Running Stateful Workloads On Kubernetes: Challenges And Solutions

Jagadish Mukku
Senior Architect, Robin.io
Stateful Application

› Application that saves client data of one session for use in next session or by other applications.

› e.g mysql, mongodb, cassandra, Key value stores
Spectrum Of Applications

Stateless Applications

Web Apps

Stateful Applications

SQL Databases

NoSQL Databases

Big Data

- splunk
- cloudera
- HORTONWORKS
- ELK Stack
- GREENPLUM

- mongoDB
- Couchbase
- cassandra
- redis
- influxdb
- Prometheus
Stateful Application Deployment

Traditional Deployment

Virtualized Deployment
Containerized Stateful Application Deployment

Container Deployment

Hardware

Operating System

Container Runtime

Container

Bin/Library

App
Kubernetes

Portable, Extensible, open-source platform for managing containerized workloads and services

Reduce Costs -> Containerize apps and Consolidate Resources

Agile -> Modularity and Scalability

Life Cycle Management -> Health Checks, Automated Rollouts, Canary deployment, Load Balancing
Application Composition

Application Composition .. The Complexity

MySQL

MariaDB

MongoDB

ElasticSearch

ELK Stack

https://github.com/helm/charts
Perils of consolidation ..

Software Defined Storage
Application awareness

Software Defined Storage

CSI

Flash Storage

Redo

Data

PVC

HR

Sales

Application awareness

Software Defined Storage

Flash Storage

Redo

Data

PVC

HR

Sales
Consolidation complexity

Imagine managing 100s of PODs and 1000s of PVCs allocated from 100s of Drives
Challenges of running Stateful Apps on Kubernetes

› Servers/Disks fail and humans make errors → Apps become unavailable, Data is lost

› Data must be protected and secured → Expects users to become storage experts

› Unpredictable performance from sharing resources → Application SLAs are not met, unhappy users

PersistentVolumes are just the first step to running stateful apps in production
Challenges of running Stateful Apps

› Unable to share data between prod and dev/test → Low productivity and slower innovation

› Data gravity prevents app mobility across environments → Cloud or Infrastructure Lock-in
How Robin helps you run stateful apps on Kubernetes

ANY STATEFUL & ENTERPRISE APP
RDBMS, NoSQL, Queues, Timeseries, Big Data, ...

STORAGE, DATA MANAGEMENT and
APP WORKFLOW MANAGER
ROBIN

ANY KUBERNETES DISTRO
Openshift, GKE, Anthos, PKS, AKS, EKS

ANY INFRASTRUCTURE
Baremetal, VMs
AWS, Azure, Google Cloud
Local Flash, Disks, SAN
EBS, PD, Block, Object
Robin Architecture Overview

1-click application Deploy, Snapshot, Clone, Scale, Upgrade, Backup
Application workflows configure Kubernetes, Storage & Networking

Application Workflow Manager

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

Works any where
ROBIN Application Aware Provisioning and QoS

- Correct binding between storage and application volumes
- Failure tolerance
  - Disk, Node, Rack, Data Centre
- QoS guarantees
  - I/O is tagged all the way to the storage stack.
  - Min IOPS, Max IOPS
  - On-demand scale up of resources
  - Ideal to meet temporary demand
- Performance isolation of apps and their Thin Clones
Storage Provisioning

Just Volume Provisioning

Application Aware Provisioning
Protect And Collaborate

› Rock-solid Data Availability and Resiliency
  › Strictly-consistent Replication, with automatic resync on fall-behind
  › Detects Disk and IO errors
  › Disk and Volume rebuilds

› Data Security
  › Encryption with Key Management Integration

› Application Snapshots
  › Consistency volume group
  › Protect against corruption, user errors
  › Rollback to any point

› Thin Clones
  › Quick and Simple to Collaborate
Protect an entire Application, not just Storage Volumes

1. Maintain periodic checkpoints of your entire app with data
   $ robin snapshot app1 snap1

2. Rollback entire app+data to healthy state to recover from corruptions or user errors
   $ robin rollback snap1 app1

3. Backup entire app+data as into external backup targets
   $ robin backup snap1 target

4. Restore entire app+data to healthy state from catastrophic hardware and datacenter failures
   $ robin restore target snap1

- ROBIN Backups are fully self-contained
- Entire app resources can be restored in the same or different data center or cloud even if the source is completely destroyed

1. DATA PersistentVolumeClaims
2. CONFIG ConfigMap, Secret, Labels, ...
3. METADATA Pods, StatefulSets, Services, ...

Any Kubernetes Distro

Local Backup Target
Remote (Cloud) Backup Target
Make DevOps productive with Stateful Apps

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

On-prem

Kubernetes Cluster 1

ROBIN

APP

CLONE

Snapshots:
- Snapshot 1: 3 months ago
- Snapshot 2: 3 days ago
- Snapshot 3: yesterday

STEP1: `robin snapshot mysql mysql-snap`

STEP2: `robin clone mysql-snap testdev-mysql`

STEP3: `robin push mysql-snap gcs://bucket`

CLOUD OBJECT STORE
(S3, GCS, AzureBlob)

STEP4: `robin pull gcs://bucket/mysql-snap mysql`

Google Cloud Platform

Kubernetes Cluster 2

ROBIN

AWS

Kubernetes Cluster 3

ROBIN
Thankyou

www.Robin.io
Agenda

› What is Stateful application
› Why Kubernetes
› Challenges deploying stateful application in Kubernetes
› ROBIN introduction
› Solutions
› Q & A