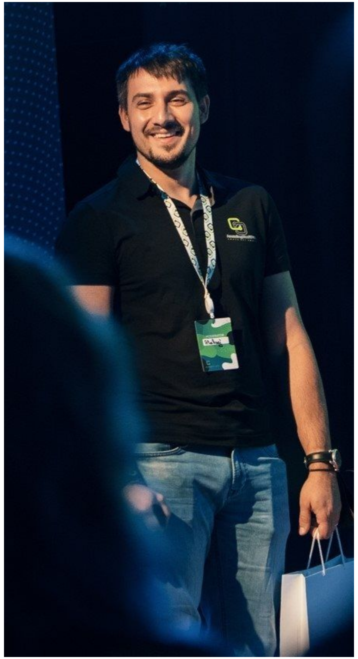
Sli.do event code:	**kuber46710**
Github repository:	**bit.ly/dbonk8s**
AmazingStuffPro Slack:	**bit.ly/slackamazingstuff**
Event hashtag:	**#amazingstuffpro**

Agenda:

● *Why do we need it?*	*15min.*
○ *QA*	*5 min.*
● *Storage for your database in k8s*	*~30min.*
● *Persistent data gravity with cross region/zone mobility*	*~15min.*
○ *QA*	*5 min.*

ALEKSEJ TROFIMOV

System Owner in Foundation Services team at Mambu



Why do we need it?

AmazingStuffPro
AMAZE NOT AMUSE

Powered by

AmazingStuffPro
AMAZE NOT AMUSE

Powered by

Why?

- In-service difference
  - Different Cloud Providers support different service functionality, like MySQL in AWS not similar to MySQL in GCP or Azure;
  - Different IAM;
  - Different backup and restore procedures.
- Who wants a vendor-lock-in?
  - DR requirement between different Cloud providers;
- Service functional limitation:
  - Like database plugins;
- Cloud providers “lags” for a new technologies roll-out:
  - CockroachDB;
  - MongoDB;

Slido: kuber46710

AmazingStuffPro
AMAZE NOT AMUSE

Powered by

What do we have?

- “Old-school DBs”: PostgreSQL (Spilo, Stolon), MySQL (Galera from Percona/MariaDB) with some limitations:
    - wasn’t designed for Kubernetes;
    - lack of tooling;
    - poor support;
    - small community who run it in Kubernetes (why??);
- “Modern DBs”: CockroachDB, Vitess, TiDB, etc. Build with Kubernetes (operators) in mind. Looks pretty cool, tries to have interface (API) similar to PostgreSQL or MySQL, but not fully compatible with old-school DBs.

YOU REMIND ME OF A SOFTWARE UPDATE
WHENEVER I SEE YOU, I USUALLY THINK "NOT NOW"

Slido: kuber46710

AmazingStuffPro
AMAZE NOT AMUSE

Powered by

Any challenges?

- Modern databases - modern solution - e.g. easier to have one huge cluster instead of hundreds of small clusters.
- Kubernetes Storage layer looks stable but with some challenges.
  - Like performance (CSI plugins can provide a unified way across Cloud Providers but adds performance penalty; Performance of storage in different clouds are different) and support;
  - Tooling around and etc.;
  - Complexity;
- Different features for Storage layer in different clouds - Good news are encryption at rest and backups are supported in some flavour in main clouds.

Slido: kuber46710

AmazingStuffPro
AMAZE NOT AMUSE

Powered by

Thank you!

Slido: kuber46710

AmazingStuffPro

AMAZE NOT AMUSE

Powered by

Questions?

Sli.do event code: **kuber46710**

UP NEXT

Storage for your database in k8s

Augustinas Stirbis

Director of Engineering

at cast

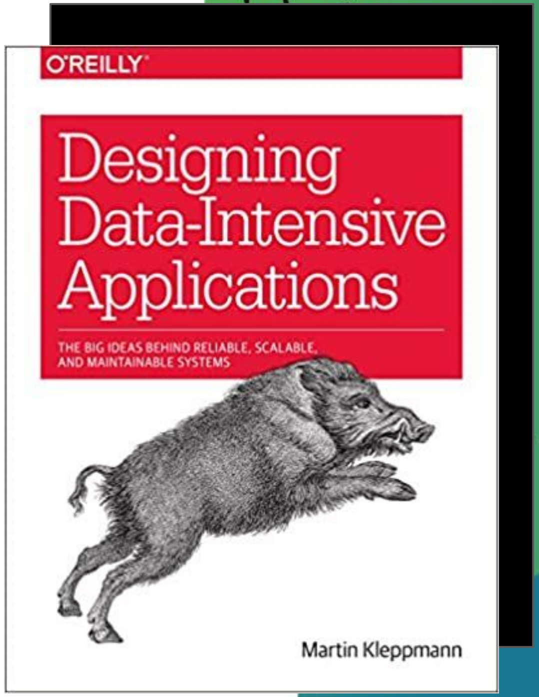
AmazingStuffPro
AMAZE NOT AMUSE

Powered by

Why do you need storage for database in 2021

Different types of databases require different types of storage:

- Traditional RDBMS (MySQL / PostgreSQL / MS SQL)
- Distributed DBs (Cassandra / ElasticSearch / Yugabyte / Cockroach)
- In memory databases (Redis)



O'REILLY

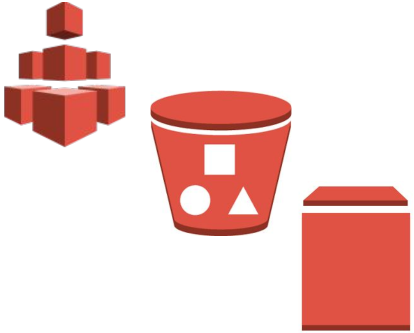
Designing Data-Intensive Applications

THE BIG IDEAS BEHIND RELIABLE, SCALABLE, AND MAINTAINABLE SYSTEMS

Martin Kleppmann

Type storage:

- File - AWS EFS
- Object - AWS S3
- Block - AWS EBS



Block based storage:

