Bringing the Database inside the Microservices World on Red Hat Openshift with Couchbase Data Platform

Daniele Paolucci - Lead Devops Engineer (Spindox)
Arduino Cascella - Solutions Engineer (Couchbase)
AGENDA

01  The Path From Monolith to Microservice Architecture
02  Microservices, Containers and Operations
03  Couchbase: The Data Platform For Containers
04  Couchbase Operator Demo
05  Example of Stateful Application
1) The Path From Monolithic to Microservices Architecture
Why Business requires Microservices
Pattern changes Between Monolithic and Microservices

Monolithic and SOA vs Microservices

- **Sharing Functions**
  - New Function needs testing whole app
  - No Real need for CI/CD
  - Complex messaging protocols (ESB)
  - Needs specific framework and application server
  - One single data store for all services
  - Dev and Ops Teams work independently

- **Decoupling Functions**
  - New Function is a separate New Service
  - CI/CD are indispensable
  - Lightweight protocol such HTTP and REST
  - many frameworks, no real need for application servers
  - Every Microservice have his own datastore
  - Dev and Ops teams work together with the same KPIs
Microservices Scalability: The Scale Cube
2) Microservices, Containers and Operations
Containers - An Evolution in Application Deployment

Containers enable:

- **Efficiency and automation** for microservices, but also support traditional applications
- **Faster and more consistent deployments** from Development to Production
- **Application portability** across 4 infrastructure footprints: Physical, Virtual, Private & Public Cloud
Microservices and Operations: new challenges

- Microservices Need complex Infrastructure Setup
- Setup may change depending on the Infrastructure type
- Create or update Microservices may need a lot of effort in IT Operations
Different Platforms, different operations
Kubernetes abstracts operations on microservices
Complete software lifecycle management with OpenShift PaaS
3) Couchbase: The Data Platform For Containers
Why Not Databases in Containers?

3 major obstacles for DB containerisation:
1. How to shard data among containers?
2. How to avoid data corruption if multiple writes in different pods?
3. How to ensure high availability, backup policies, restore?

Only state is not enough!
THE COUCHBASE DATA PLATFORM

Develop with agility. Deploy at any scale.
Couchbase designed for containerized applications

Microservice Architecture == Multi-Dimensional Scaling
Couchbase Designed for Containerized Applications

Geo Distribution

Centralized Management

Auto Provisioning

Auto Recovery

#RedHatOSD
Couchbase and Red Hat enable hybrid/multi-cloud
Customers across every industry embracing digital
digital

**GANNETT**
- 50M unique monthly visitors
- 2.5B monthly page views
- Replaced MongoDB

**Linkedin**
- 2821 nodes, 100+ clusters
- 16M entries every 5 min
- 2.5 million ops/sec. on a single cluster

**PayPal**
- 1 billion+ documents
- 10TB+ data
- Sub-200ms response time

<table>
<thead>
<tr>
<th>E-Commerce</th>
<th>Travel</th>
<th>Gaming</th>
<th>Communications</th>
<th>Financial Services</th>
<th>Digital Health</th>
<th>Digital Media</th>
<th>Industrial IoT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walmart</td>
<td>Marriott</td>
<td>EA</td>
<td>Comcast</td>
<td>ADP</td>
<td>WebMD</td>
<td>sky</td>
<td>THOMSON REUTERS</td>
</tr>
<tr>
<td>Best Buy</td>
<td>OfficeMax</td>
<td>Blizzard</td>
<td>AT&amp;T</td>
<td>FICO</td>
<td>BD</td>
<td>GANNETT</td>
<td>Chamberlain</td>
</tr>
<tr>
<td>Office Depot</td>
<td>Sabre</td>
<td>EA</td>
<td>Viber</td>
<td>Concur</td>
<td>Everyday Health</td>
<td>Fandango</td>
<td>Numerex</td>
</tr>
<tr>
<td>TEISCO</td>
<td>Orbitz</td>
<td>Blizzard</td>
<td>AT&amp;T</td>
<td>Visa</td>
<td>BD</td>
<td>GANNETT</td>
<td>Chamberlain</td>
</tr>
<tr>
<td>Staples</td>
<td>TripAdvisor</td>
<td>PlayStation</td>
<td>Viber</td>
<td>VISA</td>
<td>BD</td>
<td>GANNETT</td>
<td>Chamberlain</td>
</tr>
<tr>
<td>Tesco</td>
<td>Booking.com</td>
<td>Pokémon Go</td>
<td>Telefonica</td>
<td>VISA</td>
<td>BD</td>
<td>GANNETT</td>
<td>Chamberlain</td>
</tr>
<tr>
<td>Q</td>
<td>Tripadvisor</td>
<td>Playtica</td>
<td>Telefonica</td>
<td>VISA</td>
<td>BD</td>
<td>GANNETT</td>
<td>Chamberlain</td>
</tr>
</tbody>
</table>

#RedHatOSD
Couchbase, by the Numbers

- 350+ Employees
- 100% Open Source
- 500+ Customers
COUCHBASE AUTONOMOUS OPERATOR
Introducing Couchbase Autonomous Operator

Couchbase Autonomous Operator is an application-specific controller that extends the Kubernetes API to create, configure and manage instances of complex stateful applications on behalf of a Kubernetes user.

