

Autonomous Kubernetes Flash Management with Application Awareness





Introducing Portworx

Portworx is the most widely used data platform on Kubernetes

CUSTOMERS



100+
MORE

PARTNERS



Portworx brings a unique mix of cloud native expertise



STORAGE & DATA SERVICES

Years of expertise in
enterprise storage

Invented storage
orchestration for K8s



APPLICATION & ORCHESTRATION INTEGRATION

Deep and early Kubernetes
expertise

Co-led the founding of CSI

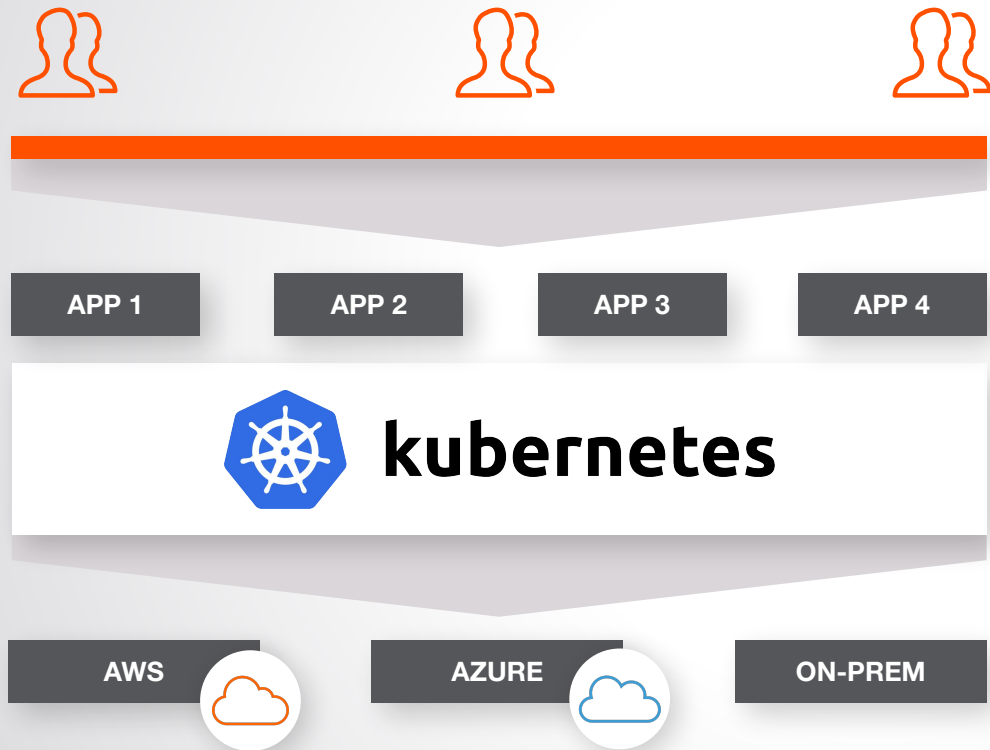


MULTI-CLOUD DATA MANAGEMENT

1st to offer multi-cloud
Kubernetes storage

1st to offer migration across
environments

What do platform owners want?



YOUR PLATFORM GOALS

- ▶ Self service for developers
- ▶ Fully automated
- ▶ Infrastructure agnostic SLAs
- ▶ Simple to adopt
- ▶ Low Touch Ops
- ▶ Optimization for cost

Portworx completes Kubernetes



APP 1

APP 2

APP 3

APP 4



kubernetes



portworx

Cloud native storage and data management platform

AWS

AZURE

ON-PREM

Any Block
Storage

Portworx is the **market leading Data Orchestration Platform** that is fully managed from within Kubernetes and also gives you the security, reliability and performance you'd expect from **enterprise class** traditional infrastructure

Challenges in running Stateful Workloads with NVMe

1. High availability and Data protection requires expensive all-flash arrays
2. Data protection across rack level failure and data center failure does not exist without a custom hardware all-flash SAN solution
3. End to end perf. QoS requires manual configuration and setup
4. Data encryption and key management can get hard to manage
5. Static provisioning can waste a lot of flash space
6. Most K8s provisioners available will do DAS provisioning
7. K8S CSI is still far away from providing end to end data management capabilities

Portworx for NVMe Data Management in Kubernetes

1. Dynamic NVMe Namespace provisioning tied to k8s volume provisioning
 - a. Works with all major NVMe drive providers
 - b. Integrations with Toshiba Kumoscale and other NVMeOF appliances
 - c. End to end hard storage performance guarantees through Kubernetes PVC mapped to NVMe namespace
2. Rack-Aware and Data center aware replication to protect against SPOF in case of NVMe DAS architectures through Portworx Operator called STORK
3. Automatic class of service classification and provisioning that identifies higher performance drives and place data that requires higher performance dynamically on the volumes residing in those pools
4. Dynamic I/O prioritization at a container granular volume level to provide multi-tenant IOPS provisioned Kubernetes volumes
5. Dynamic capacity management through Portworx Operator called Auto-pilot

