Supercharge Kubernetes to run Big Data and Databases

Challenges to overcome and Solutions

Partha Seetala

Chief Technology Officer, ROBIN.IO

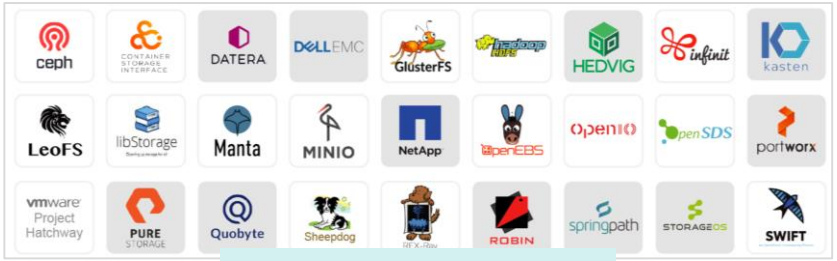
Kubernetes is a great platform to run STATELESS workloads

Imagine if you could run STATEFUL workloads on it

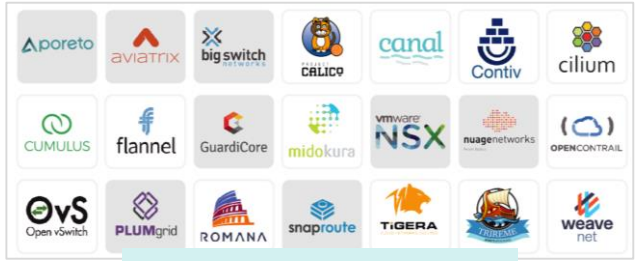
Not just any STATEFUL workloads, but ones where money & effort is being spent to achieve deployment agility and management simplicity

Isn't this a solved problem?

› There are 27 Storage vendors and 21 Network vendors providing Storage & Networking solutions for containers and Kubernetes[1]



Storage vendors



Network vendors

› CNCF[2]: 48% say Storage is a big challenge, 44% 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/2017/06/28/survey-shows-kubernetes-leading-orchestration-platform

Who am I?

Chief Technology Officer at ROBIN.IO

before that Distinguished Engineer at Veritas/Symantec

partha@robin.io

https://www.linkedin.com/in/parthaseetala

@parthaseetala

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)

CSI (Storage) challenges to overcome

Providing persistent storage to pods is a solved problem

Any storage vendor who tells you otherwise is lying ☺

4 Storage challenges to overcome:

1. Data management that is app-consistent
   Snapshots, Clones, Backup, Restore, Migrate (with app-consistency)

2. Data placement that honors fault-domain constraints of an app
   Data locality, affinity & anti-affinity for both pods and data

3. Handling apps that modify Root filesystem?
   Configuration changes to /etc, /var and / directories

4. Guarantee performance SLAs when running on a consolidated platform
   Noisy neighbor challenges when running transactional workloads

CNI (Networking) challenges to overcome

Providing connectivity, routing and networking security to pods is a largely solved problem

20+ CNI providers, 14+ Security providers

3 Networking challenges to overcome:

1. Handle IP address persistency on pod restarts/relocations
   Ephemeral IP addresses (even with StatefulSets) is a challenge when apps embed IP addresses into their config/business logic

2. Multiple NICs to a pod
   Needed to separate North/South client traffic from East/West management traffic

3. Connectivity to pods network from clients in a different L3 subnet
   Breaks apps that embed POD IP addresses in their Web UI

CRI (Runtime) challenges to overcome

Docker containers are everywhere. Everyone understands how to build one.

4 Docker challenges to overcome:

1. Docker Shim (K8S default CRI) doesn't adequately virtualize cgroups
2. JVM sees entire host memory even if you cap the memory for container
   Results in Out of Memory container kills when operating at high memory consumption
3. blkio cgroups setting is useless to avoid noisy neighbor problems
   Breaks apps that embed POD IP addresses in their Web UI
4. Raw block devices access is incorrectly done
   WWN management of devices is critical to avoid breaking correctness of a database

Kubernetes challenges to overcome

Don't StatefulSets and PersistentVolume/Claims address all challenges

3 Microservices challenges to overcome:

1. Most heavily used Big Data and Database platforms predate both Docker and Kubernetes
2. There are decades of built-in assumptions that don't easily fit into the microservices philosophy of K8S
3. Rewriting them to fit this model is not realistic

But aren't Operators solving this problem for us?

What about Operators?

- Operators is custom logic for the provisioning and scaling complex workloads on Kubernetes.
  CSI, CNI and CRI challenges called out earlier still need to be addressed outside an Operator
- Big Data and Databases rely on lifecycle management which depends on external storage providers
  Snapshots, Clones, Backups, Restores
- How to handle multi-tier apps that span multiple Operators?
  e.g., how to scale, snapshot, clone and backup a 3-tier app, when an Operator understands only one tier?

