Couchbase

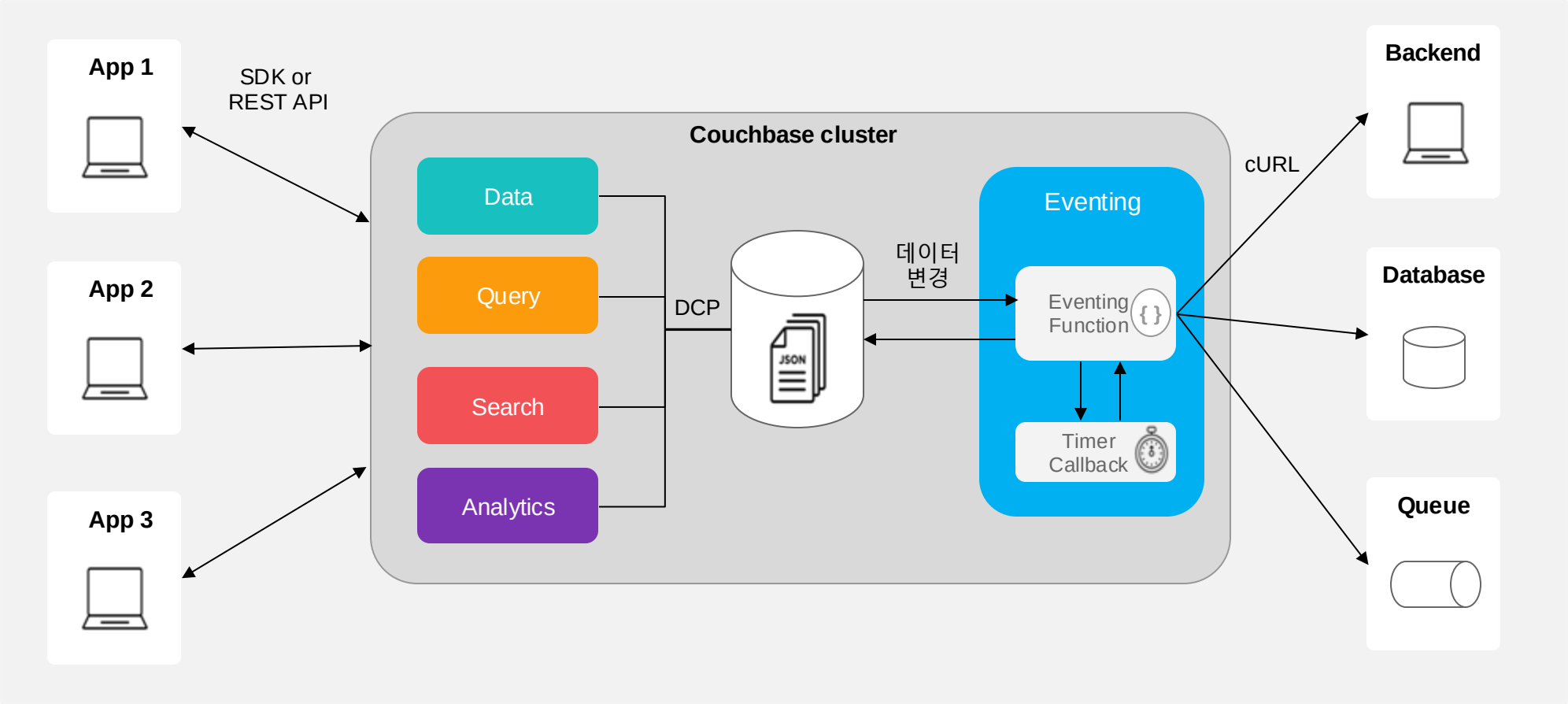
Couchbase Eventing Workshop

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.

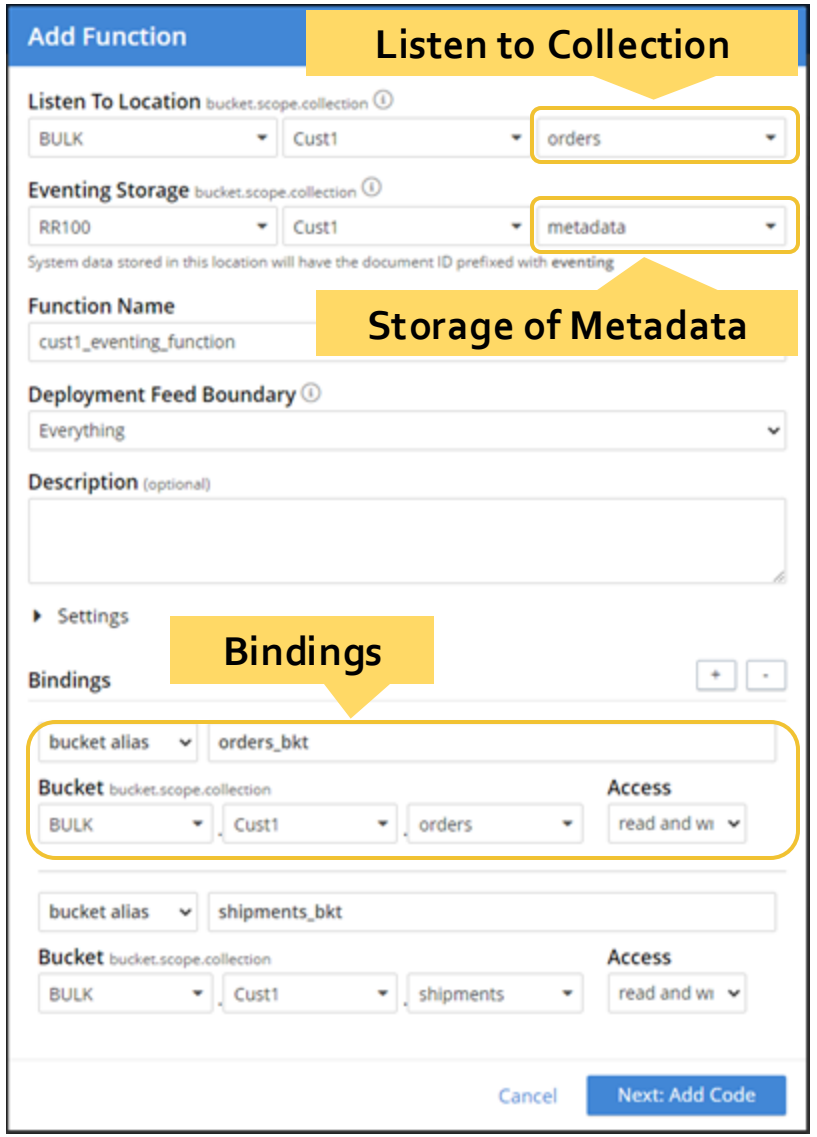
- **Functions to process and respond to data-changes in an Event-Condition-Action model.**
- **Functions are written in JavaScript code to analyze and manipulate your JSON documents on mutations.**
- **Typical Use Cases:**
  - Threshold-based alerts. 임계값 기반 경고.
  - Monitoring parameters 모니터링 매개변수
  - 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.**



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(twoMins.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

Couchbase

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 변결 되었을 때 Eventing Function이 수행됨.