1. Local disks: SATA/SAS or very fast NVMe interfaces
2. Remote storage:
   a. Fiber Channel network - on premise DC expensive SAN with HBA cards
   b. TCP/IP network - iSCSI, various Software Defined Storage proprietary solutions (Tomas demo portworx)

AmazingStuffPro
AMAZE NOT AMUSE

Powered by

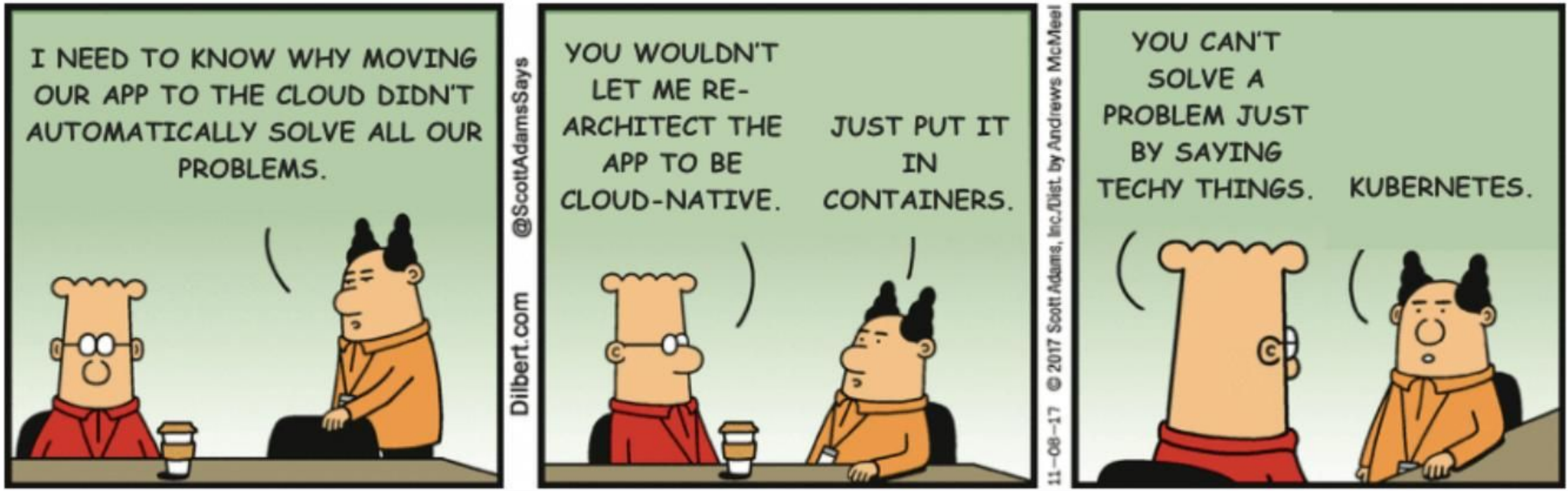
Storage for traditional server:

1. Understand your application storage requirements (size, IOps, throughput)
2. Provision Volume based of requirements
3. Attach Volume (insert disk / zoning-masking)
4. Create Partition (fdisk)
5. Format partition with filesystem (ext4)
6. Mount partition as dir
7. Configure Database to write to path
8. Profit, your database can retrieve written data!

AmazingStuffPro

AMAZE NOT AMUSE

Powered by


I NEED TO KNOW WHY MOVING OUR APP TO THE CLOUD DIDN'T AUTOMATICALLY SOLVE ALL OUR PROBLEMS.
YOU WOULDN'T LET ME RE-ARCHITECT THE APP TO BE CLOUD-NATIVE.
JUST PUT IT IN CONTAINERS.
YOU CAN'T SOLVE A PROBLEM JUST BY SAYING TECHY THINGS.
KUBERNETES.
Dilbert.com @ScottAdamsSays
11-08-17 © 2017 Scott Adams, Inc./Dist. by Andrews McMeel