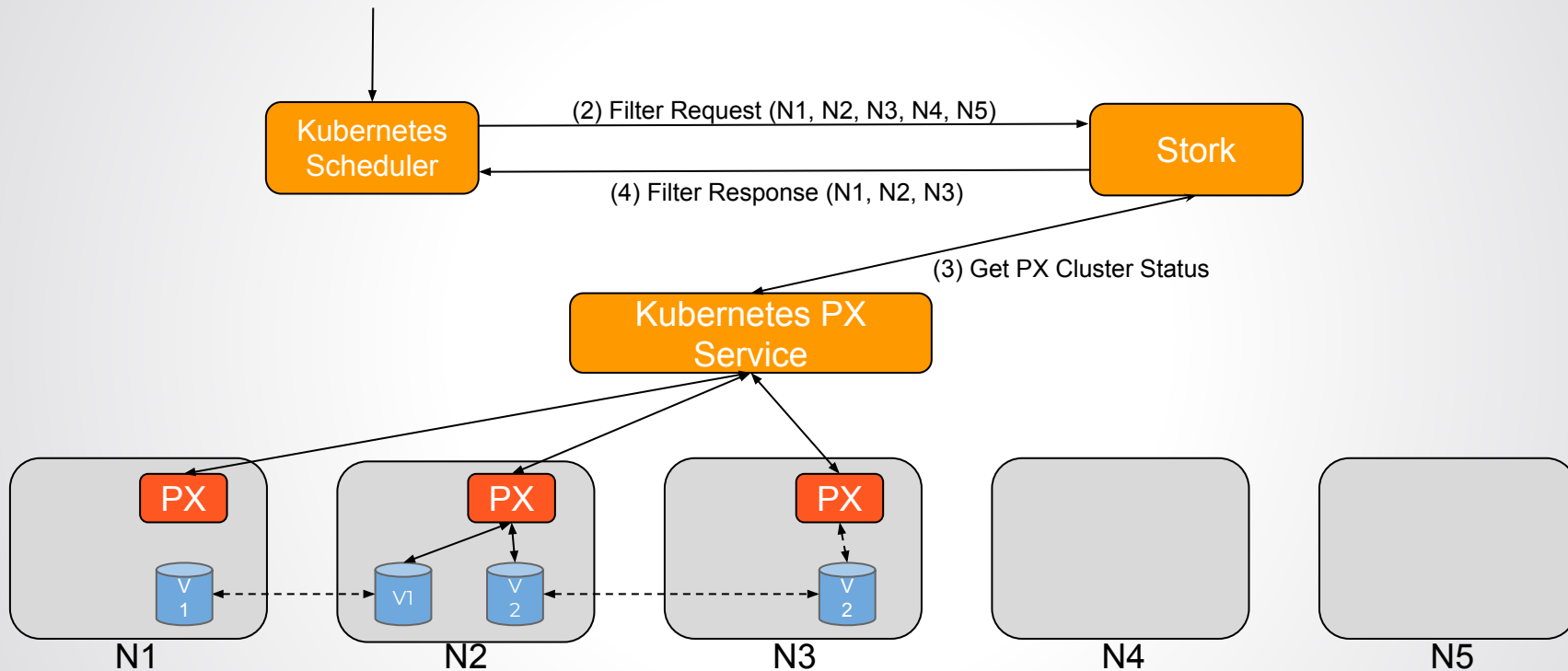


Stateful Data services with **STORK**

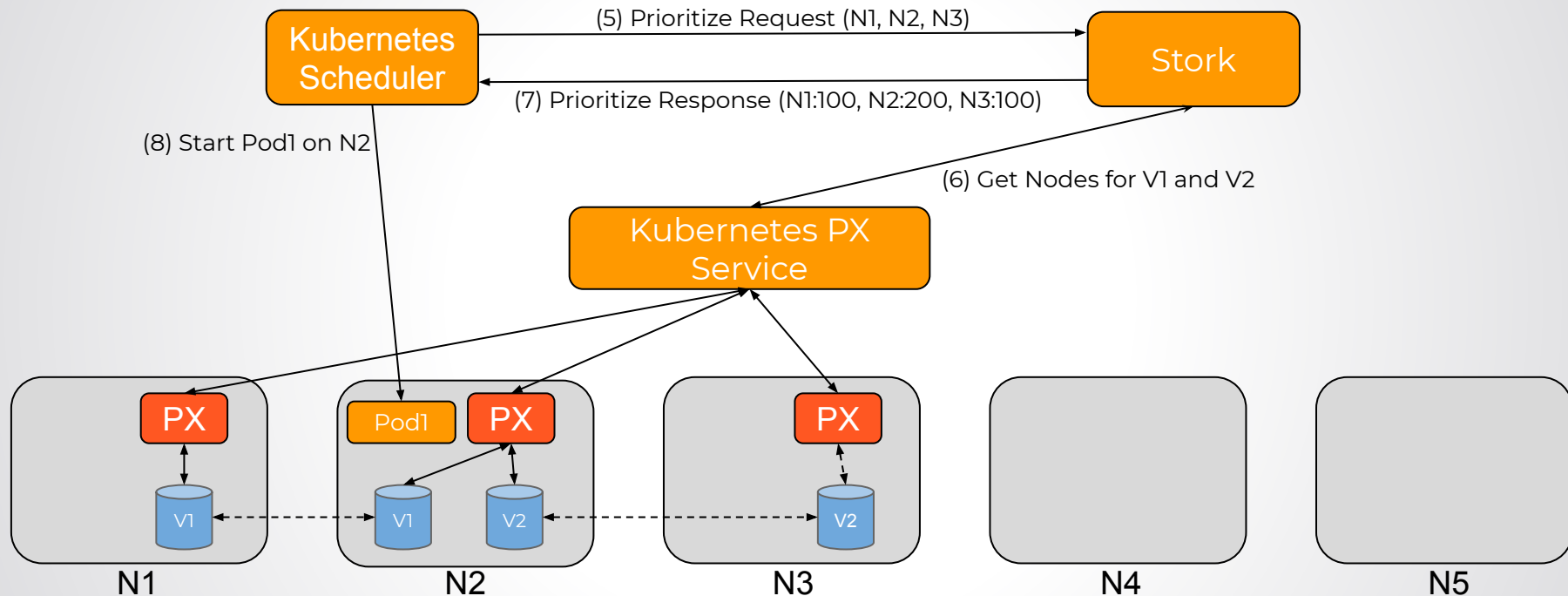
Motivation

- Help run stateful applications more efficiently on Kubernetes
 - Provide Hyper-convergence
 - Advanced health monitoring of stateful apps
- Manage lifecycle of stateful applications