This is getting too complicated!!

STORAGE (CSI)

SERVER INFRA (Baremetal, On-prem VM, AWS, Azure, GCP)

DATA (PV, PVC)

NETWORKING (CNI, Overlay)

CONFIGURATION (ConfigMap, Secrets)

CONTAINERS (docker, LXC)

MANAGEMENT (kubectl, helm)

SERVICES (Ingress, Proxy, LB)

UI

MONITORING (Heapster, HPA)

TROUBLESHOOTING (Logging, Events)

Desired

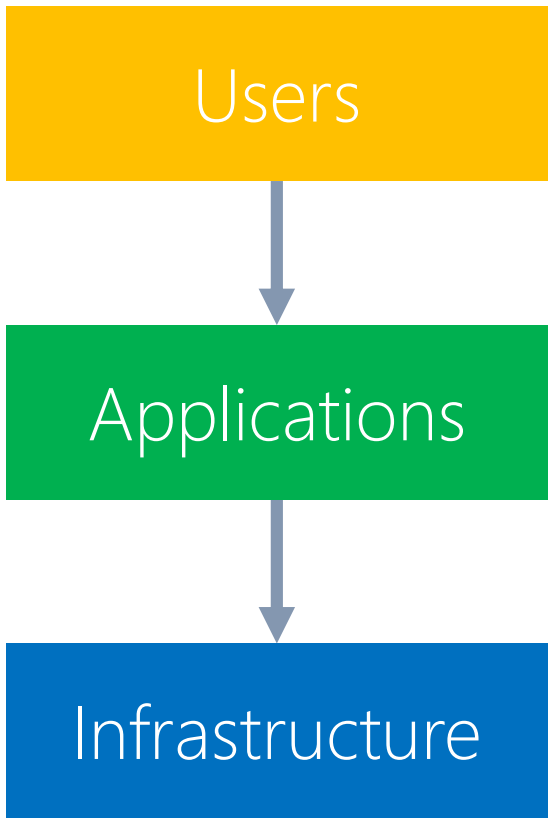
Reality

Time to reframe our thinking

Let applications drive infrastructure to meet user requirements

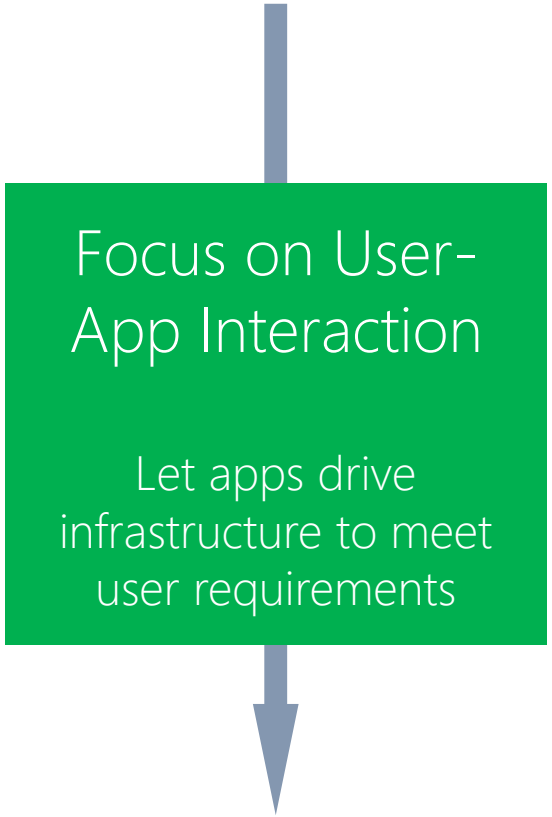
(in this model application workflows configure Kubernetes, Networking and Storage)

