



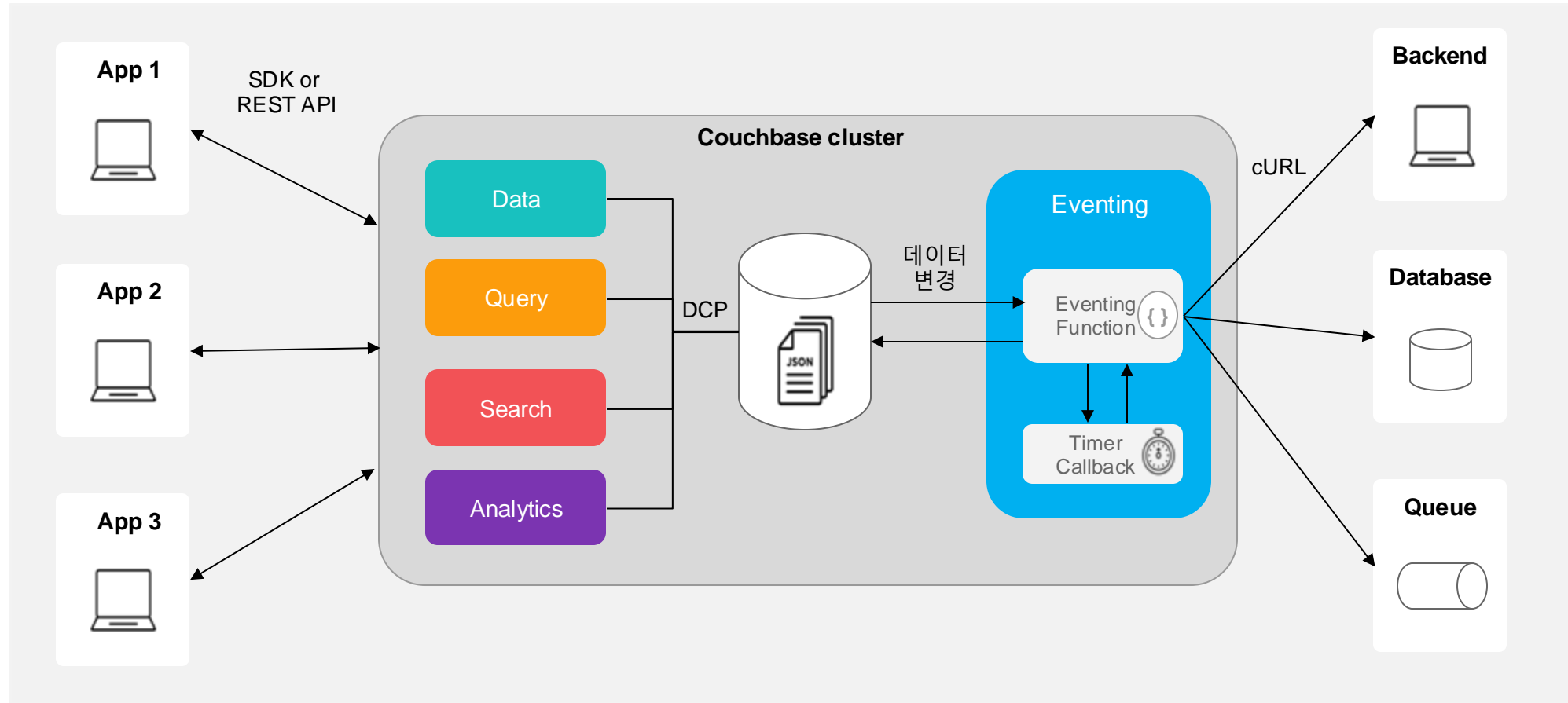
# Couchbase Eventing Workshop

2025.01.14

손 광락, Solutions Engineer

# Eventing Service 개요

- 핵심 데이터 서비스와 함께 사용자 정의 비즈니스 로직을 통해 데이터 변경에 대응하여 실시간으로 대응 조치 가능.
- MDS를 활용하여 기본 데이터 서비스에 대한 걱정 없이 필요에 따라 확장 가능.



# Eventing Functions

친숙한 JavaScript를 활용하는 Eventing을 사용하면 개발자가 데이터에 가까운 데이터 기반 비즈니스 논리를 쉽게 구축하고 유지할 수 있습니다.

