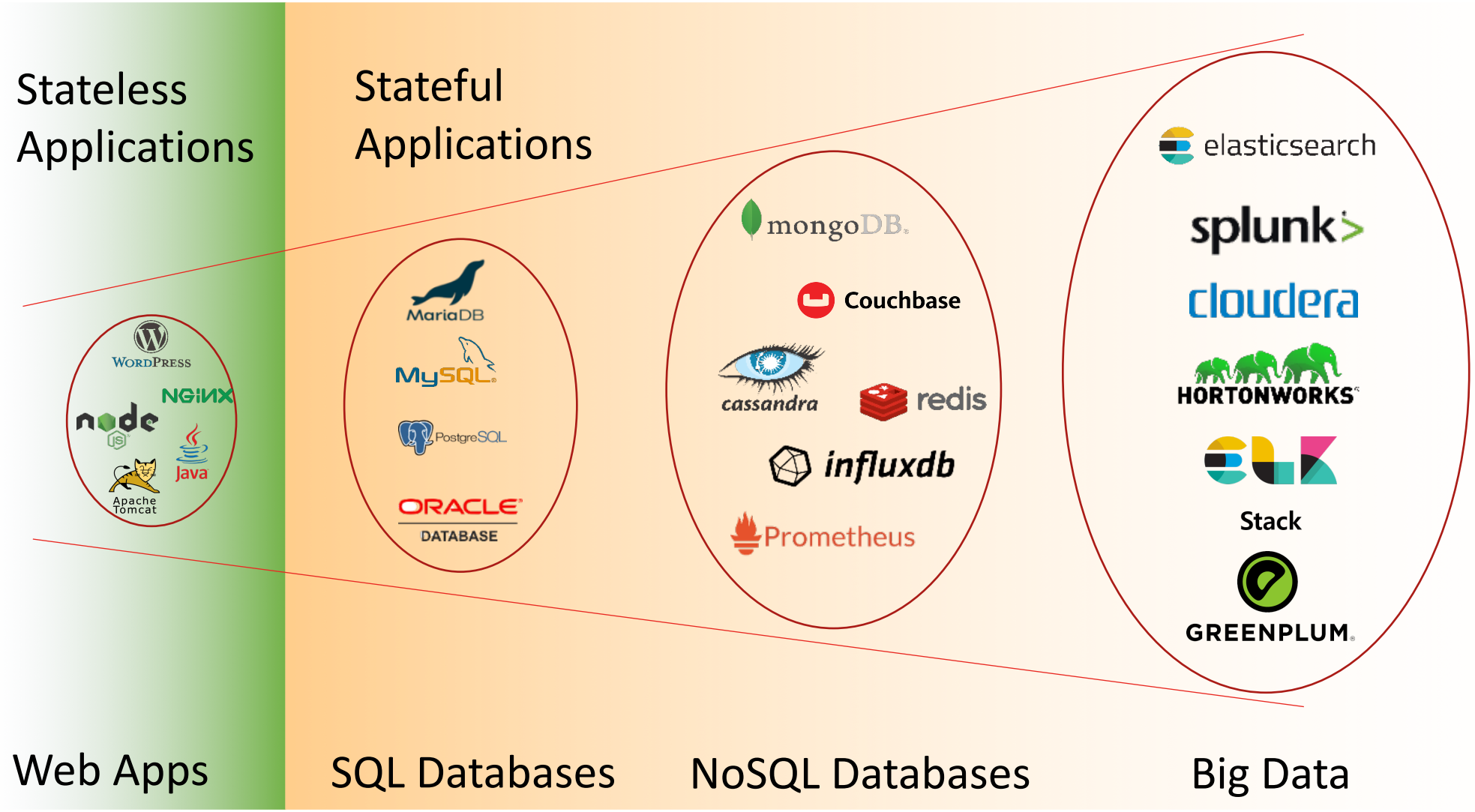
Application Modernization And Portability Across Clouds

Ravikumar Alluboyina

Senior Product Architect, Robin.io
https://get.robin.io

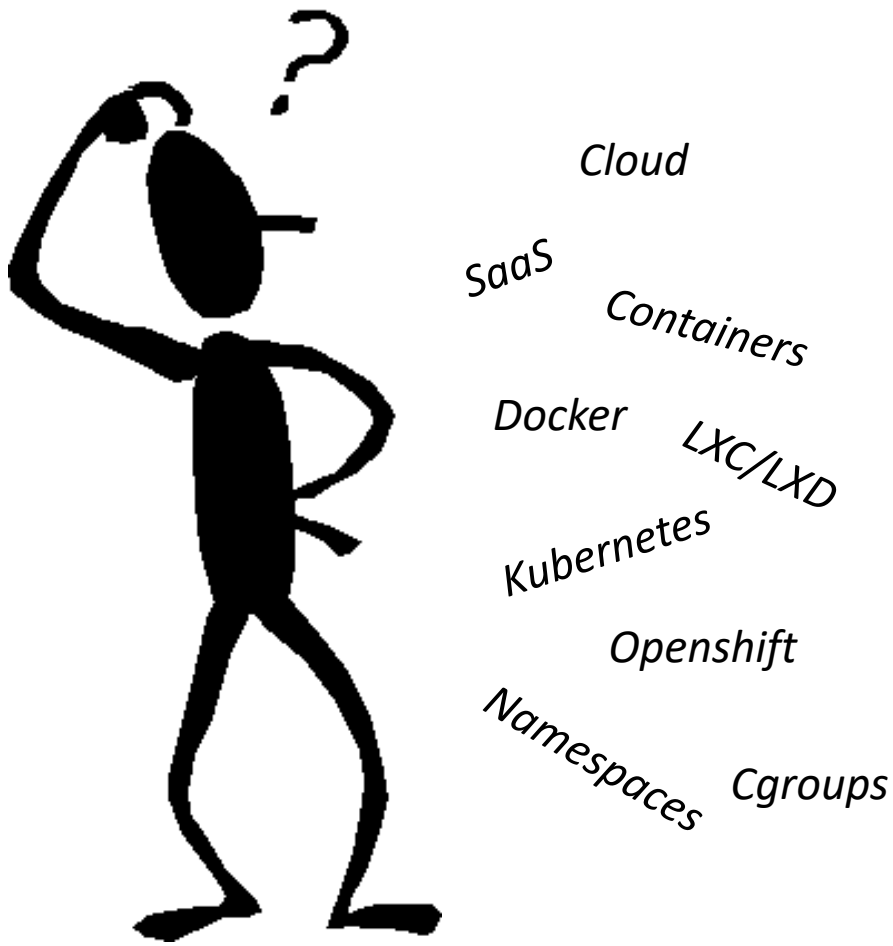
Spectrum of Applications



Application Modernization

Modernization.. The trend

- **What** is Modernization?
  - Define modernization?
  - Is there a standard?
- **Why** would I want to modernize my application?
  - What will I gain by modernizing
- **How** can I modernize my application?
  - What are the tools available?
  - What processes should I adopt?



Q. Virtualization did not force me to modernize my applications, why now? What changed?

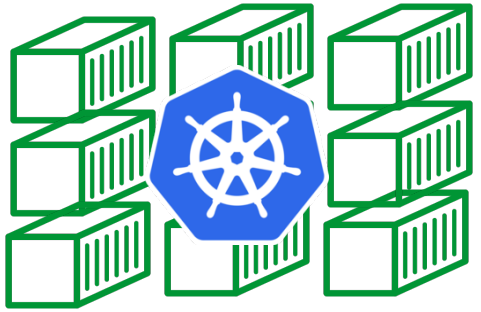
Modern Application .. The ‘What’

Application that can adapt to any environment and perform equally well

- Bare metal, Virtual machines, Cloud, Containers
- Mode of delivery (rpm, deb, tar vs runnable formats vmdk, docker container etc)
- Can integrate easily with pluggable infrastructure component.
  - Security, Performance monitoring, Logging, backup etc.,
- Is elastic, can grow and shrink depending on the load and usage
- Is agile / portable

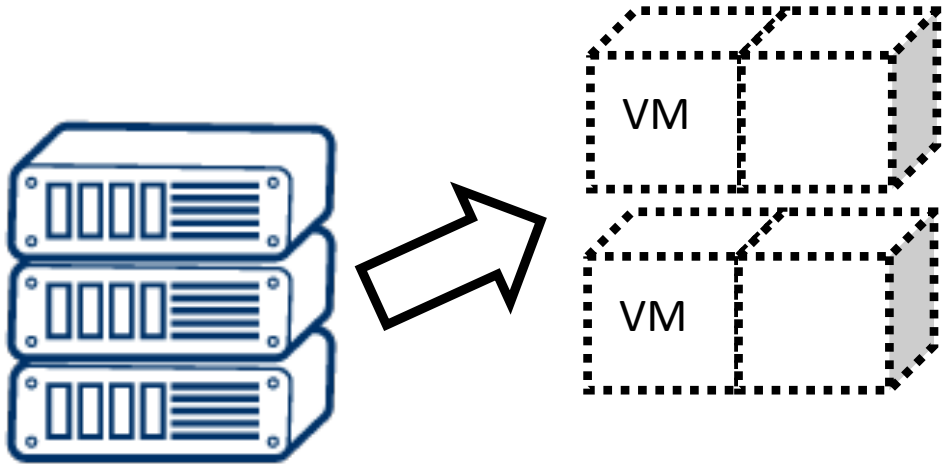
Why now, what changed?

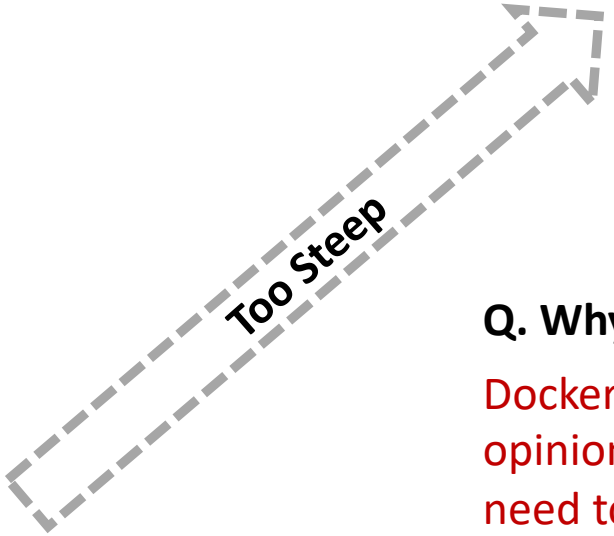
- The infrastructure landscape and tooling has changed drastically
  - We have Cloud, Virtual Machines, Containers, Dockers
  - Security, Logging, Monitoring frameworks have evolved



Q. Why is this a smooth transition for applications?

Application didn't need to change. Infrastructure emulated the baremetal

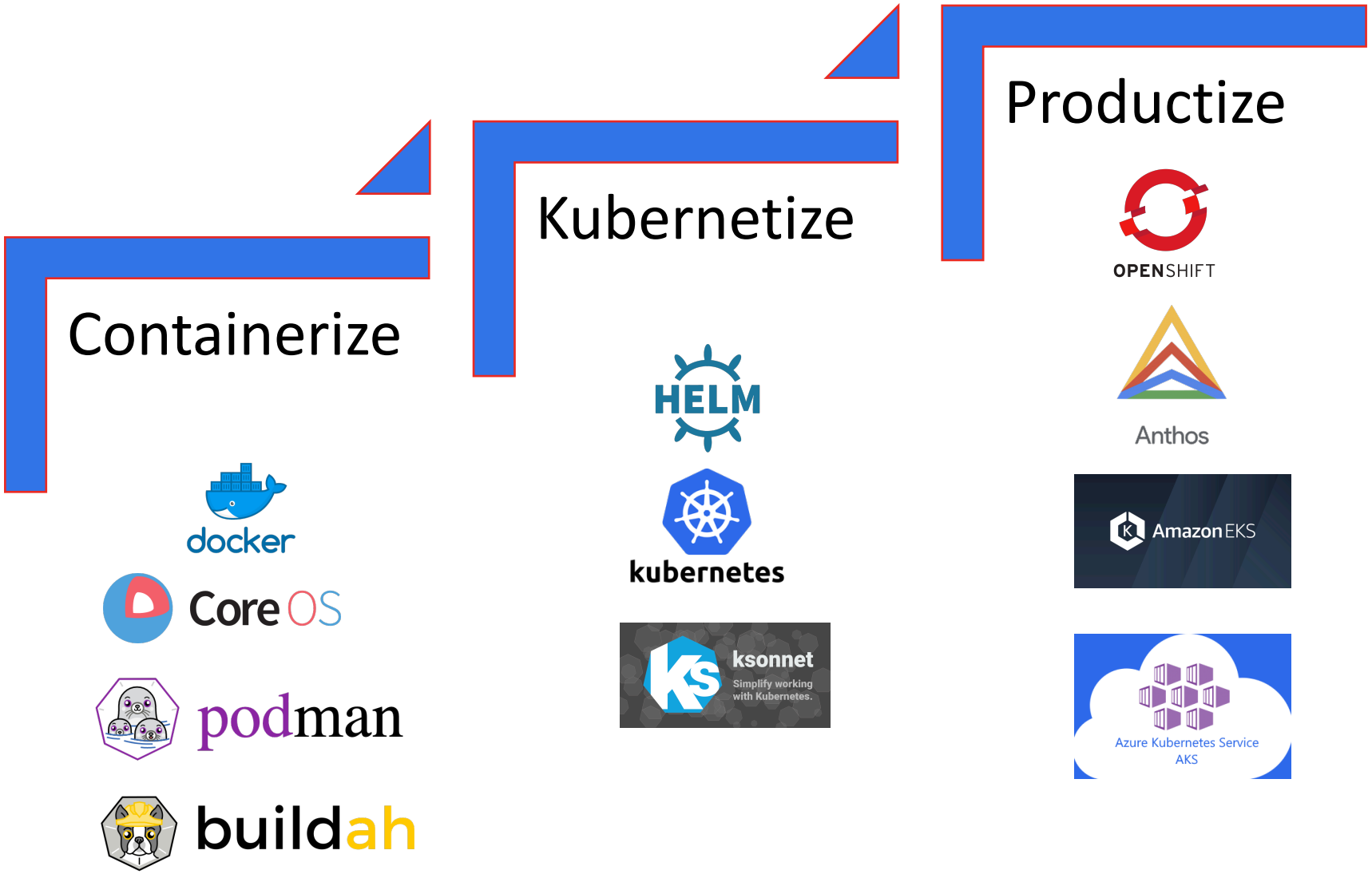




Q. Why is this a huge leap?

Docker / CRI-O, Kubernetes are all very opinionated frameworks. Applications need to be re-designed for these platforms.