What if we focus on Apps, not Infrastructure



Most vendors are looking at the problem in this direction

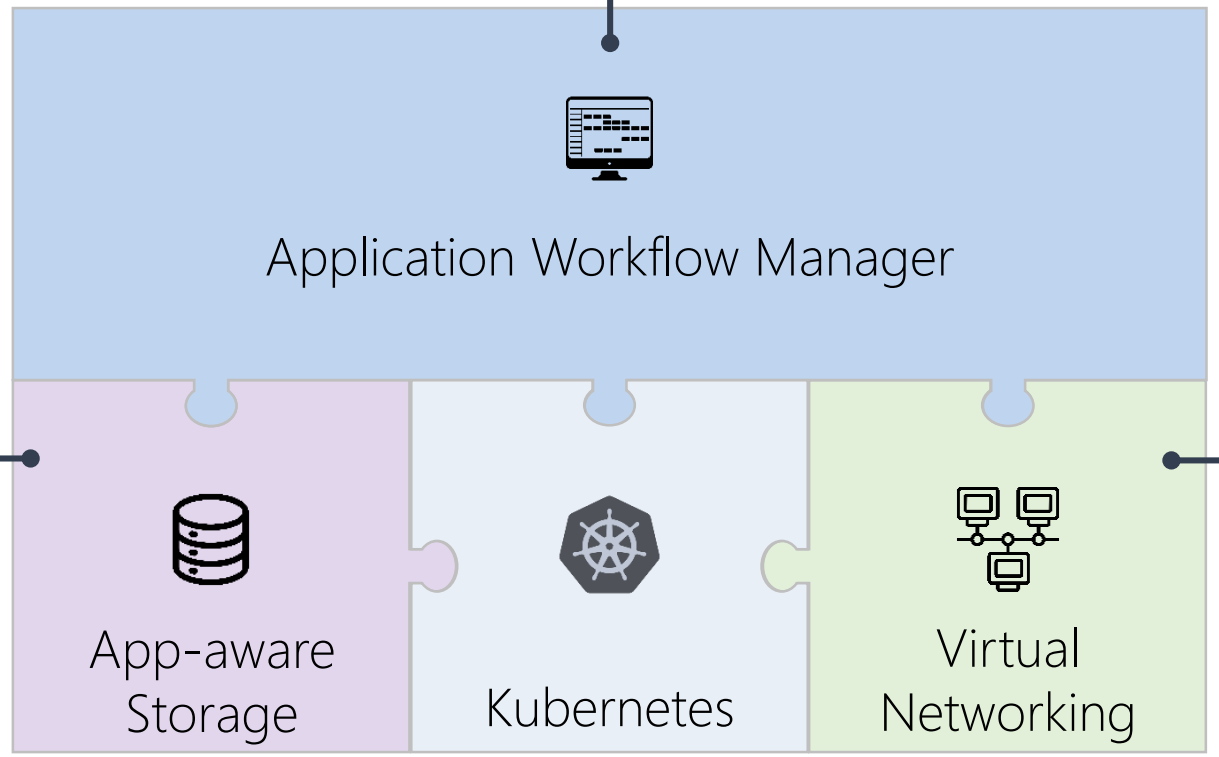
Whereas we should be looking at it in this direction



What would an ideal solution stack look like?

