



## 3교시. SDK 소개 (Python) 및 실습

- 1 SDK 소개 (Python)
- 2 SDK 실습 (Python)
- A 데이터 저장 방식 및 고가용성

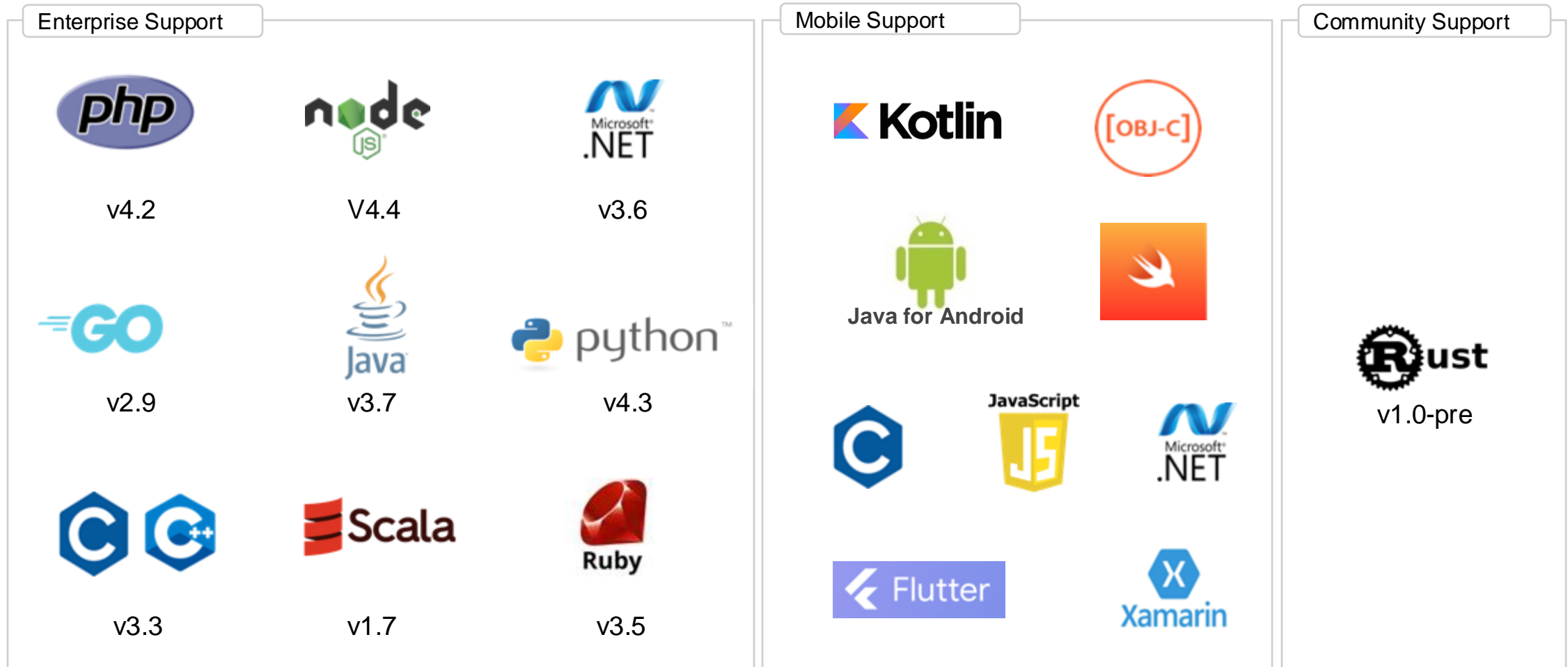


## 3-1. SDK 소개 (Python)

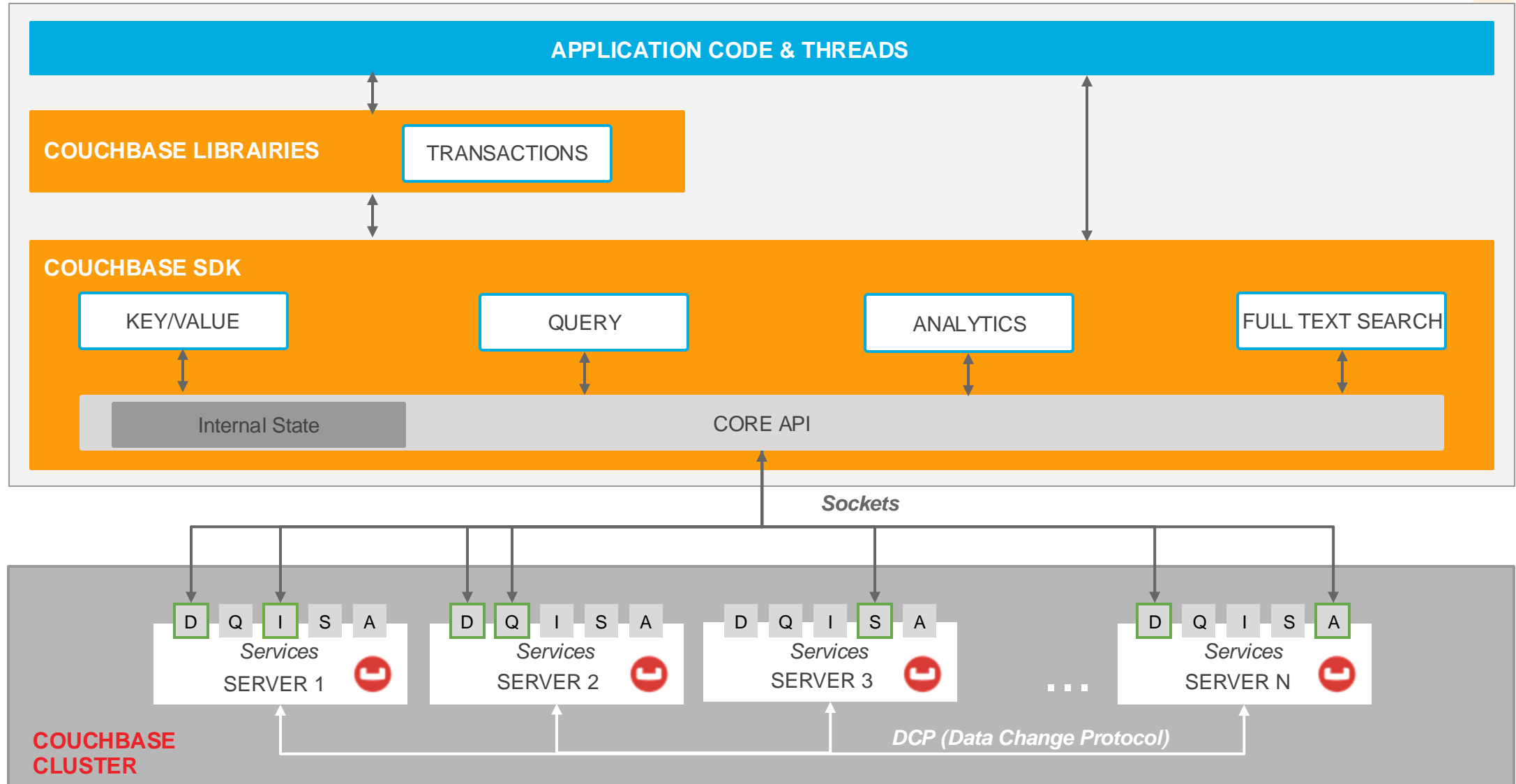


# SDK 개요

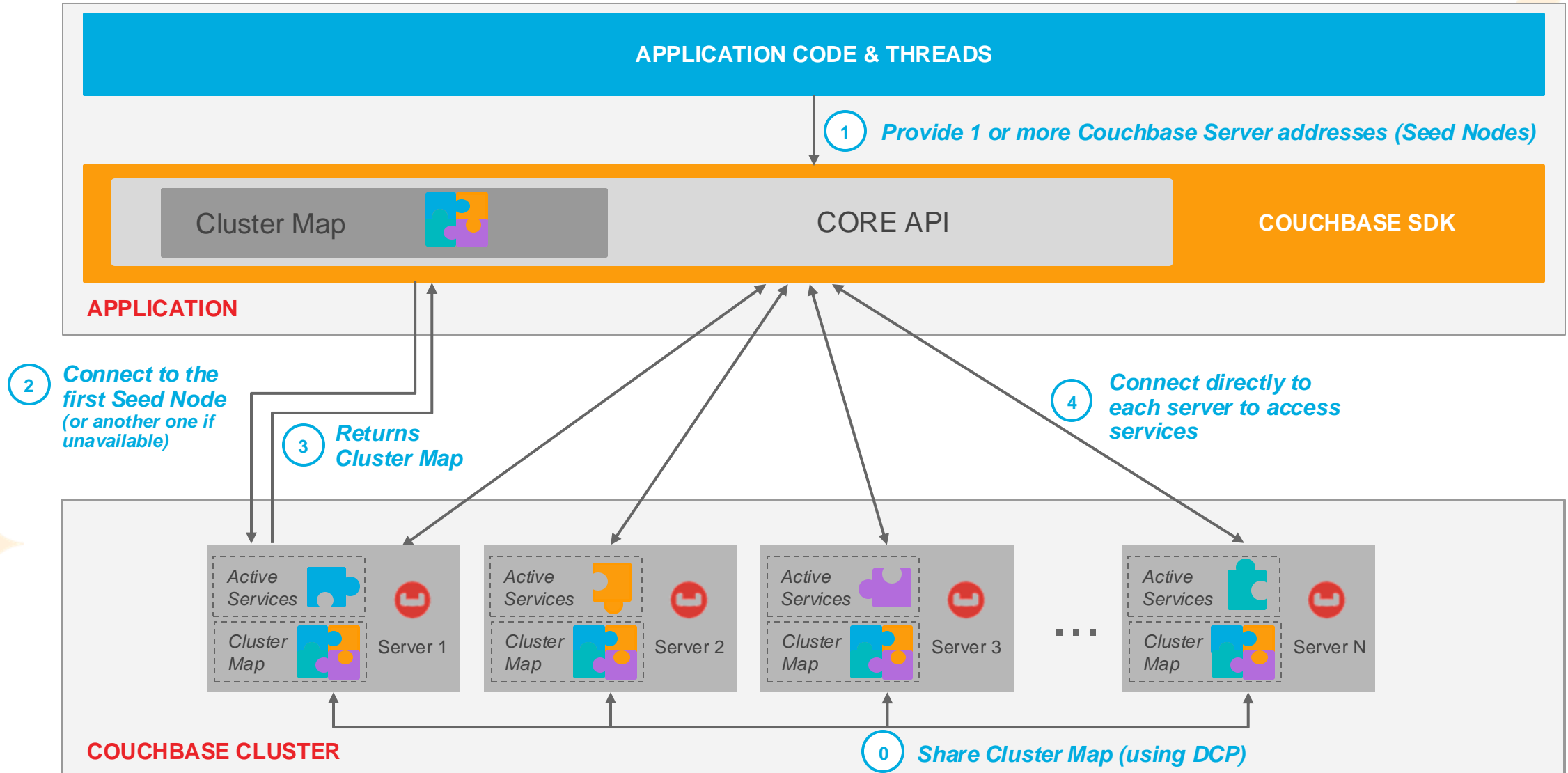
10 개 개발 언어에 대한 Couchbase 클러스터 접근을 위한 SDK(Software Development Kit) 제공



