



Couchbase

What's New in Couchbase 7.x

Modern Database for Enterprise Applications

Agenda

- 01/** Couchbase 7 for Developers
- 02/** Scopes and Collections
- 03/** Trusted SQL Transactions
- 04/** Faster Performance & Lower TCO
- 05/** High Data Density
- 06/** High Data Density
- 07/** High Data Density
- 08/** Analytics
- 09/** Query and Index
- 10/** High Availability
- 11/** Backup & Restore

1 Couchbase 7 For Developers

What Does Couchbase Server 7 Do For Developers?



With Couchbase there is no compromise with application architectures and user experiences from first device touch through to trusted transactions.

Expands into transactional applications dominated by trusted RDBMS's

- SQL based Distributed ACID transaction from N1QL query language plus application SDKs
- User-defined functions, Cost based Optimizer for JSON, new analytic and eventing functions
- Supports enterprise goals of 100% uptime, including autoscaling, & new Backup Service

Simplifies applications architectures by pervasively adding new dynamic data containment structures

- A familiar, multi-level logical data structure, *Scopes & Collections*, that maps easily to RDBMS schema
- Pervasive support across services, including RBAC Security, HA clustering, auto sharding & replication
- Simplified multitenant design that allows ISVs to balance TCO and data isolation.

Delivers faster operational performance and lowers TCO

- Collection-level data access, partitioning and index granularity to increase density & lower costs
- Parallel & portable index building & building of up to 10,000 indexes per cluster
- Cost-based optimizer a first for JSON database



Couchbase Server 7 New Features by Service

Query Service

- N1QL Collections
- N1QL Transactions (distributed multi-document ACID)
- *Memory Quotas*
- User Defined Functions
- Index Advisor Functions
- *Cost-Based Optimizer*

Indexing Service

- GSI at Collection-level
- 10k indexes per cluster
- Auto-Move Index during partitioning
- *Concurrent Index Creation*

*New Backup Service

- New service works w/MDS
- Backup & Restore Collections
- Schedule Backups in GUI
- Re-map & filter Restoration

Full Text Search

- FTS for Scopes, Collections & Multiple Collections
- FTS Quick Index
- *FTS performance & TCO improvements*

Analytics

- Analytics Collections
- User-Defined Functions
- Rollup function
- CUBE function
- Python ML models
- External Dataset: Azure Blob

Eventing

- Eventing Collections
- Operational Improvements
- Eventing Security
- *Performance Improvements*

2 Dynamic Data Containment Model Scopes & Collections

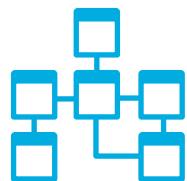
New in 7: Scopes & Collections

Multi-level dynamic data container model that maps clearly to RDBMS schema models, pervasively incorporated across Couchbase security, query, data access and replication services.



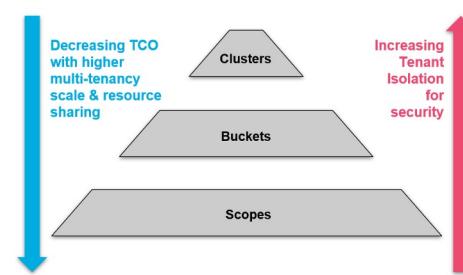
- Buckets
- Scopes
 - Collections
 - Documents

Collections and Scopes are **dynamic, logical containers within a bucket**.

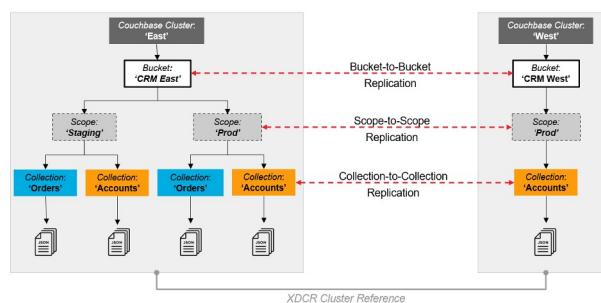


- | | |
|-------------|------------|
| Buckets | = Database |
| Scopes | = Schema |
| Collections | = Table |
| Documents | = Rows |

Developers use them to **organize data, map RDBMS models and isolate schemas**.

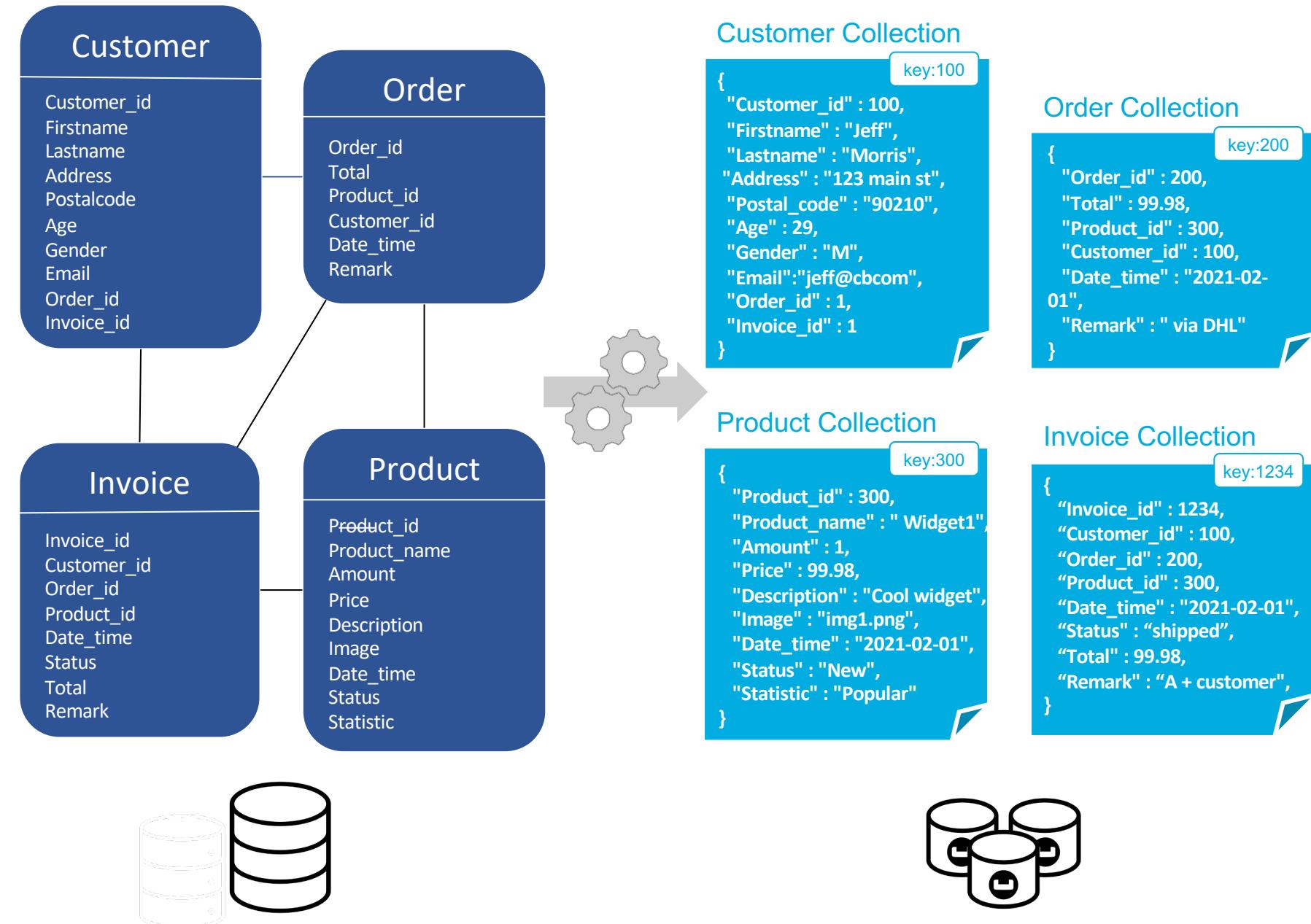


Administrators use them to **consolidate tenants, and control access or replication**.



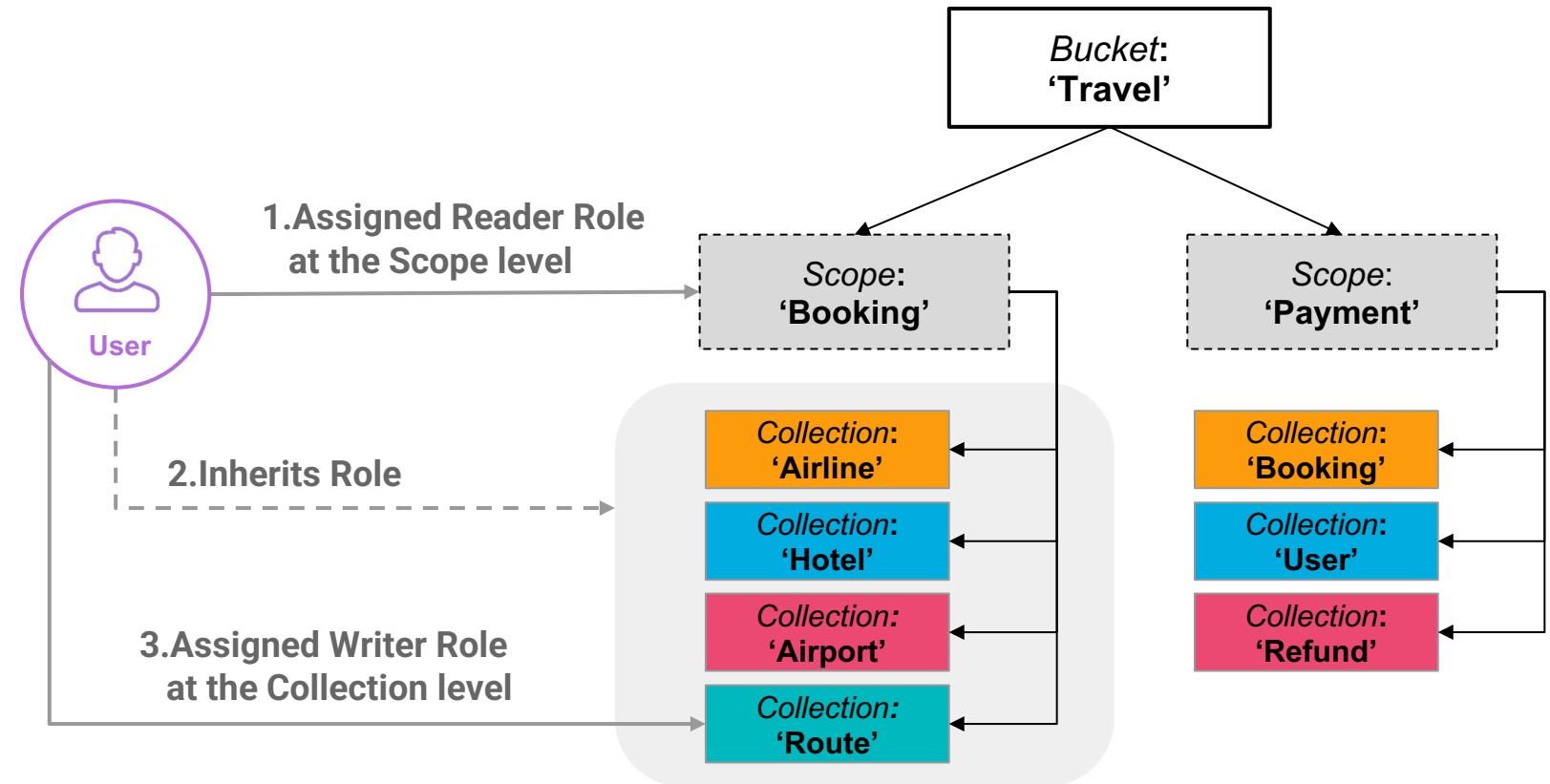
Architects design for **replication, microservices delivery, & enjoy high-performance indexes**, granular security and improved resource control.