- **Functions to process and respond to data-changes in an [Event-Condition-Action](#) model.**
- **Functions are written in [JavaScript code](#) to analyze and manipulate your JSON documents on mutations.**
- **Typical Use Cases:**
  - Threshold-based alerts. 임계값 기반 경고.
  - Monitoring parameters 모니터링 매개 변수
  - Propagating changes 변경사항 전파
  - Enriching documents 문서 강화
  - Scheduling future actions 향후 작업 예약
  - Data cleansing point tools 데이터 정리 포인트 도구

```
function OnUpdate(doc, meta) {  
  log('document', doc);  
  doc["ip_num_start"] =  
    get_ip3o(doc["ip_start"]);  
  doc["ip_num_end"] = get_ip3o(doc["ip_end"]);  
  tgt[meta.id]=doc; //updates doc  
}  
  
function get_ip3o(ip) {  
  var return_val = 0;  
  if (ip) {  
    var parts = ip.split('.');  
    //IP = Ax(256*256*256)+Bx(256*256)+Cx256+D  
    return_val = (parts[0]*(256*256*256)) +  
                  (parts[1]*(256*256)) +  
                  (parts[2]*256) +  
                  (parseInt(parts[3]));  
    return return_val;  
  }  
}
```

Data Enrichment example

# Eventing Service 구성 요건

- All Eventing Functions listen to a single collection.
- All Eventing Functions require a scratchpad **collection** known as “**Eventing Storage**” for internal metadata.
  - Dedicated to the Eventing Service.
  - Shareable across all of a tenant’s Functions
  - For performance this collection should be **100% resident**.
- Optional “**Bucket Bindings**” allow access to other collections.

The screenshot shows the 'Add Function' configuration page for the 'Listen to Collection' function. The interface includes the following sections and callouts:

- Listen to Location:** A dropdown menu showing 'orders' is highlighted by a yellow callout labeled 'Listen to Collection'.
- Eventing Storage:** A dropdown menu showing 'metadata' is highlighted by a yellow callout labeled 'Storage of Metadata'. Below this, a note states: 'System data stored in this location will have the document ID prefixed with eventing'.
- Function Name:** The field contains 'cust1\_eventing\_function'.
- Deployment Feed Boundary:** A dropdown menu showing 'Everything'.
- Description:** An optional text field.
- Settings:** A section header.
- Bindings:** A section with a yellow callout labeled 'Bindings'. It contains two binding entries:
  - Binding 1:** 'bucket alias' is 'orders\_bkt'. The 'Bucket' dropdown shows 'BULK', 'Cust1', and 'orders'. The 'Access' dropdown shows 'read and write'.
  - Binding 2:** 'bucket alias' is 'shipments\_bkt'. The 'Bucket' dropdown shows 'BULK', 'Cust1', and 'shipments'. The 'Access' dropdown shows 'read and write'.
- Buttons:** 'Cancel' and 'Next: Add Code' are at the bottom right.

# onUpdate Function

```
function OnUpdate(doc, meta) {  
  <function_body>  
}
```

```
function OnUpdate(doc, meta) {  
  log('document', doc);  
  doc["ip_num_start"] = get_numip_first_3_octets(doc["ip_start"]);  
  doc["ip_num_end"]  = get_numip_first_3_octets(doc["ip_end"]);  
  tgt[meta.id]=doc;  
}  
  
function get_numip_first_3_octets(ip) {  
}
```

# onDelete Function

```
function onDelete(meta) {  
<function_body>  
}
```

```
function onDelete(meta) {  
if ((meta.id).startsWith("user_") == false) return;  
  var id = meta.id;  
  var numeric_id = parseInt(id.substring(5));  
  DELETE FROM bulk.data.transactions WHERE user_id = $numeric_id;  
}
```

# createTimer Function

**createTimer**(callback, date, reference, context)

- \* callback : timer가 종료되면 실행되는 function
- \* date : timer가 종료되는 시간
- \* reference : timer 식별자
- \* context : JavaScript Object, 대부분 도큐먼트 Key를 전달

```
function OnUpdate(doc, meta) {  
  if (meta.expiration == 0 ) return;  
  var twoMinsPrior = new Date();  
  twoMinsPrior.setSeconds(twoMinsPrior.getSeconds() + (meta.expiration -120));  
  var context = { docID : meta.id, expiration : meta.expiration };  
  createTimer(DocTimerCallback, twoMinsPrior , meta.id, context);  
}  
function DocTimerCallback(context) {  
  <function_body>  
}
```