# SDK 아키텍처 : 통합 API 제공



# SDK 아키텍처 : Connection 세부 단계(Bootstrapping)





# Connection Management | Connect to Cluster

Python provides a well thought out Object Oriented Interface

필요한  
Library 등록

```
from datetime import timedelta
# needed for any cluster connection from couchbase.auth
import PasswordAuthenticator from couchbase.cluster
import Cluster
# needed for options -- cluster, timeout, SQL++ (N1QL) query, etc.
from couchbase.options import (ClusterOptions, ClusterTimeoutOptions, QueryOptions)
```

접속 정보 정의

```
# Update this to your cluster
endpoint = "couchbase://your-ip"
username = "Administrator"
password = "password"
bucket_name = "travel-sample"
```

인증 정보

```
# User Input ends here.
# Connect options - authentication
auth = PasswordAuthenticator( username, password )
```

카우치베이스 접속  
및 대기시간 정의

```
# Get a reference to our cluster
# NOTE: For TLS/SSL connection use 'couchbases://<your-ip-address>' instead
cluster = Cluster(endpoint, ClusterOptions(auth))

# Wait until the cluster is ready for use.
cluster.wait_until_ready(timedelta(seconds=5))
```

사용할  
scope>collection  
지정

```
# get a reference to our bucket
cb = cluster.bucket(bucket_name)
cb_coll = cb.scope("inventory").collection("airline")
```



# Key Value Operations | Get/Upsert/Insert

Python allows reading JSON docs directly

Key로 문서를  
가져오는 Get 함수 정의

Get 함수 호출

Json 문서 정의

Key, Value(Json문서)를  
저장하는 Upsert 함수 정의

Upsert 함수 호출

```
# get document function
def get_airline_by_key(key):
    print("\nGet Result: ")
    try:
        result = cb_coll.get(key)
        print(result.content_as[str])
    except Exception as e:
        print(e)
```

```
get_airline_by_key("airline_8091")
```

```
airline = {
    "type": "airline",
    "id": 8091,
    "callsign": "CBS",
    "iata": None,
    "icao": None,
    "name": "Couchbase Airways"
}
```

```
def upsert_document(doc):
    print("\nUpsert CAS: ")
    try:
        # key will equal: "airline_8091"
        key = doc["type"] + "_" + str(doc["id"])
        result = cb_coll.upsert(key, doc)
        print(result.cas)
    except Exception as e:
        print(e)
```

```
upsert_document(airline)
```

Type과 id로 Key 생성

Upsert가 아니라 insert로 해도 됨  
`result = cb_coll.insert(key, doc)`



# Key Value Operations | SQL++

SQL 호출시, 파라미터 사용

입력 받은 값으로  
SQL 실행하여  
결과를 출력하는  
함수 정의

lookup 함수 호출

```
# query for new document by callsign

def lookup_by_callsign(cs):
    print("\nLookup Result: ")
    try:
        inventory_scope = cb.scope('inventory')
        sql_query = 'SELECT VALUE name FROM airline WHERE callsign = $1'
        row_iter = inventory_scope.query(
            sql_query,
            QueryOptions(positional_parameters=[cs]))
        for row in row_iter:
            print(row)
    except Exception as e:
        print(e)

lookup_by_callsign("CBS")
```

위치 정의 파라미터





# Key Value Operations | SQL++ , Metrics 확인

SQL 호출시, 수행한 시간, 결과 건수 등의 메트릭 정보 확인

```
from couchbase.cluster import Cluster
from couchbase.options import ClusterOptions, QueryOptions
from couchbase.auth import PasswordAuthenticator
from couchbase.exceptions import CouchbaseException

cluster = Cluster.connect(
    "couchbase://your-ip",
    ClusterOptions(PasswordAuthenticator("Administrator", "password")))
bucket = cluster.bucket("travel-sample")
collection = bucket.default_collection()

try:
    result = cluster.query(
        "SELECT * FROM `travel-sample`.inventory.airport LIMIT 10", QueryOptions(metrics=True))

    for row in result.rows():
        print(f"Found row: {row}")

    print(f"Report execution time: {result.metadata().metrics().execution_time()}")

except CouchbaseException as ex:
    import traceback
    traceback.print_exc()
```

다른 예시  
QueryOptions(read\_only=True)

SQL 실행하여  
결과를 출력시,  
건수와 실행 시간  
확인



# Key Value Operations | SQL++ , 결과 출력 방법

SQL 호출시, 결과를 원하는 방식으로 출력하는 방법

SQL 실행



```
from couchbase.cluster import Cluster
from couchbase.options import ClusterOptions, QueryOptions
from couchbase.auth import PasswordAuthenticator
from couchbase.exceptions import CouchbaseException

cluster = Cluster.connect(
    "couchbase://your-ip",
    ClusterOptions(PasswordAuthenticator("Administrator", "password")))
bucket = cluster.bucket("travel-sample")
collection = bucket.default_collection()

result = cluster.query(
    "SELECT * FROM `travel-sample`.inventory.airline LIMIT 10")
```

결과 출력



```
# iterate over rows
for row in result:
    # each row is an instance of the query call
    try:
        name = row["airline"]["name"]
        callsign = row["airline"]["callsign"]
        print(f"Airline name: {name}, callsign: {callsign}")
    except KeyError:
        print("Row does not contain 'name' key")
```



# Key Value Operations | SQL++ , Sub-Document 처리

Sub Documents에 대한 검색, 수정, 추가 방법

key: **customer123**

<https://docs.couchbase.com/python-sdk/current/howtos/subdocument-operations.html>

```
{
  "name": "Douglas Reynholm",
  "email": "douglas@reynholmindustries.com",
  "addresses": {
    "billing": {
      "line1": "123 Any Street",
      "line2": "Anytown",
      "country": "United Kingdom"
    },
    "delivery": {
      "line1": "123 Any Street",
      "line2": "Anytown",
      "country": "United Kingdom"
    }
  },
  "purchases": {
    "complete": [
      339, 976, 442, 666
    ],
    "abandoned": [
      157, 42, 999
    ]
  }
}
```

```
result = collection.lookup_in('customer123',
                              [SD.get('addresses.delivery.country')])
country = result.content_as[str](0) # 'United Kingdom'

result = collection.lookup_in('customer123', [SD.exists('purchases.pending[-1]')])
print(f'Path exists: {result.exists(0)}.')
# Path exists: False.

result = collection.lookup_in('customer123', [SD.get('addresses.delivery.country'),
                                              SD.exists('purchases.complete[-1]')])

print('{0}'.format(result.content_as[str](0)))
print('Path exists: {}'.format(result.exists(1)))
# path exists: True.

collection.mutate_in('customer123', [SD.upsert('fax', '311-555-0151')])

collection.mutate_in('customer123', [SD.insert('purchases.pending', [42, True,
'None'])])

try:
    collection.mutate_in('customer123', [SD.insert('purchases.complete', [42, True,
'None'])])
except PathExistsException:
    print('Path exists, cannot use insert.')

collection.mutate_in('customer123', (SD.remove('addresses.billing'),
                                     SD.replace('email', 'dougr96@hotmail.com')))
```



# Key Value Operations | Full Text Search

Full Text Search에도 다양한 검색을 수행할 수 있음.

<https://docs.couchbase.com/python-sdk/current/howtos/full-text-searching-with-sdk.html>

```
from couchbase.cluster import Cluster
from couchbase.options import ClusterOptions,
SearchOptions
from couchbase.auth import PasswordAuthenticator
from couchbase.exceptions import
CouchbaseException
import couchbase.search as search

auth = PasswordAuthenticator('Administrator',
'password')
cluster = Cluster.connect('couchbase://your-ip',
ClusterOptions(auth))
bucket = cluster.bucket('travel-sample')
scope = bucket.scope('inventory')
collection = scope.collection('hotel')
```

```
try:
    result = cluster.search_query('travel-sample-index',
                                search.QueryStringQuery('Paris'))

    for row in result.rows():
        print(f'Found row: {row}')

    print(f'Reported total rows: {result.metadata().metrics().total_rows()}')

except CouchbaseException as ex:
    import traceback
    traceback.print_exc()

result = cluster.search_query('travel-sample-index',
                              search.PrefixQuery('swim'),
                              SearchOptions(fields=['description']))

for row in result.rows():
    print(f'Score: {row.score}')
    print(f'Document Id: {row.id}')

# print fields included in query:
print(row.fields)
```

미리 만들어 둔 검색 인덱스 명

Keyword 검색

Prefix(접두어) 검색

# 개발 지원 문서

The image shows a screenshot of the Couchbase documentation website. The browser address bar displays the URL <https://docs.couchbase.com/sdk-api/couchbase-python-client/index.html>, which is highlighted with a red dashed box. The page content includes a sidebar with a list of SDKs (C, Go, Java, Kotlin, Node.js, PHP, Python, Ruby, Scala, C++ Transactions, Elasticsearch Connector, Kafka Connector, Spark Connector, Tableau Connector, Power BI Connector) and a main content area titled "Welcome to the Couchbase Python SDK documentation!". The main content area lists various API categories: USING THE COUCHBASE PYTHON SDK, SYNCHRONOUS API, Core couchbase API, Query (SQL++), Analytics, Full Text Search, Transactions, and Range Scan. The Python SDK version 4.3.2 is also visible in the sidebar.