Dynamic, Flexible Data Organization



Role Based Access Control

- Out-of-box **roles at multiple levels**: Collection, Scope, Bucket
- **Hierarchical**: Roles assigned at a higher level are percolated to the lower levels

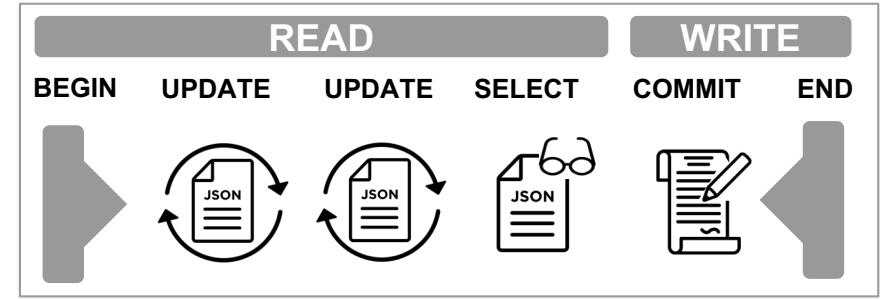


3 Trusted SQL Transactions

N1QL Support for Multidocument ACID Transactions

Ensures database consistency when data modifications are performed with multiple documents in a single or multiple N1QL statements

- Ensure data integrity for application transaction that consists of multiple operations where multiple concurrent activities can occur
- Support all or nothing for multi documents N1QL DML operations
- Statement Level Atomicity
- Each transaction supports multiple document updates via N1QL DML on multiple collections, scopes and buckets
- No central coordinator - Performance overhead is isolated at query node level



```
BEGIN WORK;  
UPDATE x1 SET a = a + 1 WHERE b < 10;  
UPDATE x1 SET a = a + 15 WHERE b < 10;  
SELECT a, b, c FROM x1 WHERE b < 20;  
COMMIT WORK;
```

N1QL support for Transactions

Comparison against RDBMS

RDBMS

```
START TRANSACTION;  
UPDATE customer SET balance = balance + 100  
WHERE cid = 4872;  
SELECT cid, name, balance from customer;  
UPDATE customer SET balance = balance - 100  
WHERE cid = 1924;  
SELECT cid, name, balance FROM customer;  
COMMIT ;
```

COUCHBASE 7.0

```
START TRANSACTION;  
UPDATE customer SET balance = balance + 100  
WHERE cid = 4872;  
SELECT cid, name, balance from customer;  
UPDATE customer SET balance = balance - 100  
WHERE cid = 1924;  
SELECT cid, name, balance FROM customer;  
COMMIT
```

```
START TRANSACTION;  
UPDATE customer SET balance = balance + 100  
WHERE cid = 4872;  
SELECT cid, name, balance FROM customer;  
SAVEPOINT s1;  
UPDATE customer SET balance = balance - 100  
WHERE cid = 1924;  
SELECT cid, name, balance FROM customer;  
ROLLBACK WORK TO SAVEPOINT s1;  
SELECT cid, name, balance FROM customer;  
COMMIT ;
```

```
START TRANSACTION;  
UPDATE customer SET balance = balance + 100  
WHERE cid = 4872;  
SELECT cid, name, balance FROM customer;  
SAVEPOINT s1;  
UPDATE customer SET balance = balance - 100  
WHERE cid = 1924;  
SELECT cid, name, balance FROM customer;  
ROLLBACK WORK TO SAVEPOINT s1;  
SELECT cid, name, balance FROM customer;  
COMMIT ;
```



Couchbase 7 as Dependable as Traditional RDBMS

Transactionality

- Distributed ACID transactions accessible from both N1QL and SDKs
- Multi-document and multi-node guarantees
- Scales linearly

Granularity

- Granularity pushed from Bucket to Collections for all Couchbase Services.
- 30 buckets, 1,000 collections & scopes, 10k indexes per cluster
- Improves service efficiency and performance
- Tightens security
- Improves programmability

100% Uptime

- Automatic, scheduled backups
- Auto-sharding & data rebalancing
- In-place upgrading
- Multidimensional scaling
- Linear scaling of indexes, transactions, rebalancing, failover results in dense clusters driving low TCO

Programmability

- SQL Skills
- User-defined functions
- Eventing & CDC
- 10 SDKs & 12 frameworks
- Kafka, Spark, Elasticsearch & CData connectors
- Cloud-native Kubernetes ecosystem integrations
- Microservices-friendly
- Mobile-friendly

Enterprise Security

- Scope and Collection-level RBAC

- Field-level Encryption in SDKs

- In-transit TLS

4 Faster Performance & Lower TCO

Performance Improvements

Couchbase Server 7 offers faster operational, querying, indexing and transactional performances

Failover: 4x faster

- 30 bucket, 3M documents, failover time is 400% faster than Couchbase 6.6.1
- From 26 sec, to 6 sec.

Rebalance: reduced hours to minutes

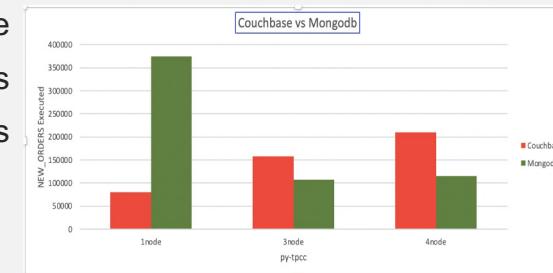
- 3 node -> 4 node, 3M documents, rebalance time is linear and 40% to 2000% faster as buckets increase
- Up to 50 buckets test improved from 2:45 hrs to 7.5 mins

Index Builds: constant

- Initial and incremental build times are flat and indistinguishable between 100 to 1000 indexes
- 6.6.1 index build times rise exponentially as # of indexes increase, while CB7 are constant

py-tpcc benchmark of N1QL transactions scales linearly vs. MongoDB 4.2

- MongoDB 4x faster on single node
- Couchbase 50% faster on 3 nodes
- Couchbase 80% faster on 4 nodes
- MongoDB range sharding does not scale as nodes increase



py-tpcc benchmark of N1QL Transactions: 1, 2, 4, 6 nodes

- Couchbase N1QL transactions scales linearly as nodes increase

5 High Data Density



Magma | Motivations

Couchstore

The Data Storage Engine pre 7.1

- Couchstore is designed for a High Performance **Memory-Oriented** database.
- Recommendations to store your data in memory

Motivations for a New Data Storage Engine

- Customers having larger and **larger amounts of data** and having larger amount of memory becomes very expensive
- Not all use cases require memory speed and disks are becoming faster



Magma

The New Data Storage Engine in 7.1

- Magma is designed for a High Performance **Disk-Oriented** database.
- Magma solves problems of
 - High Data Density
 - Disk access performance
 - High-end hardware



Magma | New Data Storage Engine

Magma enables Couchbase to become a high performance disk-oriented database for large volumes of data

High Data Density at Low TCO

Support 1% memory to data ratio

- 10 GB of memory to store 1 TB of data on disk
- Maximum of 10 TB data per node in Couchbase Server 7.1

High Disk Access Performance

4x throughput improvement

- High sustained write throughput
- Better disk read latency

Scale Up with High-End Hardware

Scale with NVMe SSDs

- Scale better with higher disk bandwidth for write heavy workloads
- Lower read latency with NVMe drives



Magma | TCO Reduction for High Data Density

10x Fewer Servers Required, 4x Faster (higher throughput) for disk access

Example: 100 TB data, with Couchbase Servers deployed on m5d.8xlarge (128GB RAM) with storage on EBS

Pre Couchbase 7.1



1TB per node

min 10% memory ratio



100

of servers required



\$3 million per year

Annual server cost

Couchbase 7.1 (Magma)



10 TB per node

min 1% memory ratio



10

of servers required



\$300K per year

Annual server cost



Magma | Numbers at a glance

How Magma improves upon Couchstore

	Couchstore	Magma	Improvement
Minimum Memory to Data Ratio	10%	1%	10x
Maximum Data size per node	3TB	10TB	3.3x
Throughput <i>(Sustainable for mixed YCSB workloads)</i>	32K ops/sec on 4 node cluster with NVMe SSD	158K ops/sec on 4 node cluster with NVMe SSD	4x
Tail Read Latency <i>(99.9 percentile for disk heavy workload)</i>	< 300ms on regular SSD	< 10ms on regular SSD	>10x
Durable Writes <i>(Persist Majority)</i>	Throughput: 10K ops/sec Latency: 4.03ms	Throughput: 26K ops/sec Latency: 2.57ms	2.5x 2x
Compression on disk	30% for 1KB documents	50% for 1KB documents	20%
Space amplification <i>(50% fragmentation for write heavy workloads)</i>	3x of compressed data	2x of compressed data	33%



Magma | General Considerations

Adjust memory residency and data density per node to meet the SLA requirements

In-memory performance

- Accessing data from memory will still be faster.
- Accordingly adjust memory residency of dataset (minimum 1%) to meet SLAs.

Amount of data on each node

- Rebalance time is proportional to amount of data on a node. Rebalance of 1TB is approximately 2 hours.
- Choose data density per node as per acceptable SLA for rebalance.



Magma or Couchstore?

Choosing Magma or staying with Couchstore depends on the use case

Use Cases for Couchstore

- **Caching** Use Cases requiring memory speed with close to 100% working set
- **2-speed** use cases with more than 20% working set
- Development and use cases on **low-end hardware**
- Use cases with a **large number of buckets** on low-end hardware

Use Cases for Magma

- **1-speed DGM** (Data Greater than Memory) use cases
 - Disk speed is considered OK
- **High Data Density** use cases
 - Main goal from customers is Low TCO with memory < 1% of data size
 - Disk speed is considered OK
- **Transactional** use cases

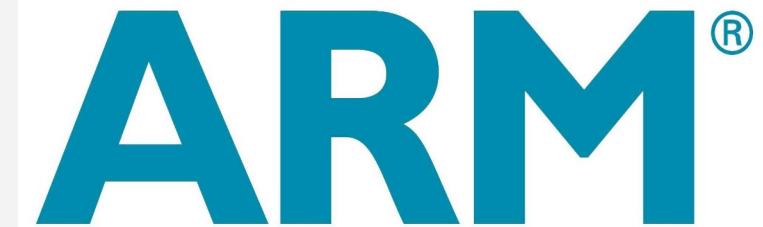
6 Platform



Support for ARM

Couchbase Server 7.1 adds support for clusters running Linux on ARM v8 processors

- ARMv8 processors support **boosts performance and reduces** power consumption over Intel X86 processors
- ARMv8 processors are gaining in **popularity** among Apple and major Cloud Service Providers
 - e.g. AWS Graviton 2 and Graviton 3 EC2 instances
- ARM is supported when running Couchbase Server on Amazon Linux 2
- Some use-cases may **improve to 2X ROI** by moving to ARM based cloud instances

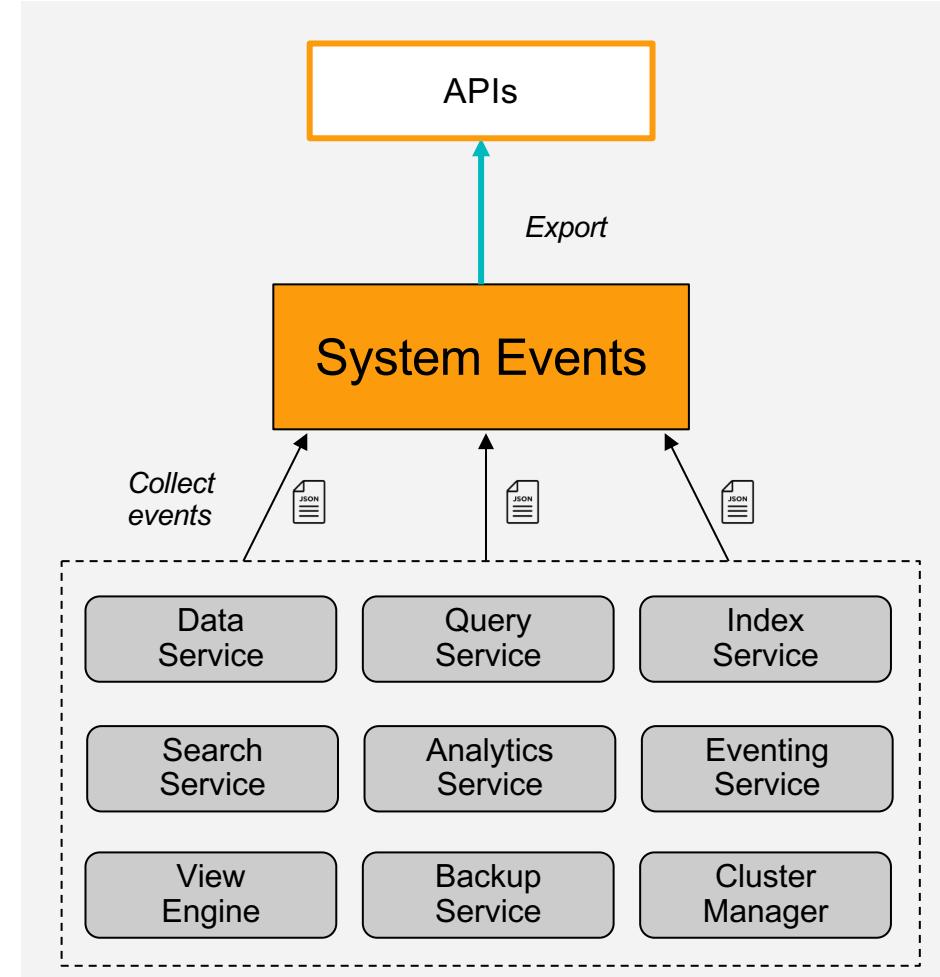




System Events

Provides standard JSON format data and unlike the existing logs it is designed for customers