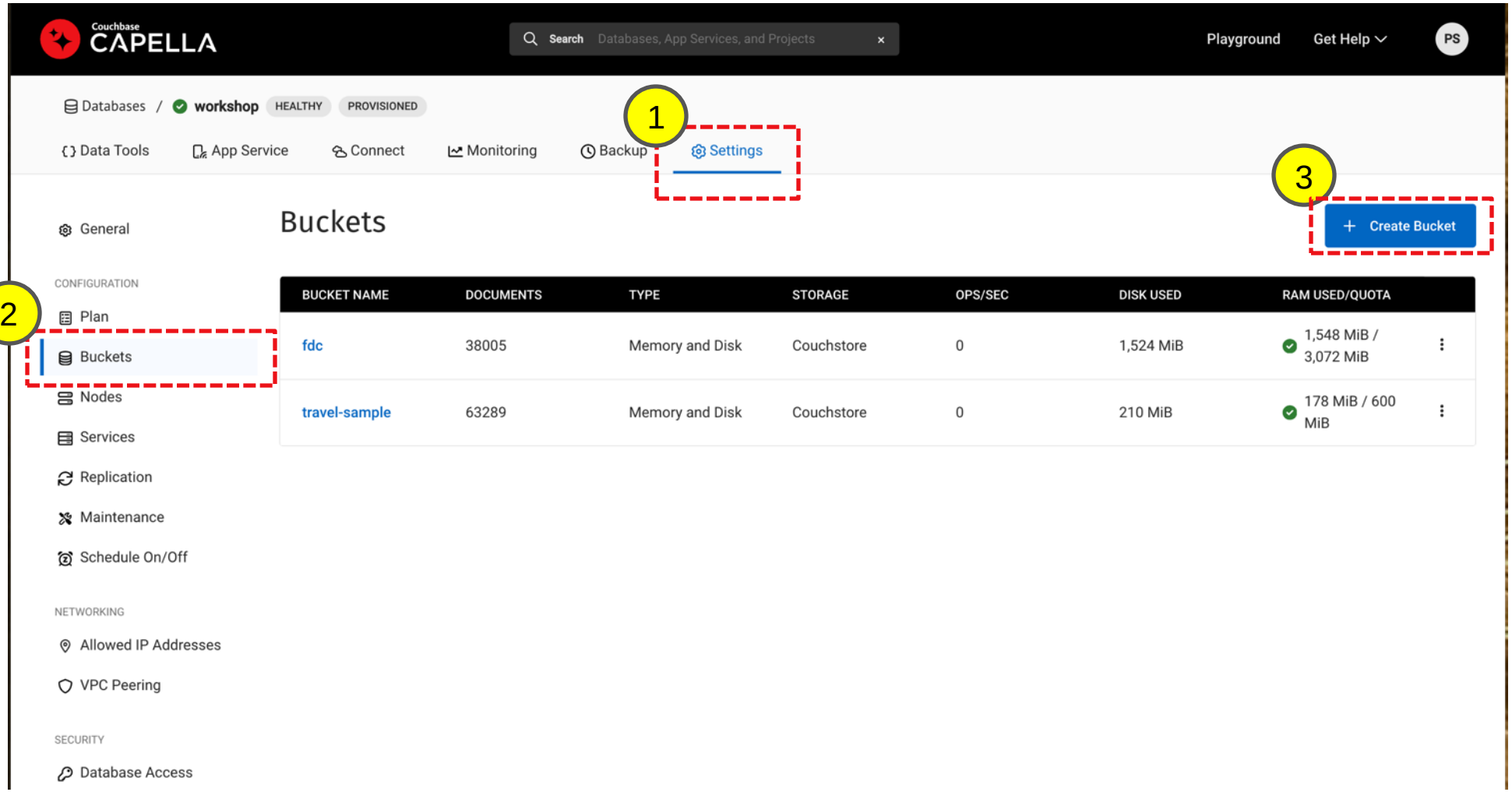
3. Binding(Alias) 설정.
- Eventing Function 에서 기능 구현을 쉽게 하기 위해 Alias 지정
- 지정하지 않아도 되며, 하나 혹은 그 이상을 지정할 수 있음.
- 버킷, 범위 또는 컬렉션에 액세스, 외부 엔드포인트 URL 및 자격 증명 허용, 상수 정의 등을 환경변수와 같이 지정