Couchbase Documentation

SERVER MOBILE CAPELLA CLOUD-NATIVE

.NET SDK 3.4 v

C SDK

Go SDK 2.7 v

Java SDK 3.5 v

Kotlin SDK 1.2 v

Node.js SDK 4.2 v

PHP SDK 4.1 v

Python SDK 4.1 v

Ruby SDK 3.4 v

Scala SDK 1.5 v

C++ Transactions

Elasticsearch Connector

Kafka Connector 4.2 v

Spark Connector 3.3 v

Tableau Connector

Power BI Connector

Start

The Couchbase Python SDK offers a wide range of APIs for interacting with Couchbase. In this guide, we will focus on the Python SDK.

How to Use the Python SDK

How to Install the Python SDK

How to Configure the Python SDK

Getting Started

Start Using the .NET SDK

Data Operations

Query

Search

Sample Application

Transactions

Further Data Ops

Managing Couchbase

Errors & Diagnostics

Learn

References

Project Docs

C SDK

Go SDK 2.7 v

Java SDK 3.5 v

Kotlin SDK 1.2 v

Node.js SDK 4.2 v

PHP SDK 4.1 v

Couchbase Python Client Library 4.3.2

Search docs

USING THE COUCHBASE PYTHON SDK

Using the Python SDK

SYNCHRONOUS API

Analytics

BinaryCollection

Core couchbase API

Datastructures

Diagnostics

Query (SQL++)

Management

Full Text Search

Range Scan

Transactions

Views

GLOBAL API

Authentication

Exceptions

Management Options

Options

Results

Welcome to the Couchbase Python SDK documentation!

View page source

## Welcome to the Couchbase Python SDK documentation!

### Getting Started with the Python SDK

[Using the Python SDK](#)

Useful information for getting started and using the Python SDK.

### Synchronous API

[Core couchbase API](#)

API reference for Cluster, Bucket, Scope and Collection objects.

[Query \(SQL++\)](#)

API reference for query (SQL++) operations.

[Analytics](#)

API reference for analytics operations.

[Full Text Search](#)

API reference for full text search (FTS) operations.

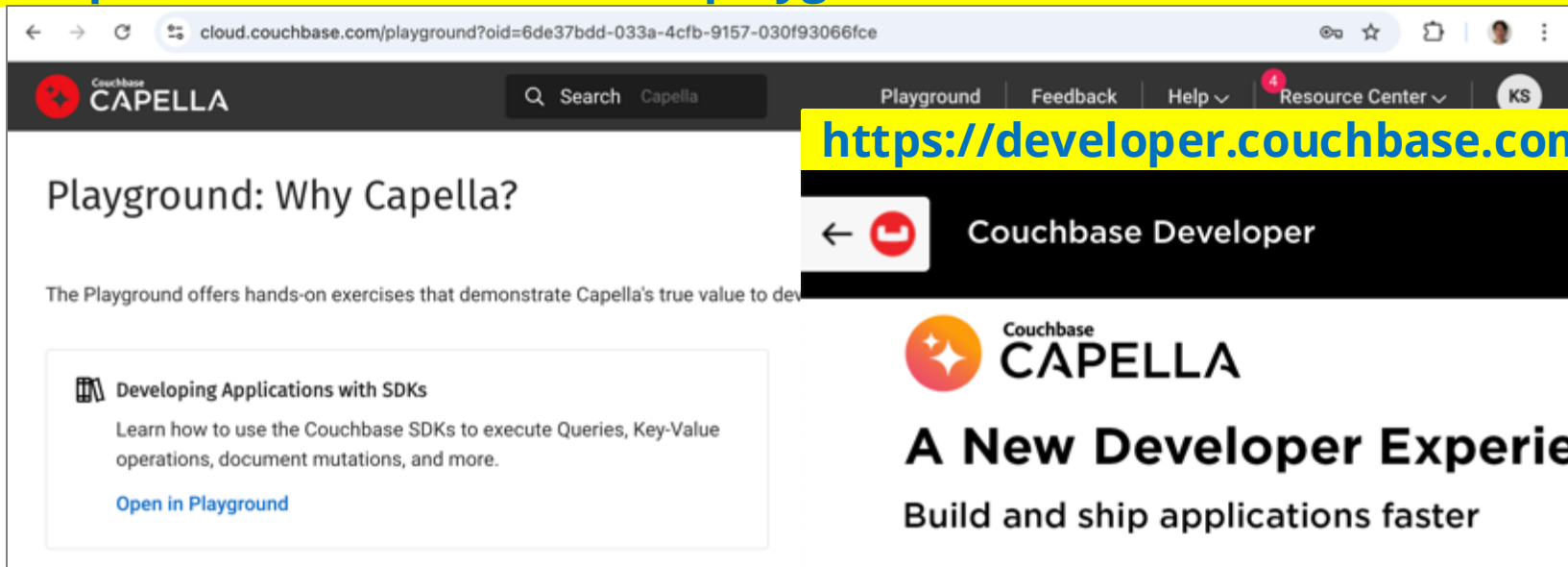
[Transactions](#)

API reference for Distributed ACID transactions with the Python SDK.

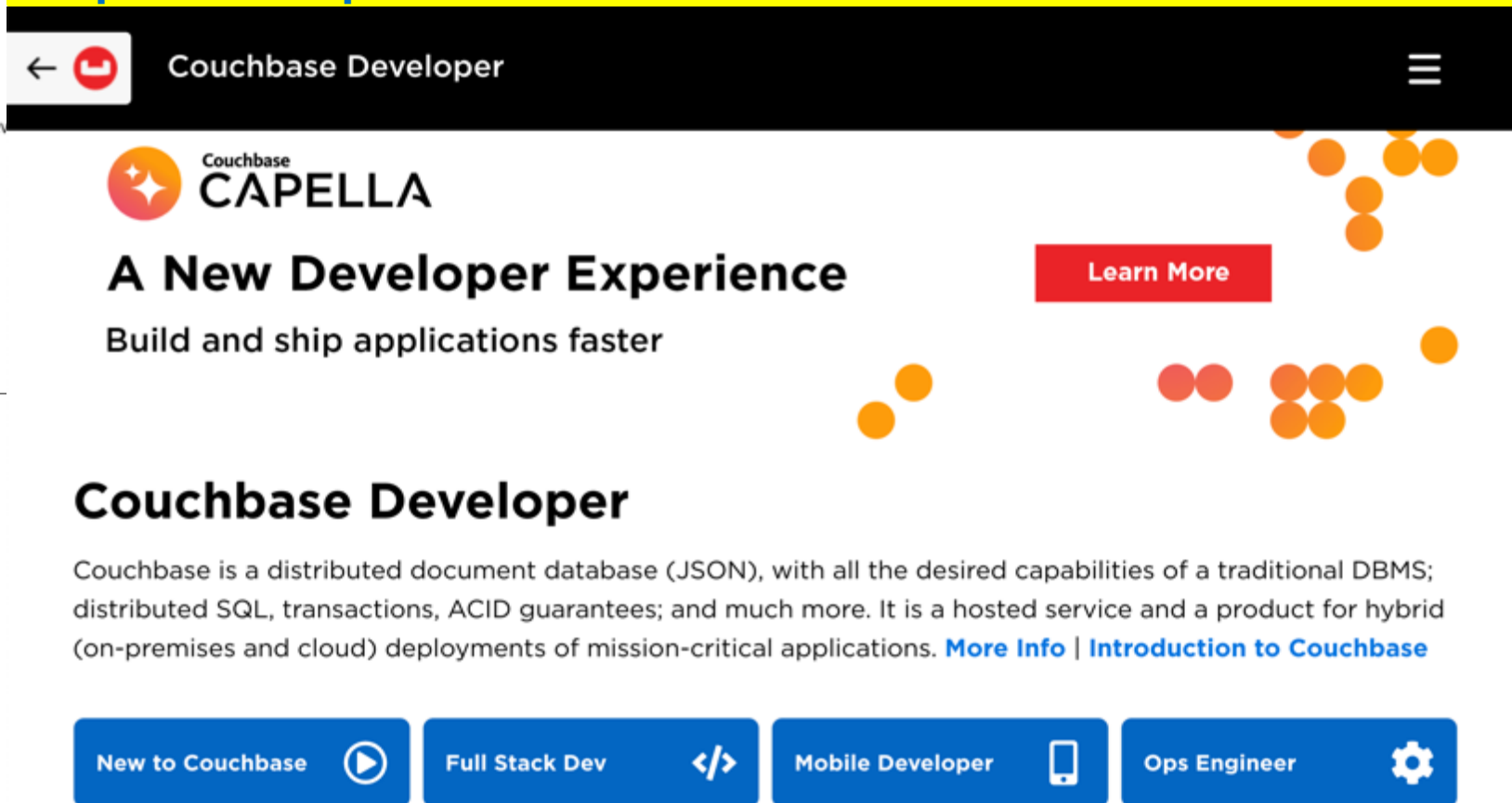
[Range Scan](#)

# 개발자 포털/플레이그라운드

<https://cloud.couchbase.com/playground>



<https://developer.couchbase.com/tutorials>



## 3-2. SDK 실습 (Python, Java)



# Windows 환경에서 Python 수행 환경 만들기

## 1. Python 설치

1. **Python 다운로드**: [Python 공식 웹사이트](#)에서 Windows용 설치 파일을 다운로드합니다.
2. **설치 실행**: 다운로드한 파일을 실행합니다.
  - “Add Python to PATH” 옵션을 체크한 후 **Install Now**를 클릭합니다. PATH에 추가하면 나중에 명령 프롬프트에서 Python을 쉽게 사용할 수 있습니다.
3. **설치 확인**: 설치가 완료되면 명령 프롬프트를 열고 다음 명령어로 설치가 정상적으로 되었는지 확인합니다.

```
python --version
```

## 2. 가상 환경 생성

가상 환경을 사용하면 프로젝트마다 Python 및 패키지 종속성을 독립적으로 관리할 수 있습니다.