Storage for database on Kubernetes with CSI driver

1. Understand your application storage requirements (size, IOps, throughput)

~~2. Provision Volume based of requirements~~

~~3. Attach Volume (insert disk / zoning masking)~~

~~4. Create Partition (fdisk)~~

~~5. Format partition with filesystem (ext4)~~

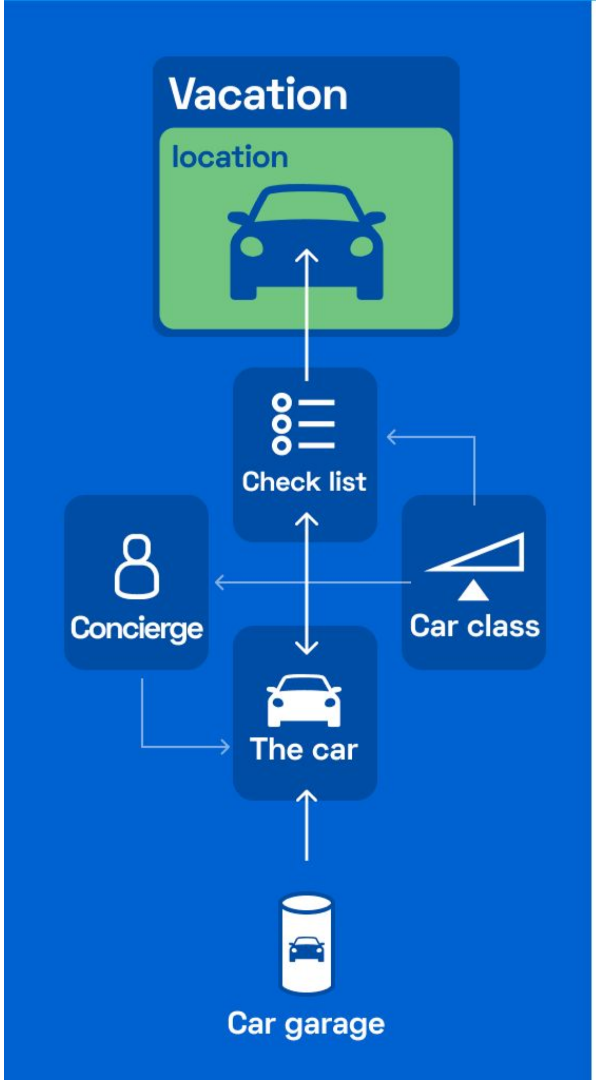
~~6. Mount partition as dir~~

7. Configure Database to write to path

8. Profit, your database can retrieve written data!

AmazingStuffPro
AMAZE NOT AMUSE

Powered by

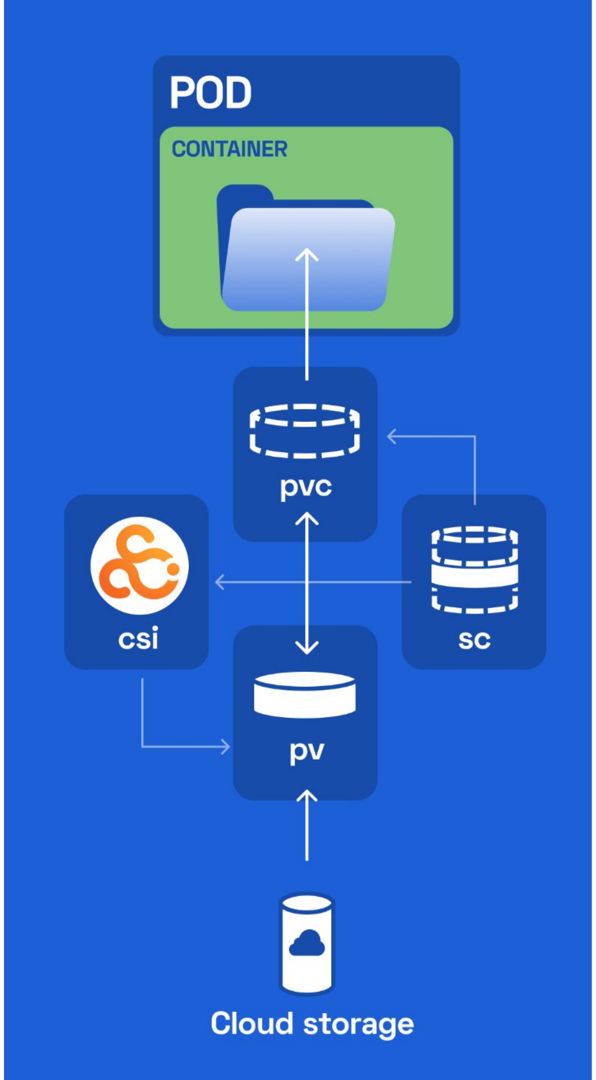


+ Car // Allocate storage

CONTAINER

AmazingStuffPro
AMAZE NOT AMUSE

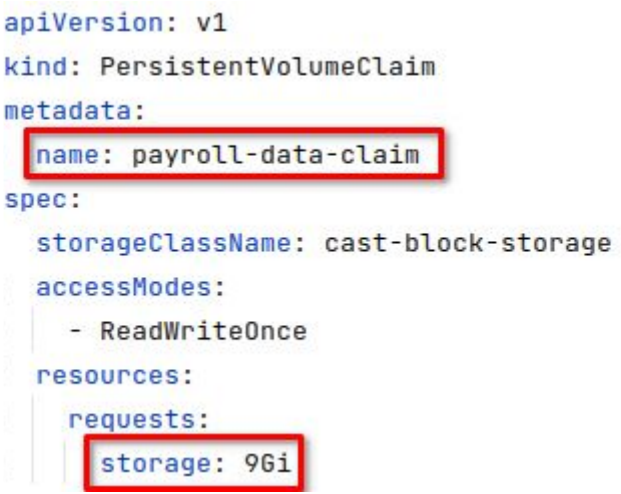
Powered by



+ 5 persons // 500GB

+ Car

Persistent Volume Claim

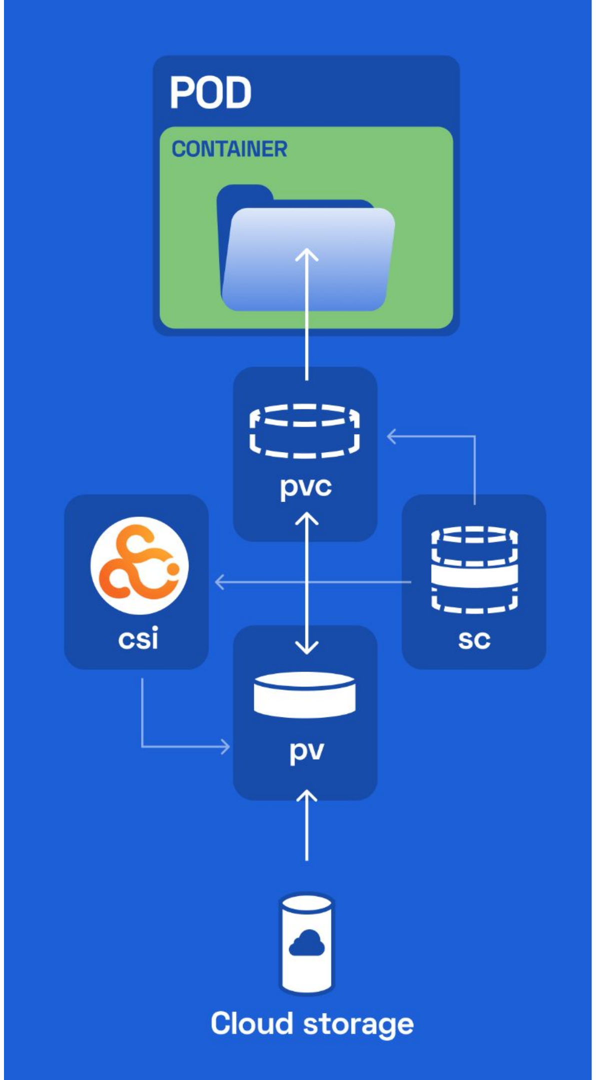


```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: payroll-data-claim
spec:
  storageClassName: cast-block-storage
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 9Gi
```