# curl Function

**curl**(method, binding, [request\_object])

\* method : HTTP method(GET | POST | PUT | HEAD | DELETE)

\* binding : HTTP endpoint URL

\* request\_object : HTTP request **관련** parameters

```
var yesterday = new Date();
yesterday.setHours(0,0,0,0);
yesterday.setDate(yesterday.getDate() - 1);
var apiReqDateUtc = yesterday.toISOString().substring(0, 10);
var request = {
    path: apiReqDateUtc
};
var response = curl('GET', exchangeRateApi, request);
```



# Hands-on Lab

## > Eventing

<https://docs.couchbase.com/server/current/eventing/eventing-overview.html>

<https://docs.couchbase.com/server/current/eventing/eventing-Terminologies.html>

<https://docs.couchbase.com/server/current/eventing/eventing-language-constructs.html>

<https://docs.couchbase.com/server/current/eventing/eventing-examples.html>

<https://docs.couchbase.com/server/current/eventing/eventing-debugging-and-diagnosability.html>



# Eventing 서비스 처리 절차

## 1. Eventing Storage 구성

- Eventing 서비스를 처리하기 위한 메타 데이터를 저장하는 Collection
- 가능한 빠른 서비스를 위해 메모리를 사용하도록 설정. (Ephemeral Bucket을 사용)
- 기존 collection을 지정해도 되고, 신규로 만들어서 지정해도 됨.
- 여러 Eventing 서비스에서 공유해서 사용이 가능함.

## 2. 소스(Listen) collection/Scope 지정.

- 특정 Collection 혹은 Scope 를 지정할 수 있음.
- 지정된 Collection/Scope에 있는 문서(Json Doc)가 변경 되었을 때 Eventing Function이 수행됨.

## 3. Binding(Alias) 설정.

- Eventing Function 에서 기능 구현을 쉽게 하기 위해 Alias 지정
- 지정하지 않아도 되며, 하나 혹은 그 이상을 지정할 수 있음.
- 버킷, 범위 또는 컬렉션에 액세스, 외부 엔드포인트 URL 및 자격 증명 허용, 상수 정의 등을 환경변수와 같이 지정