Modernization .. The ‘How’



Containerization

Containerization: The process

- Process separation
  - Separate the application into multiple processes (if possible and if it makes sense)
  - For example:
    - Separate the nodemanager and datanode in Hadoop,
    - (or) query router and config server in MongoDB
- Configuration separation
  - Find the configuration paths (/etc, /var, etc) and persist it on a different volume (on the host)
  - This is by far the most difficult process for legacy applications
- Data separation
  - Attach volumes to the container so data can be preserved across restarts
- Entrypoint
  - Add enough logic in entrypoint to handle first-start, subsequent starts and upgrades.

Did you really think this was going to be enough to Dockerize?

Docker: The real issues

Incomplete cgroups virtualization causes many Big Data and Databases to misbehave

CPU

› Contiguous core IDs, CPU ID mapping (Kudu), accurate threads: cores mapping (DB)

› NUMA aware assignment (HANA)

Memory:

› JVM sees entire host memory even if you cap the memory for container (Any JVM app)

› Memory allocation inconsistencies (hugepages, shared page cache) (Oracle)

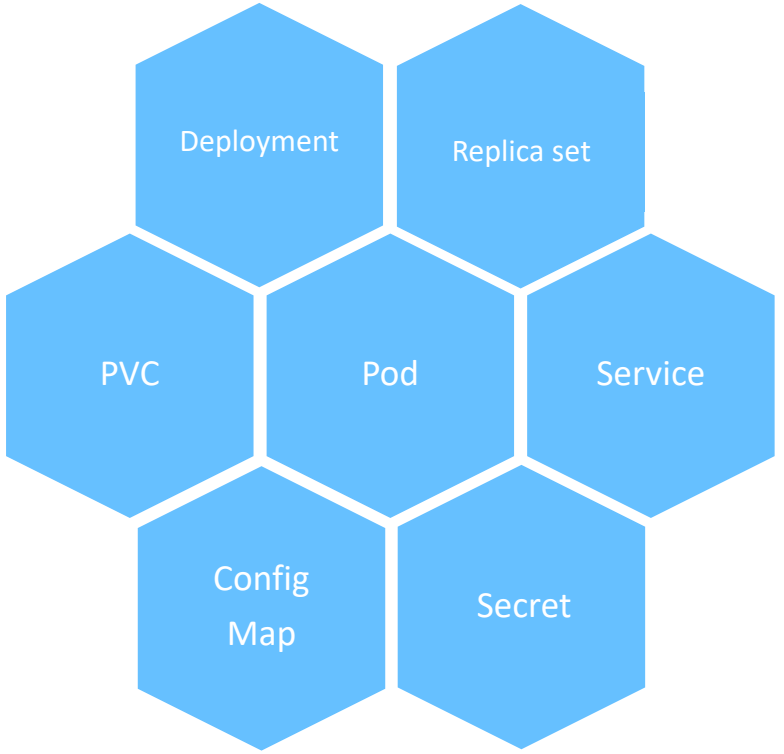
Storage

› Apps that need raw block devices need correct WWNs management (e.g., Oracle, MapR)

› `blkio` cgroups setting is useless to avoid noisy neighbor problems (All apps)

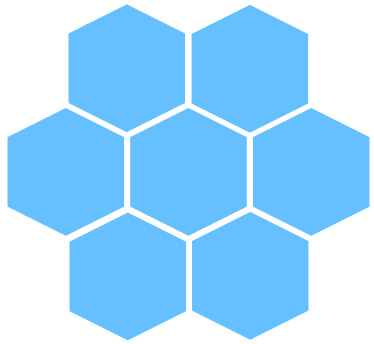
Kubernetes

Application Composition

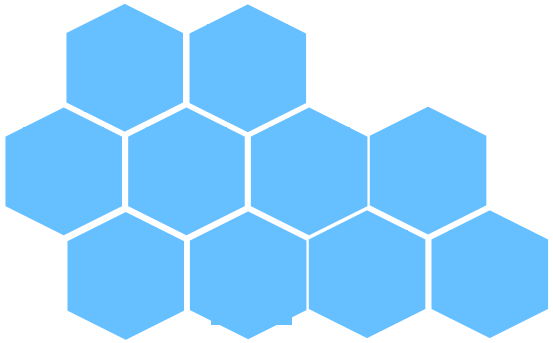


https://github.com/helm/charts/tree/master/stable/mysql

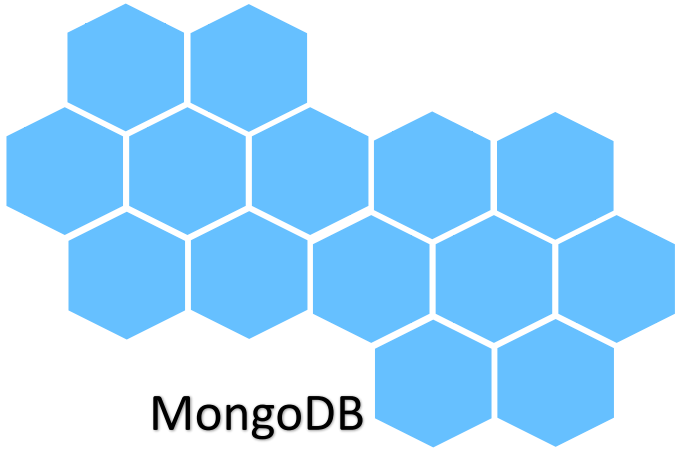
Application Composition .. The complexity



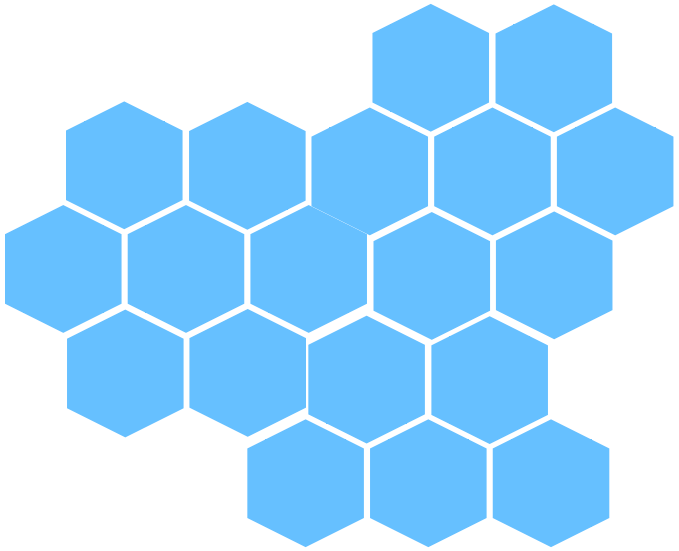
MySQL



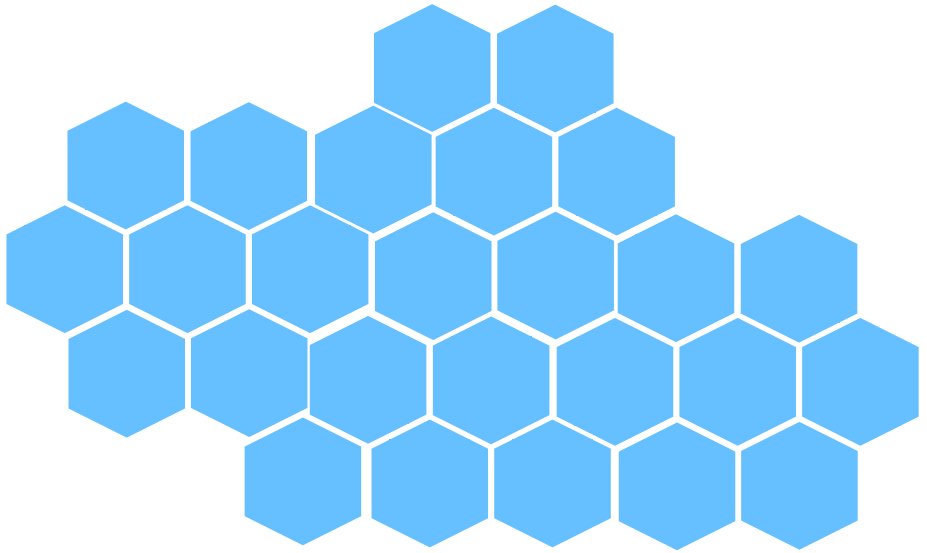
MariaDB



MongoDB



ElasticSearch



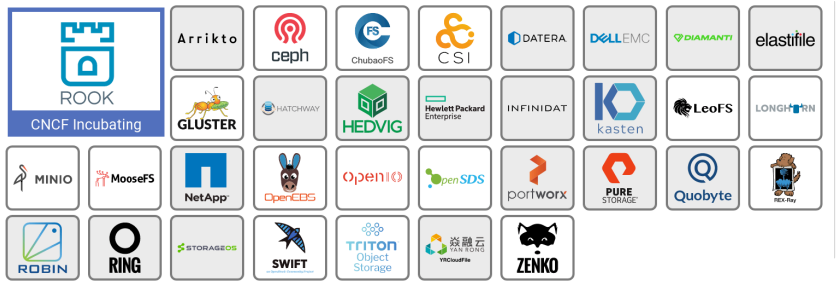
ELK Stack

https://github.com/helm/charts

Productize

State of Kubernetes eco-system

› There are **over 30 Storage vendors** and **20 Network vendors** providing Storage & Networking solutions for containers and Kubernetes[1]



Storage vendors

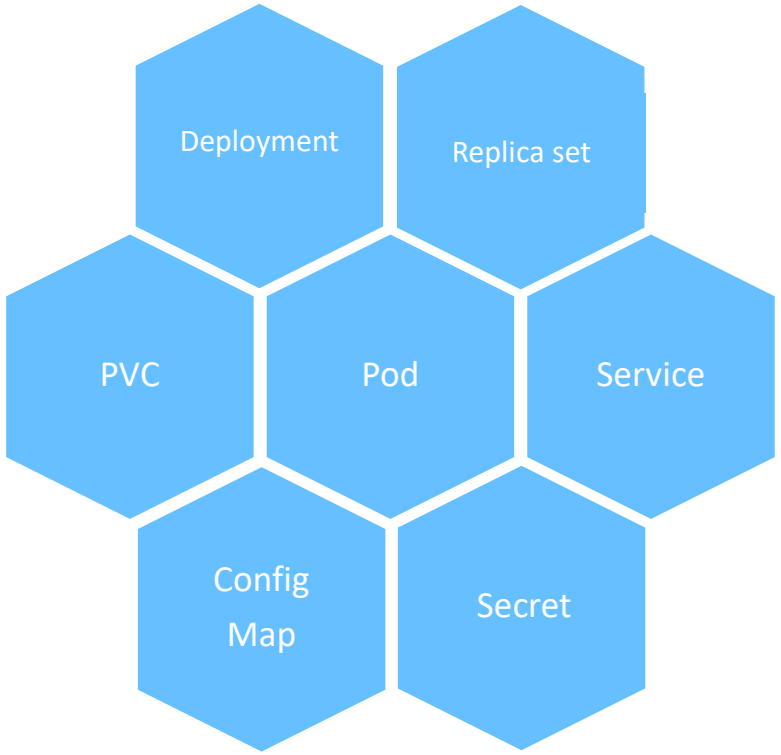
Network vendors

› CNCF[2]: 30% say Storage is a big challenge, 30% say Networking is a challenge in Kubernetes

Despite so many vendor solutions, why is it still a challenge for so many people?