AmazingStuffPro

AMAZE NOT AMUSE

Powered by



+ A class //SSD

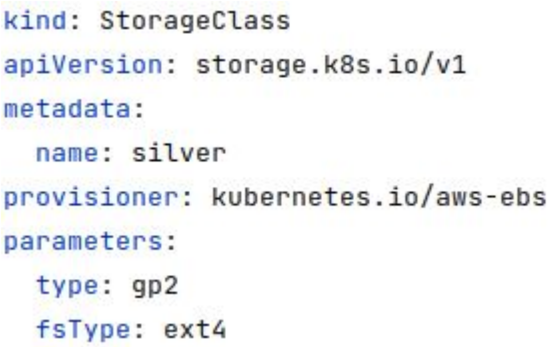
- + 5 persons
- + Car

sc

AmazingStuffPro
AMAZE NOT AMUSE

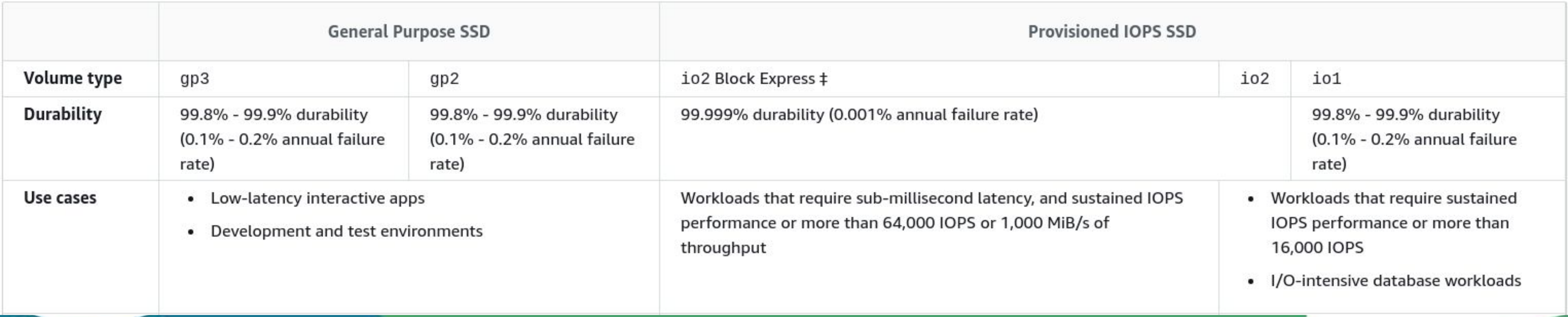
Powered by

Storage Class - Bronze, Silver, Gold

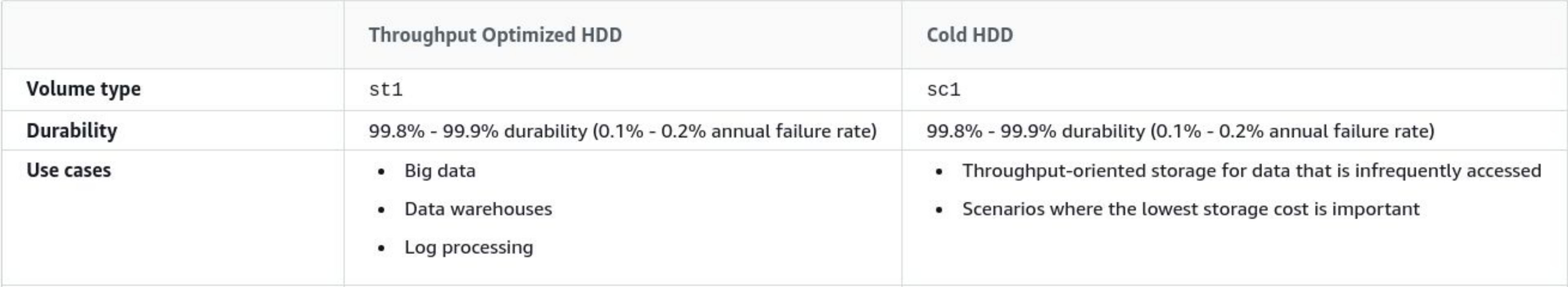


```yaml
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: silver
provisioner: kubernetes.io/aws-ebs
parameters:
  type: gp2
  fsType: ext4
```

7 volume types available in AWS EBS



	General Purpose SSD		Provisioned IOPS SSD		
Volume type	gp3	gp2	io2 Block Express ‡	io2	io1
Durability	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)	99.999% durability (0.001% annual failure rate)		99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)
Use cases	• Low-latency interactive apps • Development and test environments		Workloads that require sub-millisecond latency, and sustained IOPS performance or more than 64,000 IOPS or 1,000 MiB/s of throughput	• Workloads that require sustained IOPS performance or more than 16,000 IOPS • I/O-intensive database workloads	