## 4. Function 생성.

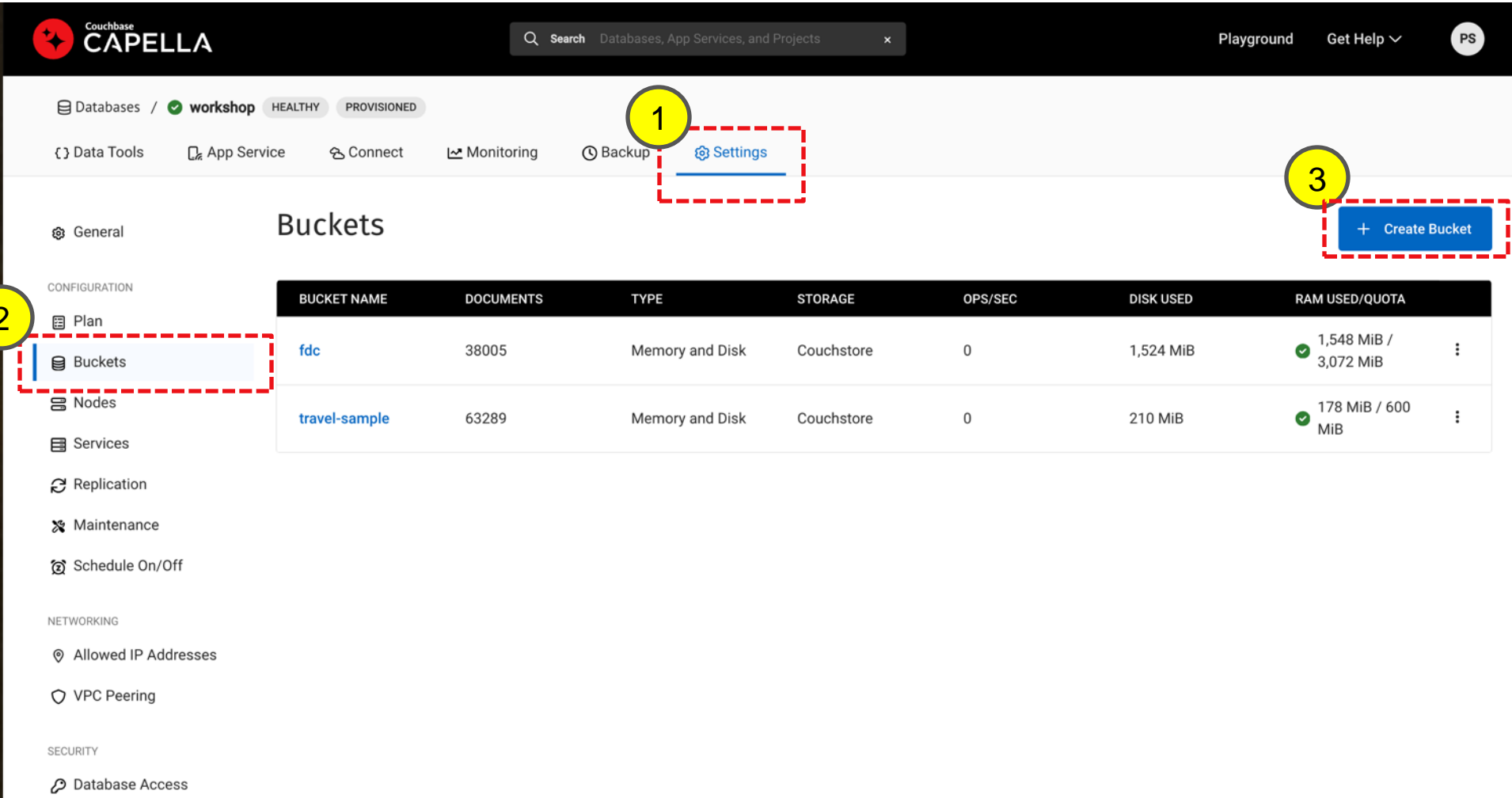
- Java Script로 생성.
- 원하는 비즈니스 로직을 구현

## 5. 생성된 Function를 Enable(Deploy) 해야 함.

- 생성 -> **Deploy** -> Deployed -> **Pause** -> Stop(수정/변경) -> **Resume** -> Deployed
- 생성 -> **Deploy** -> Deployed -> **Pause** -> Stop -> **Undeploy** -> 삭제
- **Function은 Json 파일로 Export / Import 가능**
- Deployed된 Function 에 대한 Log 확인, 처리 현황 확인 가능.

# Eventing Exercise : 준비

## 1-1. Eventing Storage Bucket/Scope/Collection 생성 : EventMeta > travel-sample > meta



**CAPELLA** Search Databases, App Services, and Projects Playground Get Help PS

Databases / **workshop** HEALTHY PROVISIONED

Data Tools App Service Connect Monitoring Backup **Settings**

General Buckets

CONFIGURATION

- Plan
- Buckets**
- Nodes
- Services
- Replication
- Maintenance
- Schedule On/Off

NETWORKING

- Allowed IP Addresses
- VPC Peering

SECURITY

- Database Access

BUCKET NAME	DOCUMENTS	TYPE	STORAGE	OPS/SEC	DISK USED	RAM USED/QUOTA
fdc	38005	Memory and Disk	Couchstore	0	1,524 MiB	✓ 1,548 MiB / 3,072 MiB
travel-sample	63289	Memory and Disk	Couchstore	0	210 MiB	✓ 178 MiB / 600 MiB

+ Create Bucket

# Eventing Exercise : Preparation

## 1-2. Eventing Storage Bucket/Scope/Collection 생성 : EventMeta > travel-sample > meta

**Couchbase CAPELLA** Search Databases, App Services, and Projects Playground Get Help PS

Databases / **workshop** HEALTHY PROVISIONED

Data Tools App Service Connect Monitoring Backup **Settings**

**Create Bucket**

**General**

CONFIGURATION

Plan

**Buckets**

Nodes

Services

Replication

Maintenance

Schedule On/Off

NETWORKING

Allowed IP Addresses

VPC Peering

SECURITY

Database Access

Security Certificate

Activity Log

Bucket Name (cannot change later) \*

ForEvent 8/100

**Memory Quota**

Memory Quota (MiB)

100

2,956 MiB available

■ This bucket 100 MiB

■ Other buckets 1,224 MiB

■ Remaining memory 2,856 MiB

**Advanced Settings**

Customize advanced settings for your bucket to change replication and backup settings, conflict resolution method, and more.