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

Application workflows configure Kubernetes, Storage & Networking



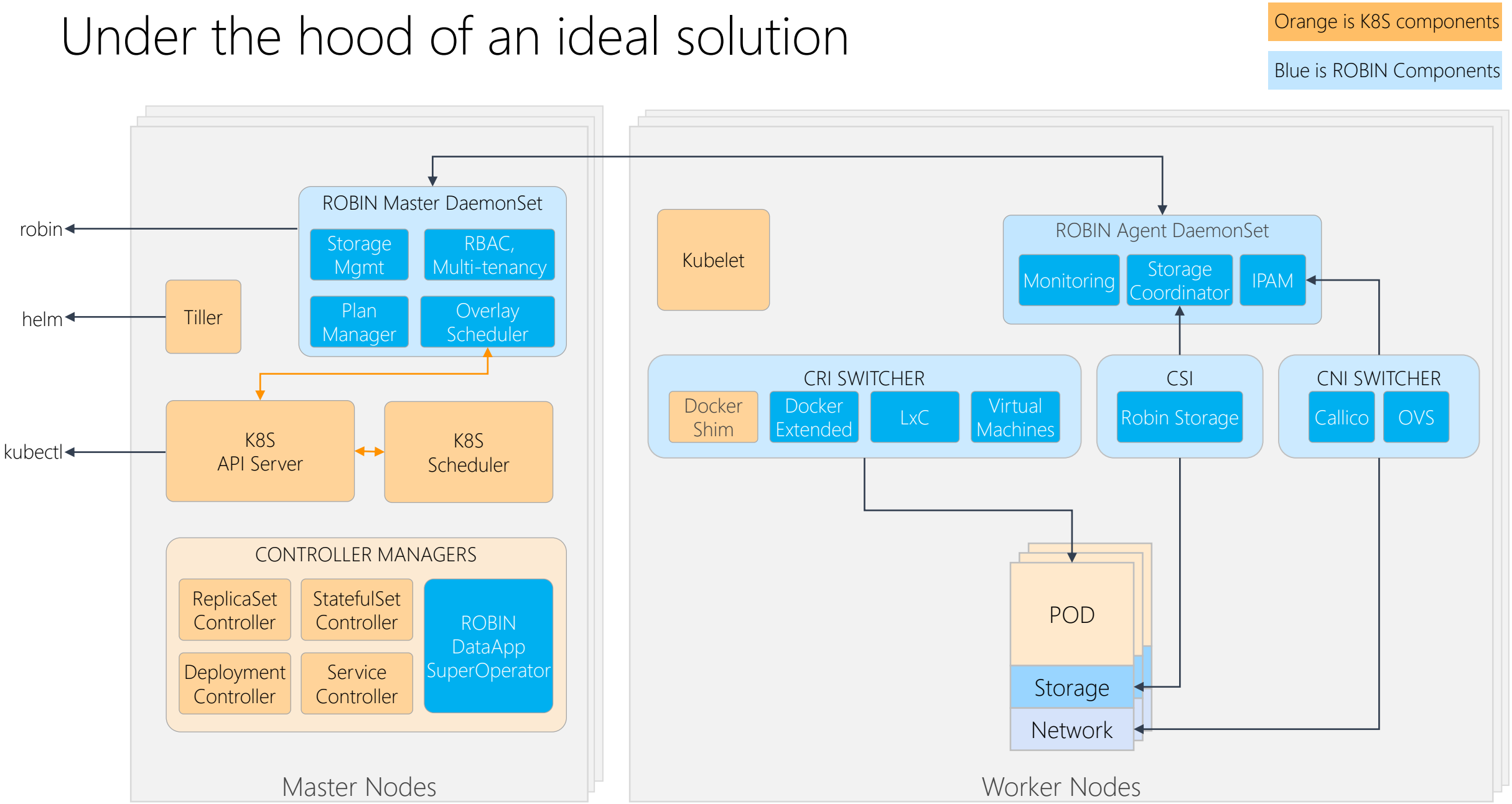
Enterprise-grade storage stack

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

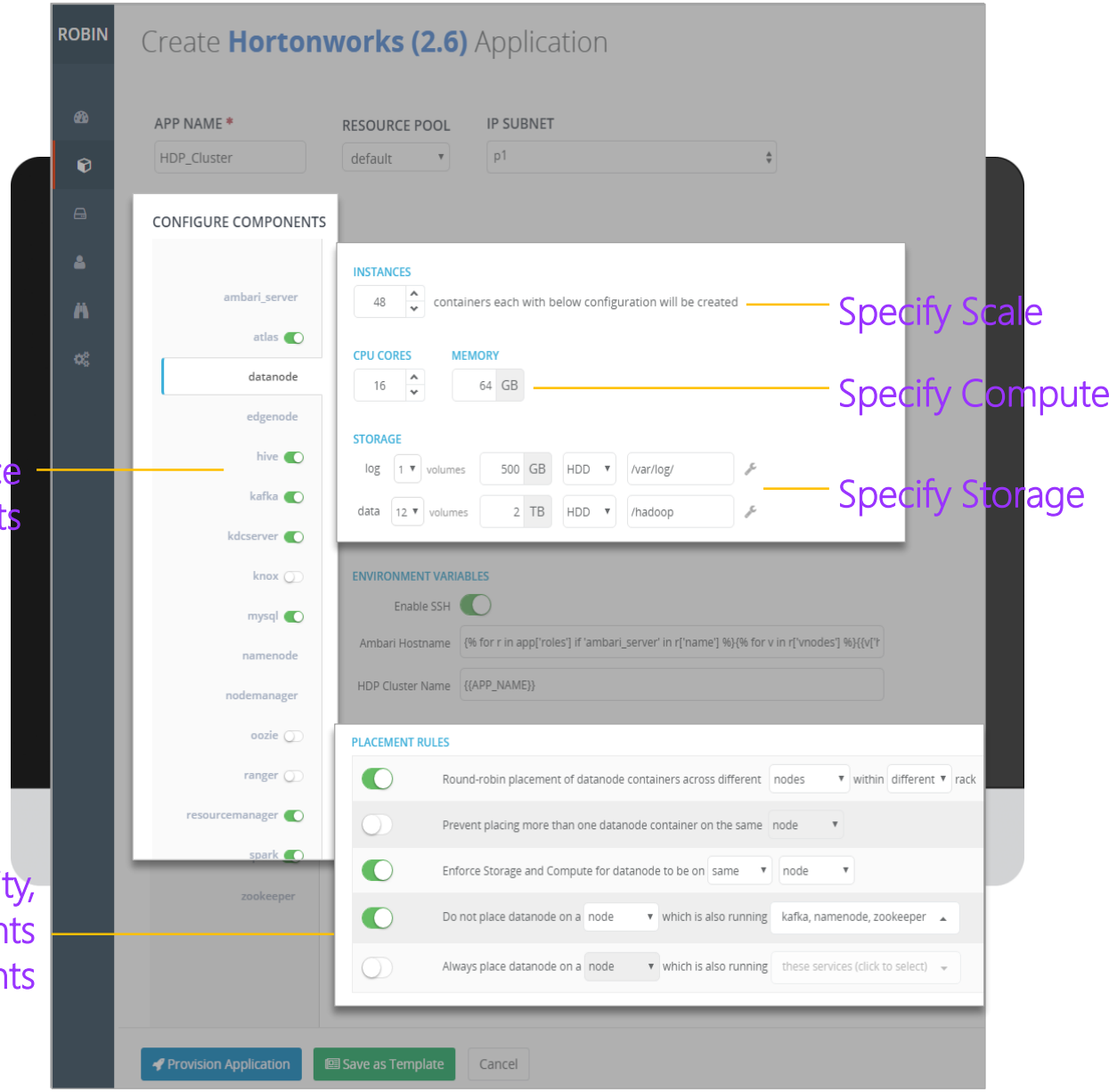
Built-in flexible networking

OVS, Calico, VLAN, Overlay networking, Persistent IPs