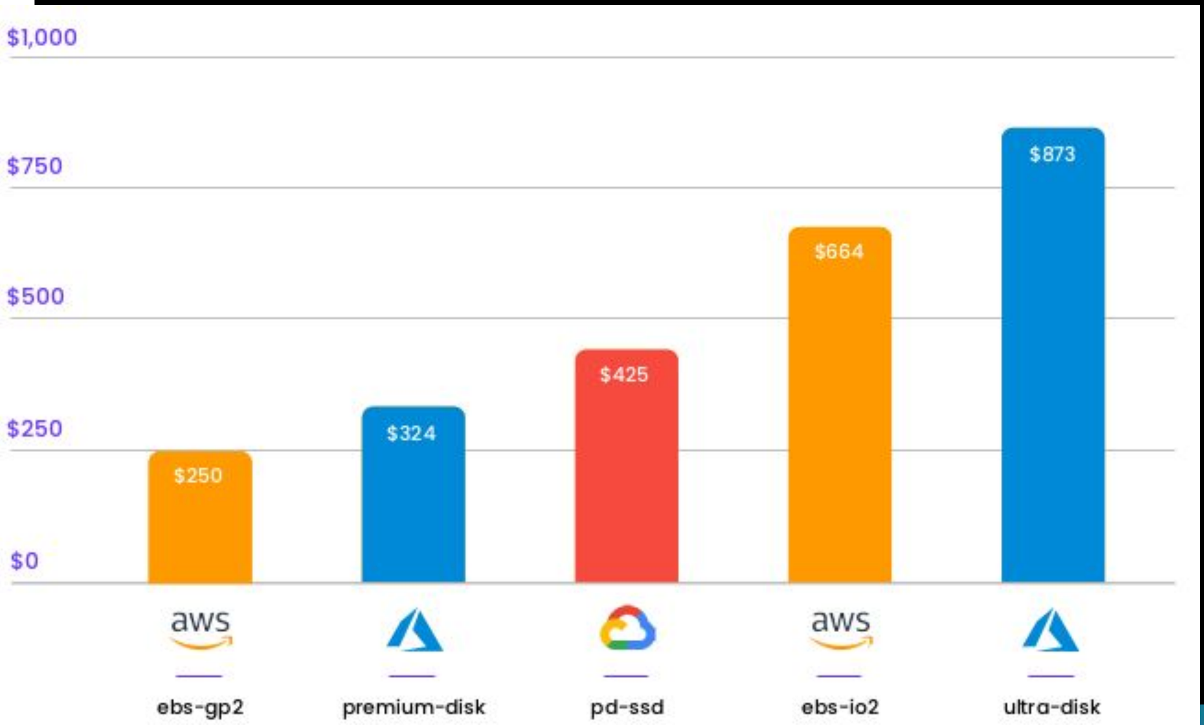


	Throughput Optimized HDD	Cold HDD
Volume type	st1	sc1
Durability	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)
Use cases	• Big data • Data warehouses • Log processing	• Throughput-oriented storage for data that is infrequently accessed • Scenarios where the lowest storage cost is important