1. **프로젝트 폴더 생성**: 먼저 작업할 폴더를 만듭니다.

```
mkdir my_project
```

```
cd my_project
```

2. **가상 환경 생성**: `python -m venv` 명령을 사용하여 가상 환경을 생성합니다.

```
python -m venv venv
```

여기서 `venv`는 가상 환경 폴더 이름으로, `my_project` 폴더 안에 `venv` 폴더가 생성됩니다.

3. **가상 환경 활성화**:

- Windows에서는 다음 명령으로 가상 환경을 활성화합니다.

```
venv\Scripts\activate
```

- 활성화되면 명령 프롬프트에 `(venv)`라는 표시가 나타납니다.

4. **가상 환경 비활성화**:

- 작업이 끝나면 `deactivate` 명령어로 가상 환경을 비활성화할 수 있습니다.

```
deactivate
```

## 3. 패키지 설치

가상 환경이 활성화된 상태에서 `pip`를 사용하여 필요한 패키지를 설치할 수 있습니다.

```
pip install package_name
```

예를 들어, `requests` 라이브러리를 설치하려면 다음과 같이 입력합니다.

```
pip install requests
```

```
pip install couchbase
```

## 4. requirements.txt로 종속성 관리하기

프로젝트의 모든 종속성을 기록하려면 `requirements.txt` 파일을 생성합니다. 나중에 다른 환경에서 동일한 패키지를 설치할 수 있도록 다음과 같이 만듭니다.

1. **requirements.txt 파일 생성**:

```
pip freeze > requirements.txt
```

2. **requirements.txt로 패키지 설치**:

다른 환경에서 `requirements.txt`를 사용하여 패키지를 설치할 수 있습니다.

```
pip install -r requirements.txt
```



# MacOS 환경에서 Python 수행 환경 만들기

## 1. Python 설치

MacOS에는 기본적으로 Python 2가 설치되어 있을 수 있지만, 최신 버전인 Python 3를 설치하는 것이 좋습니다.

### 1. Python 설치:

`brew update`

`brew install openssl@1.1 python3`

### 2. 설치 확인: `python3 --version`

### MacOS에 Homebrew 가 설치되어 있지 않는 경우

Homebrew는 MacOS용 패키지 관리자입니다. 터미널을 열고 다음 명령을 입력하여 Homebrew를 설치합니다.

`/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"`

## 2. 가상 환경 생성

가상 환경을 사용하면 프로젝트마다 Python 및 패키지 종속성을 독립적으로 관리할 수 있습니다.

### 1. 프로젝트 폴더 생성: 먼저 작업할 폴더를 만듭니다.

`mkdir my_project`

`cd my_project`

### 2. 가상 환경 생성: `python -m venv` 명령을 사용하여 가상 환경을 생성합니다.

`python3 -m venv venv`

venv라는 이름의 폴더가 프로젝트 폴더 내에 생성되며, 이 폴더가 가상 환경을 담고 있습니다. 이름은 자유롭게 지정할 수 있습니다.

### 3. 가상 환경 활성화:

`source venv/bin/activate`

활성화되면 터미널에 (venv) 표시가 나타나 가상 환경이 활성화된 것을 알 수 있습니다.

### 4. 가상 환경 비활성화:

작업이 끝나면 `deactivate` 명령어로 가상 환경을 비활성화할 수 있습니다.

`deactivate`

## 3. 패키지 설치

가상 환경이 활성화된 상태에서 pip를 사용하여 필요한 패키지를 설치할 수 있습니다.

`pip install package_name`

예를 들어, requests 라이브러리를 설치하려면 다음과 같이 입력합니다.

`pip install requests`

`pip install couchbase`

## 4. requirements.txt로 종속성 관리하기

프로젝트의 모든 종속성을 기록하려면 requirements.txt 파일을 생성합니다. 나중에 다른 환경에서 동일한 패키지를 설치할 수 있도록 다음과 같이 만듭니다.

### 1. requirements.txt 파일 생성:

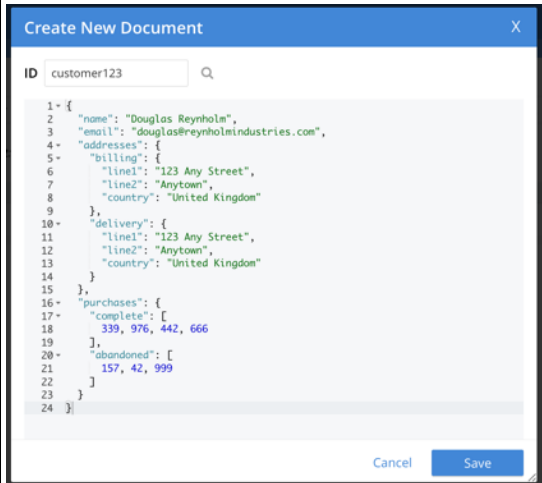
`pip freeze > requirements.txt`

### 2. requirements.txt로 패키지 설치:

다른 환경에서 requirements.txt를 사용하여 패키지를 설치할 수 있습니다.

`pip install -r requirements.txt`

# 실습

	실습 항목	상세 실습 내용	기타
1	개발 환경 (구성) 확인	<ul style="list-style-type: none"> <li>Python 실행 환경 구성</li> </ul>	pip install couchbase
2	기본 KV, SQL++	<ul style="list-style-type: none"> <li>아래에 있는 소스 실행해 보기</li> <li><a href="https://docs.couchbase.com/python-sdk/current/hello-world/start-using-sdk.html">https://docs.couchbase.com/python-sdk/current/hello-world/start-using-sdk.html</a></li> <li><a href="https://github.com/unixfree/CouchbaseTraining/blob/main/src/kv_sql_basic.py">https://github.com/unixfree/CouchbaseTraining/blob/main/src/kv_sql_basic.py</a></li> </ul>	
3	Sub-Document 처리	<ol style="list-style-type: none"> <li><a href="#">travel-sample</a> bucket 의 <a href="#">inventory</a> scope 에 <a href="#">subdoc</a> collection 생성</li> <li><a href="#">Documents</a> UI 에서 오른쪽 상단에 <b>ADD DOCUMENT</b> 클릭 후, <b>11 페이지</b>에 있는 Json 문서 등록 key : <b>customer123</b></li> <li><b>11 페이지</b>에 있는 소스 완성하여 실행해 보기 <a href="https://github.com/unixfree/CouchbaseTraining/blob/main/src/subdoc.py">https://github.com/unixfree/CouchbaseTraining/blob/main/src/subdoc.py</a></li> </ol>	
4	Range Scan ( KV, SQL )	<ul style="list-style-type: none"> <li><a href="https://github.com/unixfree/CouchbaseTraining/blob/main/src/rangescan_sql.py">https://github.com/unixfree/CouchbaseTraining/blob/main/src/rangescan_sql.py</a></li> <li><a href="https://github.com/unixfree/CouchbaseTraining/blob/main/src/rangescan_kv.py">https://github.com/unixfree/CouchbaseTraining/blob/main/src/rangescan_kv.py</a></li> </ul>	
5	Travel-sample Application	<ul style="list-style-type: none"> <li><a href="https://docs.couchbase.com/python-sdk/current/hello-world/sample-application.html">https://docs.couchbase.com/python-sdk/current/hello-world/sample-application.html</a></li> <li><a href="https://github.com/couchbaselabs/try-cb-python">https://github.com/couchbaselabs/try-cb-python</a></li> <li>한글 버전 : <a href="https://github.com/unixfree/try-cb-python">https://github.com/unixfree/try-cb-python</a></li> </ul>	Couchbase, Flask, Vue.JS

The screenshot shows the Couchbase documentation website for the Python SDK. The top navigation bar includes the Couchbase logo, 'Documentation', a search bar, 'Downloads', and a 'Try Free' button. A secondary navigation bar lists categories: SERVER, MOBILE, CAPELLA, CLOUD-NATIVE, COUCHBASE SDKS (highlighted in red), and COLUMNAR SDKS. The left sidebar for the Python SDK (version 4.3) lists sections: Getting Started (with 'Start Using the Python SDK' highlighted), Data Operations, Query, Search, Sample Application, Transactions, Further Data Ops, Managing Couchbase, Errors & Diagnostics, Learn, Reference, and Project Docs. The main content area is titled 'Start Using the Python SDK' with a 'TUTORIAL' badge. It includes a sub-header 'Python SDK / Getting Started / Start Using the Python SDK', an 'Edit on GitHub' link, and a brief introduction: 'Get up and running quickly, installing the Couchbase Python SDK, and running our Hello World example.' It states that the SDK allows Python applications to access a Couchbase cluster via a synchronous API or integration with *twisted* and *asyncio*. A list of topics to be covered in the guide includes connecting to Couchbase Capella or Couchbase Server, adding and retrieving documents, and looking up documents using SQL++ (formerly N1QL). The section 'Hello Couchbase' begins with the text: 'We will go through the code sample step by step, but for those in a hurry to see it, here it is:'. The right sidebar contains links to 'Hello Couchbase', 'Quick Installation', 'Prerequisites', 'Step-by-Step' (with sub-links for Connect, Add and Retrieve Documents, SQL++ Lookup, and Execute!), 'Next Steps' (with sub-links for Additional Resources and Troubleshooting), a feedback section 'Is this page helpful?' with 'Yes' and 'No' buttons, and a 'Leave Additional Feedback?' link.