**Bucket Type (cannot change later)**

Choose whether to persist your bucket's data to memory and disk, or to hold bucket data in memory only. [Learn More](#)

☐ Memory and Disk

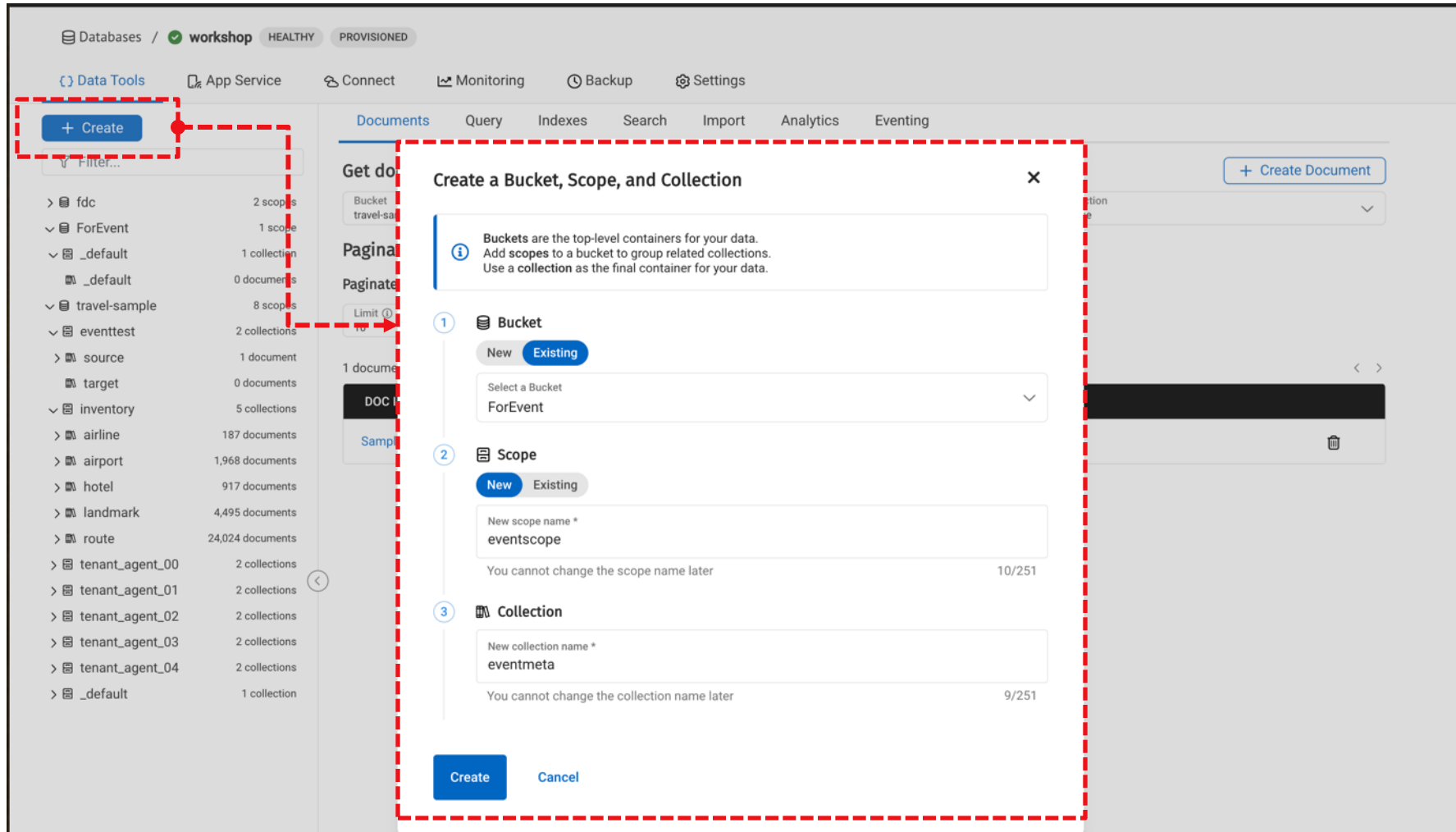
☒ **Memory Only**

You can't import, create app endpoints, or turn off a database with Memory Only buckets.