Under the hood of an ideal solution



1-Click Deployment of Big Data and Databases



Specify Scale

Specify Compute

Specify Storage

Enable Service Components

Specify Data-locality, Anti/Affinity Constraints and Placement Hints

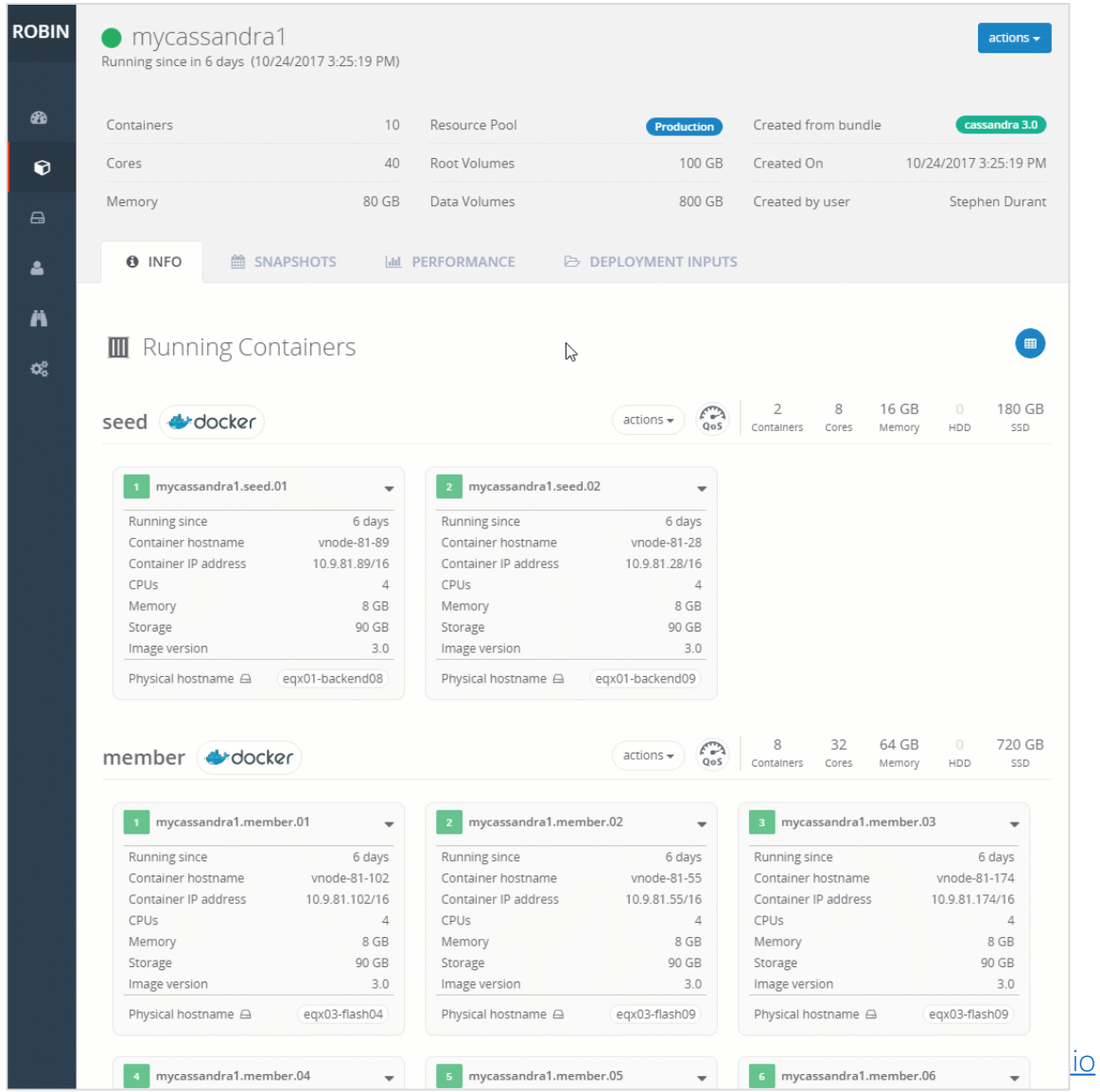
Application is up and running in minutes, not weeks