- Each **log entry** has a timestamp, UUID, component, description and an event ID code
- Designed to be **low volume** with key information
- Single point to **monitor across all services** on all nodes
- **Enabled** by default and **included** in the cbcollect zip
- The last 10k events served via a timestamp constrained **REST API** or a **streaming API** to monitor cluster activity





System Events | Comparison with CBCollect and Audit Logs

Couchbase logs and tools provide informations tailored to different purposes

- **System Events** exposes standard JSON format data for customers to monitor clusters
- **CBCollect Logs** are intended to be uploaded to Couchbase Support for detailed diagnostics and analysis
- **Audit Logs** allow admins to track administrative and configuration change as well as attempts to access and change data

Feature	CBCollect Logs	Audit Log	System Event Log
Structured Std Format	X	✓	✓
Defined Id Codes	X	✓	✓
Intended for Customer Usage	X	✓	✓
Detailed Diagnostics and Statistics	✓	X	X
Uploaded to Couchbase	✓	X	✓
Redaction	✓	X	X
Enabled by Default	✓	X	✓
Contains Records of Individual Doc/Query Access	X	✓	X
Accessible via API	X	X	✓



OS Changes in Couchbase Server 7.1.0

Couchbase is updating the OS Platforms supported

- Couchbase Server is supported on all **3 major desktop and server platforms**: Linux, Windows and MacOS
- Couchbase supports the **latest releases** of each platform, **plus one** previous release
- When a platform is no longer **supported by its vendor**, Couchbase also removes support
- Couchbase Server is supported on Linux across a **wide range of distributions**: RedHat, CentOS, Amazon Linux, Oracle Linux, Suse, Debian, Ubuntu

OS Distribution	Support in Couchbase Server 7.1.0
Debian 11	Added
Microsoft Windows Server 2022	Added
MacOS 12 Monterey	Added
RHEL 7	Deprecated
CentOS 7	Deprecated
Oracle Linux 7	Deprecated
Ubuntu 18	Deprecated
MacOS 10.15 Catalina	Deprecated
Debian 9	Removed
CentOS 8	Removed
MacOS 10.14 Mojave	Removed
Microsoft Windows Server 2016 (inc. DC Edition)	Removed

7

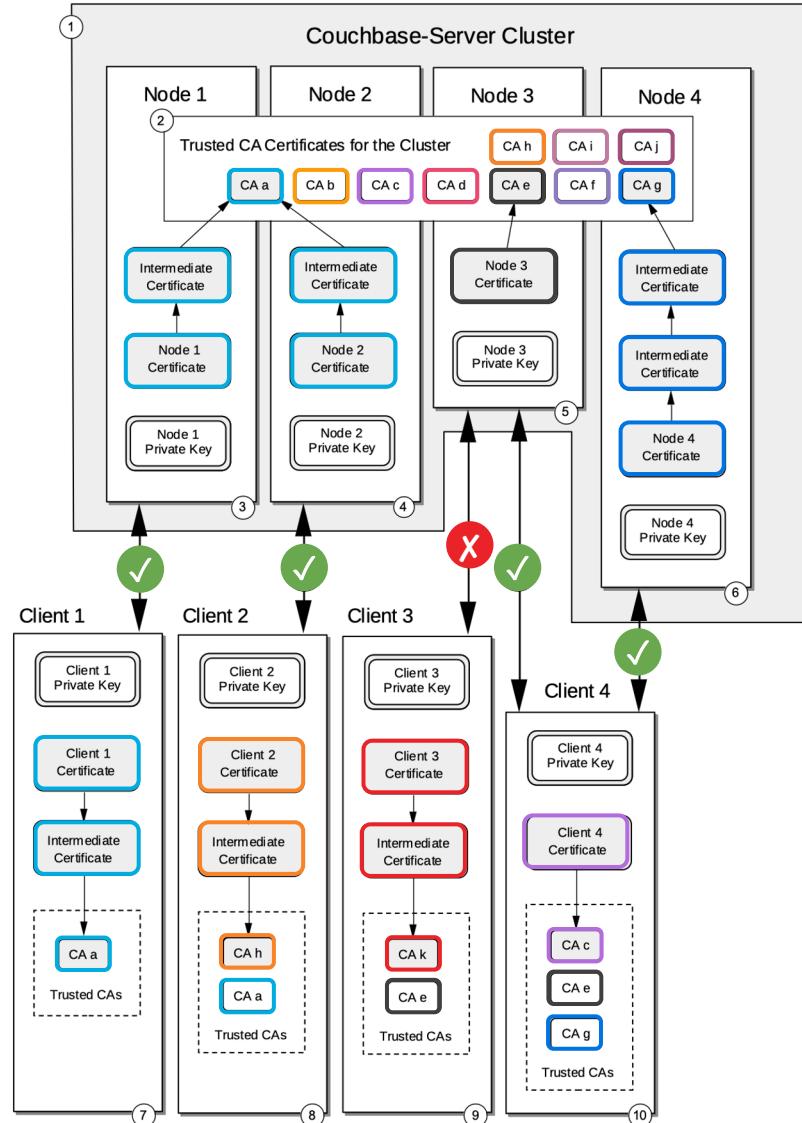
Security



Multiple Certificate Authorities

Flexibility of Multiple Certificate Authorities

- Certificates are **used by many components** such as Couchbase nodes and clients authenticating with X.509 certificates
- Couchbase Server 7.1 now supports multiple Certificate Authorities (CA)
- **100% uptime** while changing CA certificates
- **Allows for different certificate chains**
 - One or more for App -> Cluster network encryption
 - Another for client authentication

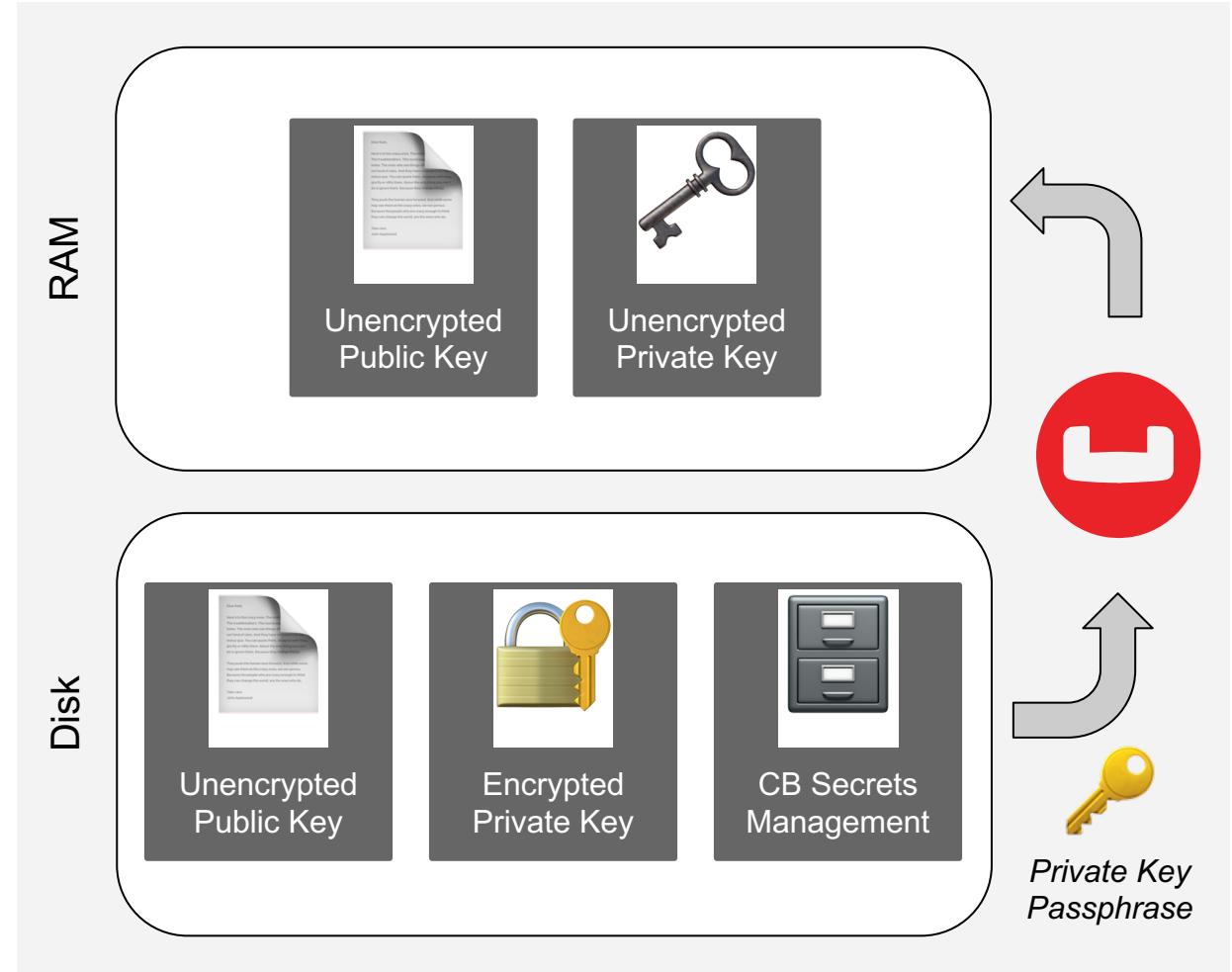




Encrypted TLS Private Keys

Prevents storing a TLS Private Key unencrypted

- Private keys are used by TLS to encrypt network communications and **must be kept safe** to prevent attacks
- Couchbase now have Encrypted Private keys **stored on encrypted files**
- The **Private Key Passphrase** can be provided in different ways
 - Stored internally with Couchbase Secrets Management (encrypted)
 - Or provided externally by a REST Call
 - Or by executing a customer provided script