**Memory Only 를 선택하거나, 아니면 가능한 메모리에 상주하도록 설정 필요**

# Eventing Exercise : Preparation

## 1-3. Eventing Storage Bucket/Scope/Collection 생성 : EventMeta > travel-sample > meta



# Eventing Exercise : Preparation

## 2-1. 테스트용 Collection 생성 : travel-sample > eventtest > source, target

The screenshot shows the Couchbase CAPELLA interface. In the left sidebar, the '+ Create' button is highlighted with a red box, and a red arrow points from it to the 'Create a Bucket, Scope, and Collection' dialog. The dialog is open, and the 'travel-sample' bucket is selected. The 'eventtest' scope is being created, and the 'source' collection is being created. The dialog has three steps: 1. Bucket (Existing), 2. Scope (New), and 3. Collection (New). The 'eventtest' scope is being created, and the 'source' collection is being created. The dialog also includes a 'Create Document' button in the top right corner.

**Create a Bucket, Scope, and Collection**

Buckets are the top-level containers for your data. Add scopes to a bucket to group related collections. Use a collection as the final container for your data.

- 1 Bucket**  
New Existing  
Select a Bucket  
travel-sample
- 2 Scope**  
New Existing  
New scope name \*  
eventtest  
You cannot change the scope name later 9/251
- 3 Collection**  
New collection name \*  
source

# Eventing Exercise : Preparation

## 2-2. 테스트 Document 생성 : source Collection

The screenshot shows the Couchbase CAPELLA web interface. On the left, a sidebar lists various collections under the 'travel-sample' bucket. The 'source' collection is highlighted with a red dashed box. In the center, the 'Create New Document' dialog is open, showing the 'Document ID' field set to 'SampleDocument' and the 'JSON' content field containing a JSON object. A red dashed box highlights the '+ Create Document' button in the top right corner of the dialog. A yellow callout box on the right contains the following text:

**Document ID : SampleDocument**  
**Document Content :**  

```
{
  "country": "AD",
  "ip_start": "5.62.60.1",
  "ip_end": "5.62.60.9"
}
```

**Save 클릭!**

# Eventing Exercise : Preparation

## 2-3. 테스트 Document 생성 확인

The screenshot displays the Couchbase CAPELLA web interface. The top navigation bar includes the Couchbase logo, a search bar, and links for 'Playground' and 'Get Help'. The main interface is divided into a left sidebar and a central content area.

**Left Sidebar:** A tree view under 'Databases / workshop' shows various collections and their document counts. The 'travel-sample' collection is highlighted, showing 8 scopes and 2 collections.

**Central Content Area:** The 'Documents' tab is active. It features a 'Get documents from' section with three dropdown menus: 'Bucket' (set to 'travel-sample'), 'Scope' (set to 'eventtest'), and 'Collection' (set to 'source'). A '+ Create Document' button is located to the right. Below this is the 'Paginate and Filter documents' section, which includes a 'Filter by' dropdown (set to 'ID'), a 'Limit' of 10, an 'Offset' of 0, and a 'DOC ID' field. A 'Get Documents' button is present. The results section shows '1 documents | limit 10 | offset 0' and a table with two columns: 'DOC ID' and 'DOCUMENT'. The first row shows 'SampleDocument' with the document content: `{ "country": "AD", "ip_start": "5.62.60.1", "ip_end": "5.62.60.9" }`. Red dashed boxes highlight the 'Get documents from' section and the document list table.



# Eventing Exercise : Eventing Workbench

## 3-1. Eventing Function 생성

Operational Clusters / ✓ ev\_test HEALTHY

[Data Tools](#) [App Service](#) [Connect](#) [Monitoring](#) [Backup](#) [Logs](#)

[Documents](#) [Query](#) [Indexes](#) [Search](#) [Import](#) [Analytics](#) [Eventing](#)

### Eventing

Use Eventing Functions to trigger your own business logic and respond in real time when application interactions create changes in your data, known as document mutations (insert, update, delete, expire). Write your business logic as a JavaScript fragment and use Timers to schedule further processing. [Learn more](#)

[Import Function](#) [+ Add Function](#)

FUNCTION NAME	SOURCE BUCKET	SOURCE SCOPE	SOURCE COLLECTION	STATUS				
ip-enrichment	travel-sample	eventing	ipsource	✓ Deployed				

# Eventing Exercise : Eventing Function

## Eventing Function 생성 : 기본 정보 설정

Name : **enrichment**

Source : **travel-sample > eventing > ipsource**

Eventing Storage : **evmeta > evmeta-scope > evmeta-ip-example**

Worker : **1**

Deployment Feed Boundary : **From Now** or **Everything**

SQL++ Consistency : **None** or **Request**

Deployment Feed Boundary : **From Now** or **Everything**

[Documents](#) [Query](#) [Indexes](#) [Search](#) [Import](#) [Analytics](#) [Eventing](#)