4. Function 생성.
- Java Script로 생성.
- 원하는 비지니스 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.

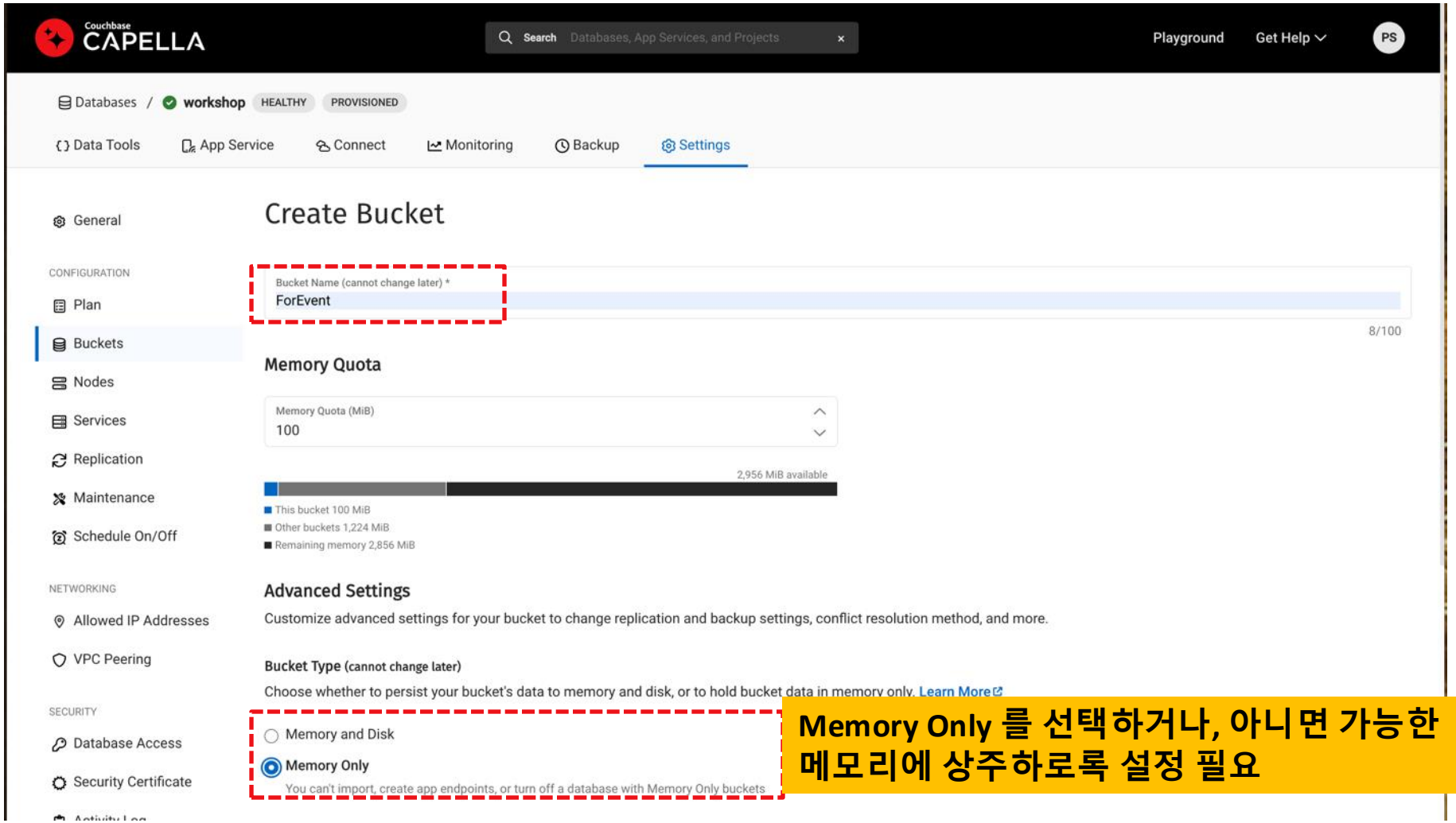
Eventing Exercise : 준비

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



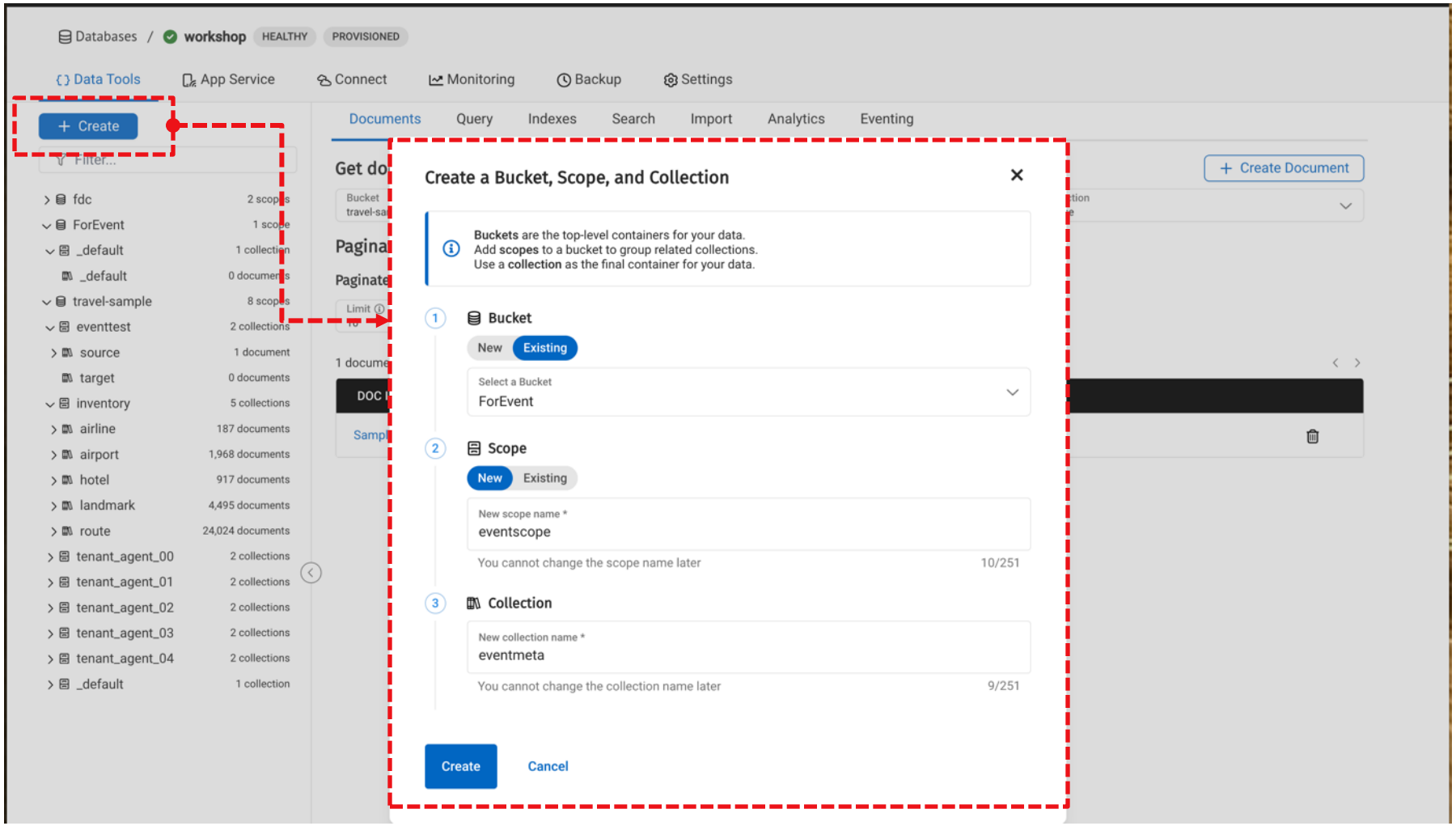
Eventing Exercise : Preparation