1 https://github.com/cncf/landscape
2 https://www.cncf.io/blog/2018/08/29/cncf-survey-use-of-cloud-native-technologies-in-production-has-grown-over-200-percent/

MySQL Application Composition



Wait .....

Performance issues !!!!

https://github.com/helm/charts/tree/master/stable/mysql

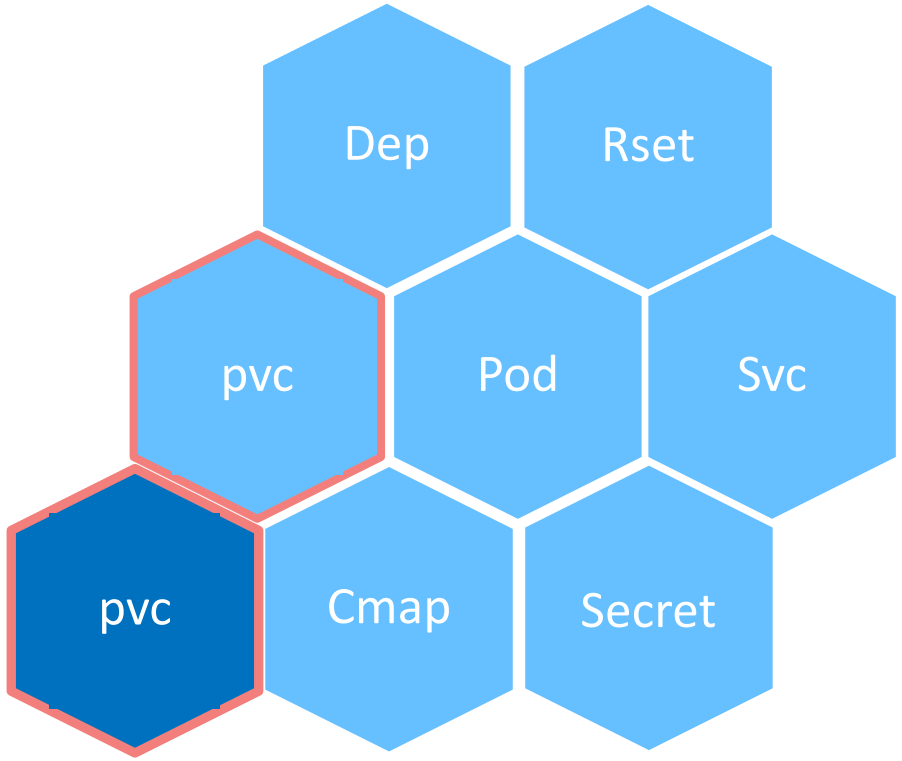
What about ….. InnoDB storage recommendation

- Non-rotational storage generally provides better performance for random I/O operations; and rotational storage for sequential I/O operations. When distributing data and log files across rotational and non-rotational storage devices, consider the type of I/O operations that are predominantly performed on each file.

- Random I/O-oriented files typically include file-per-table and general tablespace data files, undo tablespace files, and temporary tablespace files. Sequential I/O-oriented files include InnoDB system tablespace files (due to doublewrite buffering and change buffering) and log files such as binary logfiles and redo log files.

Let us be a good MySQL Admin and follow recommendations

MySQL Application Composition



Wait .....

Performance issues !!!!

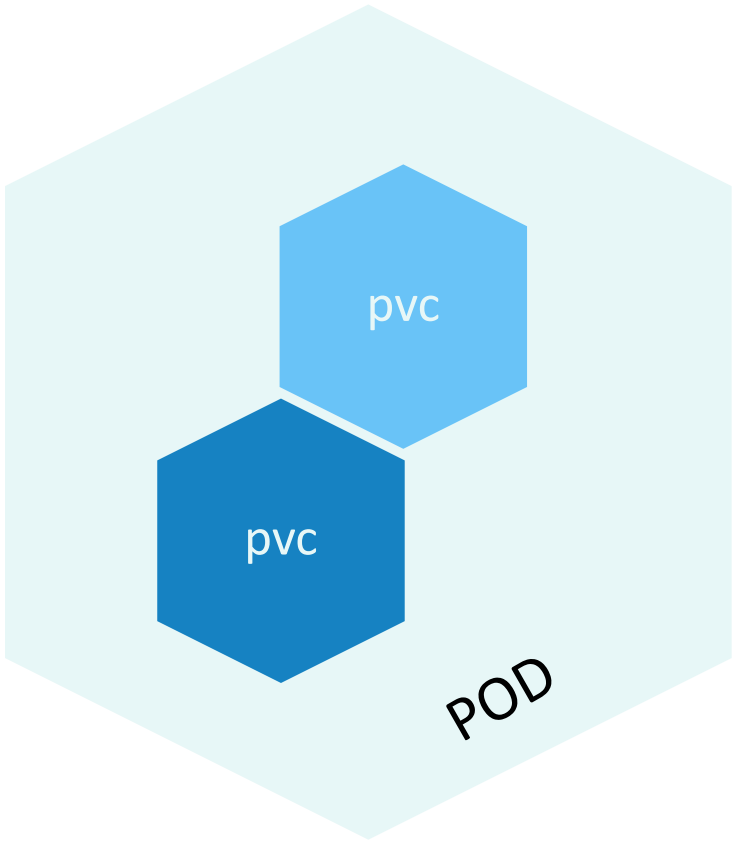
https://github.com/helm/charts/tree/master/stable/mysql

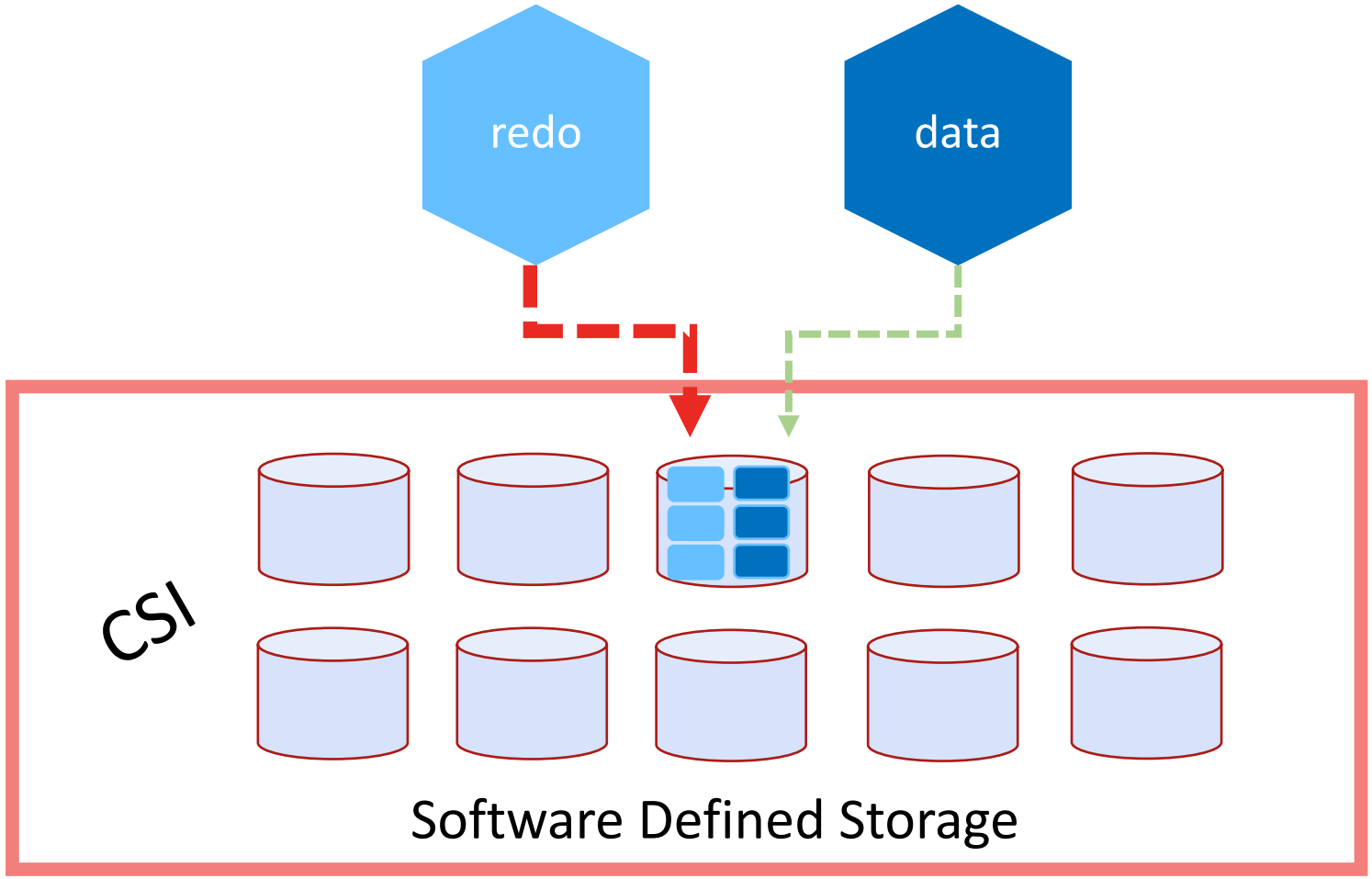
robin.io

Any guesses ??