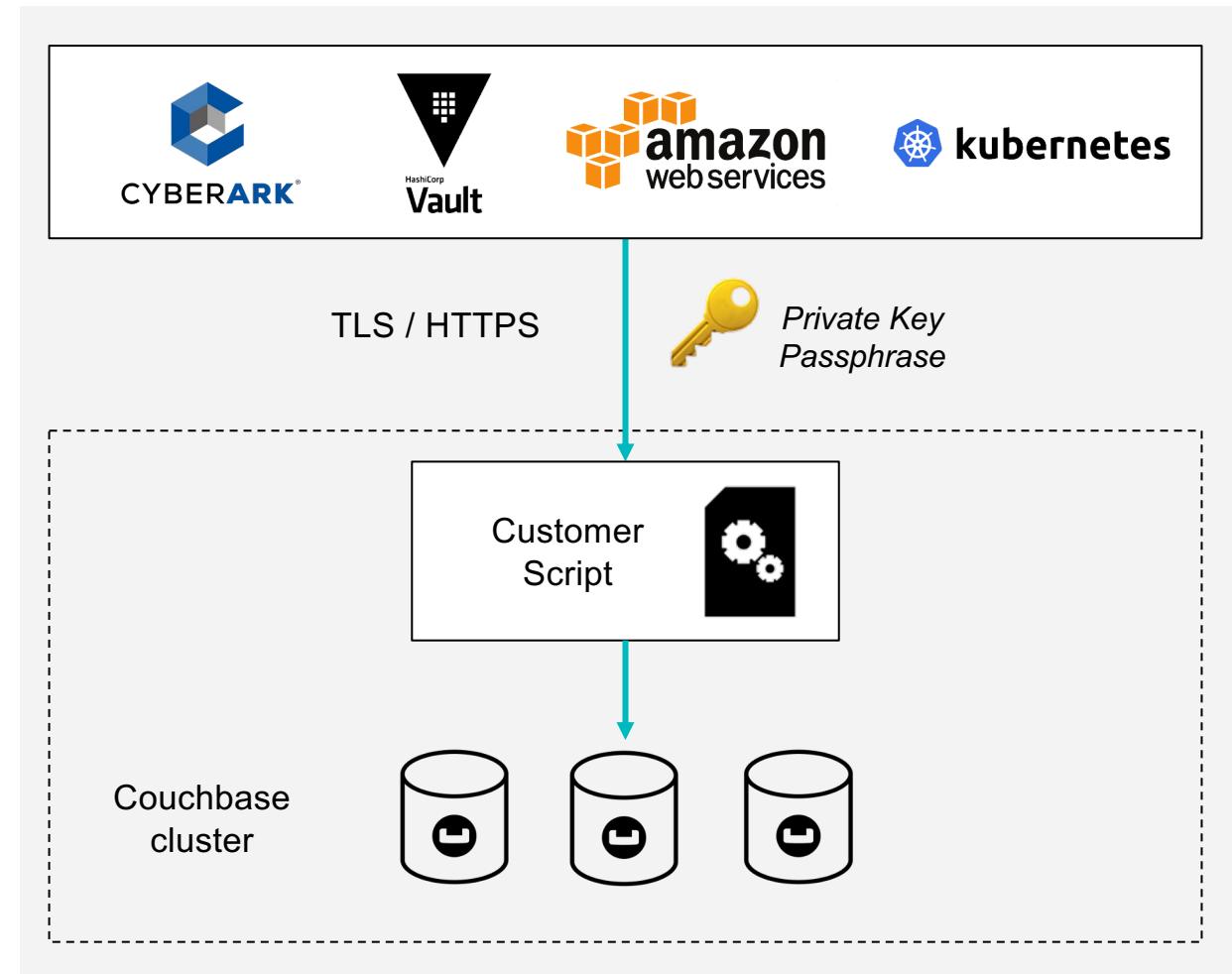




Encrypted TLS Private Keys | KMS

Many sources for the passphrase can be used

- **Secrets Managers**
 - CyberARK
 - Hashicorp Vault
- **Cloud Backends**
 - AWS Secrets Management
 - Azure Key Vault
- **Infrastructure Keystores**
 - Kubernetes Secrets
 - Docker Secrets
- **Hardware Solutions**
 - Trusted Platform (TPM)
 - Hardware Security Module (HSM)





PKCS #8 and PKCS #12 Certificate Formats

Reduces the need to convert certificates before loading them

- Public-Key Cryptography Standards (PKCS) are a set of standard protocols to enable secure information exchange
- Prior versions of Couchbase Server support only the PKCS #1 type certificates
- Couchbase Server 7.1.0 adds support for the **PKCS #8** format
- Couchbase Server 7.1.0 adds **beta** support for **PKCS #12** format

PKCS formats	Description
PKCS #1	<ul style="list-style-type: none">• RSA Cryptography Standard• Public and private keys
PKCS #8	<ul style="list-style-type: none">• Private-Key Information Syntax Standard• Encrypted or unencrypted private keys
PKCS #12	<ul style="list-style-type: none">• Personal Information Exchange Syntax Standard• Container format that can contain multiple embedded objects, such as multiple certificates



TLS 1.3

TLS 1.3 can now be set as the minimum cluster-wide level of TLS

- TLS 1.3 includes more secure ciphers for encryption and decryption and is faster than previous versions of TLS
- TLS 1.3 was introduced in Couchbase Server 7.0 but it was an incomplete solution, so the cluster-wide minimum TLS setting could not be set to require TLS 1.3
- All Couchbase services components have been upgraded so **the entire Couchbase Server 7.1.0 cluster can now use TLS 1.3**

TLS v1.3





RBAC for Sync Gateway

SGW Users and Roles managed from the Couchbase Server 7.1 Cluster

- RBAC for Sync Gateway allows to grant **specific roles** permissions to manage and access the SGW database, users, metrics, replications and app data
- 5 New RBAC roles for Sync Gateway
- Simply **add users into Couchbase Server** with appropriate SGW roles: permissions are propagated into SGW

Roles	Description
Sync Gateway Architect	<ul style="list-style-type: none">• Can manage Sync Gateway databases and users, and access Sync Gateway's /metrics endpoint.• This user cannot read application data.
Sync Gateway Application	<ul style="list-style-type: none">• Can manage Sync Gateway users and roles, and read and write application data through Sync Gateway.
Sync Gateway Application Read Only	<ul style="list-style-type: none">• Can manage Inter-Sync Gateway Replications.• This user cannot read application data.
Sync Gateway Replicator	<ul style="list-style-type: none">• Can manage Inter-Sync Gateway Replications.• This user cannot read application data.
Sync Gateway Dev Ops	<ul style="list-style-type: none">• Can manage Sync Gateway node-level configuration, and access Sync Gateway's /metrics endpoint for Prometheus integration.



Eventing Service Security Enhancements

Full RBAC in Eventing roles and tunable settings for cURL

- Admin Privileges for Eventing are no longer needed:** Eventing Functions can now run as normal users via the standard RBAC security model.
- Bucket Scope:** all Eventing functions are tagged with a “bucket.scope” identifier
- Seamless upgrade for Eventing admin roles:** all existing 7.0 functions upgraded and imported into 7.1 with a default “Bucket Scope” of “*.*”
- Privileges:** Users other than “Full admin” or “Eventing admin” require at least Data DCP Reader privileges
- To move a function** to a different “Bucket Scope” it must be exported, deleted and reimported.
- cURL** can now be disabled or enabled for Eventing functions, cluster-wide.

The screenshot shows two interface components related to Eventing security. The top component, titled "Function Settings", displays a "Function Scope bucket.scope" field containing the value "*". Below this is a section titled "Bucket Scope tags for Eventing functions". The bottom component, titled "Add New User", is a form for creating a new user account. It includes fields for "Username" (payroll), "Full Name (optional)", "Password", and "Verify Password". On the right side of the "Add New User" form is a "Roles" tab, which is currently selected, showing the "Data DCP Reader" role assigned to the "bucket... scope..." scope. Other available roles listed include Data Monitor, Views, Query & Index, Search, Analytics, Eventing, and Manage Scope Functions. The "Eventing" role is also listed under the "Manage Scope Functions" section. A "Groups" tab is also present but appears to be empty.

Function Settings

Function Scope bucket.scope

Bucket Scope tags for Eventing functions

Add New User

Username: payroll

Full Name (optional):

Password:

Verify Password:

Roles

- Data DCP Reader
- Data Monitor
- Views
- Query & Index
- Search
- Analytics
- Eventing
- Manage Scope Functions

Groups

Eventing

Adding users with Eventing privileges



Notable Changes

Notes for Couchbase Administrators on functionality changes

REST API to load a CA Certificate

- The `/controller/uploadClusterCA` endpoint has been deprecated.
- Use the new `/node/controller/loadTrustedCAs` endpoint instead.

Certificate Authority settings restrictions

- We are restricting the ability to change the Certificate Authority settings to security administrators on a localhost of a node in the existing cluster.

Adding a node for clusters using customer managed certificates

- Incoming nodes to a Couchbase Server 7.1.0 cluster must have a trusted TLS certificate present before they are added.
- This is a mandatory requirement in clusters configured to use TLS with customer managed certificates.

8 Analytics



Couchbase Analytics in 7.1: One Stop Shop for JSON Analytics

Couchbase Analysis provides Real time insights on Real time data

Optimized Analytics Engine

High Availability Support

- Always available real-time data
- Query and analyze information without disruption

Extended Data Visualization

Native connectivity to Tableau

- Analyze JSON, CSV, TSV, Parquet (DP) data together and seamlessly
- Uses tabular views, provide schemas and supply relationship information

Insights from Additional Data Lakes

Extend Analytics reach

- GA : AWS S3 and Azure Blob Storage
- DP : Azure Data Lake Gen 2, Google Cloud Storage and Parquet file format on AWS S3



Array Index Support for Analytics Service

Improves query performance with faster search for array fields

- Allow for the **creation of indexes on arrays** of primitive values and field values within arrays of objects
- Support index-acceleration of various **queries involving arrays**
 - Same optimizable predicates as for existing Analytics indexes
 - Both existential (ANY) and universal (EVERY) predicates are supported

```
1 SELECT h.name, h.reviews
2 FROM hotel h
3 WHERE h.city = "Los Angeles"
4 AND (ANY r in h.reviews
5      SATISFIES r.ratings.Overall > 3);|
```

Example of Query involving Array

```
1 CREATE INDEX hotelCityRatingIdx ON hotel (
2   city: string,
3   UNNEST reviews
4   SELECT ratings.Overall: bigint
5 ) EXCLUDE UNKNOWN KEY;|
```

Example of creation of Indexes on Array

9

Query and Index



Motivations and Benefits

SQL++ improvements in overall performances, extensibility and manageability.

Query Service Enhancements

Improves Query Performances and Resiliency

- Enhanced CBO to choose the most efficient JOIN order
- Composite index on array fields to improve query performance
- Query service graceful shutdown when removed

User Defined Functions with Javascript

Simplify migration from RDBMS stored procedures

- Developers can create custom functions with javascript
- New UI and support for embedded SQL++ statements
- Provides a comprehensive development framework

Index Service Enhancements

Improves Index Resiliency, TCO and Performances

- The Index service now supports automatic failover
- Memory compression to lower Resident Ratio of indexes
- Faster index build and rebalance



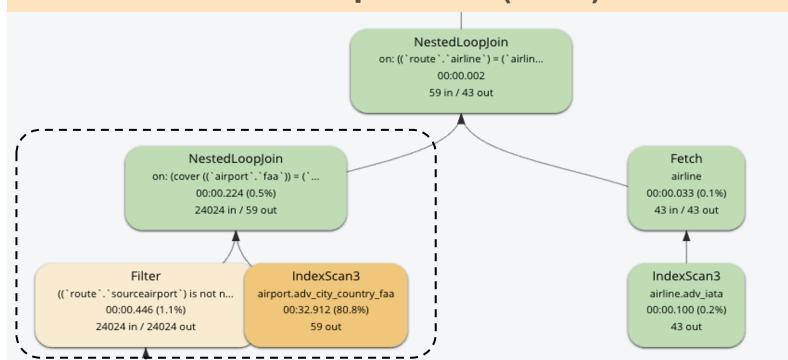
Cost Based Optimization (CBO) - JOIN Order

Enhance SQL++ CBO to choose the most efficient JOIN order

- CBO will choose the most efficient JOIN order based on the execution costs
- Remove the need for SQL developer to conscious decision on the JOIN order
- JOIN order can change depending on the data shape

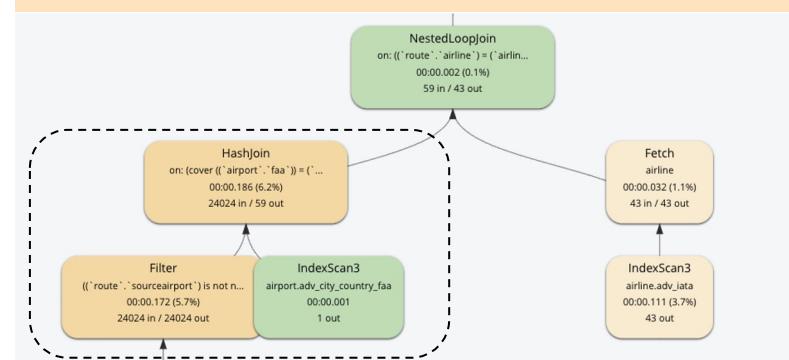
```
SELECT DISTINCT airline.name
FROM route
JOIN airport ON airport.faa = route.sourceairport
JOIN airline ON route.airline = airline.iata
WHERE airport.city = "San Jose"
AND airport.country = "United States";
```