[← Back to the Eventing Listing](#)

Add a new Function

Functions offer a computing paradigm that you can use to handle data changes. In the Couchbase cluster, you can use the Functions to process and respond to data-changes according to an Event-Condition-Action model. If you need more information on how to create a function, refer to the [Documentation](#)

1

Settings

2

Bindings

3

Javascript

Name and describe the function

Name\* ⓘ  
enrichment

Description  
Type description of the function here

Listen to Location ⓘ

Bucket\*  
travel-sample

Scope\*  
eventing

Collection\*  
ipsource

Eventing Storage ⓘ

Bucket\*  
evmeta

Scope\*  
evmeta-scope

Collection\* ⓘ  
evmeta-ip-example

Other settings of the function

Workers ⓘ  
1

Script Timeout ⓘ  
60

Timer Context Max Size ⓘ  
1024

SQL++ Consistency ⓘ  
None

Deployment Feed Boundary ⓘ  
Everything

Language Compatibility ⓘ  
7.2.0

Next >

1

# Eventing Exercise : Eventing Function

## Eventing Function 생성 : Binding 변수 설정

Documents Query Indexes Search Import Analytics Eventing

[← Back to the Eventing Listing](#)

Add a new Function

Functions offer a computing paradigm that you can use to handle data changes. In the Couchbase cluster, you can use the Functions to process and respond to data-changes according to an Event-Condition-Action model. If you need more information on how to create a function, refer to the [Documentation](#)

1

Settings

2

Bindings

3

JavaScript

Create bindings for your function

A binding is a construct that allows the separation of environment-specific variables (such as accessing buckets, scopes, or collections, allowing external endpoint URLs and credentials, and defining constants) for access in the handler's source code. An Eventing Function can have no bindings, just one binding, or several bindings. For more information, refer to [Bindings](#)

+ Add Binding

There are no bindings configured.

Use "Add Binding" above to create them.

[< Previous](#)

[Next >](#)

# Eventing Exercise : Eventing Function

## Eventing Function 생성 : Binding 변수 설정

DocumentsQueryIndexesSearchImportAnalyticsEventing

[← Back to the Eventing Listing](#)

Add a new Function

Functions offer a computing paradigm that you can use to handle data changes. In the Couchbase cluster, you can use the Functions to process and respond to data-changes according to an Event-Condition-Action model. If you need more information on how to create a function, refer to the [Documentation](#)

1

Settings

2

Bindings

3

JavaScript

Create bindings for your function

A binding is a construct that allows the separation of environment-specific variables (such as accessing buckets, scopes, or collections, allowing external endpoint URLs and credentials, and defining constants) for access in the handler's source code. An Eventing Function can have no bindings, just one binding, or several bindings. For more information, refer to [Bindings](#)

View Bindings

1

BucketURLConstant

Allows Javascript in a Function to access Couchbase collections from the Data Service.

Alias Name ⓘtgt

PermissionRead and Write

Buckettravel-sample

Scopeeventing

Collectionipsource

2

BucketURLConstant

Allows Javascript in a Function to access Couchbase collections from the Data Service.

Alias Name ⓘsum

PermissionRead and Write

Buckettravel-sample

Scopeeventing

Collectionsummary

< Previous

Next >

3

# Eventing Exercise : Eventing Function

## Eventing Function 생성 : JavaScript 작성

[← Back to the Eventing Listing](#)

Update Function Settings

Functions offer a computing paradigm that you can use to handle data changes. In the Couchbase cluster, you can use the Functions to process and respond to data-changes according to an Event-Condition-Action model. If you need more information on how to create a function, refer to the [Documentation](#)

1

Settings

2

Bindings

3

JavaScript

Edit JavaScript code for your function

Provide an OnUpdate or OnDelete function with your custom business logic.

```
1 function OnUpdate(doc, meta) {
2   log('document', doc);
3   doc["ip_num_start"] = get_numip_first_3_octets(doc["ip_start"]);
4   doc["ip_num_end"] = get_numip_first_3_octets(doc["ip_end"]);
5   tgt[meta.id]=doc;
6
7   var results =
8     | SELECT sum(ip_num_end) AS total
9     | FROM `travel-sample`.`eventing`.`iptarget`;
10
11   for (var item of results) { // Stream results using 'for' iterator.
12     | var total = item.total;
13     | log('total', total);
14   }
15   results.close();
16 }
```