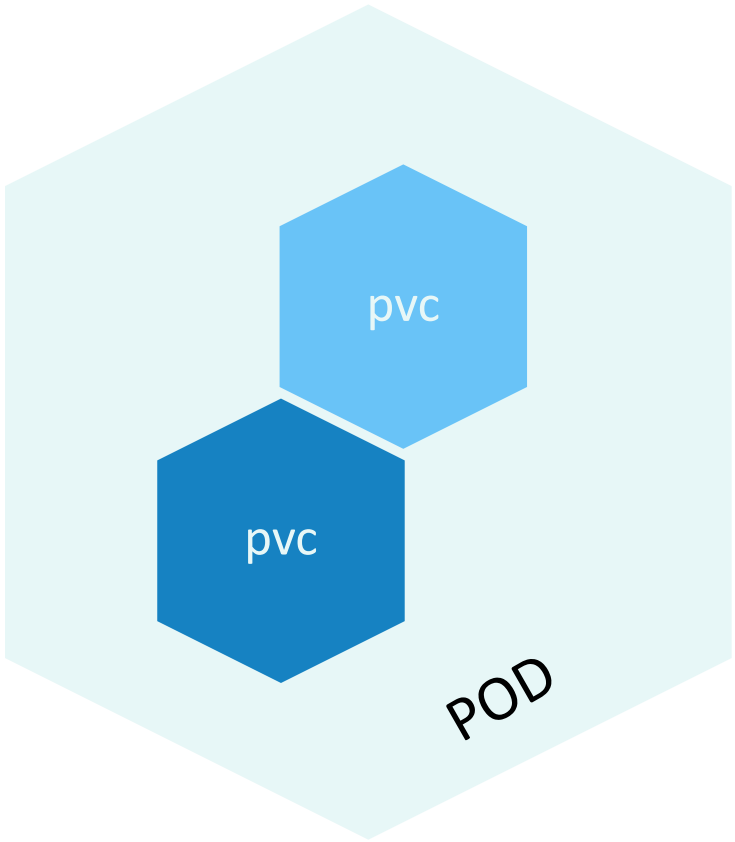
Let us look at it more closely

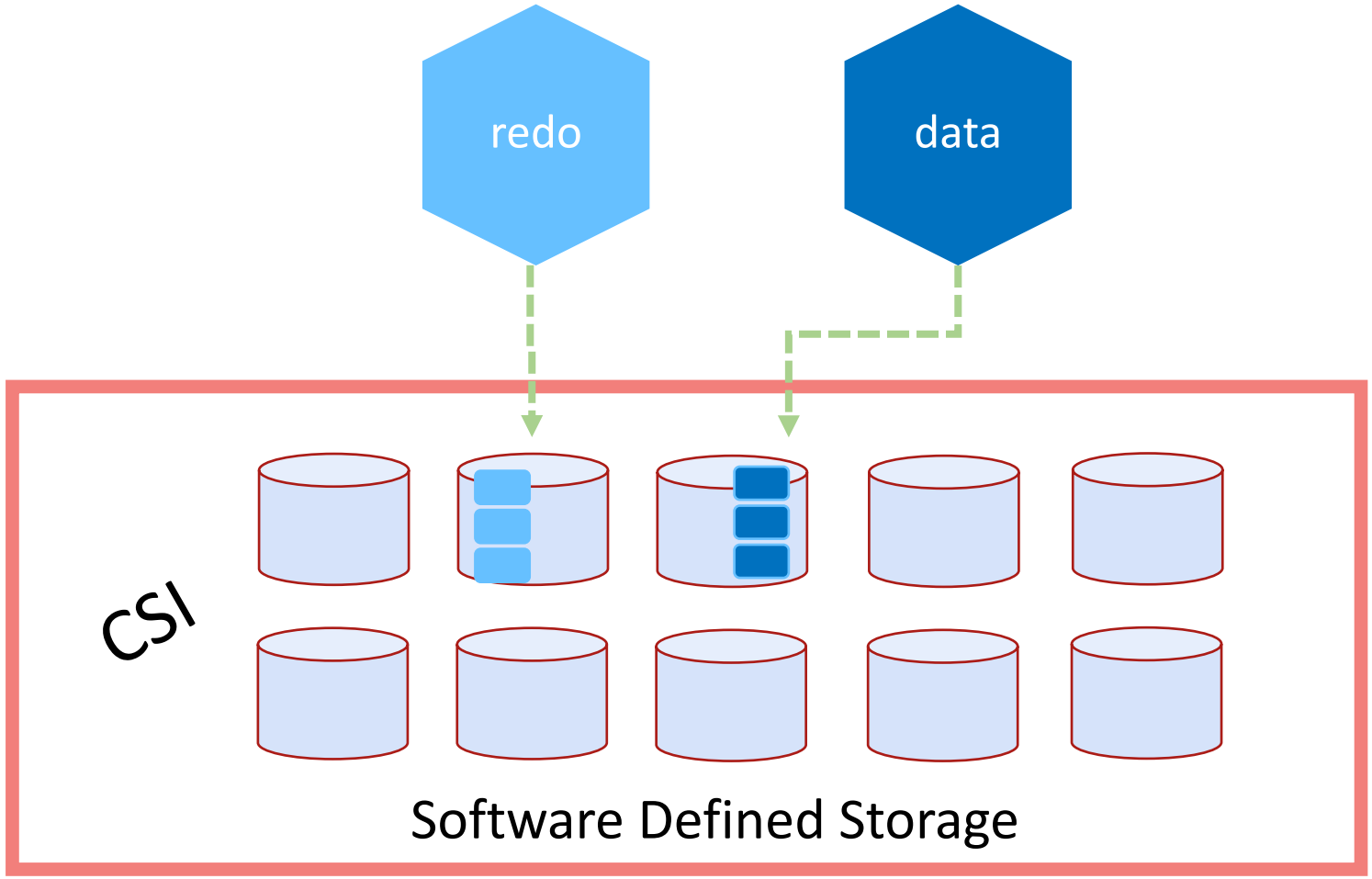
Troubleshooting .....





Let us fix it ..

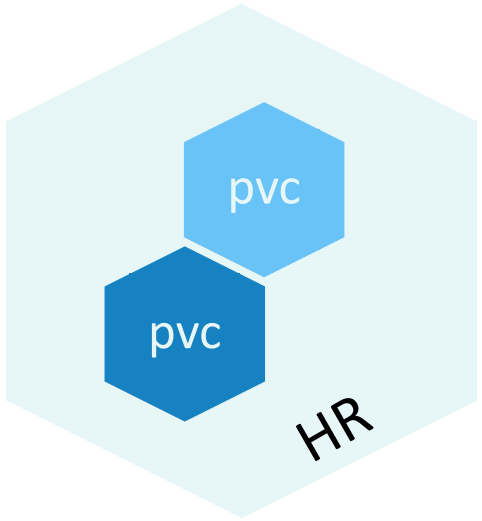


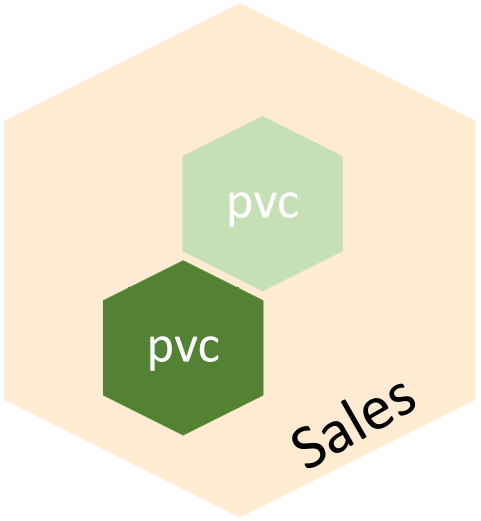


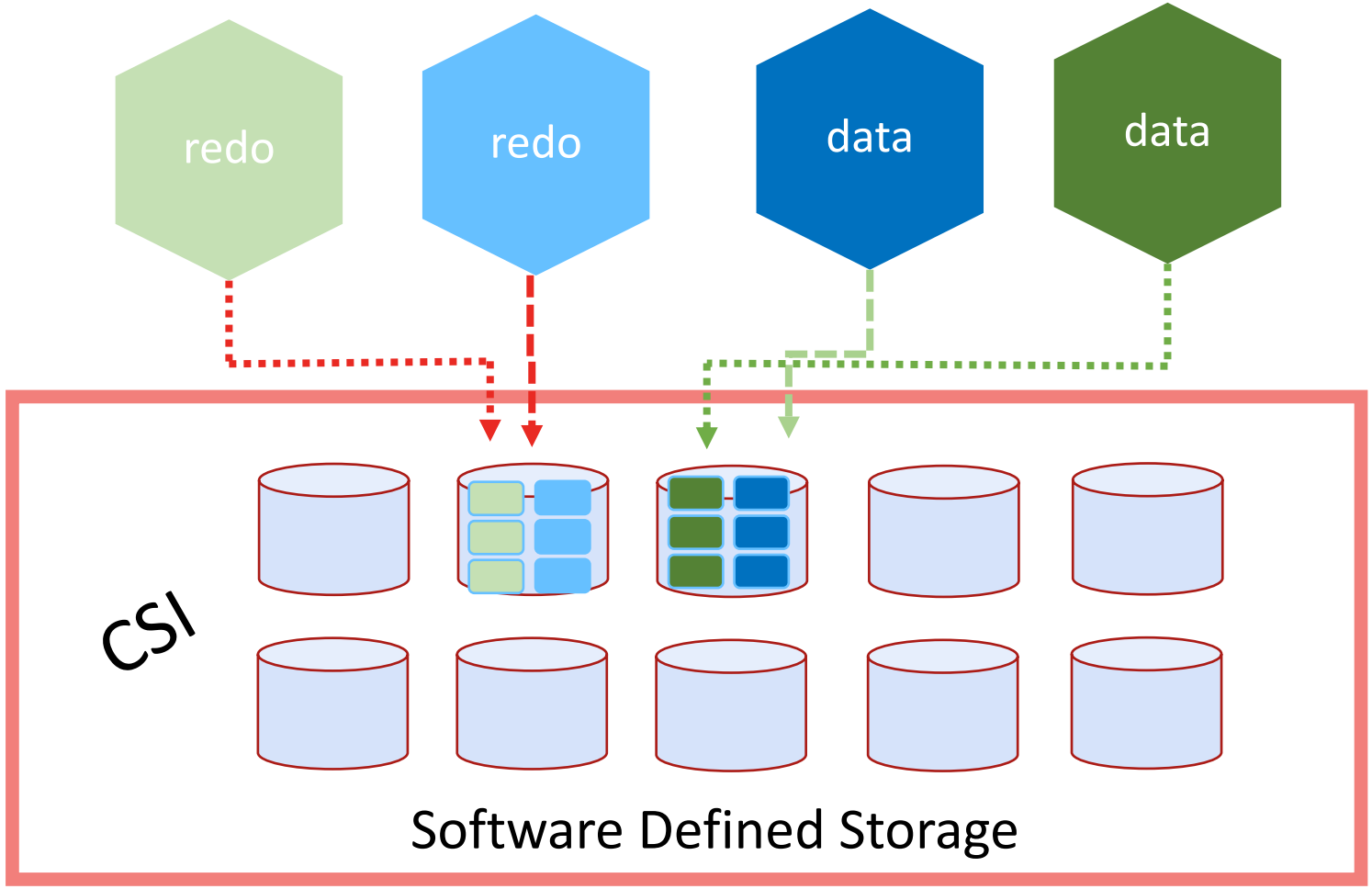
Are we good ??

Not yet.. Not completely

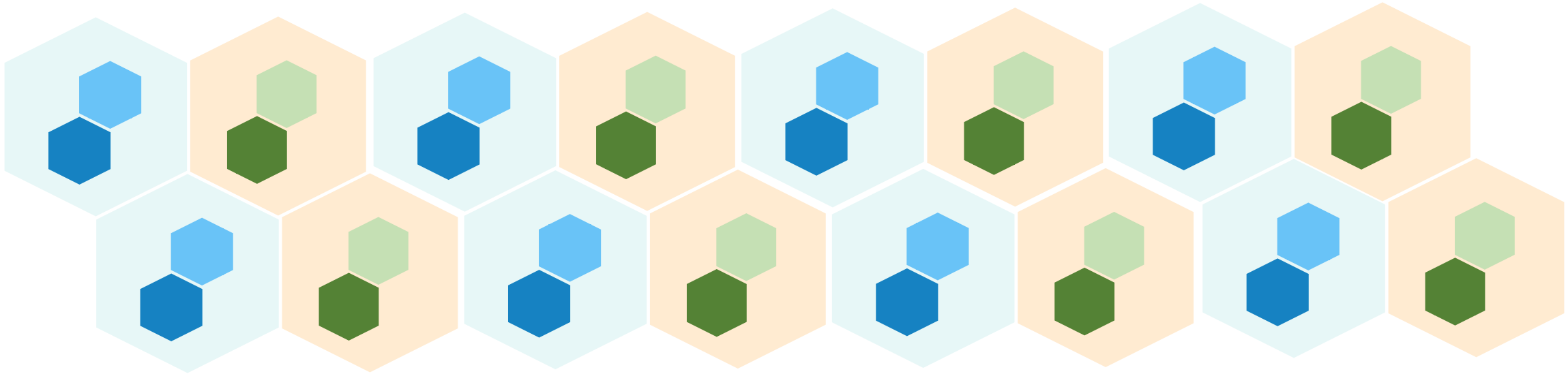
Perils of consolidation ..







Perils of consolidation ..



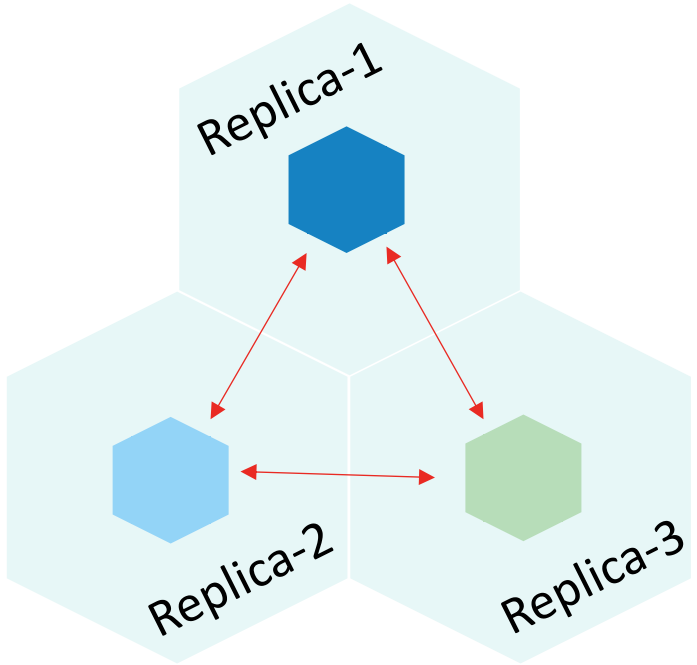
Imagine managing 100s of PODs and
1000s of PVCs allocated from 100s of Drives

Are we good ??