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



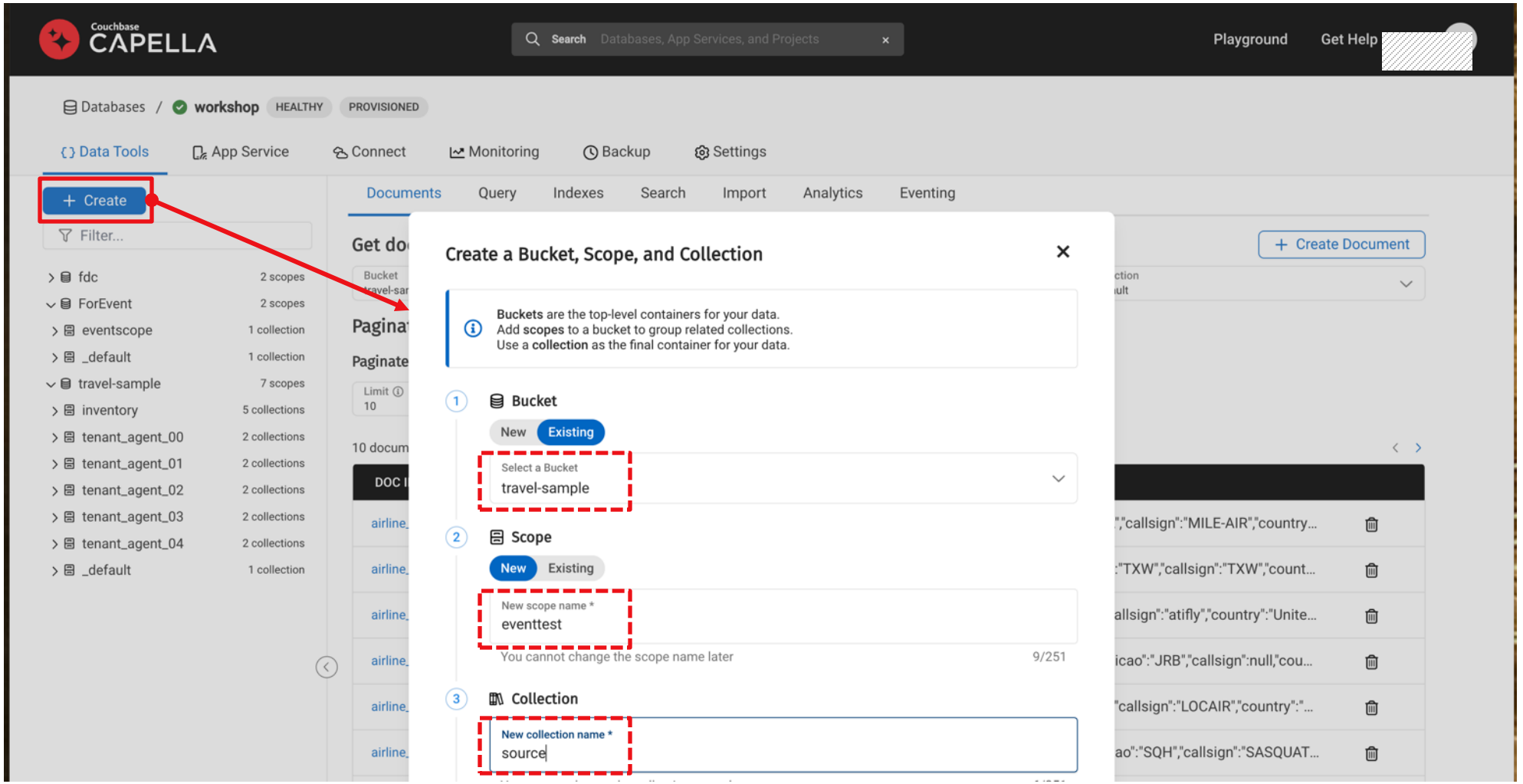
Eventing Exercise : Preparation

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



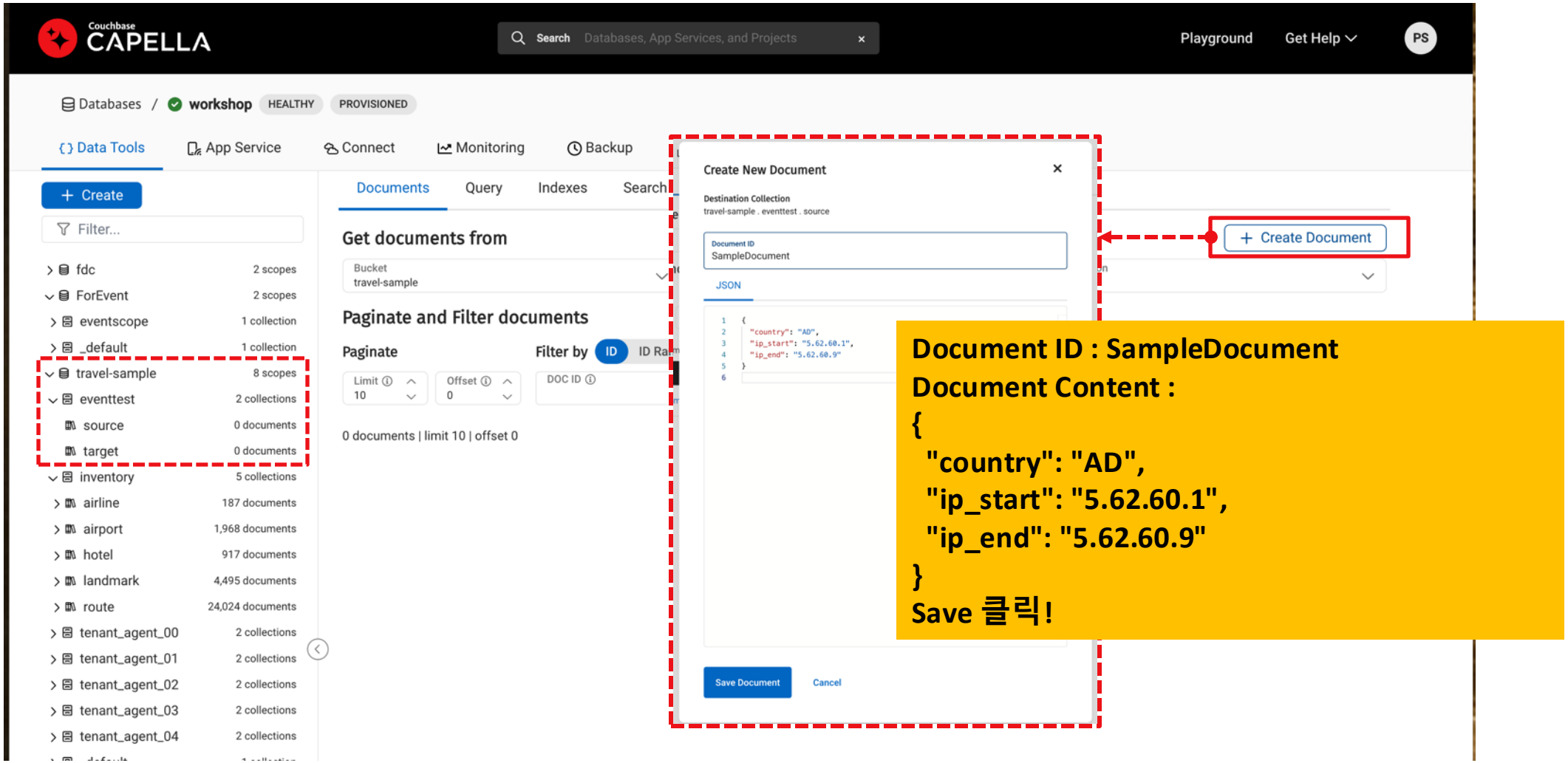
Eventing Exercise : Preparation