Rule-Based Optimizer (RBO) - 25s



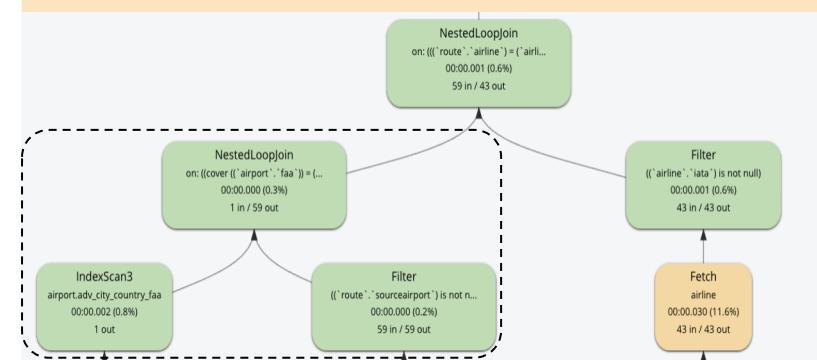
Loops on **24000** routes and for each route, scans the airports

CBO without JOIN order - 2.3s



Hash Join is faster than Nested Loop

CBO with JOIN order - 130ms



Reverses the join to loop first on **2000** airports then for each airport, scans the routes



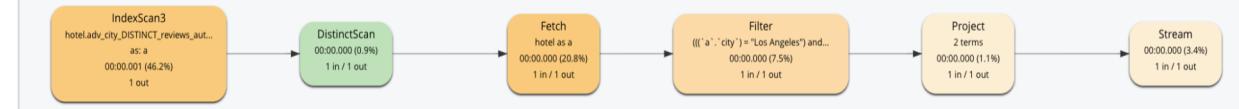
Array Flatten Keys - Composite Index on array fields

Composite index improves query performance when searching multiple fields in an array

- Improve query performance by using **composite index** for queries that have multiple search conditions on array fields
- Existing queries can benefit from the new index **without modifying the queries**
- Query execution can use **covering indexes** without the need to fetch from data service
- Performance improves when **sorting and pagination** are required on the index fields

```
SELECT a.name, a.reviews  
FROM hotel a  
WHERE a.city = 'Los Angeles'  
    AND ANY r IN a.reviews SATISFIES r.author LIKE  
        'Marie%'  
    AND r.ratings.Overall > 3 END;
```

```
CREATE INDEX ix1  
ON `travel-sample`.inventory.hotel (  
    city,  
    (distinct ( ARRAY_FLATTEN_KEYS(  
        r.author,  
        r.ratings.Overall) for r in reviews  
    end),  
    reviews  
)
```



Removes the need to define multiple array indexes



Query - Graceful Shutdown

Query service will gracefully shutdown when query node is removed from the Cluster

cluster1 > Servers

Dashboard Servers Buckets Backup XDCR Security Settings Logs Documents Query Indexes

filter servers... 172.23.120.208 Group 1 query index 2.8% 14.3% 0.0% --- 0/0 Statistics

Addresses: 172.23.120.208 (int)
Address Family: IPv4
Node-to-Node Encryption: off
OS: x86_64-pc-linux-gnu
Uptime: 47 days, 22 hours, 58 minutes, 17 seconds
Version: Enterprise Edition 7.1.0 build 2031
RAM Quotas: Index 512MiB
Index Storage Path: /opt/couchbase/var/lib/couchbase/data

Memory Disk Usage

index service used (182MiB) remaining (329MiB)

In flight queries **gracefully terminated** if node is removed

In flight queries **immediately terminated** if node is failed over

Remove Failover

- Allow in flight queries to complete before the service is **removed**
- The failover option will not be changed, i.e. the inflight queries will be immediately terminated.



User Defined Functions with JavaScript & SQL++

UDFs with JavaScript allows developers to provide custom functions to extend SQL++ capabilities

- **New UI** to create UDF with JavaScript libraries
- UDF supports **embedded SQL++ statements** to allow easy access and modify of the data
- Allows **easy migration from RDBMS** stored procedures
- UDF performance can **scale out** with the query nodes
- Developers uses JavaScript to **directly manipulate JSON documents**
- Provides a comprehensive development **framework** as UDF supports calling other UDFs, as well as recursion

The screenshot illustrates the process of creating and using a User Defined Function (UDF) in Couchbase's Query Editor.

Edit Library: A modal window titled "Edit Library" shows a code editor with the following JavaScript function:

```
1 /* a UDF library contains one or more javascript functions */
2 function getairportname(airportcode) {
3     var query = SELECT a.airportname FROM `travel-sample`.inventory.airport
4         a WHERE a.icao=$airportcode;
5     let acc = [];
6     for (const row of query) {
7         acc.push(row);
8     }
9
10    return(acc);
11 }
```

A callout bubble points to the "Create the function in javascript" button.

Edit Function: A modal window titled "Edit Function" shows the configuration for the "getname" function:

- Function Name: getname
- Namespace: global
- Parameters: ...
- Function Type: javascript
- Javascript Library: demodml
- Library Function Name: getairportname

A callout bubble points to the "Edit the function settings" button.

Query Editor: The main interface shows a query being executed:

```
1 select getname(a.icao) from `travel-sample`.inventory.airport a limit 1;
```

The results show a single document:

```
1 [
2   {
3     "$1": [
4       {
5         "airportname": "Calais Dunkerque"
6       }
7     ]
}
```

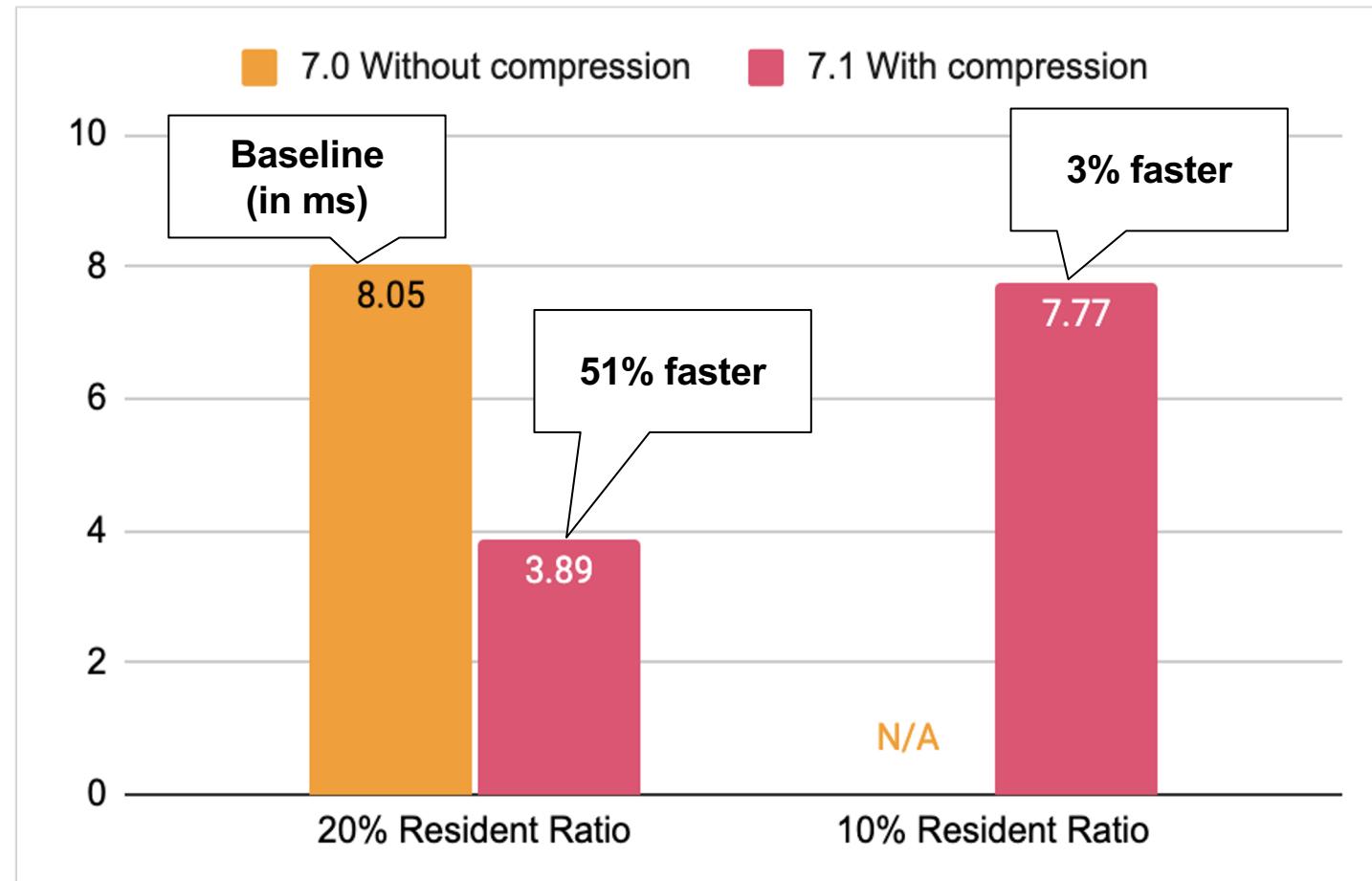
A callout bubble points to the "Use the function in SQL++ statements" button.



Index Service - Support Lower Resident Ratio

The Index service supports a lower resident ratio, which can increase index capacity per index node

- The Index service now supports **memory compression**
- This allows more indexes pages to be stored in memory
- This provides **faster index scans** hence faster query performance
- Index nodes can now use **50% less memory** to lower the TCO while still providing similar index performance