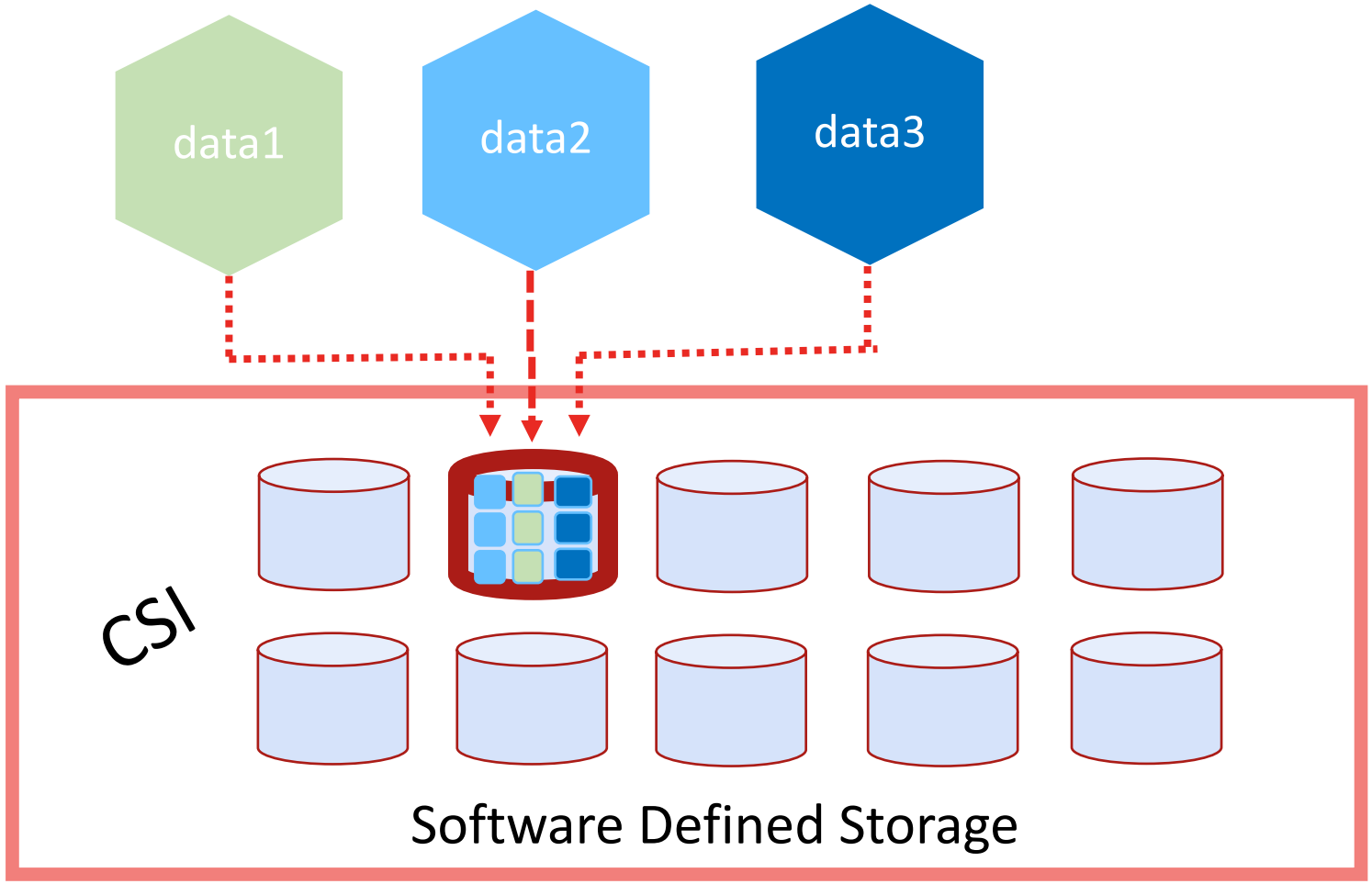
MySQL Admin is Happy

Cassandra Admin is NOT

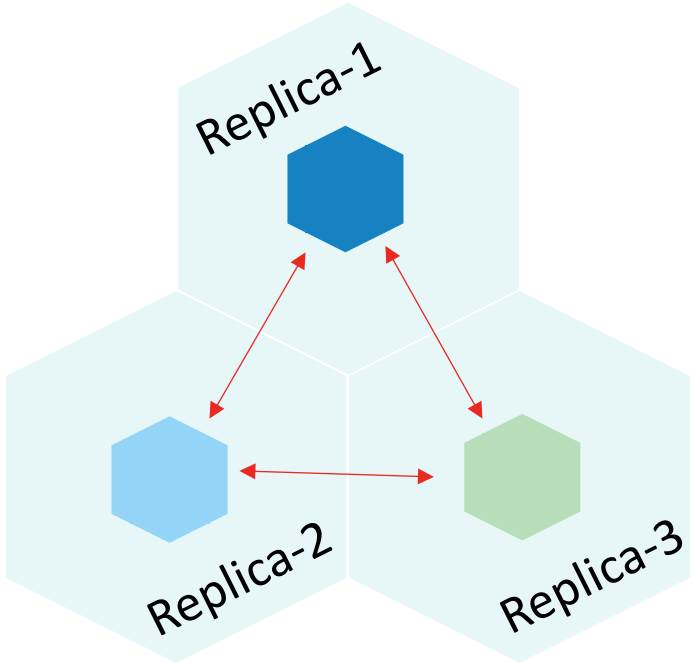
What's with Cassandra ??

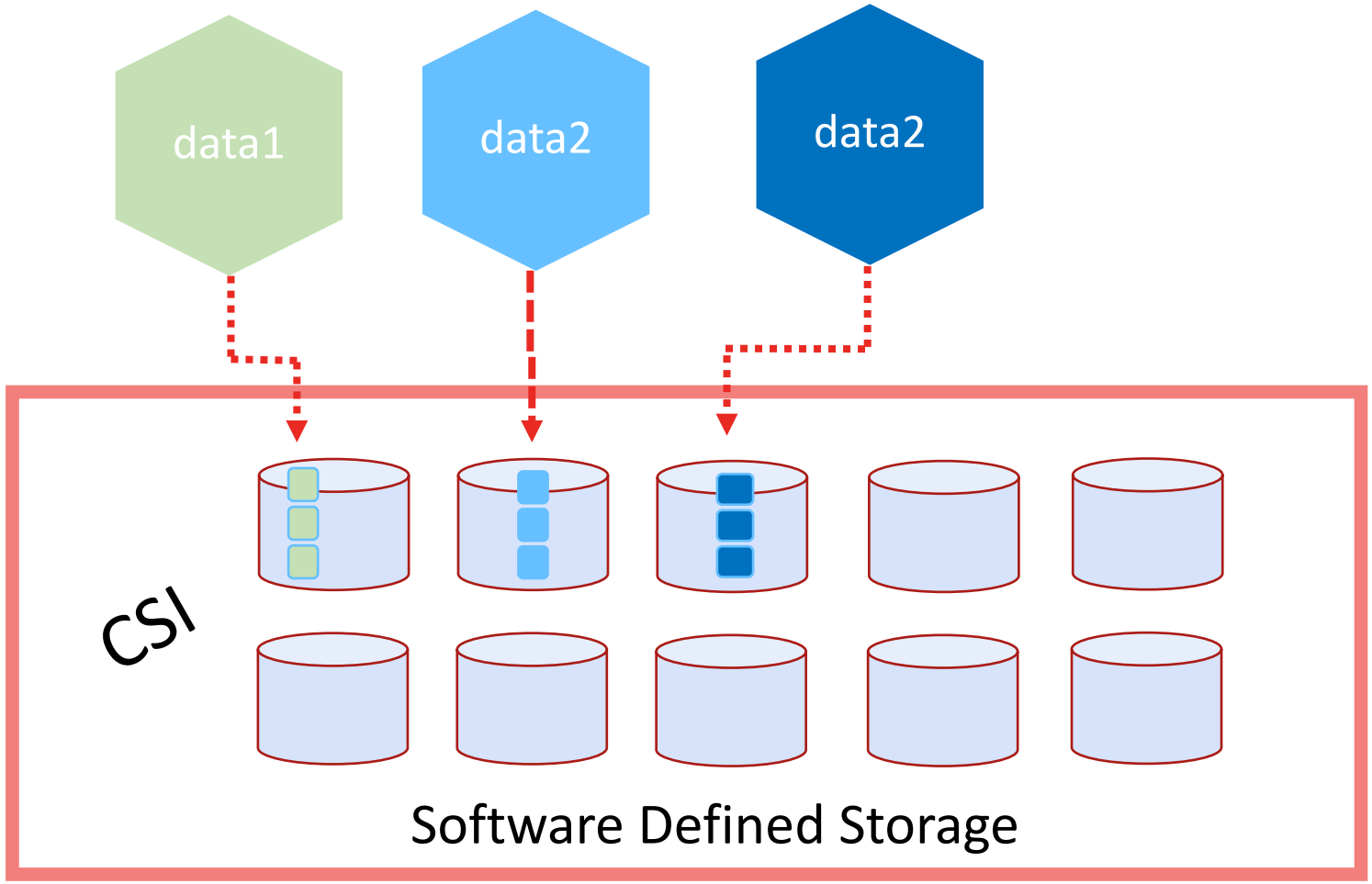


Still resilient to disk failure ???



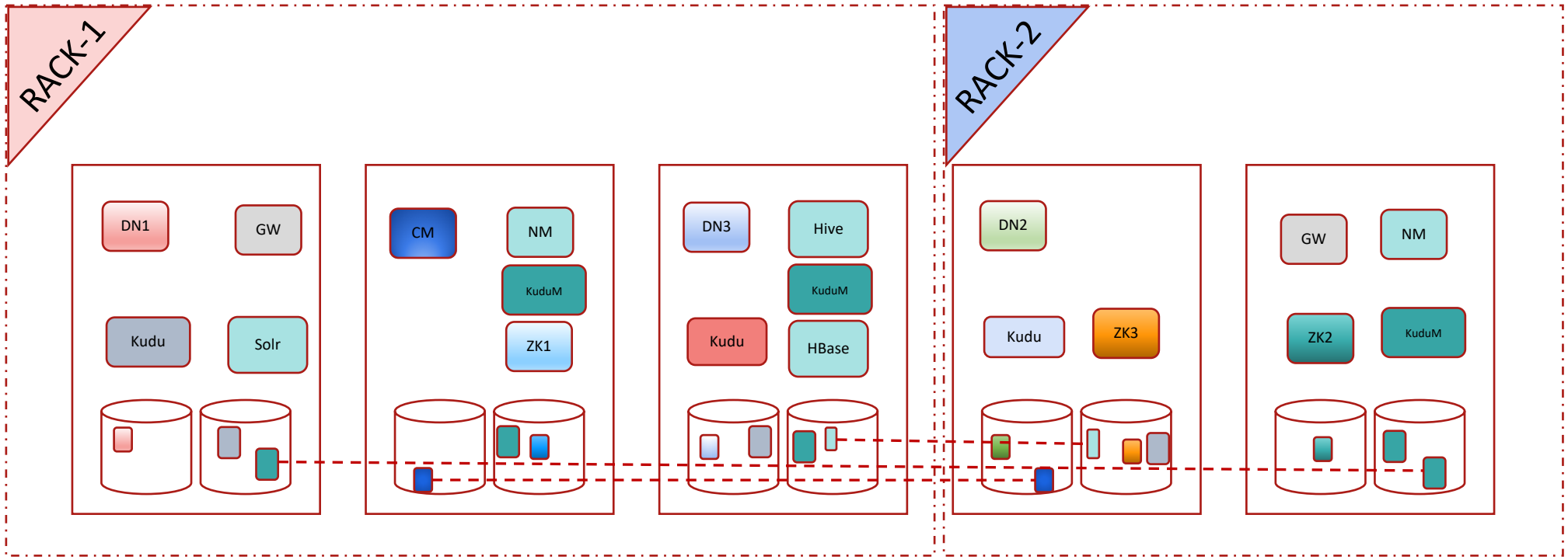
Let us fix it …


Replica-1
Replica-2
Replica-3


data1
data2
data2
CSI
Software Defined Storage

What about Hadoop (HBase, Kudu) ?