AmazingStuffPro
AMAZE NOT AMUSE

Powered by

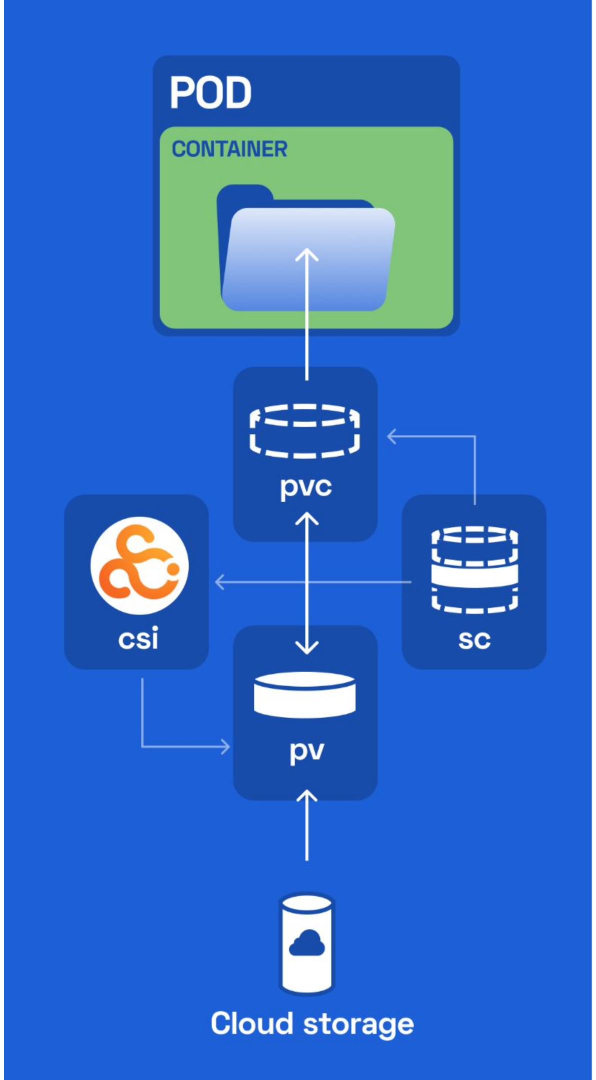
Storage prices



Cockroach Labs cloud report

AmazingStuffPro
AMAZE NOT AMUSE

Powered by



+ High durability //HA

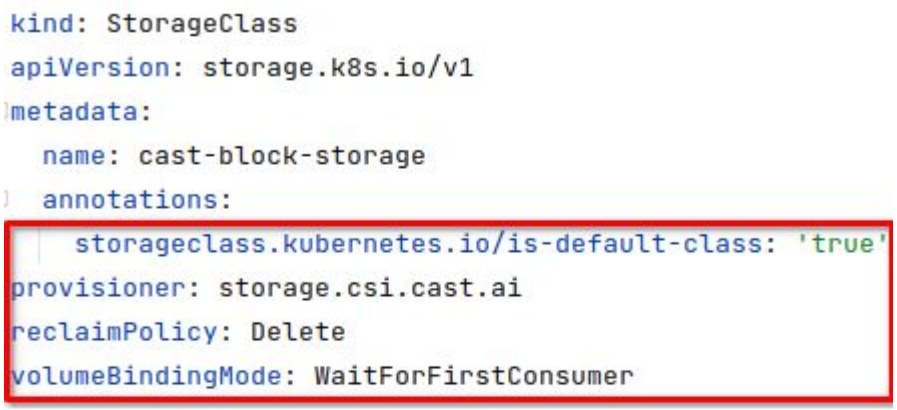
- + A class
- + 5 persons
- + Car

SC

AmazingStuffPro
AMAZE NOT AMUSE

Powered by

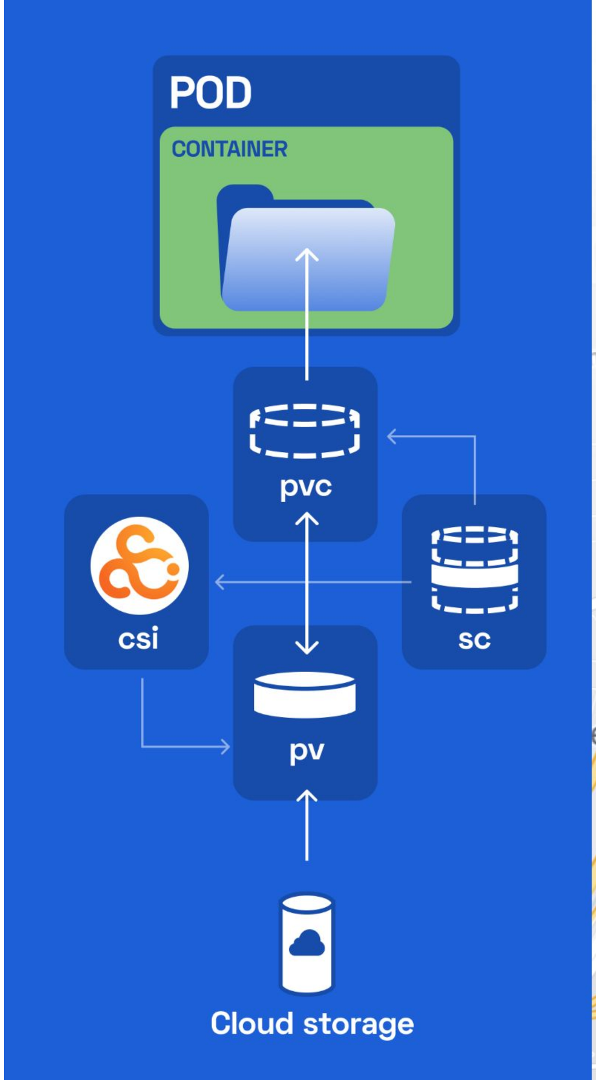
Advanced Storage Class options



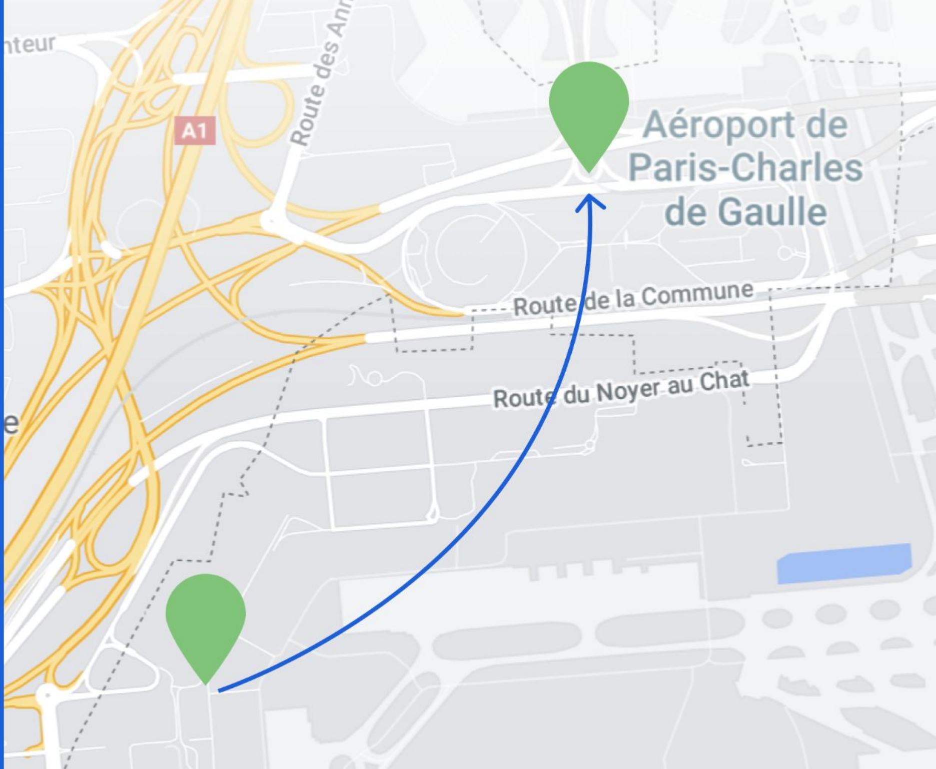
```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: cast-block-storage
  annotations:
    storageclass.kubernetes.io/is-default-class: 'true'
provisioner: storage.csi.cast.ai
reclaimPolicy: Delete
volumeBindingMode: WaitForFirstConsumer
```

AmazingStuffPro
AMAZE NOT AMUSE

Powered by



- + Full tank //Provision
- + Clean //Attach
- + Deliver //Format
- + Give keys //Mount