- ✓ K8S components auto created (StatefulSets, PVC, Services, …)
- ✓ Storage & Networking provisioned
- ✓ Data-locality, anti/affinity policies enforced
- ✓ Run workflow hooks to customize application

Common provisioning workflow for any Big Data, NoSQL, Database, AI/ML app

Agile DevOps for Big Data and Databases

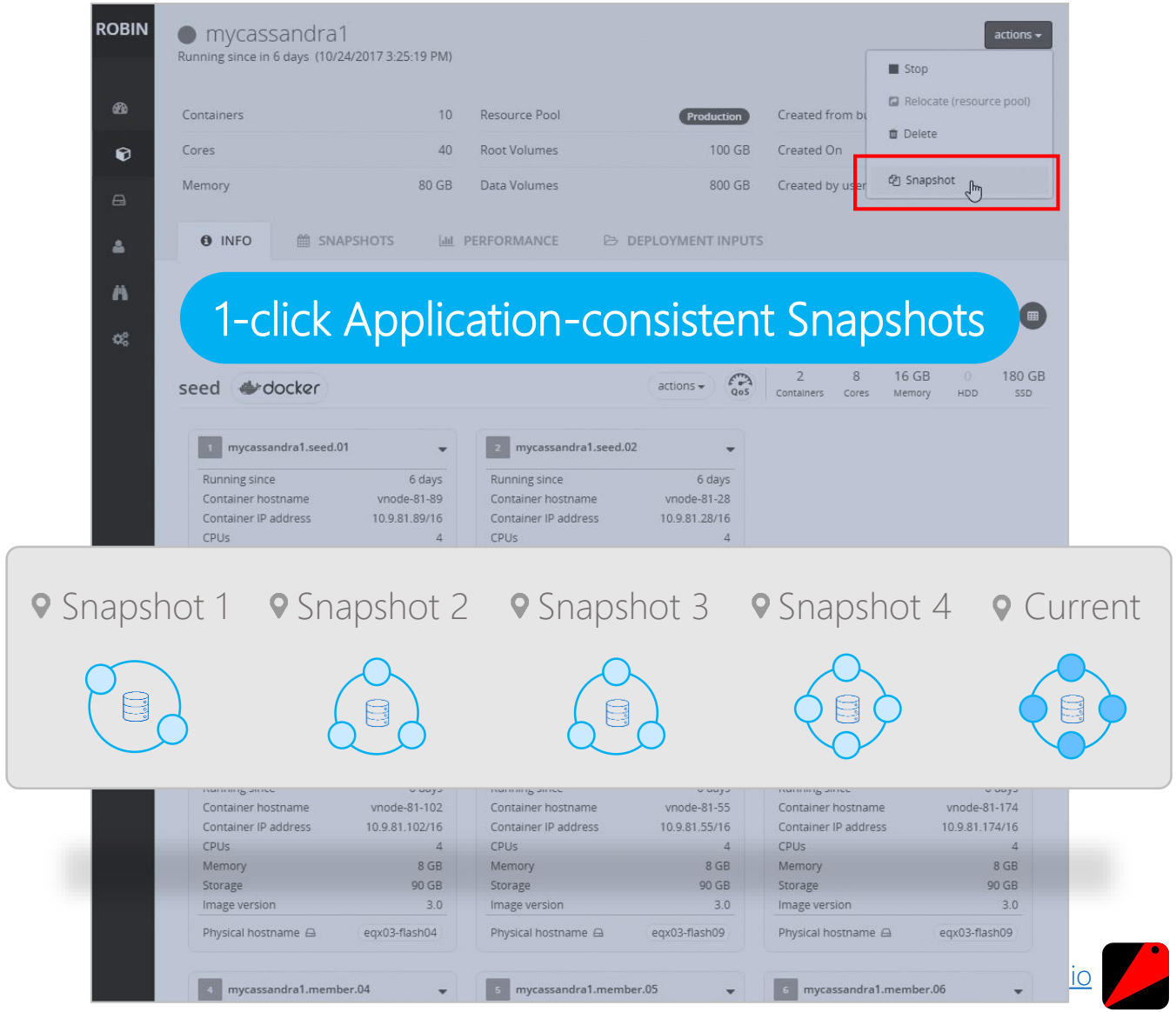
- **Time machine for applications**
  Time travel across multiple application states
- Clone and share entire applications
  for running reports, tests, and what-if analysis
- Backup and restore entire application
  avoid fear of app+data loss
- Safely upgrade application
  without fear of service disruption due to version incompatibilities
- Migrate entire applications with data to cloud



