

Key Benefits and Use Cases Supporting Scopes and Collections Using Couchbase Autonomous Operator on Kubernetes



INTRODUCTION

In the past, developers encountered multiple challenges when working with microservice-based applications with Couchbase, specifically on container and Kubernetes-based environments. For example:

- **Database silos:** Typically, enterprises moving toward a microservices architecture adopted containers and Kubernetes technologies. They often isolated microservices applications running in containers from their stateful application, i.e., database on premises or VMs, building new silos in their infrastructure.
- **Everything in a single bucket:** Due to Couchbase Server's limitations on the number of available buckets, users ended up storing all the data for an application in a single bucket. If the application had more than one microservice, there was no data isolation. We solved the database silo problem with Couchbase Autonomous Operator for Kubernetes, where Couchbase runs as a containerized stateful application on the Kubernetes platform and stateless applications.

For the second challenge, we didn't have an elegant solution. We initially recommended users of Couchbase Autonomous Operator to create multiple Couchbase clusters and have data isolation for each microservice. This model is typically called "data services for a microservices application."



MICROSERVICES DEPLOYMENT STRATEGIES IN KUBERNETES

The following describes the different microservice deployment strategies with Couchbase that customers have adopted.

Shared Microservices with Centralized Database

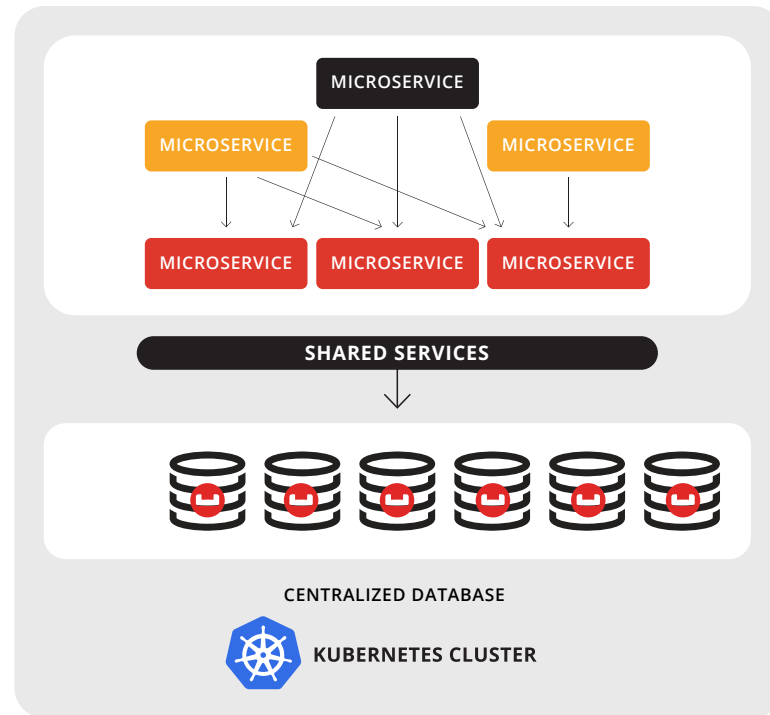
In this deployment model the microservice acts as a shared service and a single source of truth as a "centralized database."

KEY POINTS

- Single source of truth where data is consistent across all the microservices
- The database coexists on the same Kubernetes platform as the microservices so networking, deployment, monitoring, and management becomes much easier
- Couchbase is built for specific data models and has flexible schemas that allow developers to create and manage different microservices independently within the same cluster



- [Before Couchbase Server 7.0] Due to the limitations of the number of buckets in Couchbase, customers wanting to deploy microservices requiring logical isolation and access isolation, security could not achieve this with a single Couchbase cluster
- [With Couchbase Server 7.0] With scopes and collections, users can achieve logical and access isolation (security) within the single Couchbase cluster



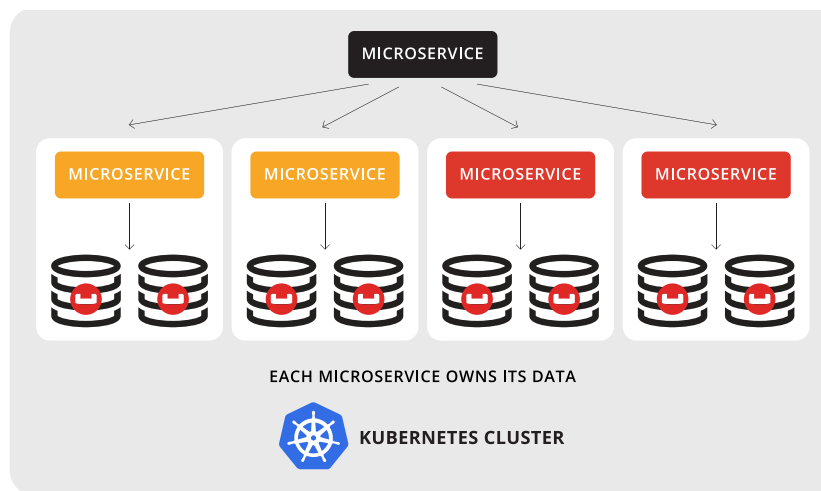
Database per microservice

In this deployment model each microservice manages its own data. This means that no other microservice can access the data directly.