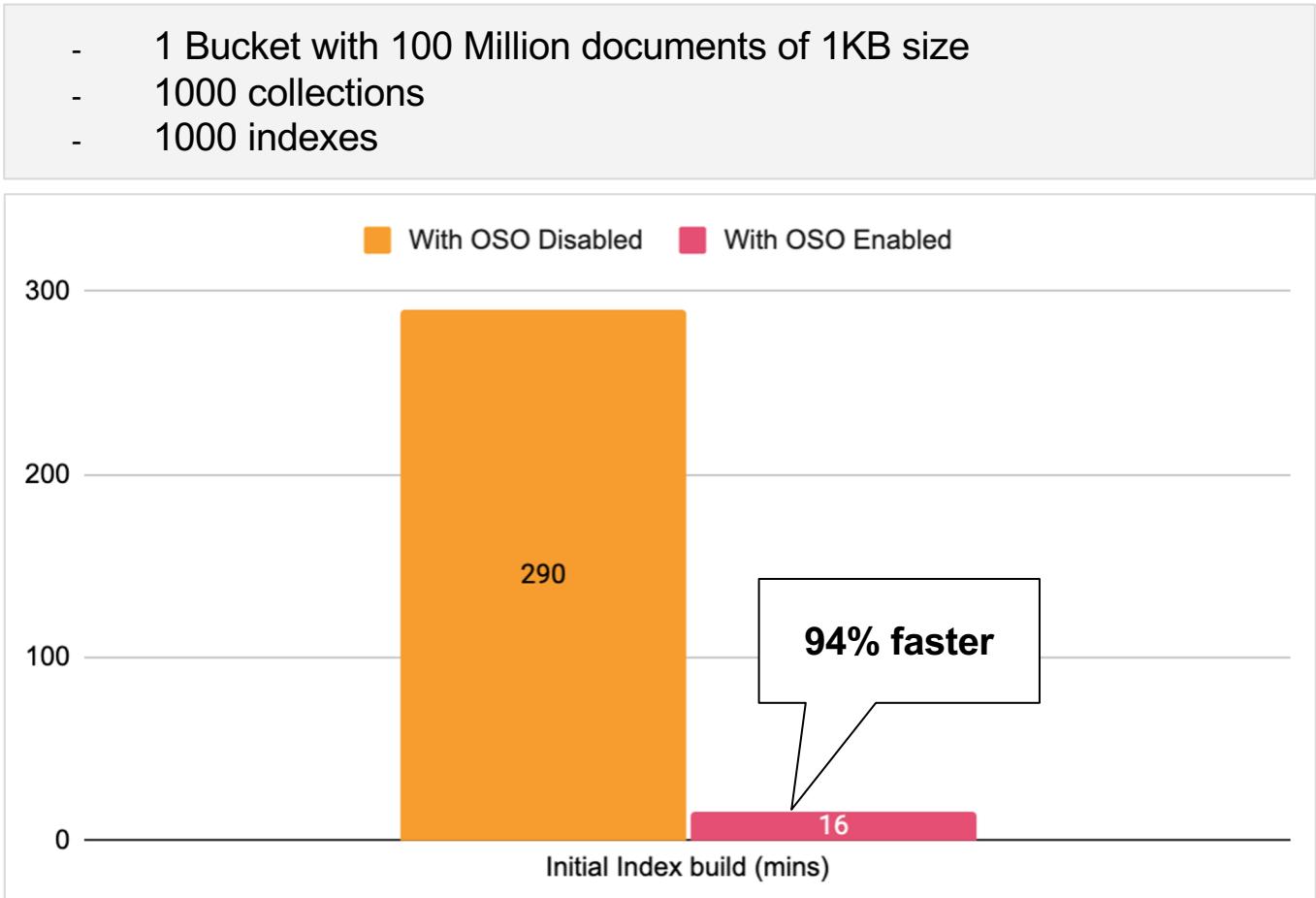




Faster Index Build with DCP Out-Of-Sequence Order

Index Build using DCP OSO resulted in significant performance improvements

- **Index Build** using DCP Streaming OSO is **based on collection**, not bucket mutations
- Available only for collections and when **requested from zero**
- This improvement will be seen in create/build index, as well as during **rebalance**
- Initial index build using OSO is the **default mode** in 7.1 - can be overridden

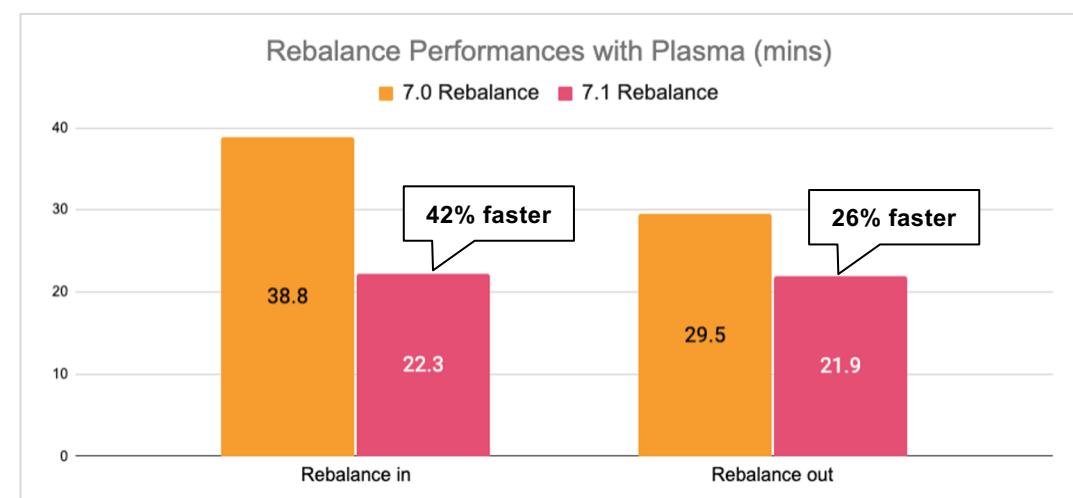
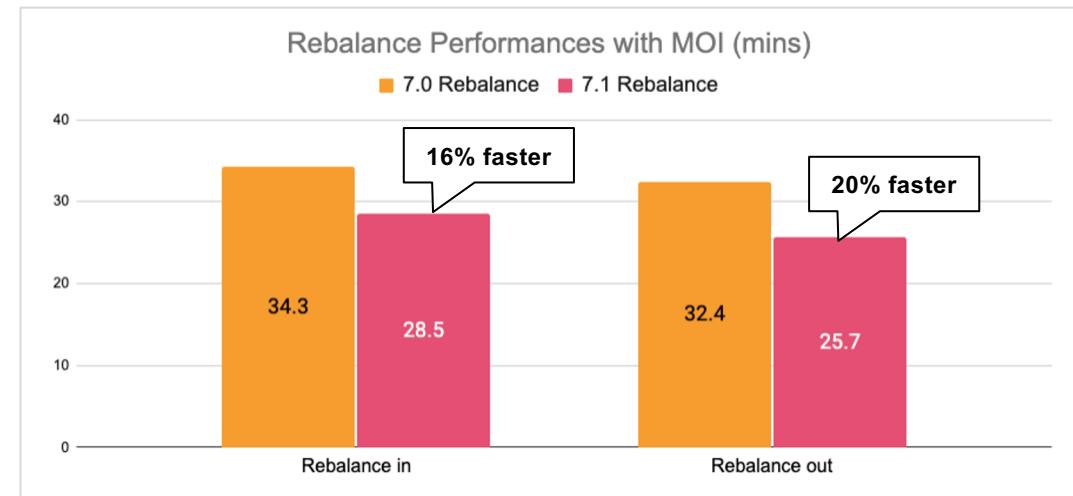




Smart Batching for Index Rebalance

Enhanced algorithm to improve Index Rebalance

- Phase 1: planning to optimize the **desired layout** of indexes across nodes to minimize costs
- Phase 2 : **indexes movement** during rebalance are executed in batches
- **Increases rebalance performance** by enhancing the pipeline parallelism
- **Reduces resource consumption** by optimizing the data streaming
- **Reduces the impact on production workload** by repairing replicas before moving indexes and by reducing concurrency on heavily loaded nodes



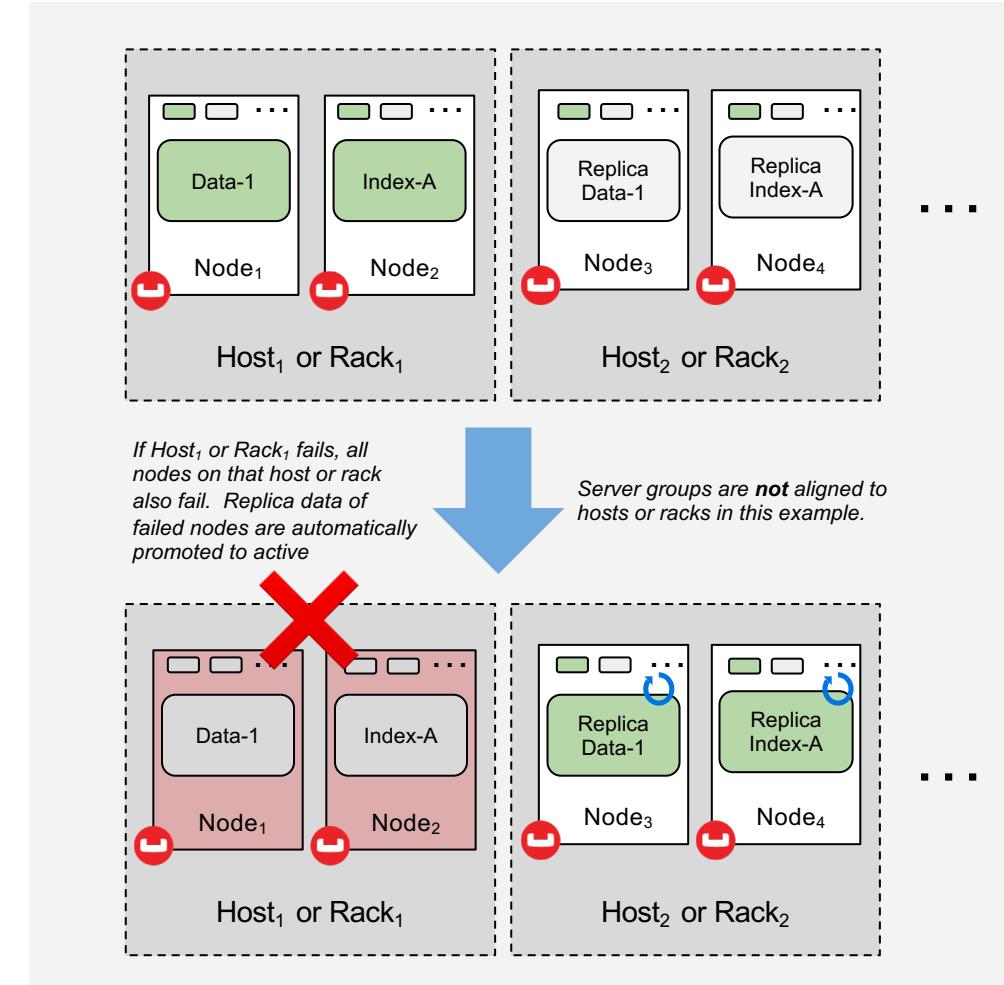
10 High Availability



Auto-failover of multiple nodes on concurrent failures

Improved High Availability without human intervention for correlated failures

- **Multiple nodes** can fail concurrently and will be **automatically failed-over**
 - Previously was only possible if nodes were in same server group and entire group failed.
 - Usual auto-failover safety caveat applies -- i.e. there should be no data loss after failover.
- **Improves HA** for many common deployment scenarios
 - Multiple VMs on a single host - Likely in public and private cloud infrastructure
 - Multiple racks in a single Availability Zone (AZ) with AZ = Couchbase Server Group





Index Service - Auto Failover

The Index service now supports automatic failover without any administrator intervention

- The Index service can detect when its node becomes unresponsive, and auto failover its service **seamlessly to another indexing service.**
- **Applications will not be interrupted** when the index service fails over.
- Index auto failover is supported when the index service is **colocated** with the data service **and when it is not.**
- Index auto failover will not be done if any indexes or partitions will be lost -- e.g. no more replicas.
- *Exception:* when Index Service is co-located with the Data Service, Index is not consulted on failing over the node due to Data Service preference

Node Availability ⓘ

Auto-failover after seconds for up to node
 Auto-failover for sustained data disk read/write failures after seconds

Auto-failover settings - same for Data, Index service

name	group	services	CPU	RAM	swap	disk used	items	Statistics
172.31.2.202	Group 1	index	0.5%	3.4%	---	---	0/0	Statistics

Addresses: 172.31.2.202 (int)
Address Family: IPv4
Node-to-Node Encryption: off
OS: x86_64-pc-linux-gnu
Uptime: 25 minutes, 32 seconds
Version: Enterprise Edition 7.1.0 build 2549
RAM Quotas: Index 256MiB
Index Storage Path: /opt/couchbase/var/lib/couchbase/data

Disk Usage

Disk used by indexing services is shown per index on their pages.
Some disk use is currently unreported.

[Remove](#) [Failover](#)

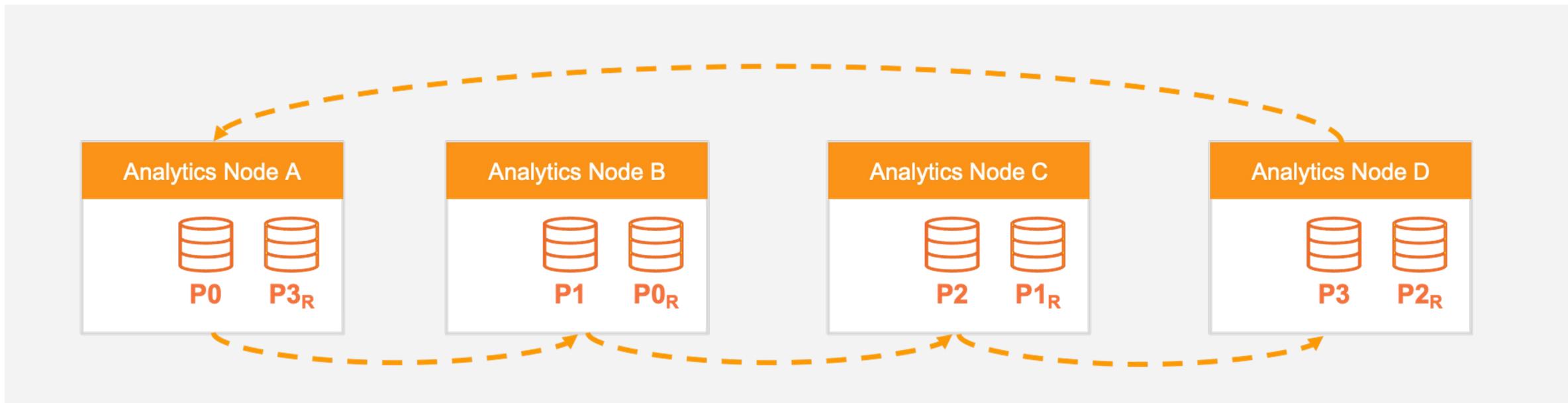
Manually failing over a node with Index service



High Availability (HA) Support for Analytics Service

Always available near real-time data - High Availability without disruptions

- Query and analyze information without disruption using **replicas**
- Data **available** when secondary analytics node(s) are down/failed-over
- Minimal disruption in time to insights during cluster topology changes (failover, etc.)