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



Eventing Exercise : Preparation

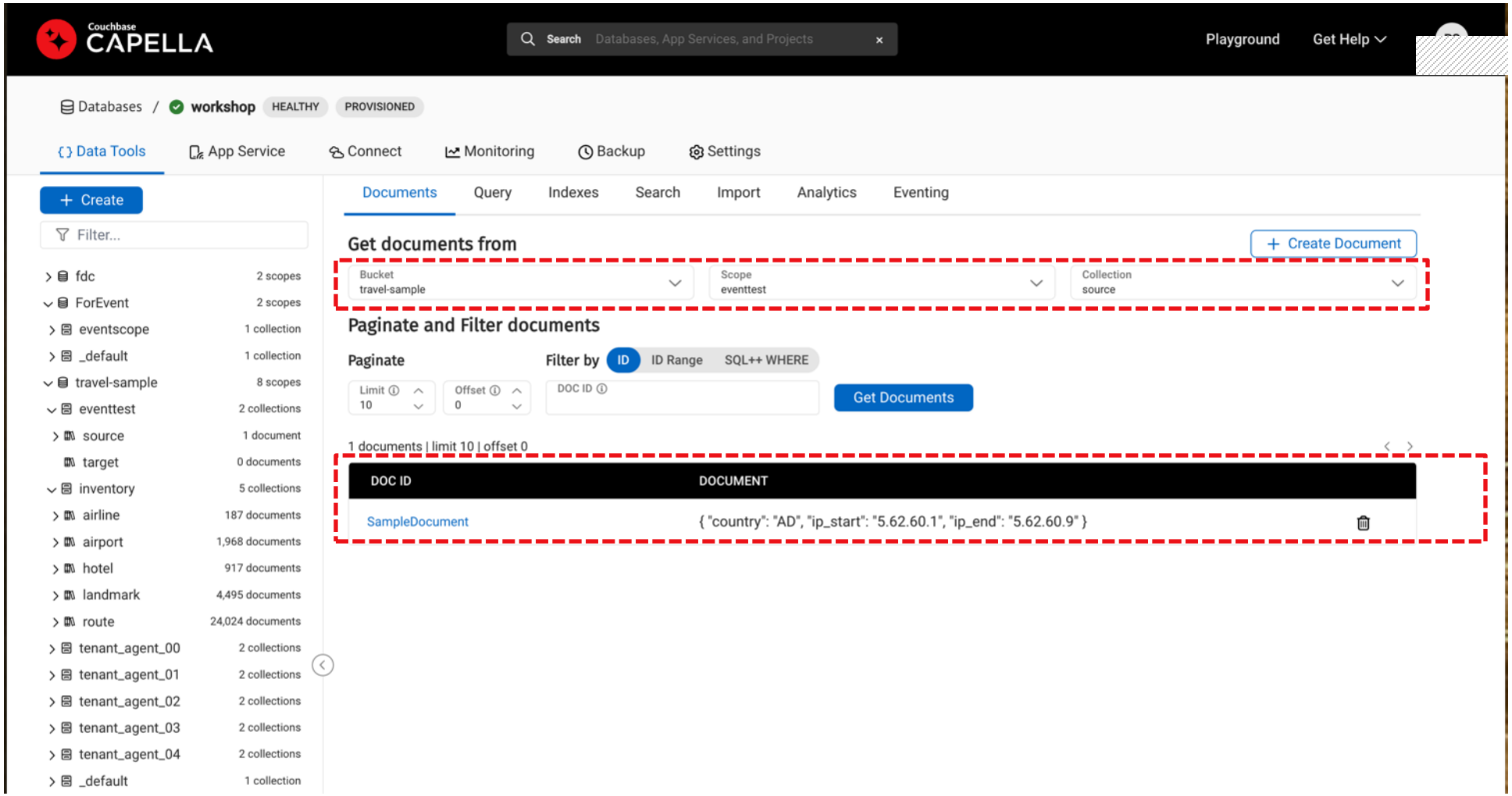
2-2. 테스트 Document 생성 : source Collection



Document ID : SampleDocument
Document Content :
{
"country": "AD",
"ip_start": "5.62.60.1",
"ip_end": "5.62.60.9"
}
Save 클릭!