[< Previous](#)[Save](#)

1

```
function OnUpdate(doc, meta) {
  log('document', doc);
  doc["ip_num_start"] = get_numip_first_3_octets(src["ip_start"]);
  doc["ip_num_end"] = get_numip_first_3_octets(src["ip_end"]);
  tgt[meta.id]=doc;
}

function get_numip_first_3_octets(ip) {
  var return_val = 0;
  if (ip) {
    var parts = ip.split('.');
    //IP Number = A x (256*256*256) + B x (256*256) + C x 256 + D
    return_val = (parts[0]*(256*256*256)) + (parts[1]*(256*256)) + (parts[2]*256) +
    parseInt(parts[3]);
    return return_val;
  }
}
```

Save 클릭!

# Eventing Exercise : Eventing Function

## Eventing Function 생성 확인 및 구동(Deploy)

DocumentsQueryIndexesSearchImportAnalyticsEventing

Eventing

Use Eventing Functions to trigger your own business logic and respond in real time when application interactions create changes in your data, known as document mutations (insert, update, delete, expire). Write your business logic as a JavaScript fragment and use Timers to schedule further processing. [Learn more](#)

Import Function+ Add Function

FUNCTION NAME	SOURCE BUCKET	SOURCE SCOPE	SOURCE COLLECTION	STATUS
ip-enrichment	travel-sample	eventing	ipsource	Paused

Deployment Statistics

Success

0

Failure

0

Timeout

0

Backlog

0

Log

Edit JavaScript

Settings

Undeploy

Export

Resume

Undeploy

Export

Pause

# Eventing Exercise : Eventing Function

## Eventing Function 적용 확인

The screenshot shows the Couchbase CAPELLA console interface. A 'Create New Document' dialog is open in the center. The dialog has a title bar with a close button. It contains a 'Destination Collection' field with the value 'travel-sample . eventtest . source'. Below that is a 'Document ID' field with the value 'AnotherSampleDocument'. The 'JSON' tab is selected, showing a JSON document structure with the following content:

```
1 {  
2   "country": "RU",  
3   "ip_start": "7.12.60.1",  
4   "ip_end": "7.62.60.9"  
5 }
```

In the background, the console shows a list of databases and collections. A red dashed box highlights the 'Create New Document' dialog. A yellow circle with the number '1' is placed over the '+ Create Document' button in the background. A yellow circle with the number '2' is placed over the 'Document ID' field in the dialog.

source Collection에 Document 신규 생성!

Document ID : AnotherSampleDocument

Document Content :

```
{  
  "country": "RU",  
  "ip_start": "7.12.60.1",  
  "ip_end": "7.62.60.9"  
}
```

# Eventing Exercise : Eventing Function

## Eventing Function 적용 확인

The screenshot displays the Couchbase CAPELLA web interface. The top navigation bar includes the Couchbase logo, a search bar, and links for 'Playground', 'Get Help', and a user profile 'PS'. The main interface is divided into a sidebar on the left and a main content area. The sidebar shows a tree view of databases and collections, with 'workshop' selected. The main content area has tabs for 'Documents', 'Query', 'Indexes', 'Search', 'Import', 'Analytics', and 'Eventing'. The 'Documents' tab is active, showing a 'Get documents from' section with a dropdown for 'Bucket' (travel-sample) and 'Scope' (eventtest). A 'Collection' dropdown is set to 'target', highlighted with a red box and a yellow circle labeled '1'. Below this is a 'Paginate and Filter documents' section with a 'Filter by' dropdown set to 'ID', a 'Limit' of 10, and an 'Offset' of 0. A 'Get Documents' button is highlighted with a red box and a yellow circle labeled '2'. A table of documents is shown below, with one document highlighted by a red dashed box and a yellow circle labeled '3'. The document is titled 'AnotherSampleDocument' and contains a JSON object with fields like 'country', 'ip\_start', 'ip\_end', 'ip\_num\_start', and 'ip\_num\_e...'. A yellow banner at the bottom right reads 'target Collection에 Document 신규 생성 확인'.





# Thank you!



[Paul.Son@couchbase.com](mailto:Paul.Son@couchbase.com)

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

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



**Couchbase**