 - Application consistent snapshots
 - Migrate applications between clusters
 - Backup Data + K8s resources
- Plugin model, can be extended to work with any storage driver

Scheduling stateful services efficiently



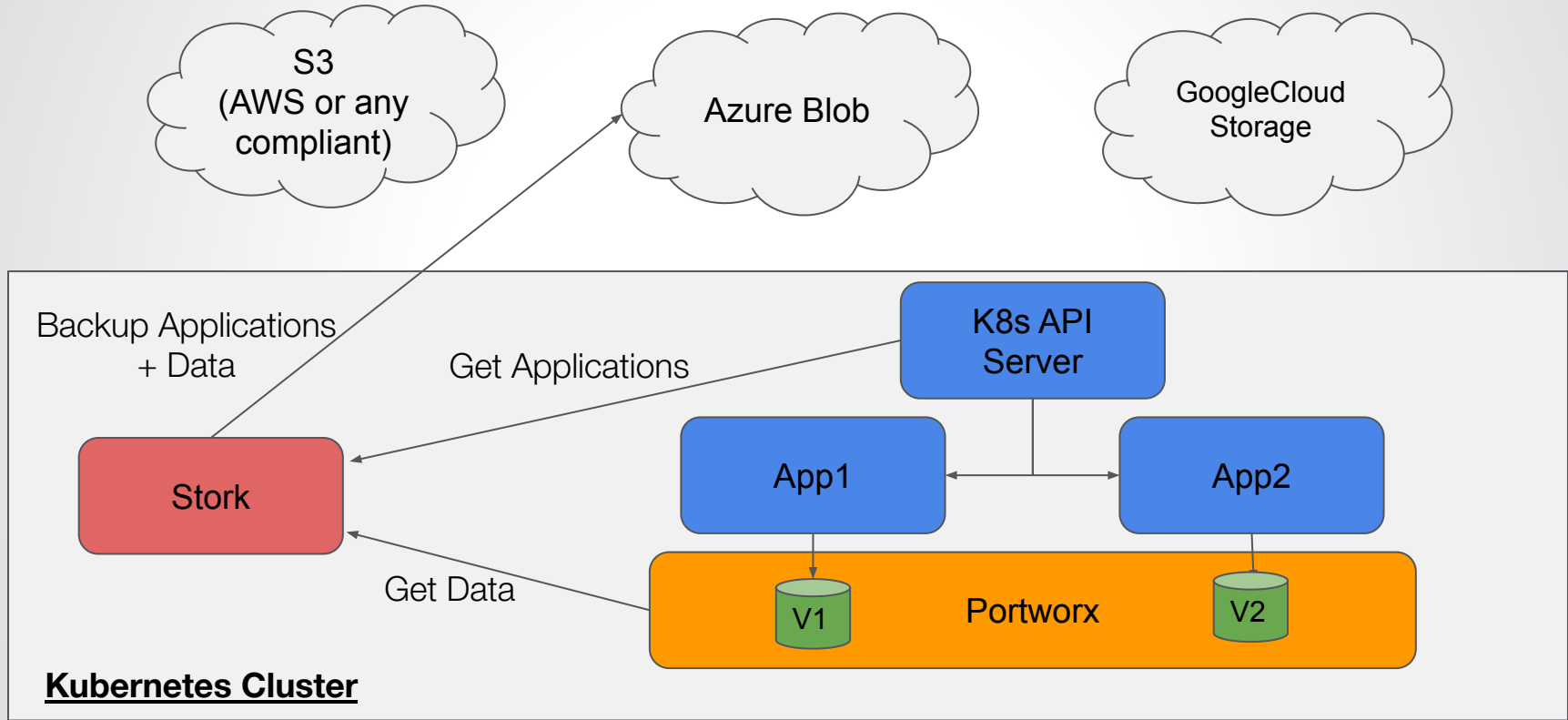
Scheduling stateful services efficiently