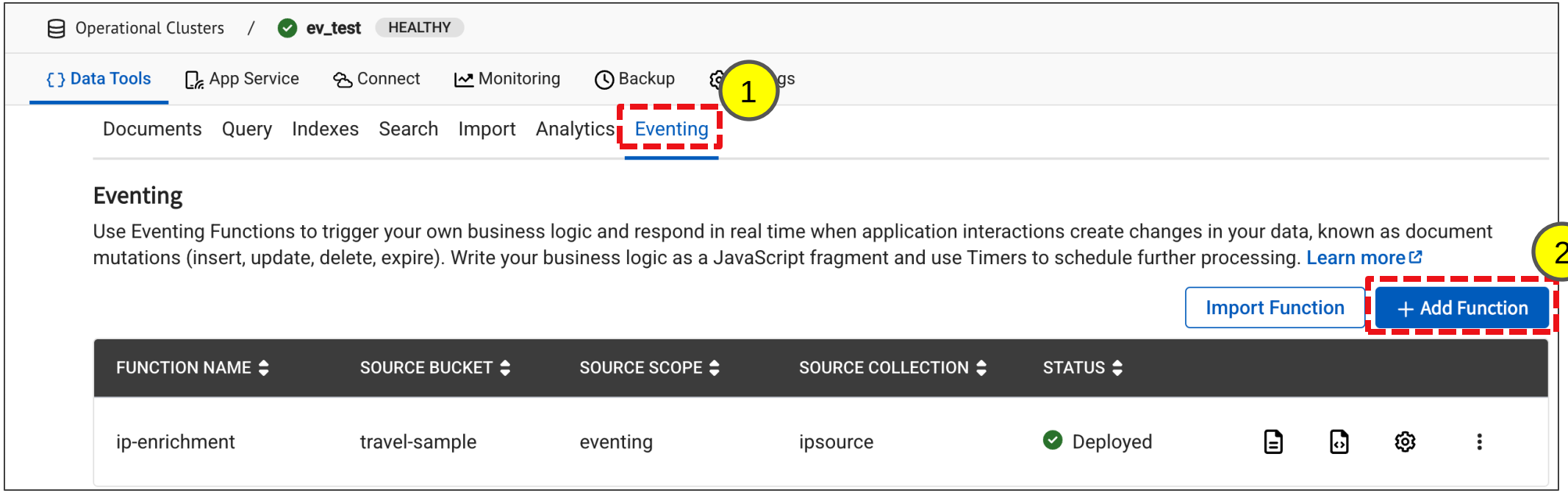
Eventing Exercise : Preparation

2-3. 테스트 Document 생성 확인



Eventing Exercise : Eventing Workbench

3-1. Eventing Function 생성



Eventing Exercise : Eventing Function

Eventing Function 생성 : 기본 정보 설정

Name : enrichment

Source : travel-sample > eventing > ipsource

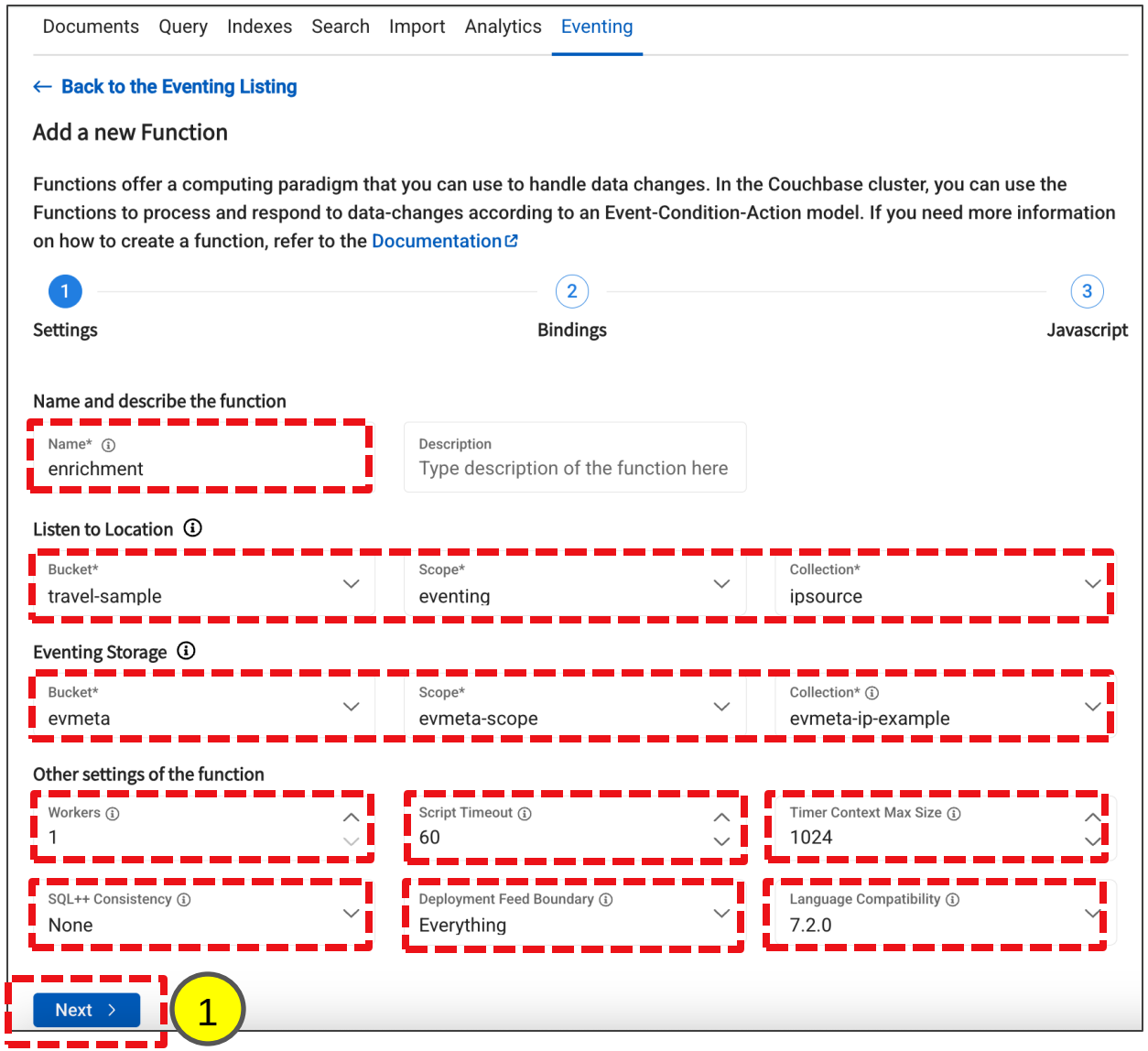
Eventing Storage : evmeta > evmeta-scope > evmeta-ip-example