io

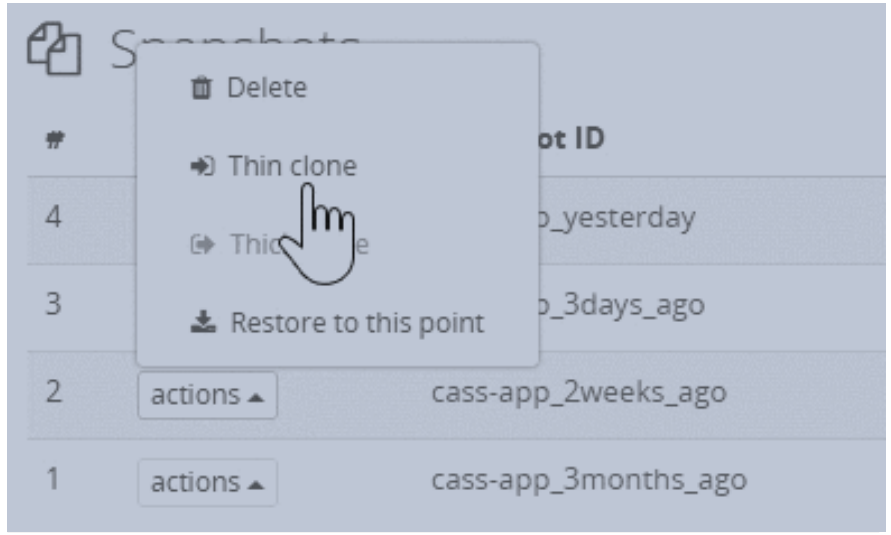
Agile DevOps for Big Data and Databases

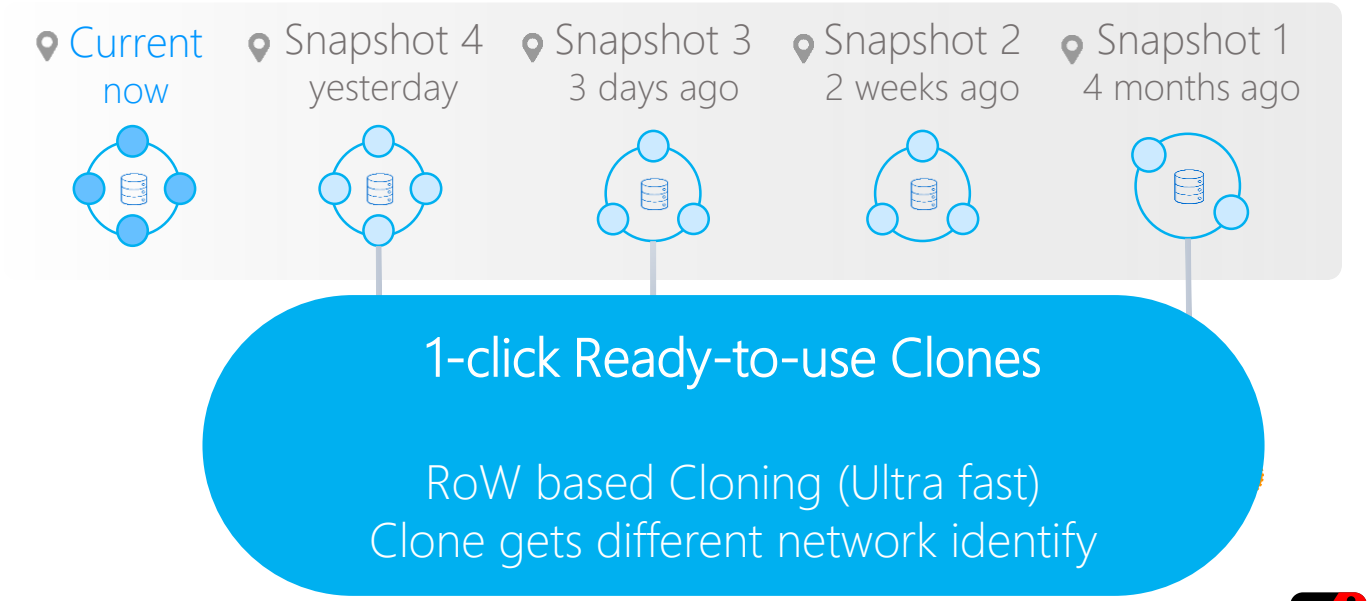
- **Time machine for applications**
  Time travel across multiple application states
- Clone and share entire applications
  for running reports, tests, and what-if analysis
- Backup and restore entire application
  avoid fear of app+data loss
- Safely upgrade application
  without fear of service disruption due to version incompatibilities
- Migrate entire applications with data to cloud



Agile DevOps for Big Data and Databases

- Time machine for applications
  Time travel across multiple application states
- **Clone and share entire applications**
  for running reports, tests, and what-if analysis
- Backup and restore entire application
  avoid fear of app+data loss
- Safely upgrade application
  without fear of service disruption due to version incompatibilities
- Migrate entire applications with data to cloud