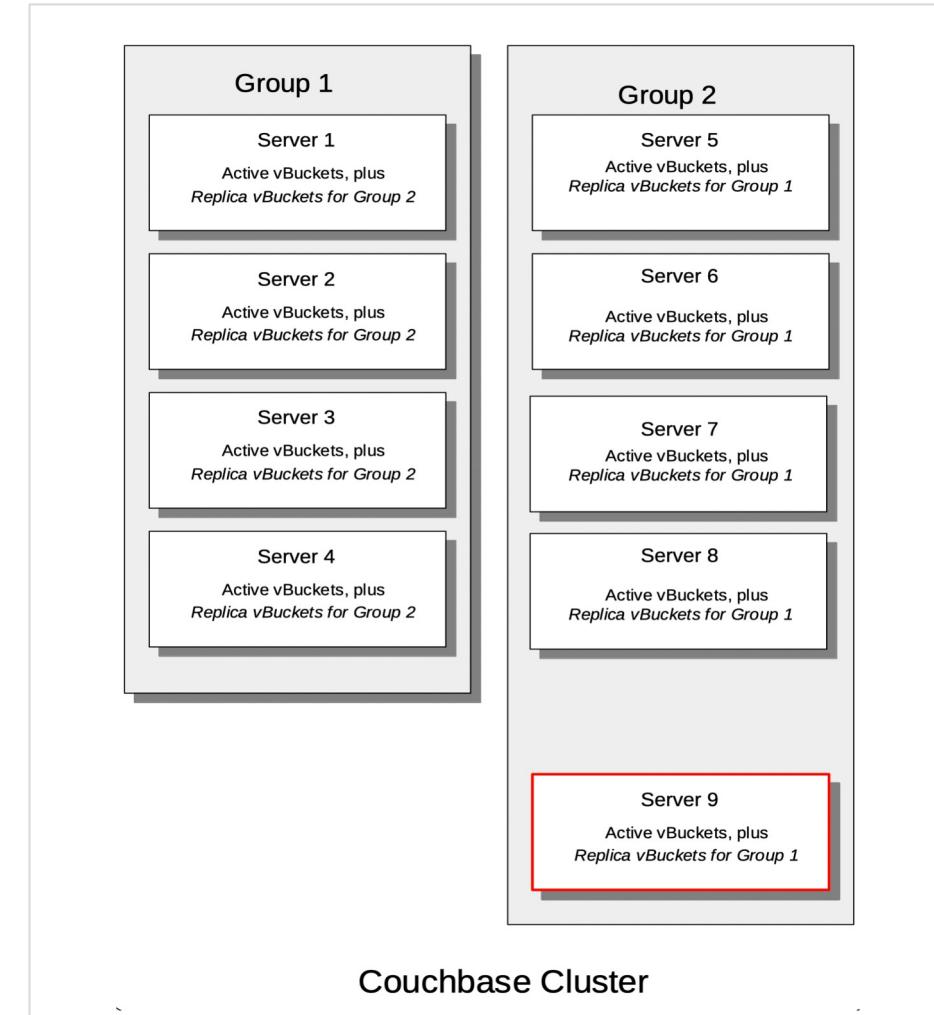




Enhanced HA support for Unequal Server Groups

Unequal Server Groups will also ensure failover can happen when a server group fails

- Previously in case of unequal groups, the cluster provided a best effort distribution of replicas vBuckets
- Since Couchbase Server 7.1, active and replica vBuckets will be **placed in separate server groups** even if server groups are unequal
- No data loss** ever on server group failure
- Failover is safe** even with unequal groups
- Note: Effort should be made to keep server groups equal and unequal server groups should be a temporary state.
Performance of nodes in smaller groups will be lower as they carry more load.





XDCR Incremental on Cluster Topology Change

XDCR Improved Performance on source cluster topology change

- XDCR computes and stores checkpoint information for each vBucket to determine starting/restart/resume point of replication
- A checkpoint contains high sequence number for each vBucket successfully replicated to the target cluster

Previous Behavior

Resource intensive if source cluster topology changes

- XDCR checkpoints are not replicated to replicas or to new nodes on the source cluster
- On a source cluster topology change (e.g. rebalance, failover), XDCR checkpoint information may not be available for all vBuckets
- Rechecking if each document needs to be replicated to the target is resource intensive, including network

Enhanced Behavior in 7.1

Improved performance on source topology changes

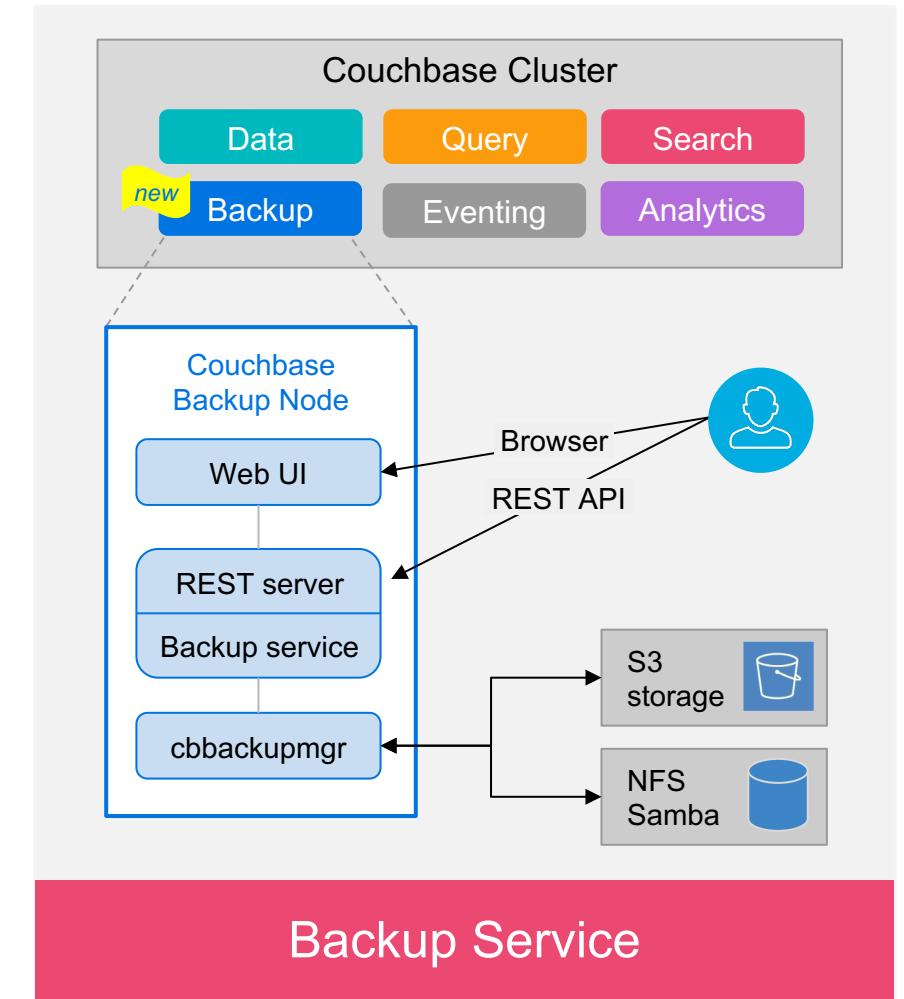
- New design where XDCR checkpoints are replicated to all replicas and nodes in a cluster
- Now, XDCR checkpoint information is always available -- even after rebalance, failover
- Result is efficient (resource-wise) and performant solution -- XDCR improved performance on source cluster topology change

11 Backup & Restore

Backup Service

New service to effortlessly manage backing up your cluster in a few clicks

- New operational access service supported by MDS
- Graphical interface within the cluster UI
- REST API
- Backup scheduling
- Task history
- Examine backup versions
- Utilizes cbbackupmgr to save to cloud storage repository
- Restore bucket, scopes or collections to any destination

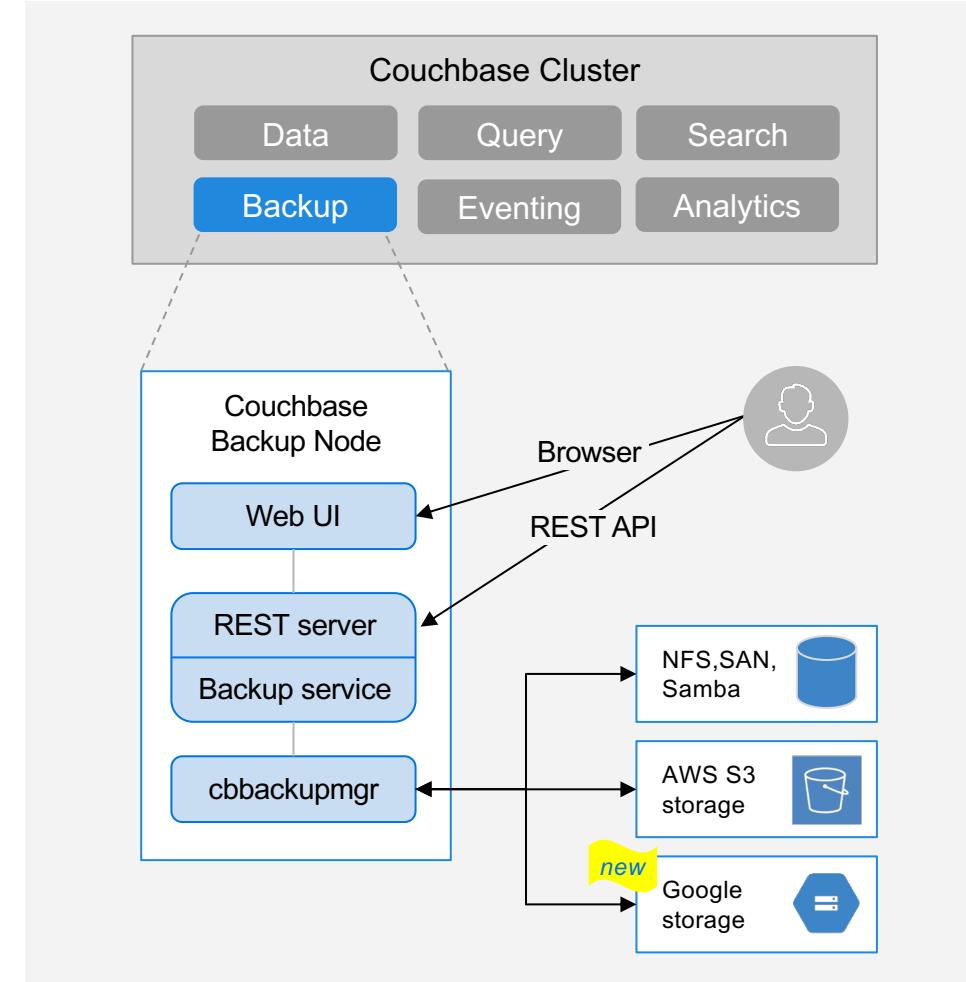




Direct Backup to Google Cloud Storage

Provides native direct backup of large volumes of data to cloud storage

- You can now backup your data to Google Cloud Storage
- Configure archive location: `gs://<bucket>/<path>`
- Uses staging directory to temporarily store metadata for fast, low-latency interaction
- Data buffered in memory and streamed directly to Google Cloud Storage
- Available in `cbackupmgr` CLI and Backup Service
- Supports different GCP credentials types (e.g. Service Account Authentication files)