Couchbase | Documentation

Search Docs

Downloads

Try Free

SERVER MOBILE CAPELLA CLOUD-NATIVE **COUCHBASE SDKS** COLUMNAR SDKS

Python SDK 4.3

Getting Started

[Start Using the Python SDK](#)

► Data Operations

Query

Search

Sample Application

Transactions

Further Data Ops

Managing Couchbase

Errors & Diagnostics

Learn

Reference

Project Docs

Python SDK / Getting Started / Start Using the Python SDK

[Edit on GitHub](#)

## Start Using the Python SDK TUTORIAL

Get up and running quickly, installing the Couchbase Python SDK, and running our Hello World example.

The Couchbase Python SDK allows Python applications to access a Couchbase cluster. It offers a traditional synchronous API as well as integration with *twisted* and *asyncio*.

In this guide, you will learn:

- How to [connect to Couchbase Capella or Couchbase Server](#).
- How to [add and retrieve Documents](#).
- How to [lookup documents](#) with the [SQL++ \(formerly N1QL\)](#) query language.

## Hello Couchbase

We will go through the code sample step by step, but for those in a hurry to see it, here it is:

[Hello Couchbase](#)

[Quick Installation](#)

[Prerequisites](#)

[Step-by-Step](#)

- [Connect](#)
- [Add and Retrieve Documents](#)
- [SQL++ Lookup](#)
- [Execute!](#)

[Next Steps](#)

- [Additional Resources](#)
- [Troubleshooting](#)

Is this page helpful?

[Yes](#) [No](#)

[Leave Additional Feedback?](#)