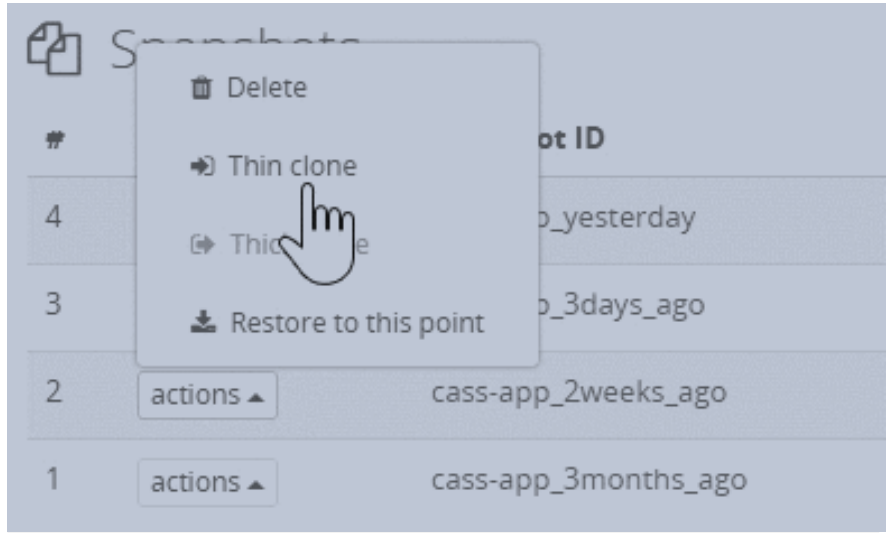


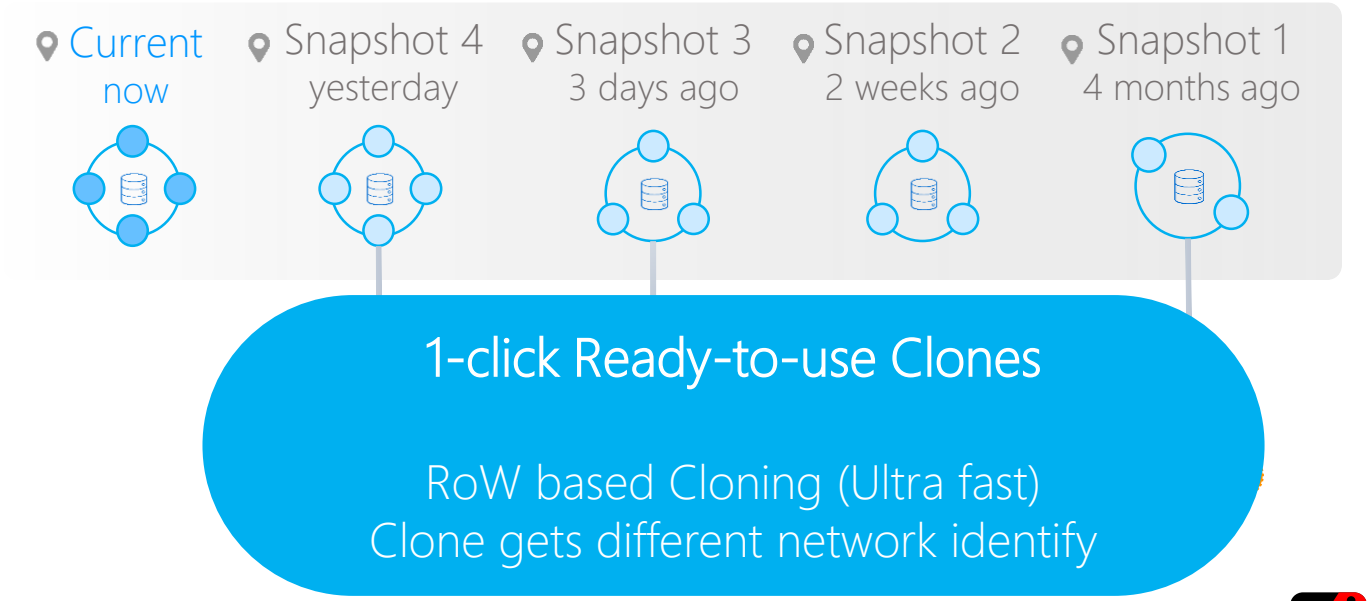


robin.io

Agile DevOps for Big Data and Databases

- Time machine for applications
  Time travel across multiple application states
- **Clone and share entire applications**
  for running reports, tests, and what-if analysis
- Backup and restore entire application
  avoid fear of app+data loss
- Safely upgrade application
  without fear of service disruption due to version incompatibilities
- Migrate entire applications with data to cloud





robin.io

Q&A

partha@robin.io

https://www.linkedin.com/in/parthaseetala

@parthaseetala

ROBIN.IO

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

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