Hadoop components



Compute anti-affinity

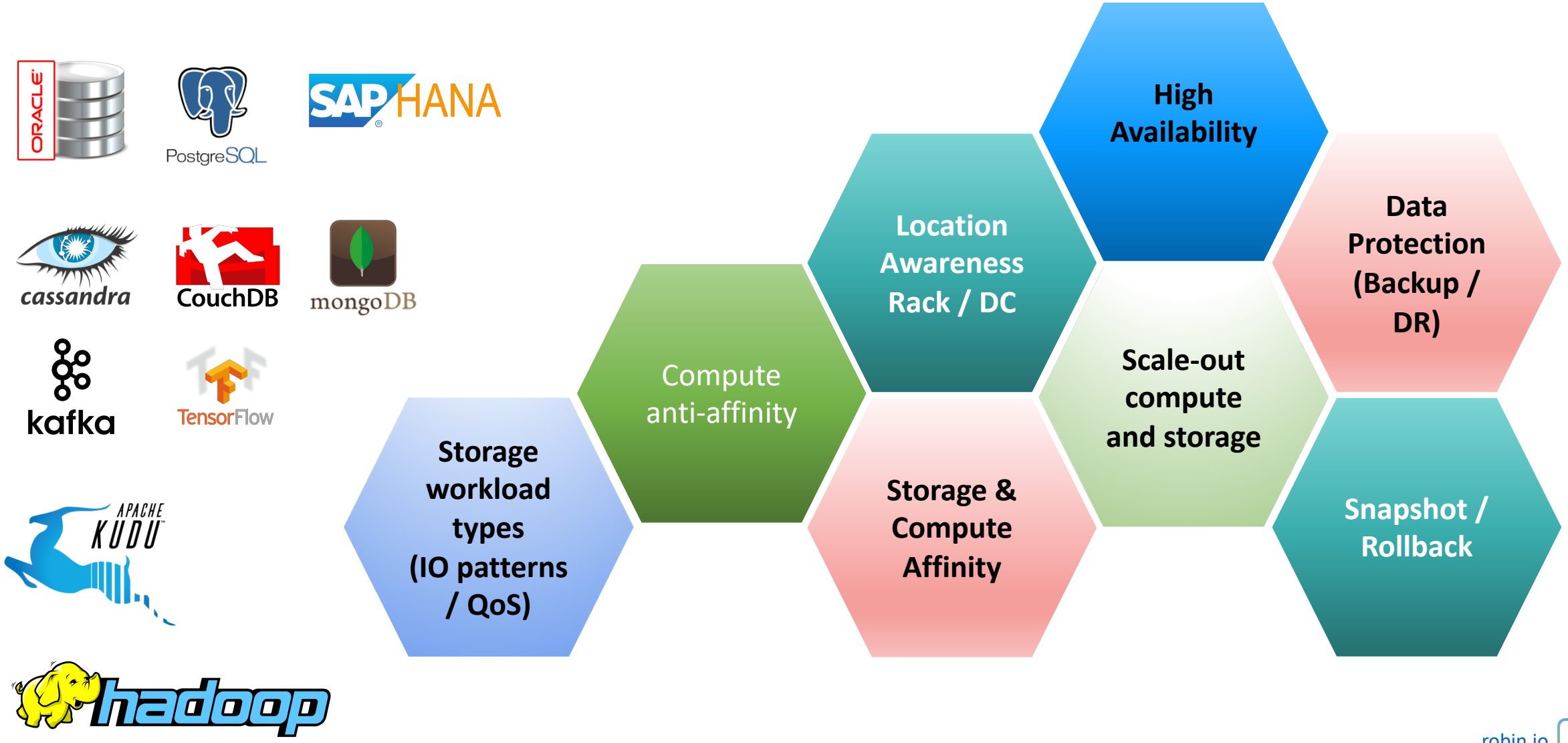
Location Awareness Rack / DC

Storage & Compute Affinity

IO patterns QoS

High Availability

Big data deployment and management challenges



Recap

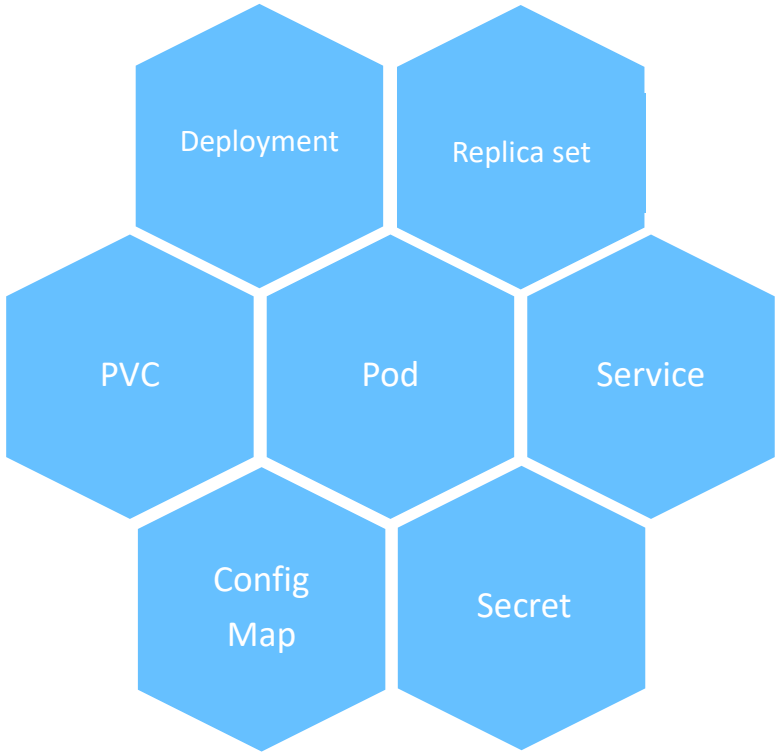
- Data-heavy applications deal with Multiple volumes
- Every volume will have different IO characteristics
- Consolidation (packing) makes the problem even harder
- Application Replication (Cassandra / Mongo) makes the allocation tricky

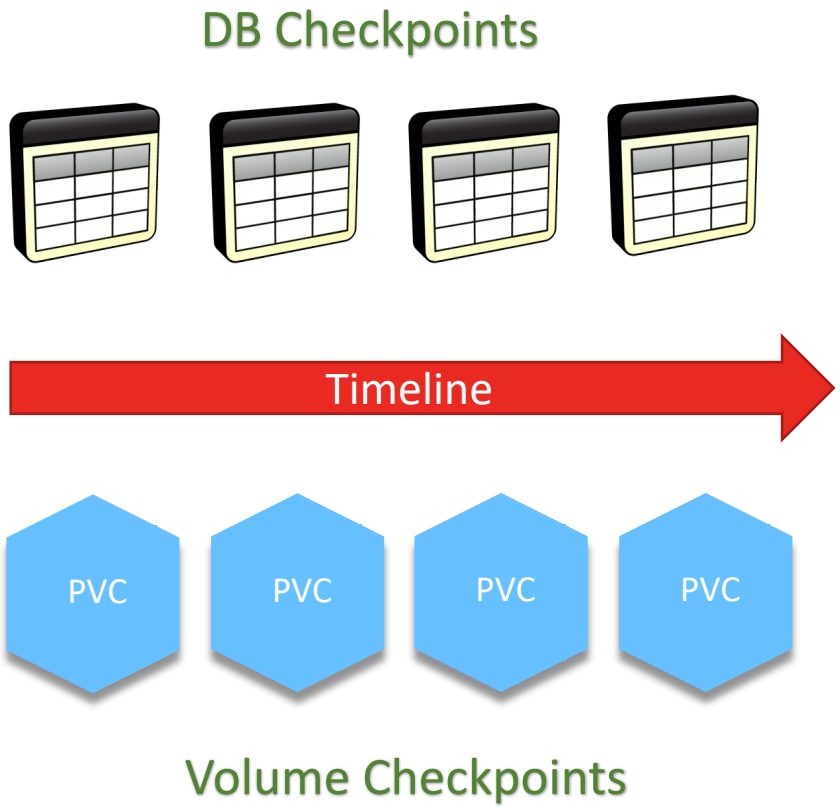
What are we looking for…..???

Application Aware Storage Provisioning

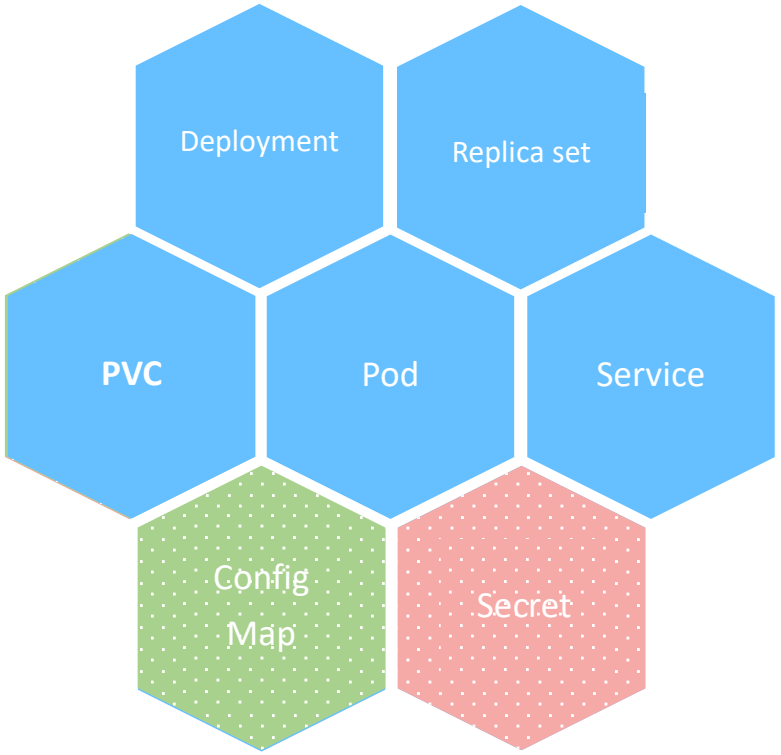
Data Protection and Cloud Portability

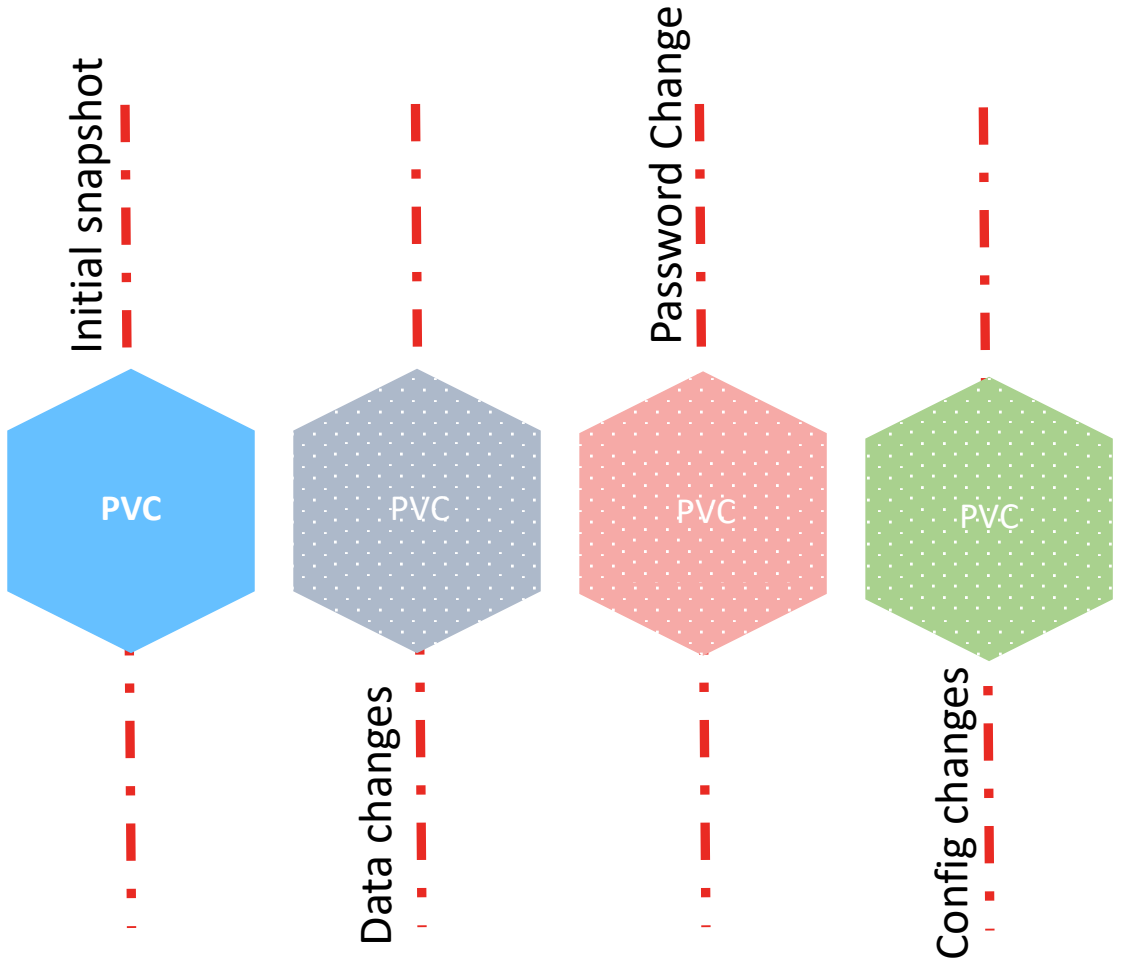
Let us talk Data Protection





Volume snapshots

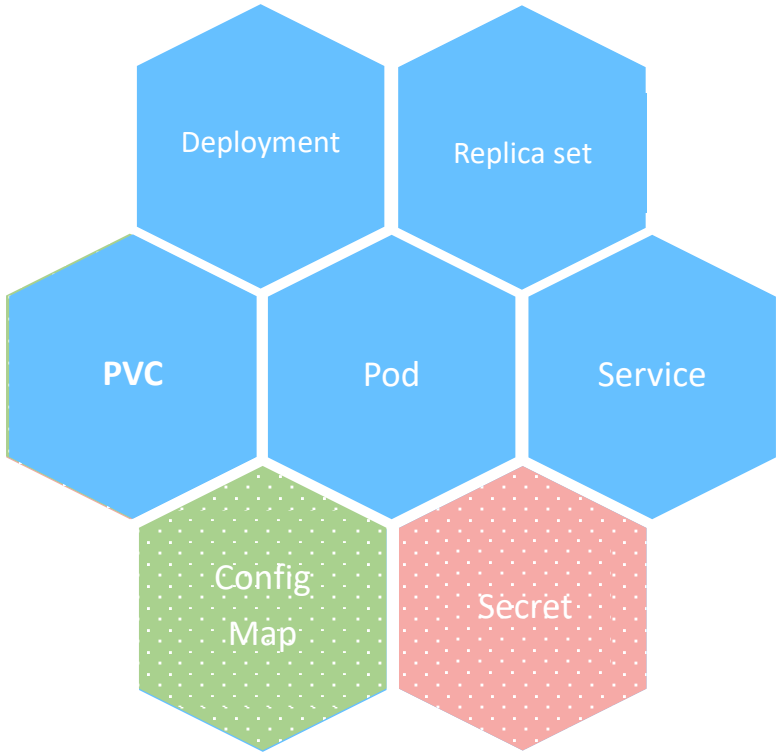




Rollback to this snapshot

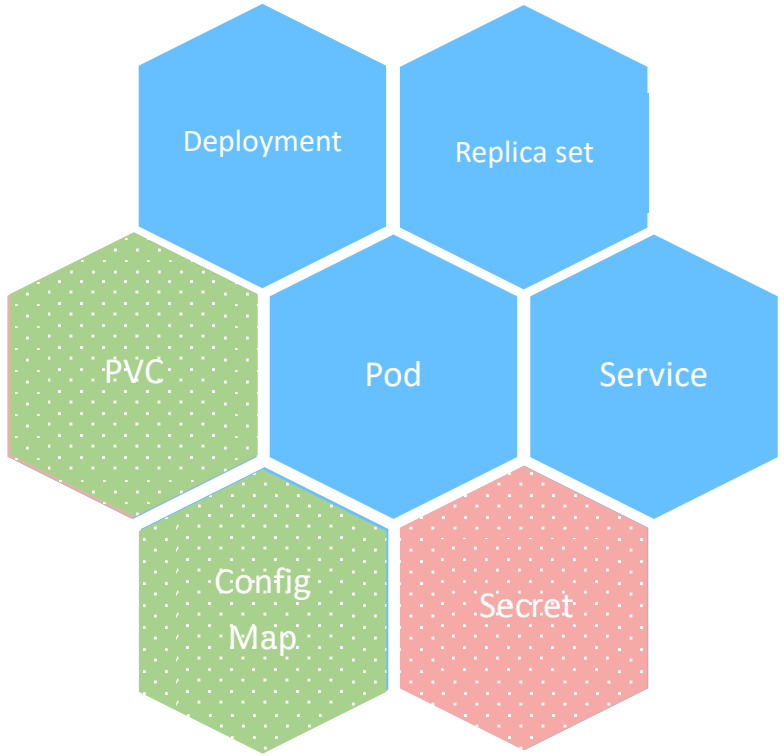
What is the problem here ?