KEY POINTS

- The Couchbase cluster acts as a data service provider for each microservice independently, and depending on the use case it can act as a cache, in-memory database, key-value store, document database with N1QL and indexing, document database with search, etc.
- Couchbase gets deployed with all microservices in the same namespace, zone, or region, which is highly performant
- Significantly higher TCO when utilizing a dedicated Couchbase cluster since the minimum number of nodes for data is 3, and multi-dimensional scaling, query, and indexing increases the resource requirements





SCOPES AND COLLECTIONS IN COUCHBASE SERVER 7.0



Scopes and collections is a significant architectural improvement to Couchbase Server 7.0 that offers many important benefits to customers (covered in detail under [Use Cases and Headlines](#) in Collections PRD).

Data isolation and high application density use cases are crucial for using Couchbase Autonomous Operator with modern application development, i.e., microservices-based architecture.

Developers use scopes and collections to organize data as logical containers within a Couchbase bucket for two use cases, namely:

- 1. Isolate schemas** – A mechanism to group all collections for a single application, similar to the schema concept in Oracle and other relational databases. This enables application consolidation by eliminating name collisions or reliance on naming conventions. Collections for one application would live in their own schema/scope.
- 2. Microservice-based isolation** – Microservices require logical isolation and access isolation from each other. Collections with role-based access control (RBAC) enable customers to consolidate their microservices-based applications onto a single Couchbase cluster and significantly reduce their TCO.

SHOULD AUTONOMOUS OPERATOR BE SCOPE AND COLLECTIONS AWARE?

The goal of the Couchbase Autonomous Operator is to fully auto-manage one or more Couchbase deployments so that users don't need to perform manual operations and worry about the operational complexities of running Couchbase. Not only does the Autonomous Operator automatically administer the Couchbase cluster, it also self-heals, upgrades, automatically scales, etc.



With the introduction of scopes and collections in Couchbase Server 7.0 there are significant changes to the operational aspect of the system. Below is the list of those modifications and the implications of each:

	Bucket	Scope	Collections
Memory quotas	✓	✗	✗
Disk partition mapping	✓	✗	✗
Number of replicas	✓	✗	✗
Eviction policy	✓	✗	✗
RBAC	✓	✓	✓
Statistics	✓	✓ (subset)	✓ (subset)
DCP	✓	✓	✓
Cross datacenter replication (XDCR)	✓	✓	✓
Conflict resolution	✓	✗	✗
Backup	✓	✓	✓
Restore	✓	✓	✓
TTL	✓	✓	✓
Create/Drop	✓	✓	✓
Flush	✗	✗	✓
Index (GSI or FTS)	✗	✗	✓
Eventing function	✗	✗	✓
Analytics dataset	✗	✗	✓

Currently, there are some operations only performed at the bucket level and similarly some only at the collection level, such as flush, indexes, eventing functions, and the analytics dataset.

For Couchbase Autonomous Operator to meet the goals/objectives of fully auto-managing one or more Couchbase deployments it must be scopes and collection aware.



USE CASES



Write once, run anywhere

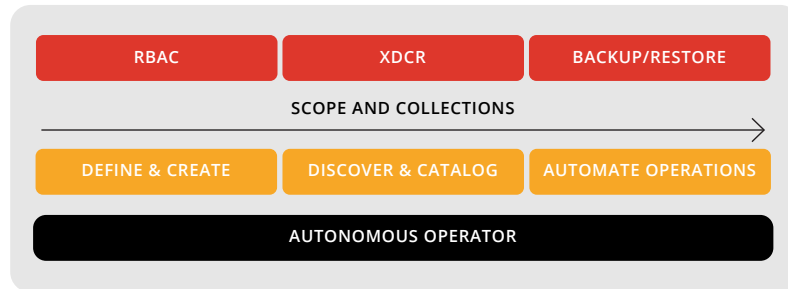
In inherently complex and rigid enterprise IT environments, breaking down barriers spurs true scalability and innovation. The ability to write once and deploy anywhere lets you run your Couchbase cluster, unmodified, on existing on-prem or private cloud (dev, test, pre-prod, prod) or in the public cloud.

For this use case, Couchbase configurations, resource specification, topology, and database configuration all need to be defined “cataloged” and then can be deployed to any other environments or clouds.

Discover and automate

With a microservices architecture, new microservices are constantly getting developed and deployed. Developers can choose to create new scopes and collections programmatically to support their new microservices. The expectation, in this case, is that new scopes and collections can get created dynamically, but once created, they need to be managed autonomously.

SCOPES AND COLLECTIONS LIFECYCLE WITH OPERATOR



The scopes and collections lifecycle with Couchbase Autonomous Operator involves three areas:

- 1. Define and create** – Users should define the scopes and collections they want to deploy in the Couchbase cluster. The CAO using the definition should be able to create those scopes and collections appropriately.
- 2. Discover and catalog** – Developers should be able to create scopes and collections dynamically in the Couchbase cluster. CAO should be able to discover the newly created scopes and collections and also catalog them.
- 3. Automate operations** – Whenever new scopes and collections are created, CAO should be able to automatically manage them unless the user explicitly disables them.





Modern customer experiences need a flexible database platform that can power applications spanning from cloud to edge and everything in between. Couchbase's mission is to simplify how developers and architects develop, deploy and consume modern applications wherever they are. We have reimagined the database with our fast, flexible and affordable cloud database platform Capella, allowing organizations to quickly build applications that deliver premium experiences to their customers—all with best-in-class price performance. More than 30% of the Fortune 100 trust Couchbase to power their modern applications.

For more information, visit www.couchbase.com and follow us on Twitter.

© 2023 Couchbase. All rights reserved.