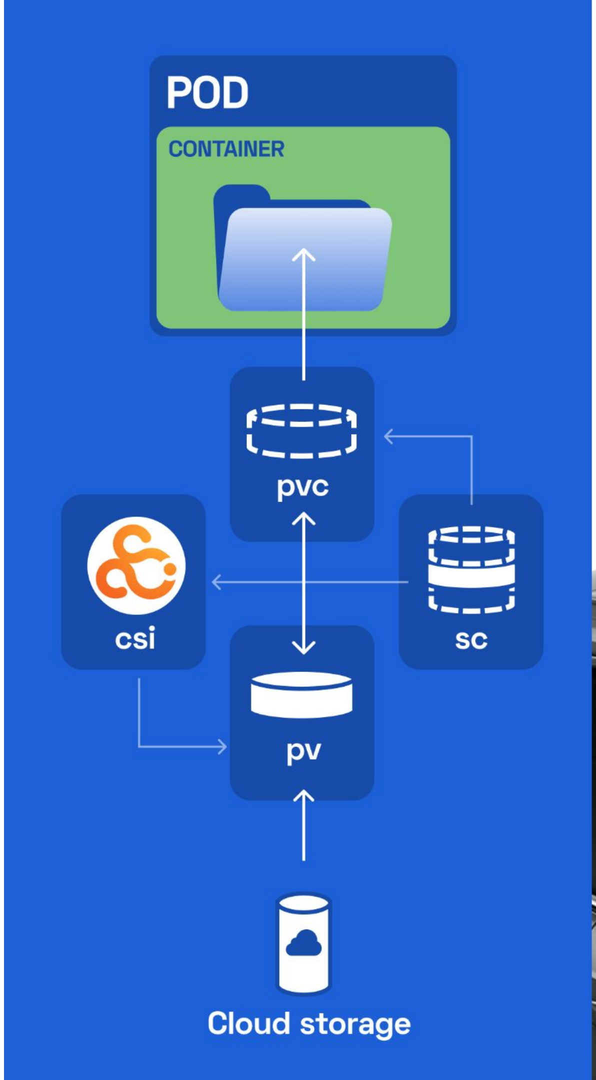


csi

AmazingStuffPro

AMAZE NOT AMUSE

Powered by



+ Expandable //+1TB

- + High durability
- + A class
- + 5 persons
- + Car

pvc

AmazingStuffPro
AMAZE NOT AMUSE

Powered by

Demo

or it does not exist!

AmazingStuffPro

AMAZE NOT AMUSE

Powered by

Thank you!

Slido: kuber46710

AmazingStuffPro
AMAZE NOT AMUSE

Powered by

Questions?

Sli.do event code:

kuber46710

UP NEXT

Persistent data gravity with cross region/zone mobility

Tomas Vaiciunas

SRE Core Platform at Mambu

Slido: kuber46710

AmazingStuffPro

AMAZE NOT AMUSE

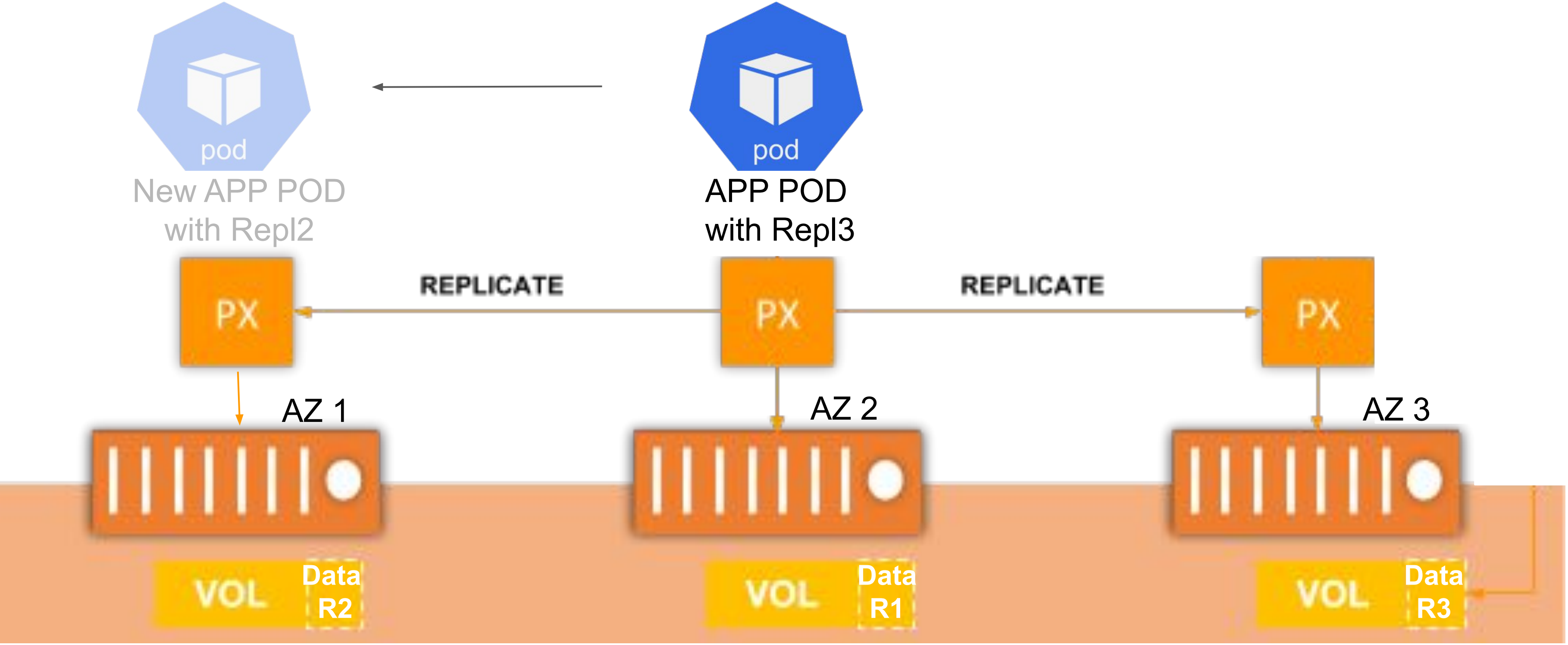
Powered by

How to make sure your persistent storage can tolerate AZ failure?

Slido: kuber46710

AmazingStuffPro
AMAZE NOT AMUSE

Powered by



Slido: kuber46710

AmazingStuffPro

AMAZE NOT AMUSE

Powered by

You can use SLIDO

For your questions!

Slido: kuber46710

AmazingStuffPro

AMAZE NOT AMUSE

Powered by

Things to consider:

- Network latency
- Cluster topology (no single AZ)
- Amount of data copies you need
- Overhead latency that is going to be introduced
- More expensive because of pre allocated storage on other AZ's

Slido: kuber46710

AmazingStuffPro

AMAZE NOT AMUSE

Powered by

Why would you consider running storage replication on top of distributed application?

- Even if application is capable of performing replication of data on its own, it will be more time consuming to recover the node thereby resulting in a negative impact on the performance. Running Sync replication in the background will improve recovery time.