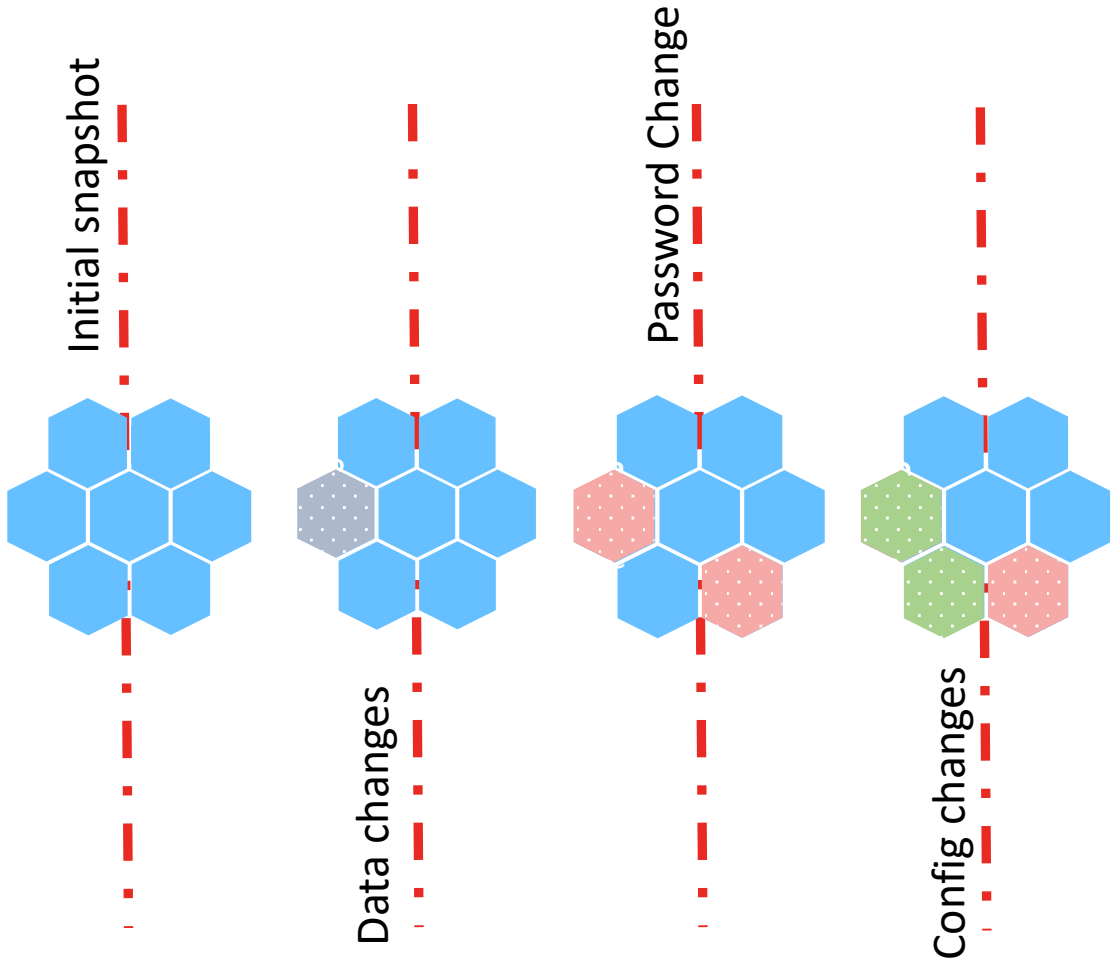
Volume snapshots



Config Drift !!!

Let us fix it …


Deployment
Replica set
PVC
Pod
Service
Config Map
Secret


Initial snapshot
Data changes
Password Change
Config changes

Recap (Data Protection)

- Snapshots and backups are not just data dumps
- Not all application have checkpoints and snapshots
- Data snapshots are prone to config drift issues
- Consistency group is a very critical construct
- Application buffers / FS page cache will need to be flushed to disk

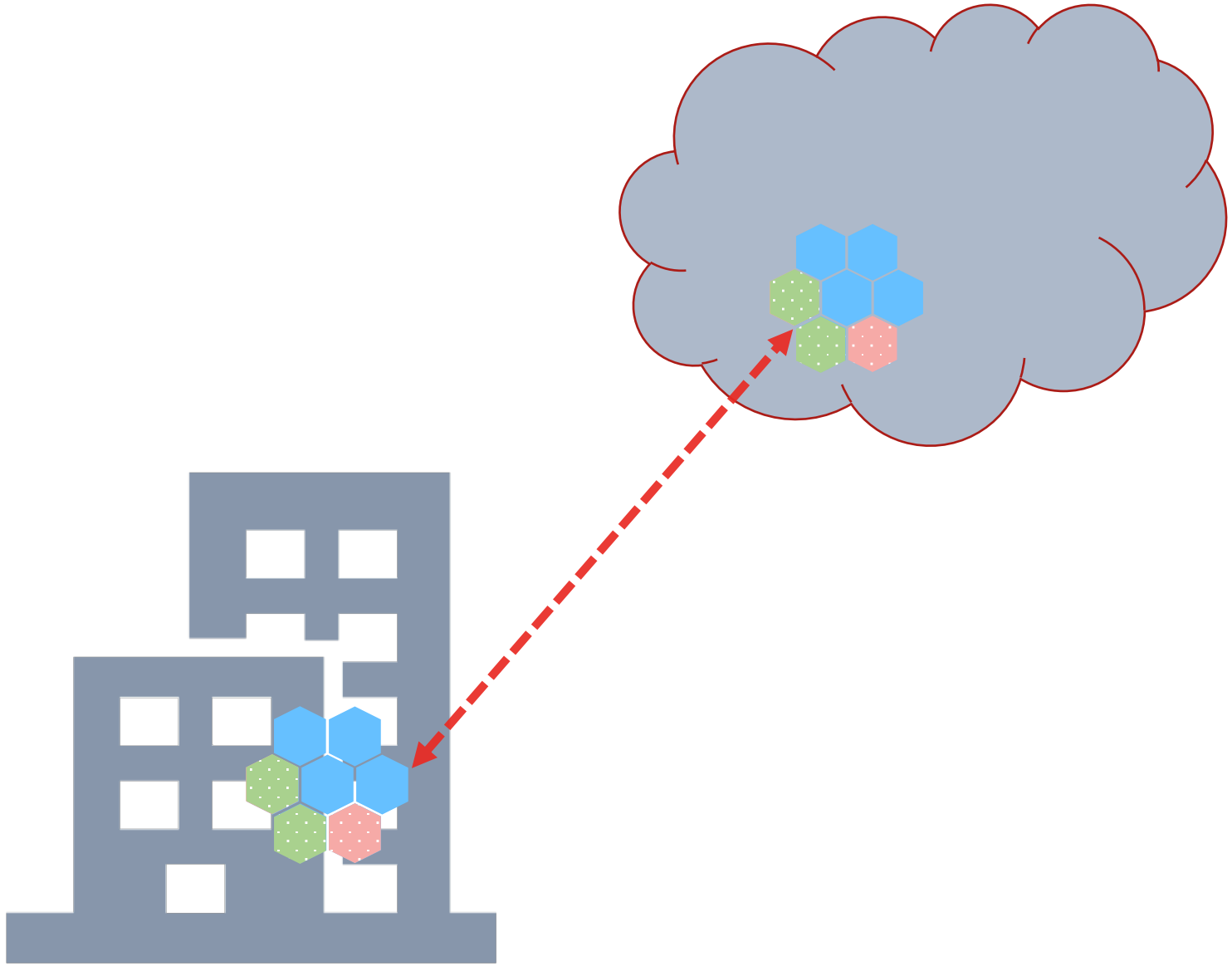
What are we looking for…..???

Application Snapshots

Any other challenges

Let us talk about portability

- Why do we need this?
  - Hardware refresh
  - Datacenter migration
  - Vendor lock-in
  - Performance
  - Test / Dev setups
  - Upgrade firedrills



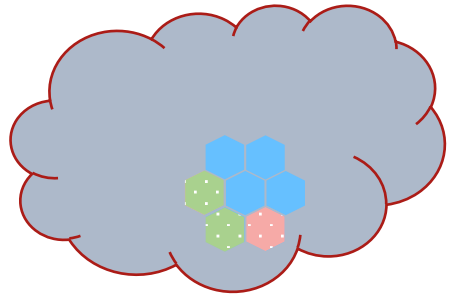
Application backups

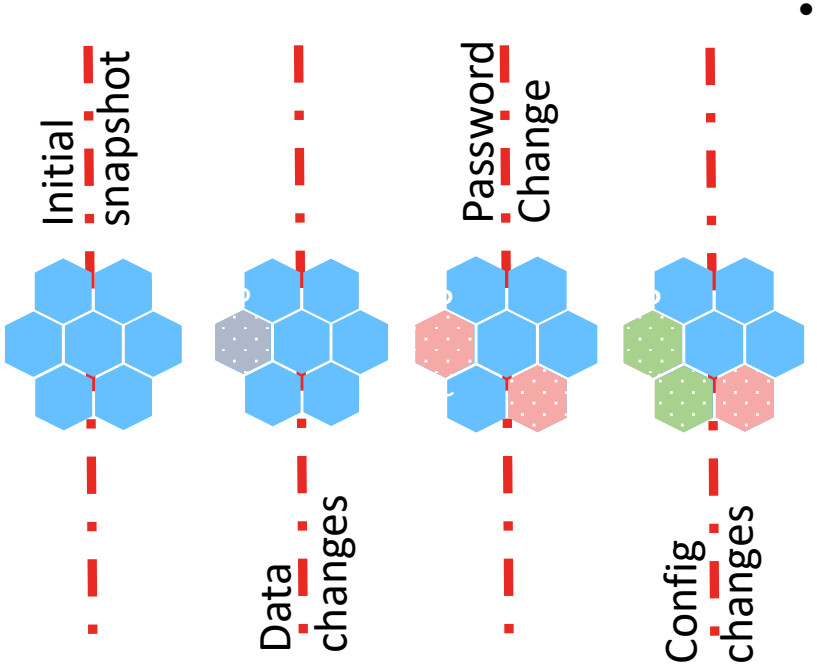
Time:

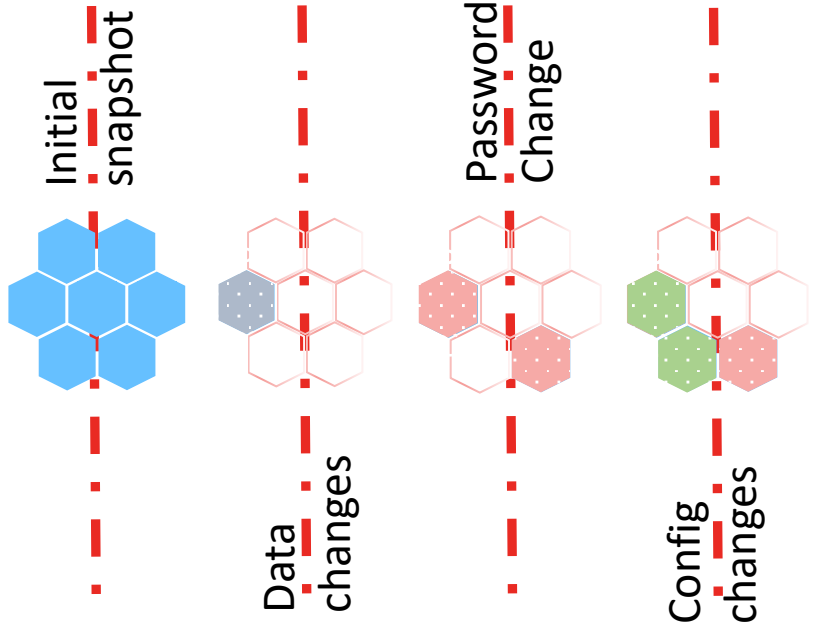
- Avoid full rehydration to Block
- Rehydrate on demand