Worker : 1

Deployment Feed Boundary : From Now or Everything

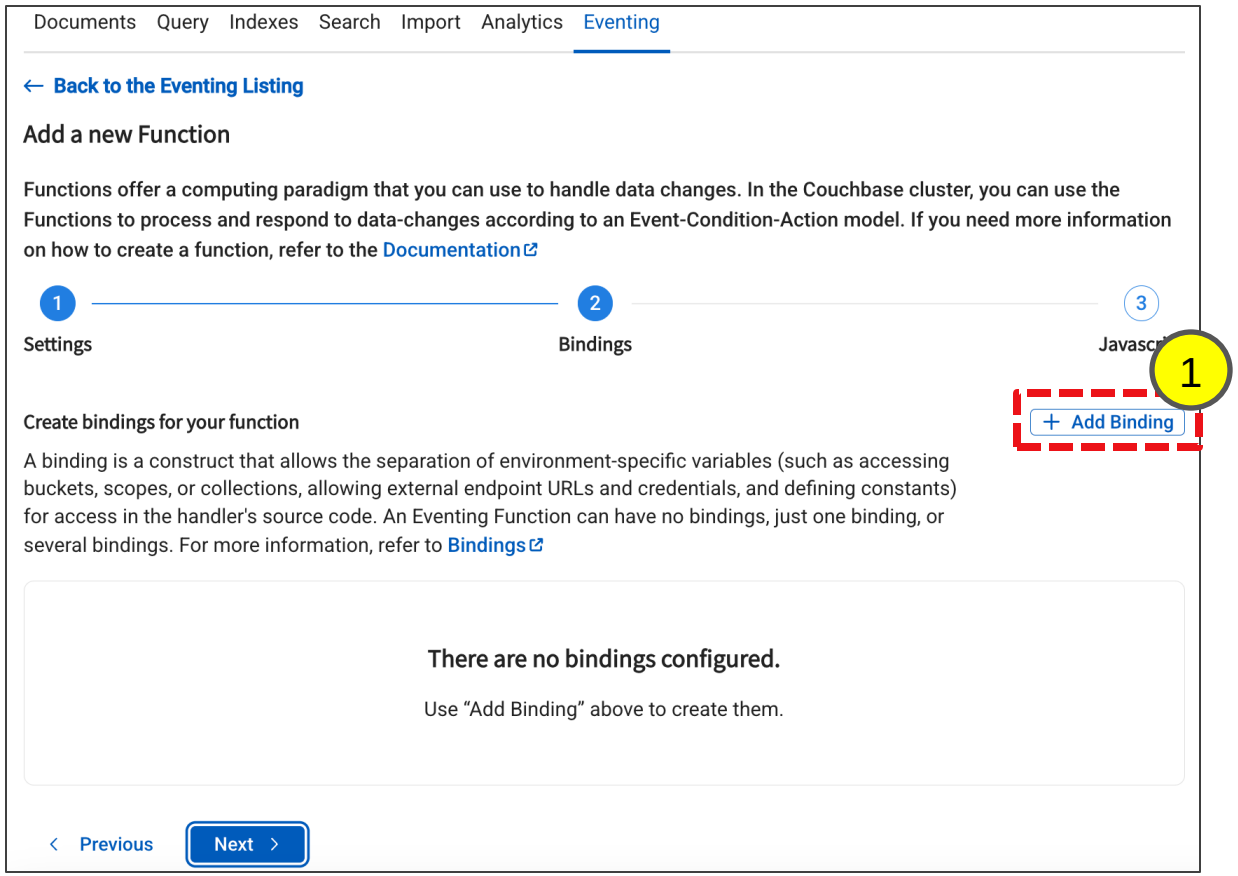
SQL++ Consistency : None or Request

Deployment Feed Boundary : From Now or Everything



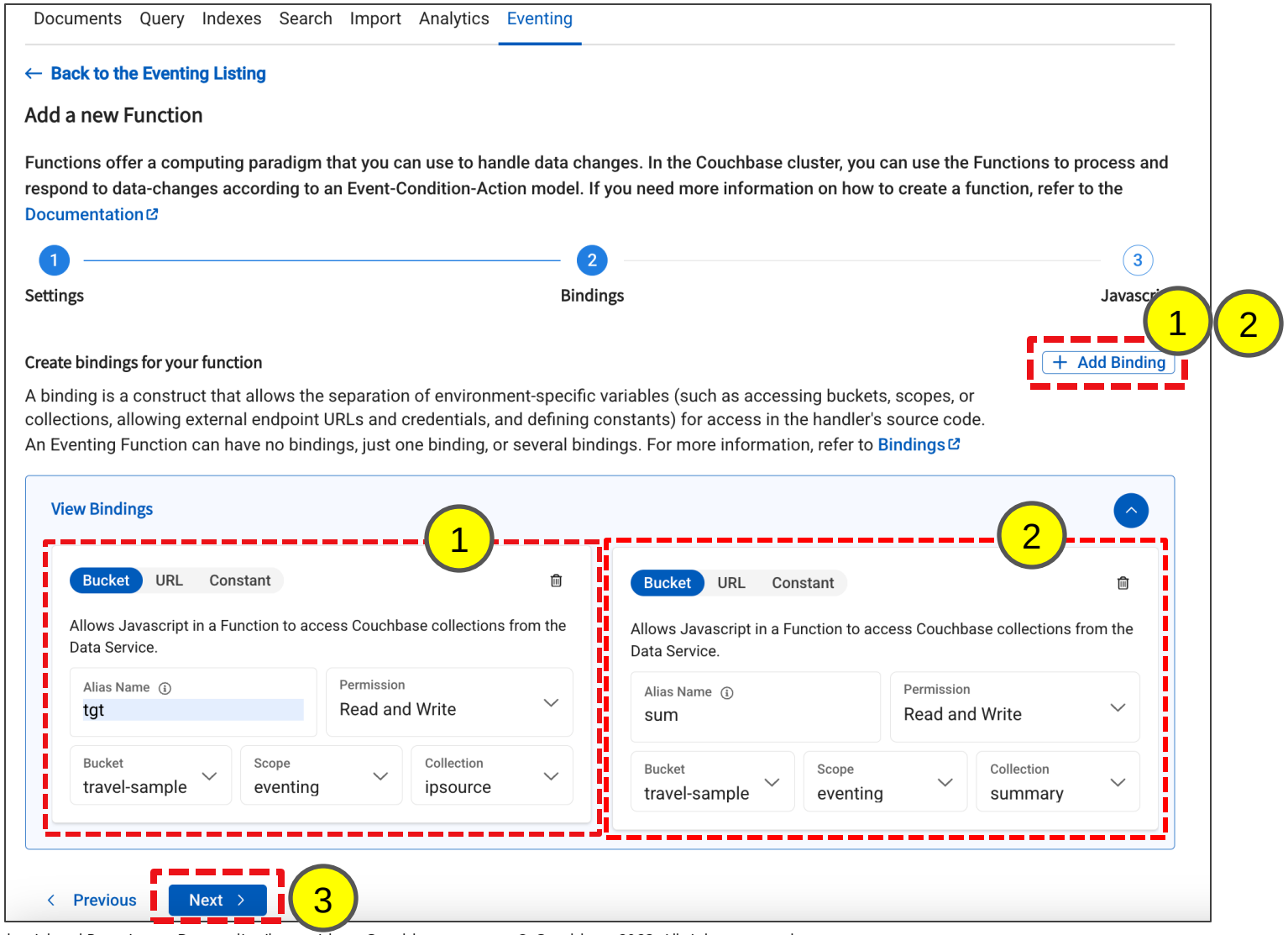
Eventing Exercise : Eventing Function

Eventing Function 생성 : Binding 변수 설정



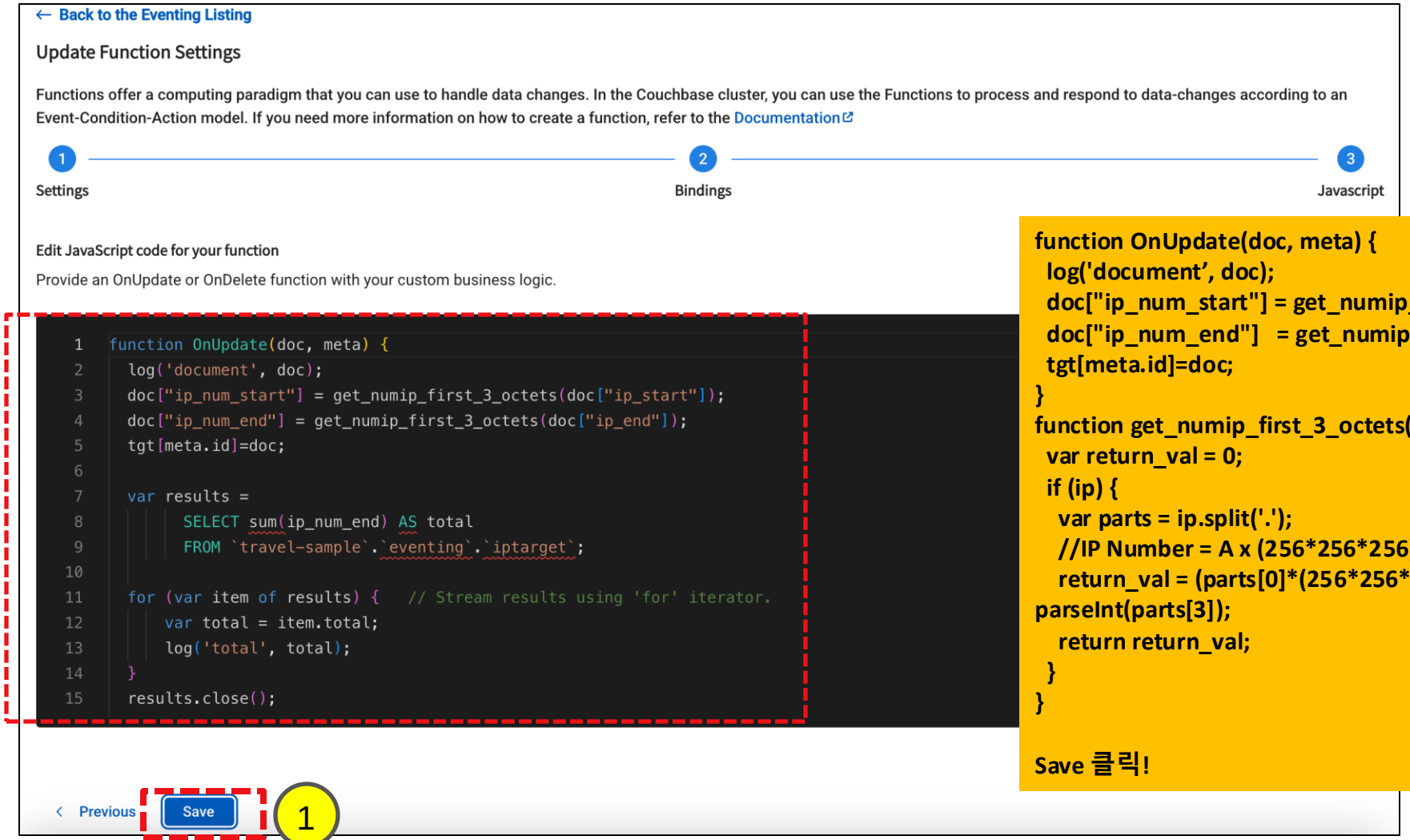
Eventing Exercise : Eventing Function