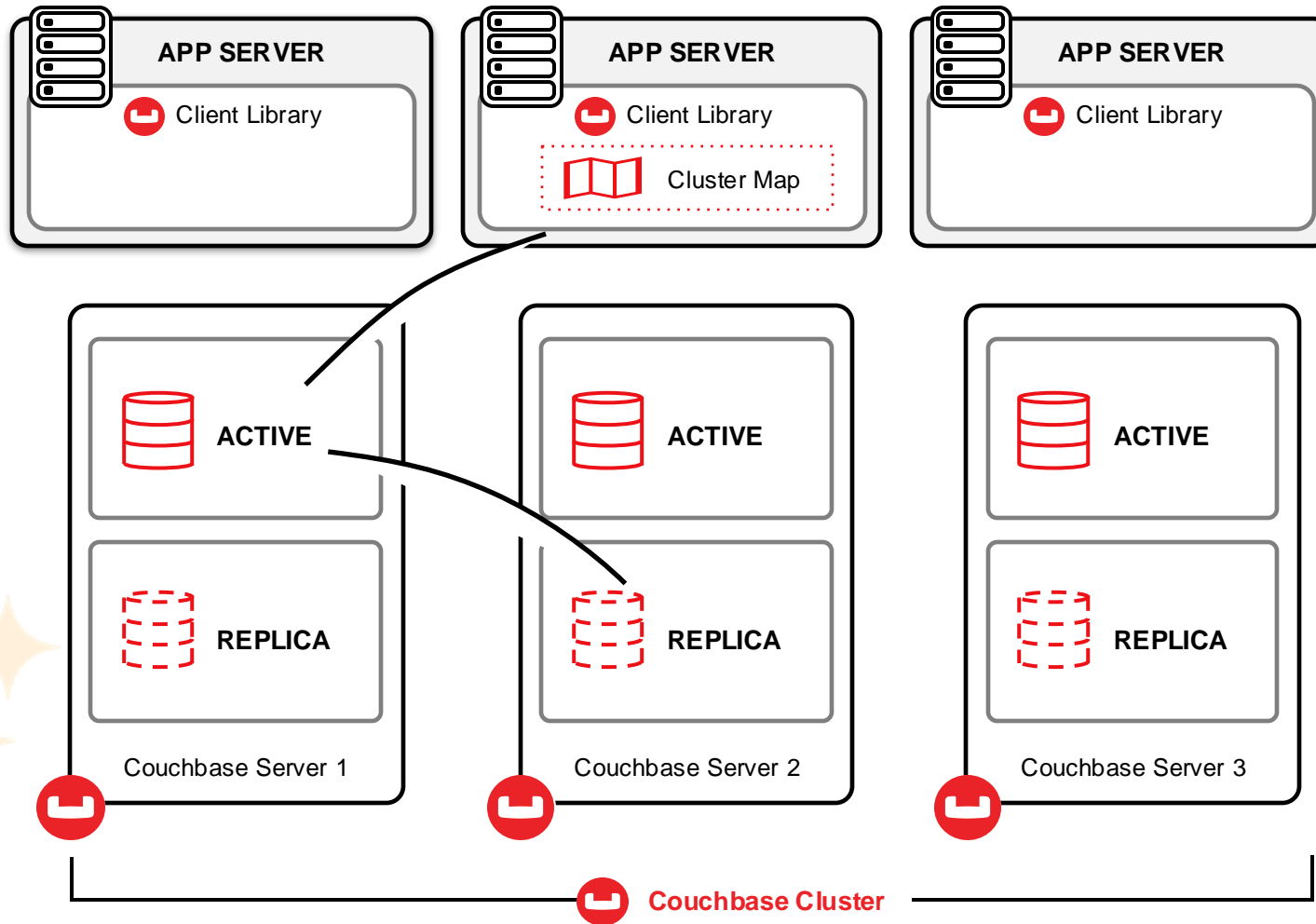


# Appendix. 데이터 저장 방식 및 고가용성

>

1

# A Fully Distributed Database

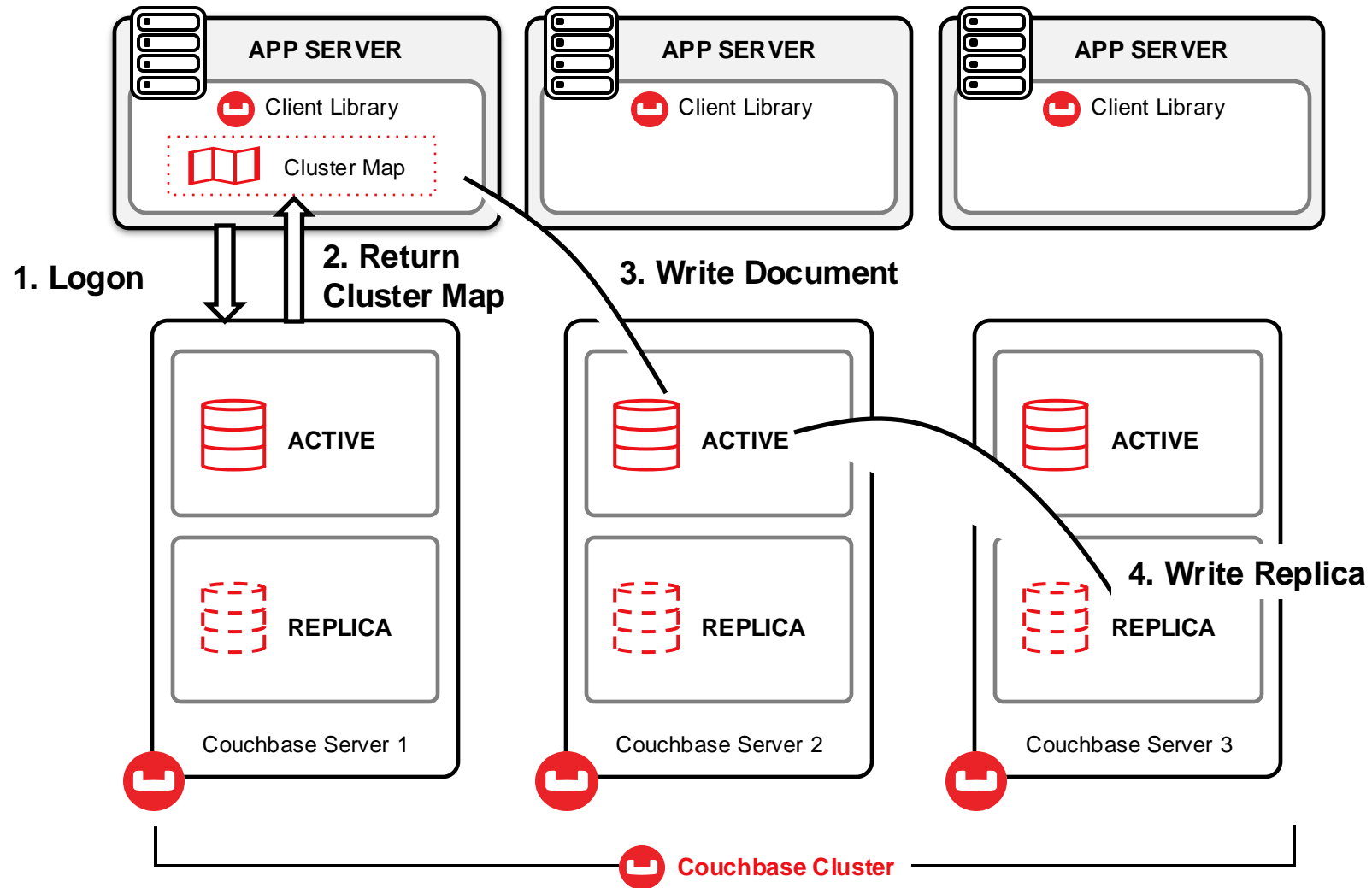


**No Master-Slave**  
Single node type  
Direct client communication

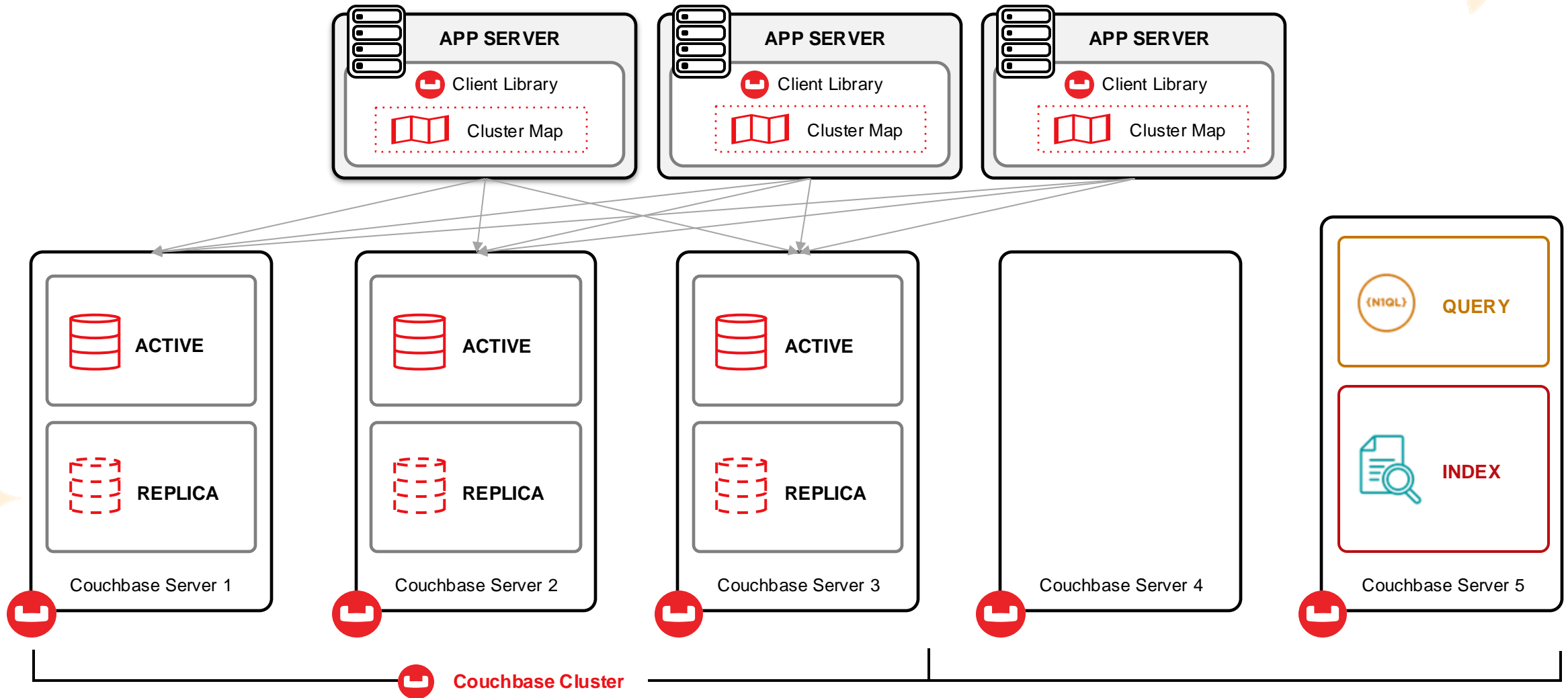
**Automatic sharding**  
Maximum cluster efficiency

**Automatic failover**  
Continuous replication  
for no disruption

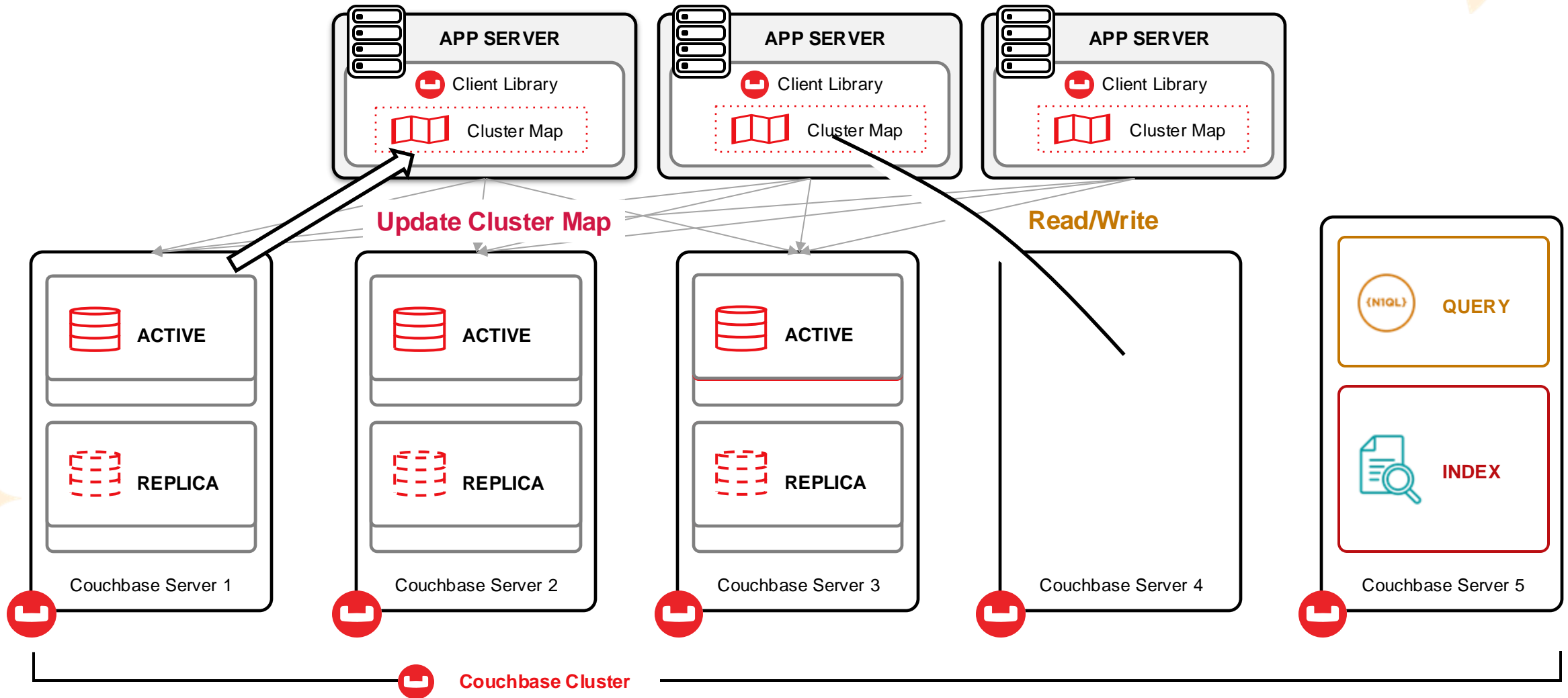
# Basics Operations



# Scalability - Horizontal & Vertical



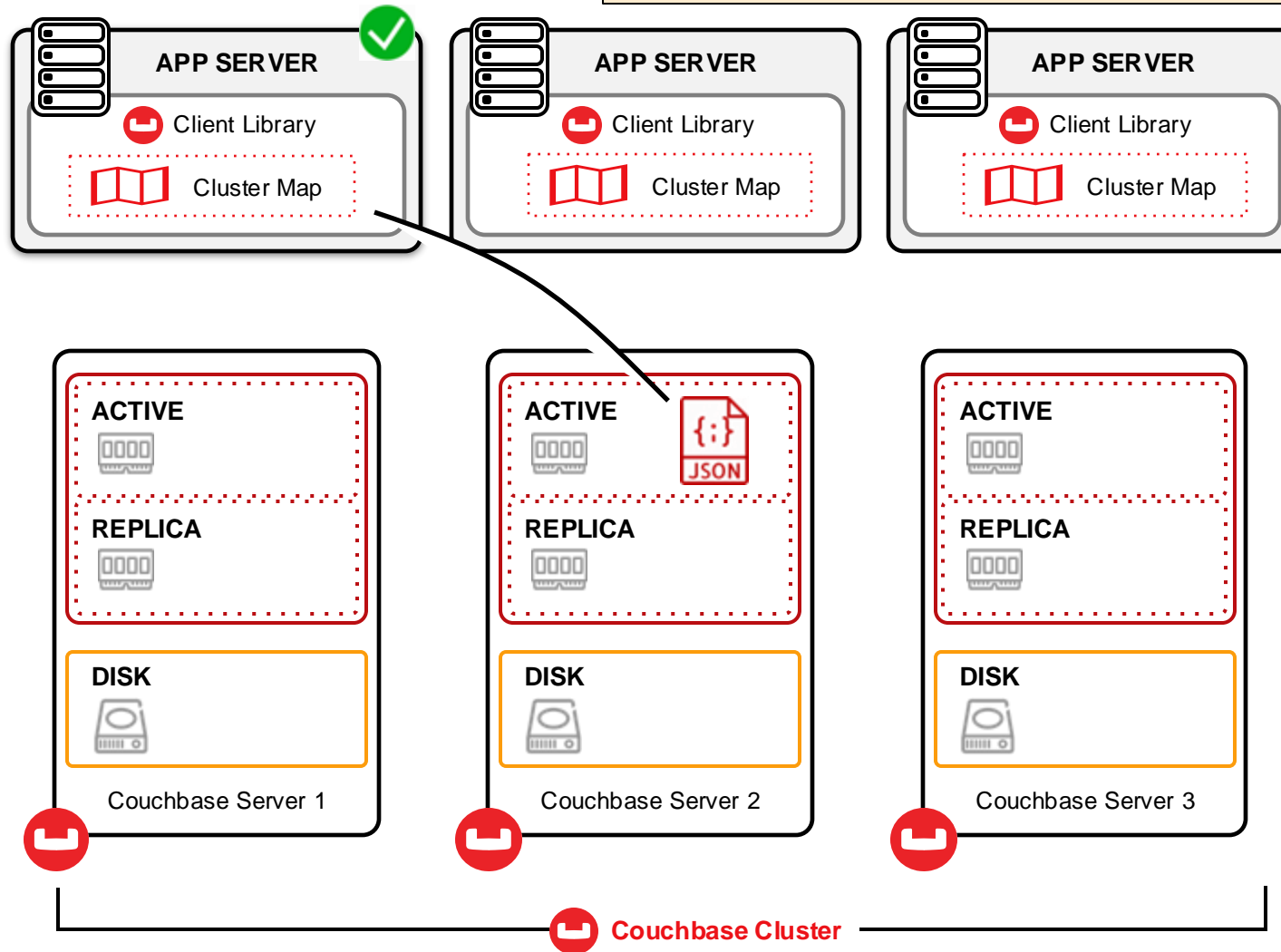
# Scalability - Horizontal & Vertical





# Durability

```
# Upsert with Durability (Couchbase Server >= 6.5) level Majority
document = dict(foo="bar", bar="foo")
opts = UpsertOptions(durability=ServerDurability(Durability.MAJORITY))
result = collection.upsert("document-key", document, opts)
```