Cost:

- Use Object store (Cheap)
- Send differentials

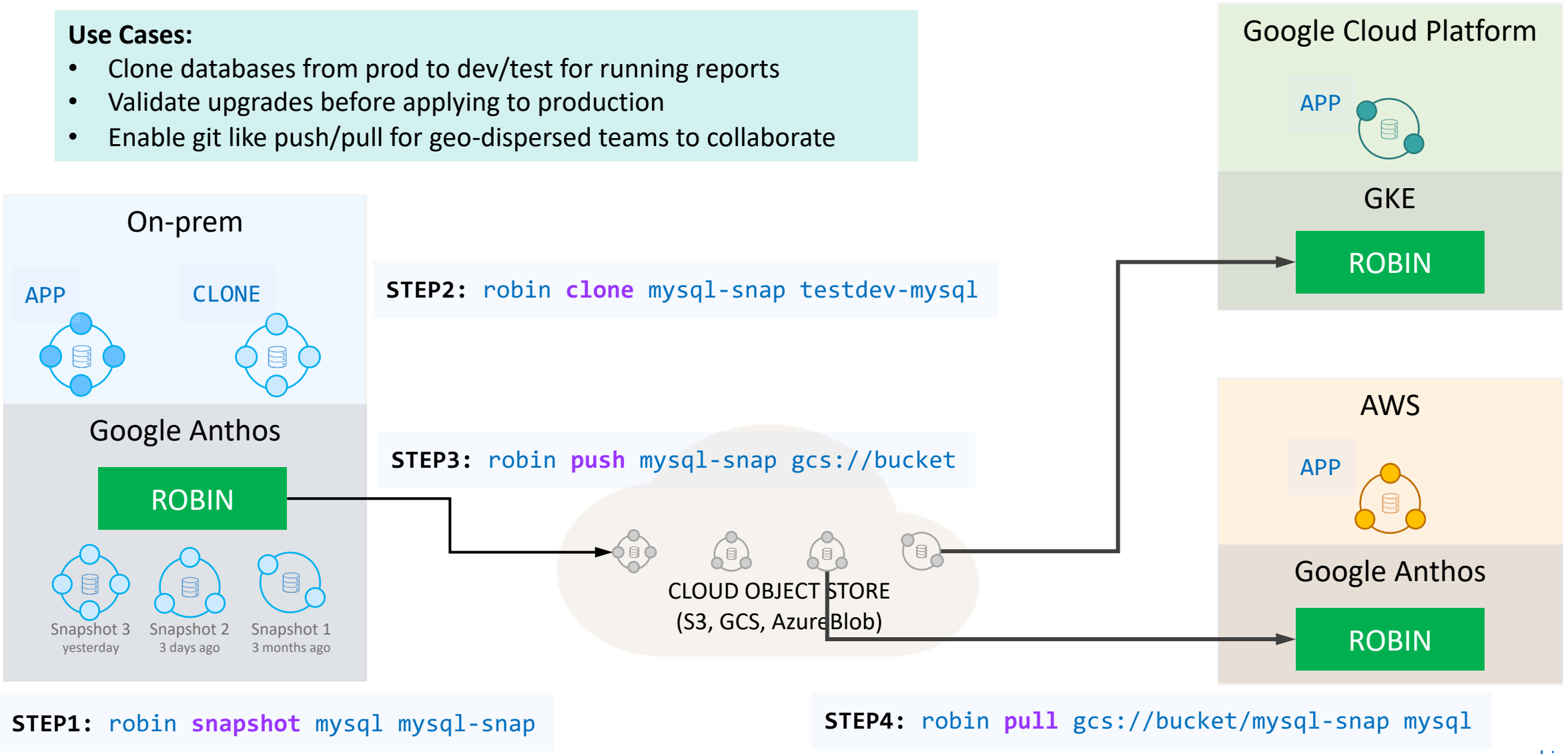






Collaborate on Applications using a Git-like workflow

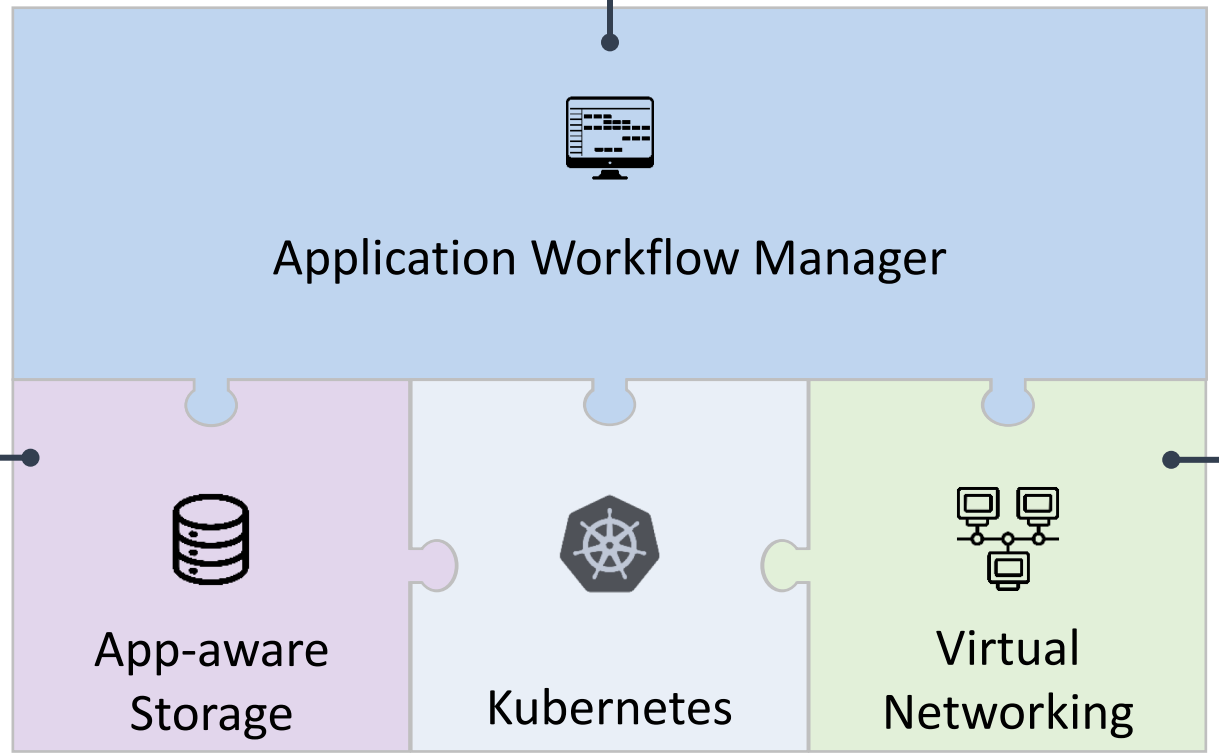
Use Cases:
- Clone databases from prod to dev/test for running reports
- Validate upgrades before applying to production
- Enable git like push/pull for geo-dispersed teams to collaborate



Robin Architecture Overview

1-click application Deploy, Snapshot, Clone, Scale, Upgrade, Backup

Application workflows configure Kubernetes, Storage & Networking



Robin's built-in enterprise-grade storage stack

Snapshots, Clones, QoS, Replication, Backup, Data rebalancing, Tiering, Thin-provisioning, Encryption, Compression

Built-in flexible networking

OVS, Calico, VLAN, Overlay networking, Persistent IPs

vm aws Azure Google Cloud Platform

Works any where

ROBIN software allows you run complex Big Data and Databases on Kubernetes

(Storage + Networking + Application Workflow Management + Kubernetes)

DEPLOYMENT PROOF POINTS

11 billion security events ingested and analyzed a day

(Elasticsearch, Logstash, Kibana, Kafka)

6 petabytes under active management in a single ROBIN cluster

(Cloudera, Impala, Kafka, Druid)

400 Oracle RAC databases managed by a single ROBIN cluster

(Oracle, Oracle RAC)

Product Demos

Demo Series 1

ROBIN's Kubernetes-native Storage and Data Management stack to deploy and manage day-2 operations of Stateful Apps

Snapshot and Rollback

Clone

Backup

Cloud-portability

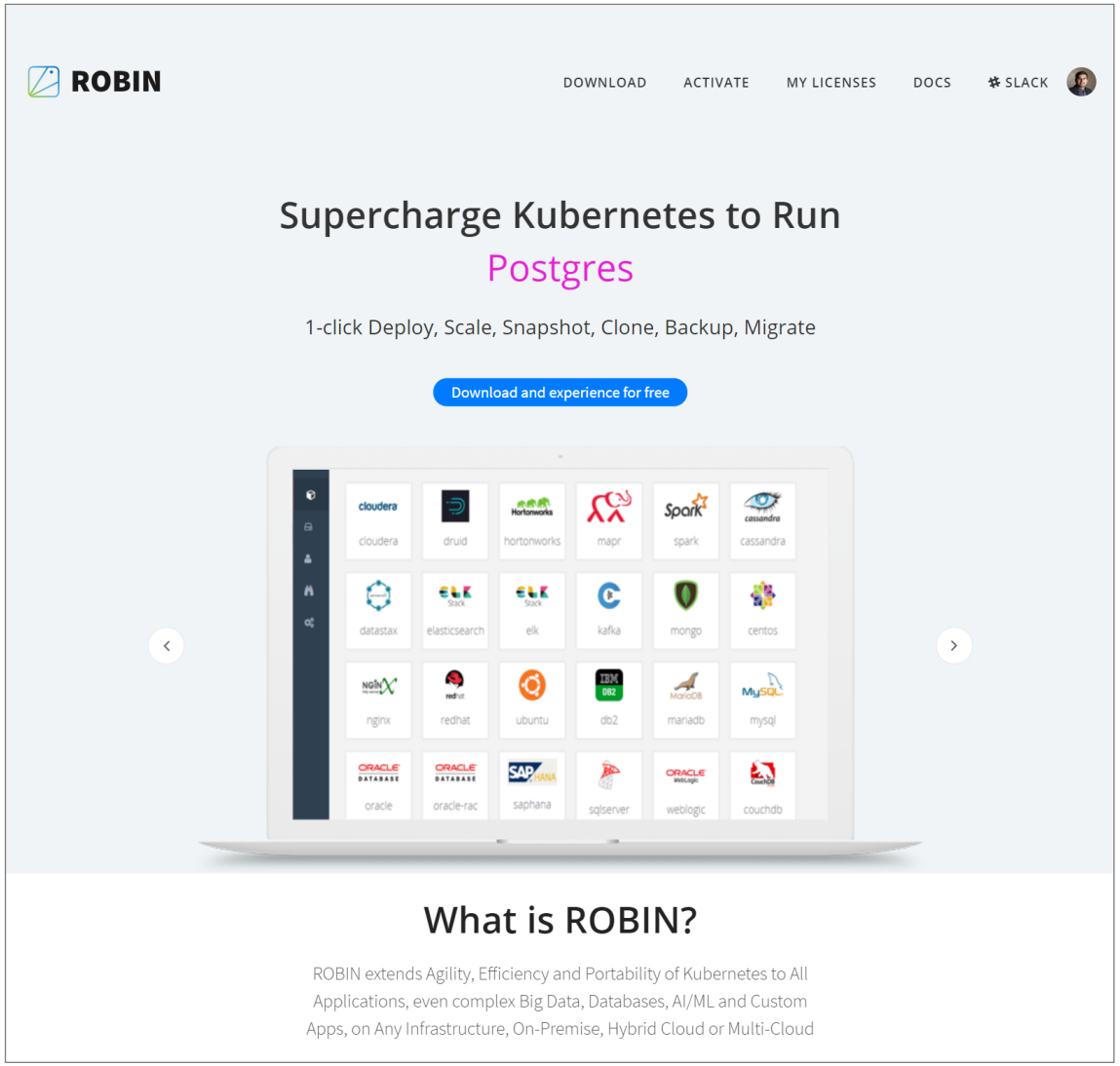
Demo Series 2

ROBIN's Super-Operator for Enterprise Apps

1-click Deploy, Cloud-Sync and Cloud-Motion

get.robin.io Portal

- get.robin.io: Download Robin Storage for free
- docs.robin.io: Read product documentation
- slack.robin.io: Interact with Robin engineering



ROBIN.IO

Supercharge Kubernetes to Deliver Big Data and Databases as-a-Service

1-click Deploy, Scale, Snapshot, Clone, Upgrade, Backup, Migrate