Storage Health Monitoring

- Monitors the health of storage driver on all nodes
- Storage driver offline?
 - Reschedule pods using storage driver
- Rescheduled on another node with volume replica
 - Continue with local disk performance
- Without this, pods will get stuck in Pending, or not able to access storage
- For stateful sets this also deals with scenarios where kubelet reports offline on a node

ApplicationBackup

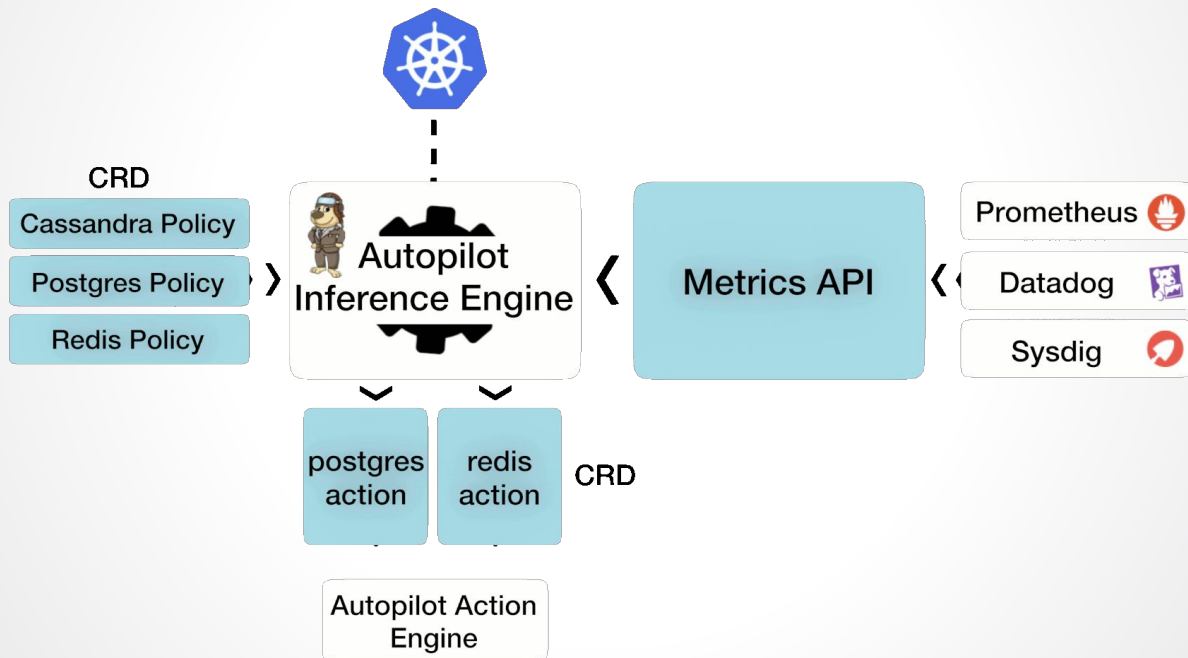




Autogrow storage pools with Autopilot

What is autopilot?

Autopilot is a monitor-and-react engine



Autopilot use cases

- **Autogrow volumes**
- **Autogrow Portworx storage pools**
- Autoscale applications when load increases
- Rebalance volumes when certain nodes experience latencies
-

The Autopilot Operator framework is extensible to watch for **any set of metrics** and perform **user defined actions**.