replica = 1

## Level of Durability

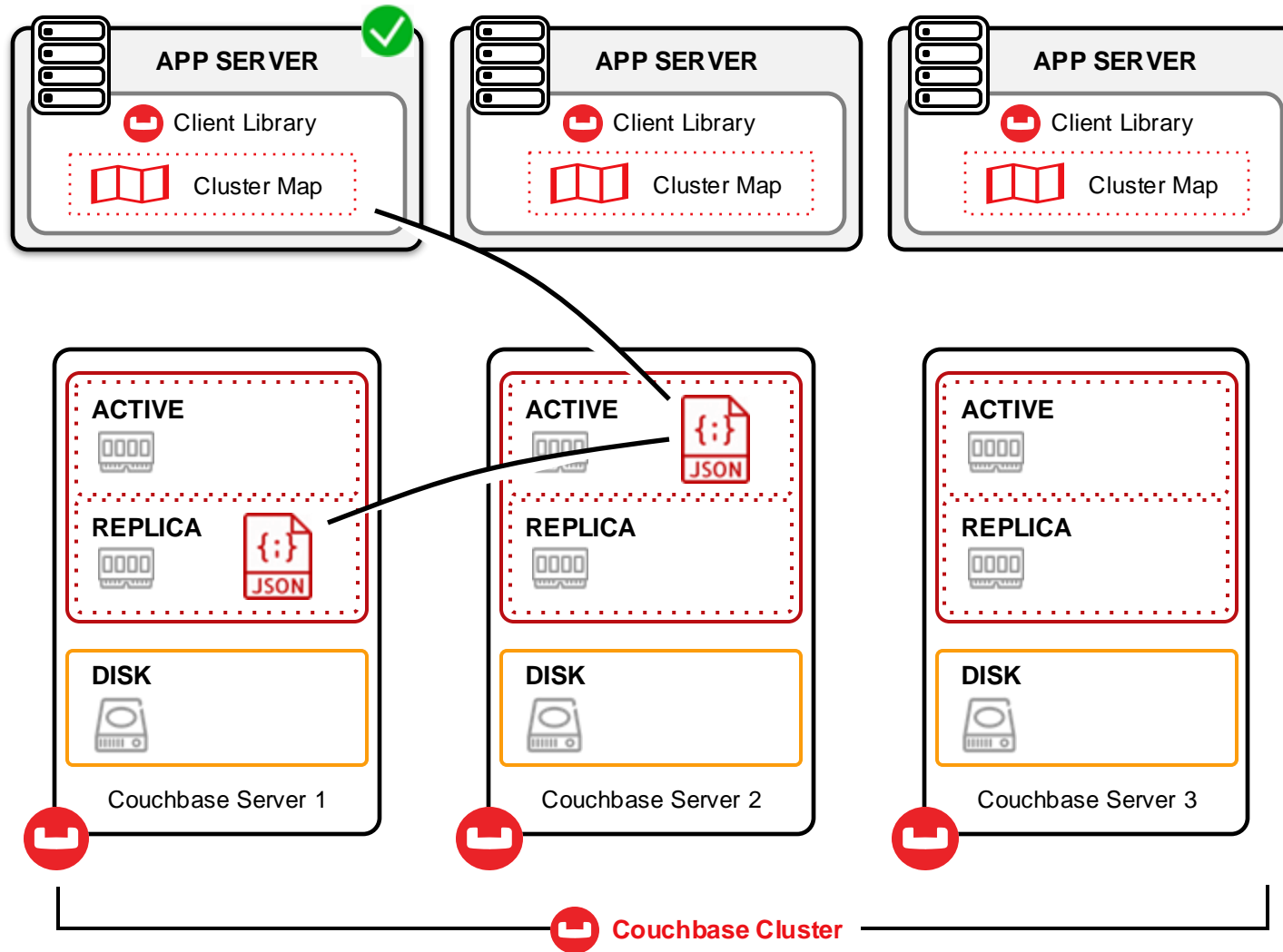
None

Majority

MajorityAndPersistToActive

PersistToMajority

# Durability



replica = 1

## Level of Durability

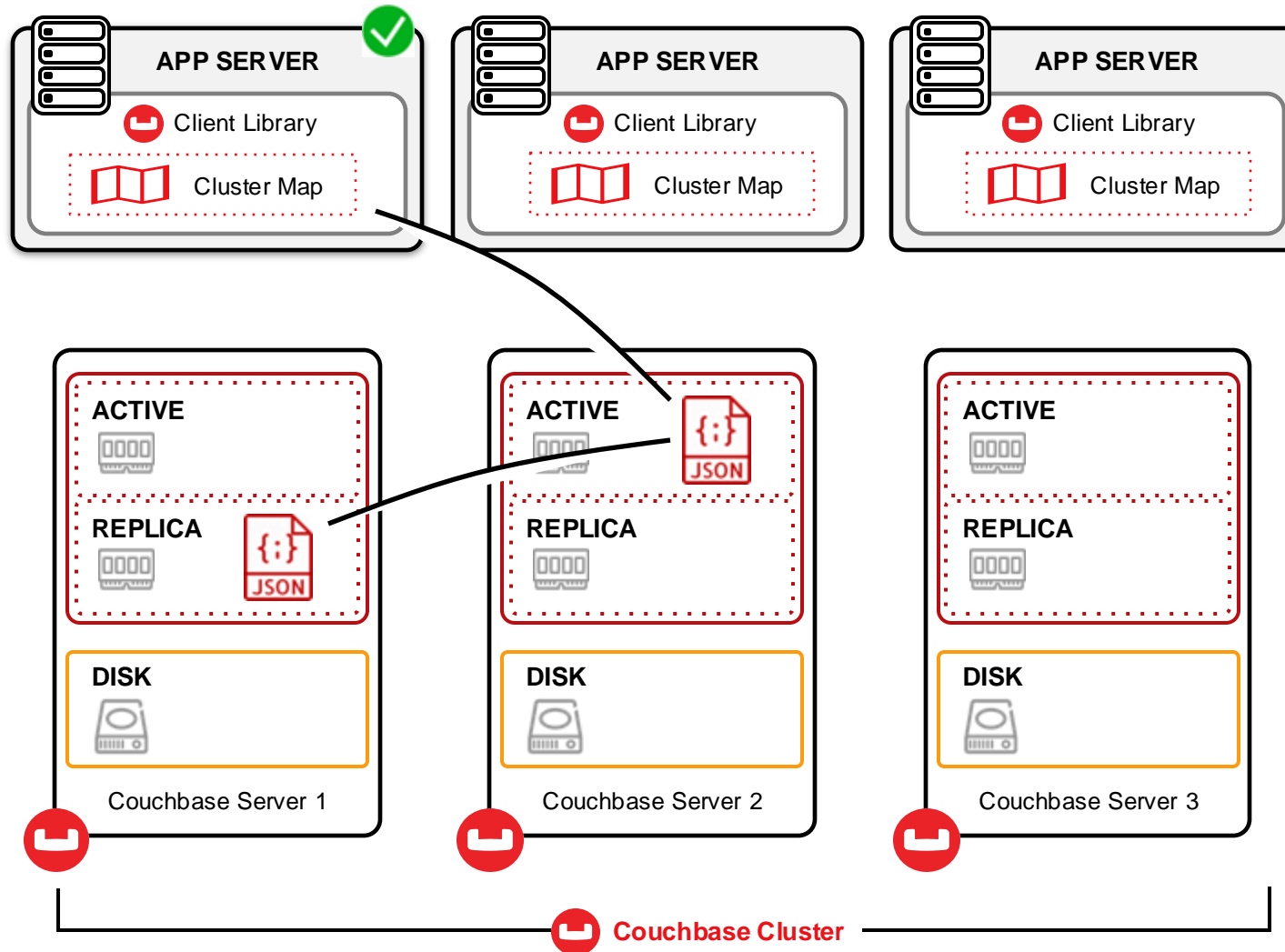
None

**Majority**

MajorityAndPersistToActive

PersistToMajority

# Durability



replica = 1

## Level of Durability

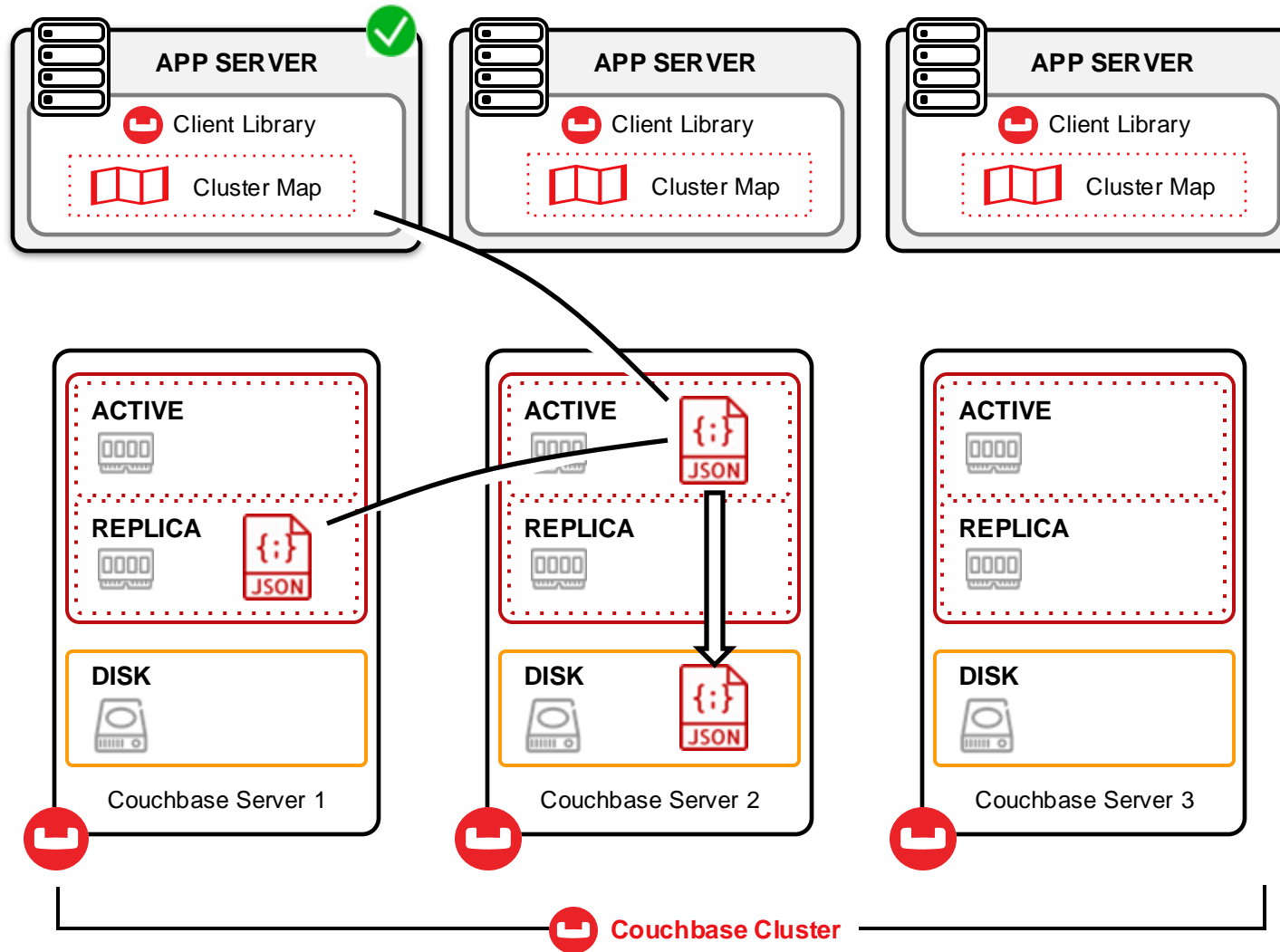
None

**Majority**

MajorityAndPersistToActive

PersistToMajority

# Durability



replica = 1