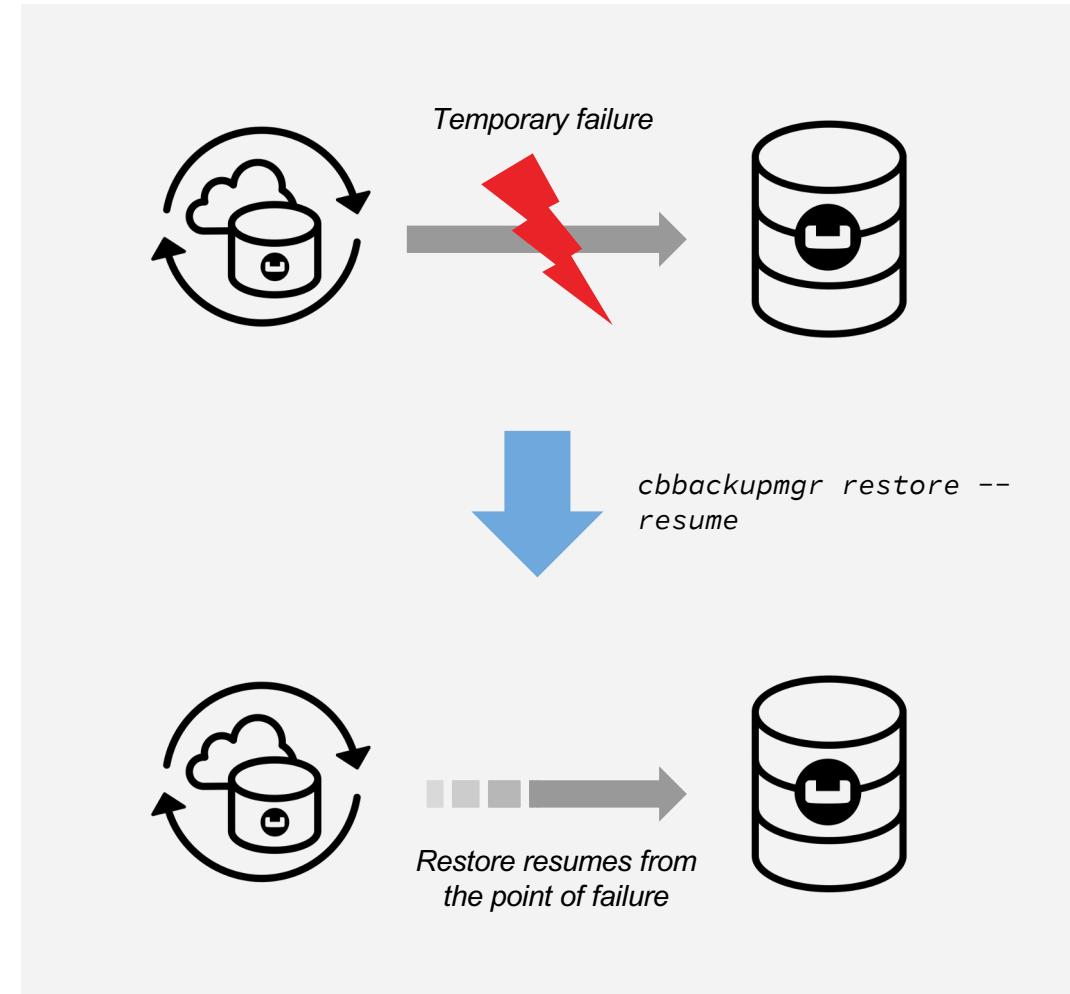




cbackupmgr restore --resume option

Improved Resilience and Increased Conformance to Recovery Time Objective (RTO)

- Previously, no support for resuming restores
 - If the restore failed midway for example, due to environmental reasons, it needed to start over
- New cbackupmgr restore --resume allows restores to **resume after a temporary failure**
 - Resumes from the point of failure
- Especially **useful for cloud restore**
 - Reduces cost because retrieving data from cloud involve significant data transfer costs
 - Increased environmental factors that may cause failures, such as network temporary issues
- Improving time to restore helps **achieve RTO**

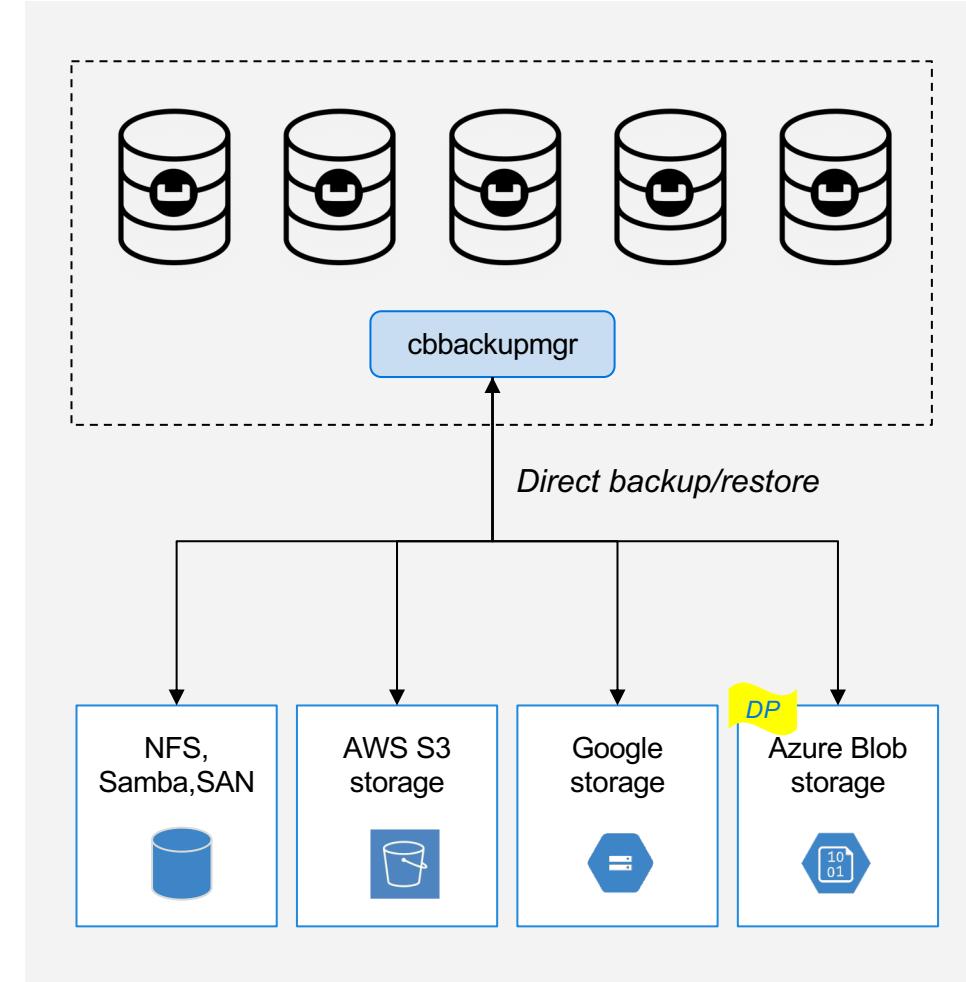




Developer Preview: Direct Backup to Azure Blob Storage

Provides native direct backup of large volumes of data to cloud storage

- You can now backup your data to Azure Blob Storage
- Available in cbbackupmgr CLI only in Preview
- Configure archive location: **az://<container>/<path>**
- Uses staging directory to temporarily store metadata for fast, low-latency interaction
- Data buffered in memory and streamed directly to Azure Blob Storage
- Options to support different Azure credentials types





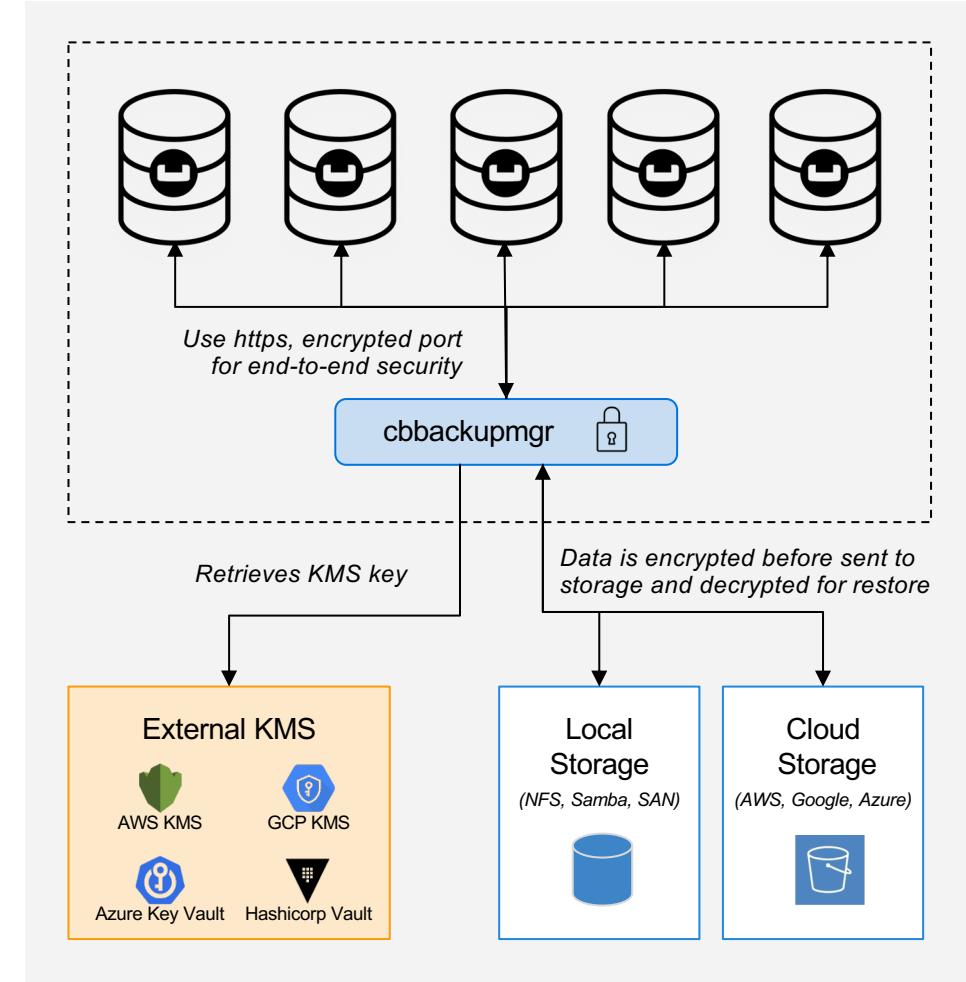
Developer Preview: cbbackupmgr Encrypted Backups

Enhanced Security for Backups

- cbbackupmgr encrypts data before sent to storage
- KMS key used to encrypt the backup repository key
 - Only one KMS API call per backup, restore, etc.
 - Passphrase for development use only
- External Key Management Service (KMS) support

Provider	KMS service supported
AWS	AWS KMS
Google	GCP KMS
Azure	Azure Key Vault
HashiCorp	Vault Transit Secrets Engine

- Encryption algorithms supported
 - AES256GCM (default)
 - AES256CBC





Developer Preview: cbbackupmgr Encrypted Backups (Details)

Enhanced Security for Backups

Data Encryption

- To avoid issues with repeated nonces, cbbackupmgr creates derived keys per Backup, Bucket, and vBucket.
- While using AES_256_GCM, cbbackupmgr can safely encrypt up to 4.3 trillion documents per bucket in each backup – after that, the chances of repeated nonces increase, weakening the encryption

Requires passphrase or KMS Key

- cbbackupmgr config
- cbbackupmgr backup
- cbbackupmgr restore
- cbbackupmgr merge
- cbbackupmgr examine

Additional Info

- **cbbackupmgr info** does not require the passphrase or the KMS key (does not access user data)
- **Backup Service** supports encrypted backups in Couchbase Server Developer Preview mode

THANK YOU



Couchbase