Downside:

- More storage will be needed to accommodate all copies = more expensive overall

Slido: kuber46710

AmazingStuffPro

AMAZE NOT AMUSE

Powered by

Demo context

Portworx + Minio

Slido: kuber46710

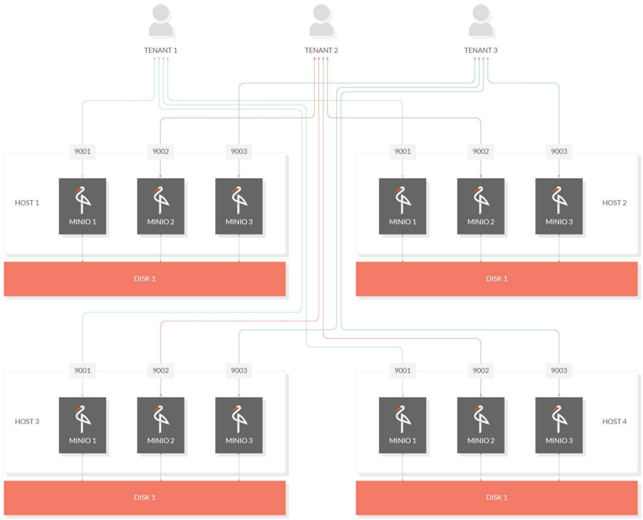
AmazingStuffPro

AMAZE NOT AMUSE

Powered by

What is Minio?

MinIO is a High Performance Object Storage (like AWS S3). When running in distributed mode it ensures that data is replicated on multiple drives tolerating m/2 loss of servers



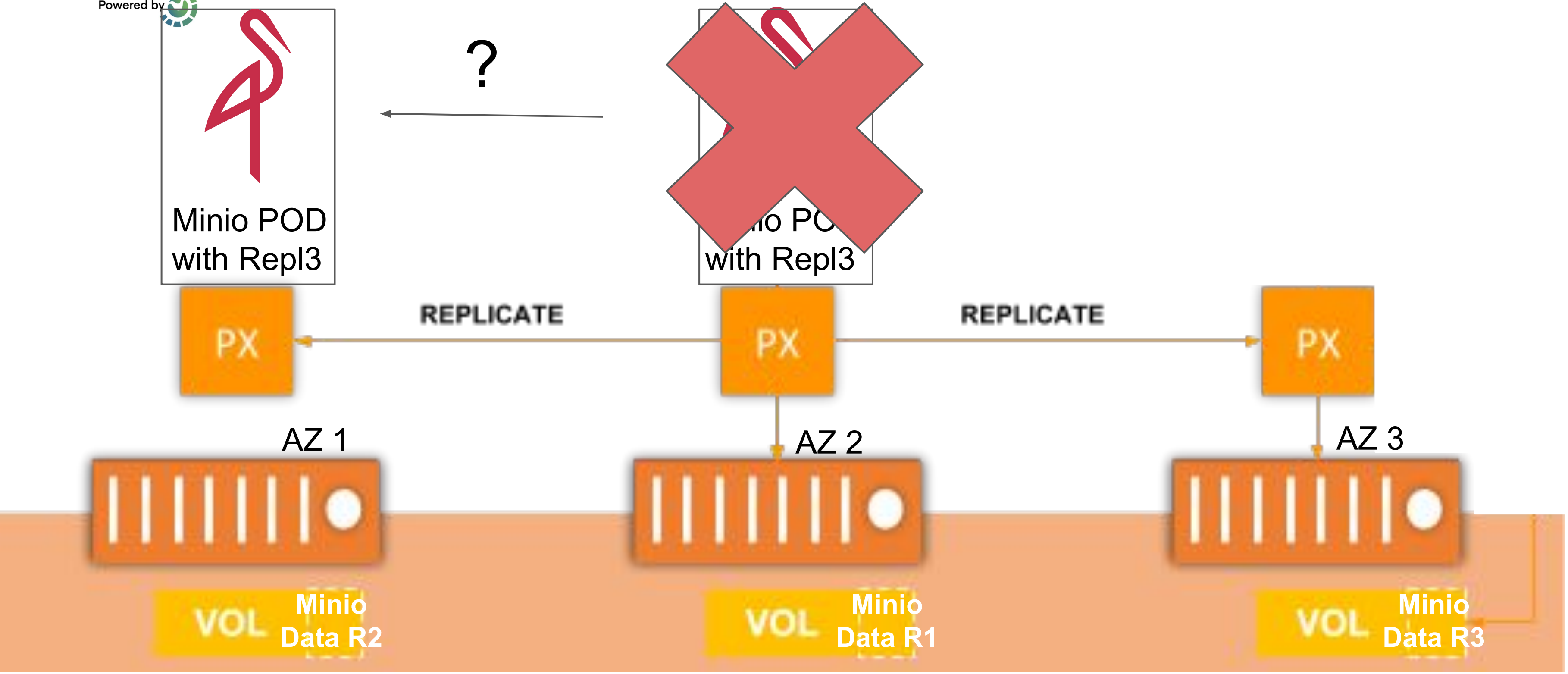
Example 3: 4 node distributed setup

Slido: kuber46710

What is Portworx?

Portworx is a software defined persistent storage solution designed and purpose built for applications deployed as containers, via container orchestrators such as Kubernetes, Marathon and Swarm. It is a clustered block storage solution and provides a Cloud-Native layer from which containerized stateful applications programmatically consume block, file and object storage services directly through the scheduler.

Slido: kuber46710


?
Minio POD
with Repl3
io PO
with Repl3
REPLICATE
REPLICATE
PX
PX
PX
AZ 1
AZ 2
AZ 3
VOL
Minio
Data R2
VOL
Minio
Data R1
VOL
Minio
Data R3

AmazingStuffPro

AMAZE NOT AMUSE

Powered by

Demo.

Slido: kuber46710

AmazingStuffPro

AMAZE NOT AMUSE

Powered by

Thank you!

Slido: kuber46710

AmazingStuffPro
AMAZE NOT AMUSE

Powered by

Questions?

Sli.do event code: **kuber46710**

AmazingStuffPro
AMAZE NOT AMUSE

Powered by

Upcoming
#AmazingStuffPro

Running Databases on Kubernetes

- Database operators - do they help?
- MySQL on Kubernetes (with demo)

AmazingStuffPro
AMAZE NOT AMUSE

Powered by

We value Your feedback.

http://bit.ly/feedbackamazingstuff