## Level of Durability

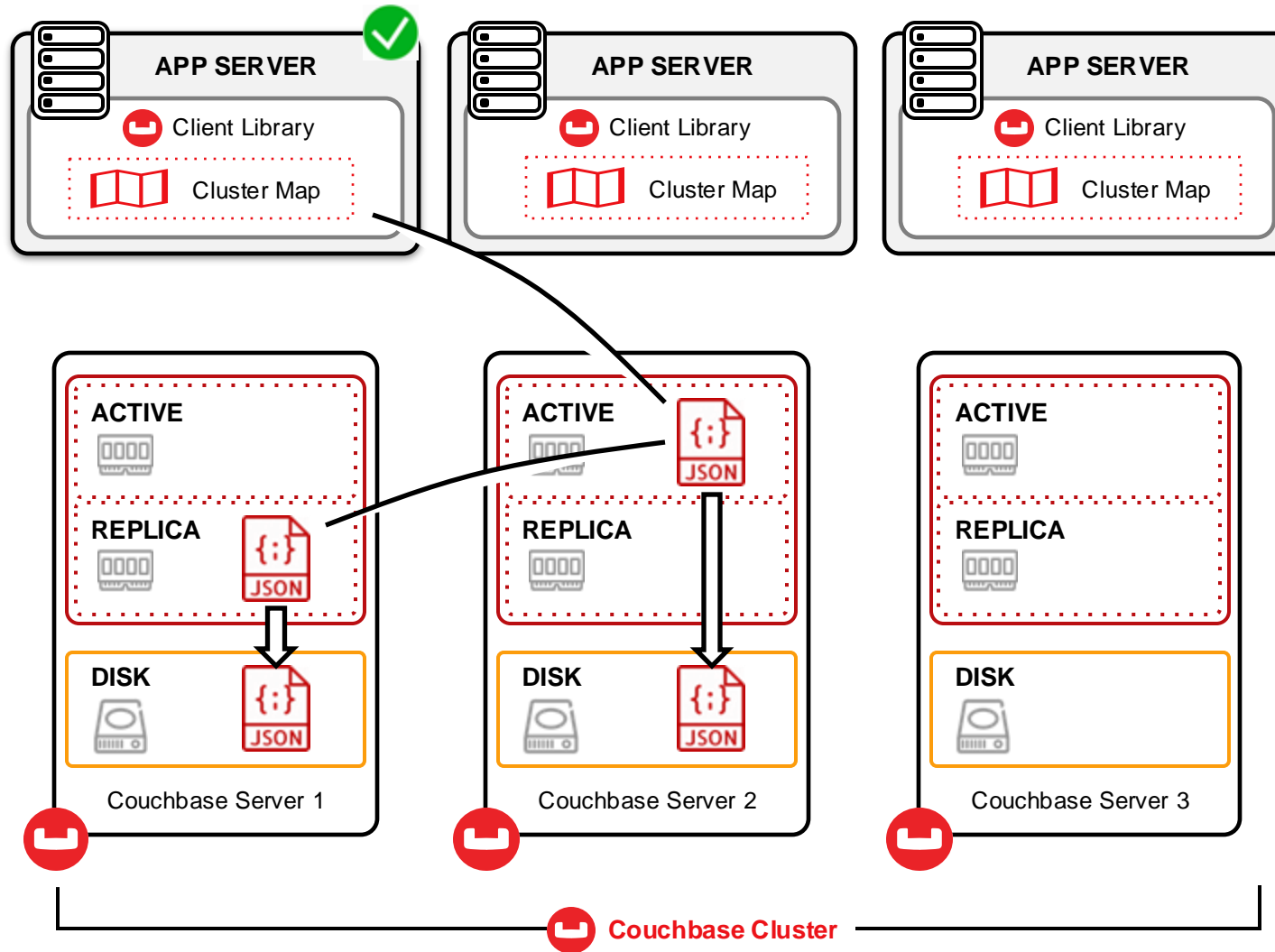
None

Majority

**MajorityAndPersistToActive**

PersistToMajority

# Durability



replica = 1

## Level of Durability

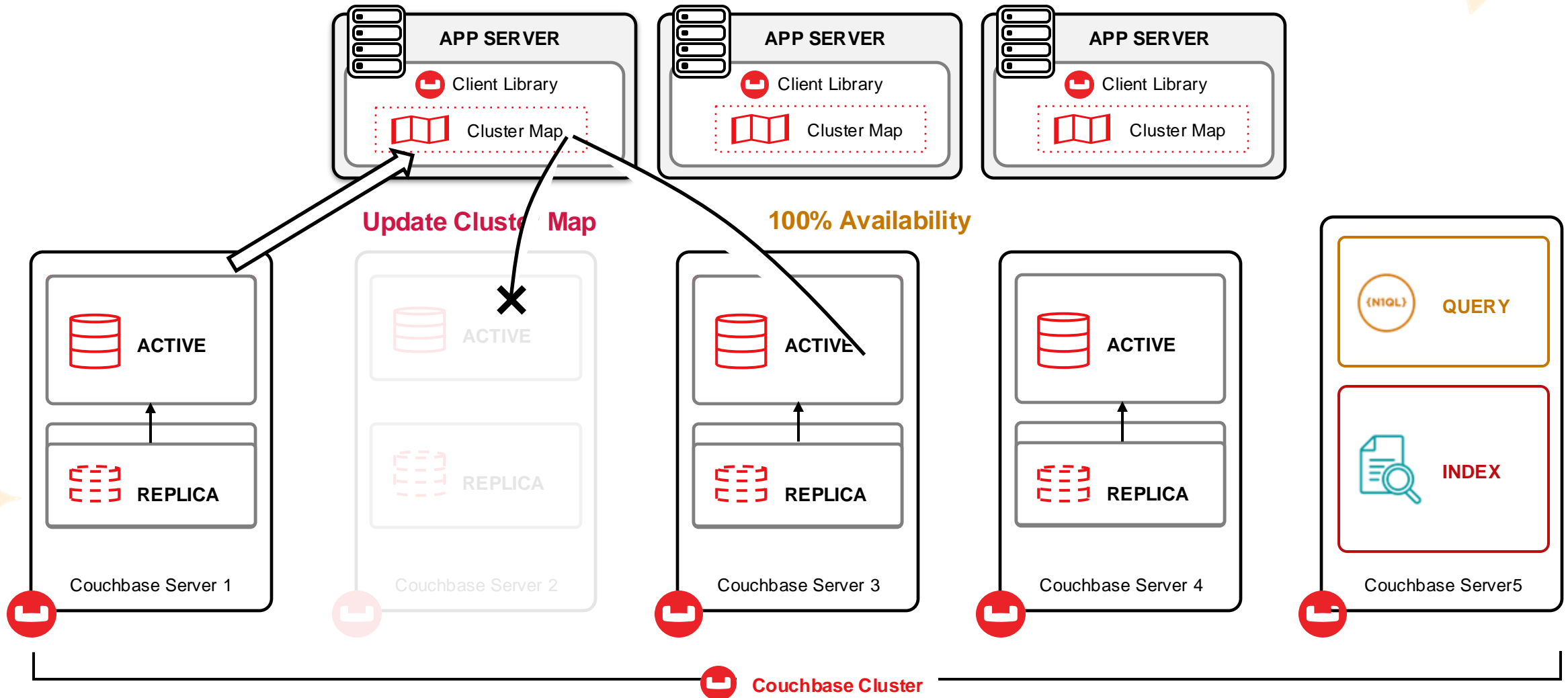
None

Majority

MajorityAndPersistToActive

**PersistToMajority**

# Node Failover





# 수고하셨습니다.



[paul.son@couchbase.com](mailto:paul.son@couchbase.com)

[www.couchbase.com](http://www.couchbase.com)

[cloud.couchbase.com](http://cloud.couchbase.com)



**Couchbase**