It builds upon the basic Kubernetes resource and controller concepts, but also includes domain or application-specific knowledge to automate common tasks better managed by computers.
Multi-Dimensional Scaling

```yaml
indexStorageSetting: memory_optimized
autoFailoverTimeout: 30
buckets:
  - name: couchbase-sample
type: couchbase
memoryQuota: 128
replicas: 3
ioPriority: high
evictionPolicy: fullEviction
conflictResolution: segno
enableFlush: true
enableIndexReplica: false
servers:
  - size: 2
    name: data_and_index
    services:
      - data
      - index
dataPath: /opt/couchbase/var/lib/couchbase/data
indexPath: /opt/couchbase/var/lib/couchbase/data
  - size: 1
    name: query_and_search
    services:
      - query
      - search
dataPath: /opt/couchbase/var/lib/couchbase/data
indexPath: /opt/couchbase/var/lib/couchbase/data
```
Architecture
Key Benefits

1. Best way to Run and Manage Couchbase in Your Data Center – Up to 95% Reduction in Operational Overhead

2. Automated Best Practices - Zero-Downtime Ops

3. Elastic scaling - Push a button Dynamic Scaling
4) Demo: The Couchbase Operator in Action
## Pods

<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
<th>Containers Ready</th>
<th>Container Restarts</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>couchbase-operator-7f689bdbfc-v7nrc</td>
<td>Running</td>
<td>1/1</td>
<td>0</td>
<td>a few seconds</td>
</tr>
</tbody>
</table>
```
apiVersion: couchbase.database.couchbase.com/v1beta1
kind: CouchbaseCluster
metadata:
  name: cb-example
spec:
  baseImage: couchbase/server
  version: enterprise-5.0.1
  authSecret: cb-example-auth
  exposeAdminConsole: true
  cluster:
    dataServiceMemoryQuota: 256
    indexServiceMemoryQuota: 256
    searchServiceMemoryQuota: 256
    indexStorageSetting: memory_optimized
    autoFailoverTimeout: 30
  buckets:
    - name: default
      type: couchbase
      memoryQuota: 128
      replicas: 1
      ioPriority: high
      evictionPolicy: fullEviction
      conflictResolution: seqno
      enableFlush: true
      enableIndexReplica: false
  servers:
    - size: 3
      name: all_services
      services:
      - data
      - index
      - query
      - search
    dataPath: /opt/couchbase/var/lib/couchbase/data
    indexPath: /opt/couchbase/var/lib/couchbase/data
```
OpenShift Web Console - Mozilla Firefox

OpenShift Container Platform

couchbase-operator

Pods

<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
<th>Containers Ready</th>
<th>Container Restarts</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>cb-example-0001</td>
<td>Container Creating</td>
<td>0/1</td>
<td>0</td>
<td>a few seconds</td>
</tr>
<tr>
<td>cb-example-0000</td>
<td>Running</td>
<td>1/1</td>
<td>0</td>
<td>a few seconds</td>
</tr>
<tr>
<td>couchbase-operator-7f689bdbf-v7nrc</td>
<td>Running</td>
<td>1/1</td>
<td>0</td>
<td>a minute</td>
</tr>
<tr>
<td>Name</td>
<td>Status</td>
<td>Containers Ready</td>
<td>Container Restarts</td>
<td>Age</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------</td>
<td>------------------</td>
<td>--------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>cb-example-0002</td>
<td>Running</td>
<td>1/1</td>
<td>0</td>
<td>a few seconds</td>
</tr>
<tr>
<td>cb-example-0001</td>
<td>Running</td>
<td>1/1</td>
<td>0</td>
<td>a few seconds</td>
</tr>
<tr>
<td>cb-example-0000</td>
<td>Running</td>
<td>1/1</td>
<td>0</td>
<td>a minute</td>
</tr>
<tr>
<td>couchbase-operator-7f689dbbf-v7nrc</td>
<td>Running</td>
<td>1/1</td>
<td>0</td>
<td>2 minutes</td>
</tr>
</tbody>
</table>
OpenShift Container Platform

couchbase-operator

cb-example-0001

Status:
- Status: Running
- IP: 10.131.2.91
- Node: oce-0scp-app-2.demolab.locat(10.60.0.211)
- Restart Policy: Never

Container: couchbase-server
- State: Running since Jun 3, 2016 8:27:26 PM
- Ready: true
- Restart Count: 0

Template: couchbase-server
5) Use Case: Running a stateful MPI Application
Use case

Microservices Architecture for the implementation of a pan-European Customer Portal to provide to Customers an easy way to see their own contractual information and to interact with back-end services as requested.
More info on Couchbase Autonomous Operator

   blog post on **Couchbase** and our use of the **Kubernetes StatefulSet API**.

2. [https://blog.couchbase.com/introducing-couchbase-operator/](https://blog.couchbase.com/introducing-couchbase-operator/) -
   blog post on the **Couchbase Operator + OpenShift** beta announcement.

3. [Red Hat Technical Implementation Guide (TIG) for Couchbase on OpenShift 3.9](https://www.redhat.com/software/technical-implementation-guide-
   couchbase-on-openshift-3.9)

   blog post “**Couchbase on OpenShift in Action**”

   blog post on creating and installing a Couchbase cluster with the **Couchbase Operator in AKS** (Azure Container Service)
GRAZIE PER L’ATTENZIONE

Daniele Paolucci - Lead Devops Engineer (Spindox)
Arduino Cascella - Solutions Engineer (Couchbase)