Eventing Function 생성 : Binding 변수 설정



Eventing Exercise : Eventing Function

Eventing Function 생성 : JavaScript 작성

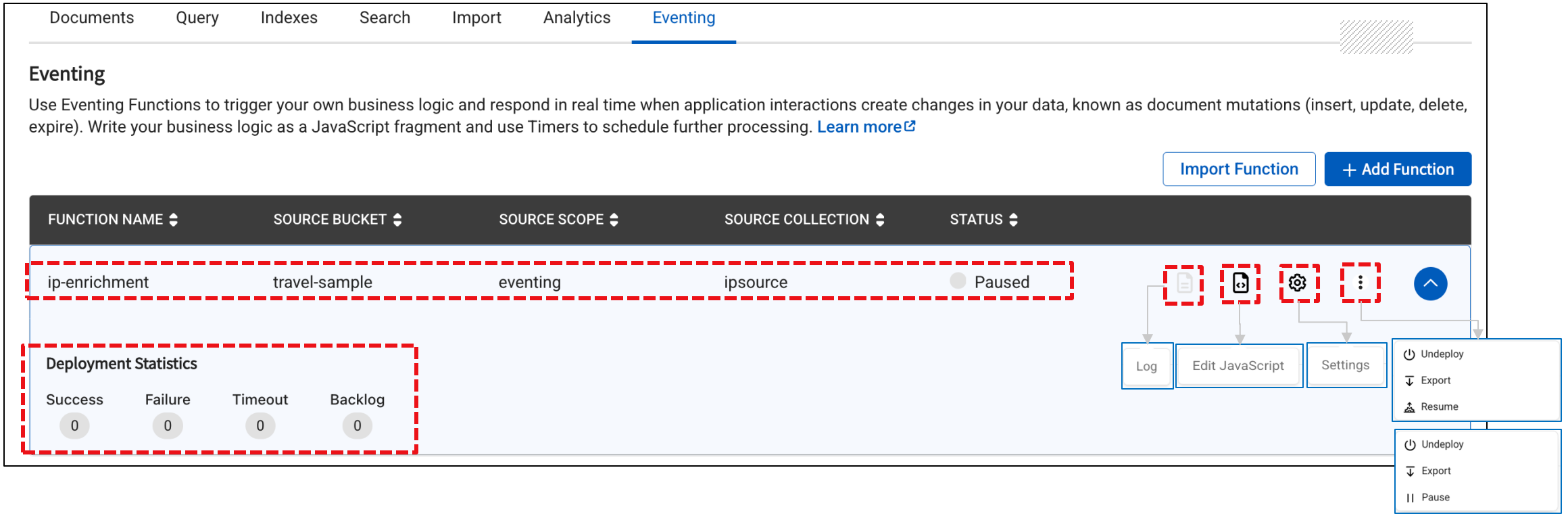



```
function OnUpdate(doc, meta) {
 log('document', doc);
 doc["ip_num_start"] = get_numip_first_3_octets(src["ip_start"]);
 doc["ip_num_end"]   = get_numip_first_3_octets(src["ip_end"]);
 tgt[meta.id]=doc;
}
function get_numip_first_3_octets(ip) {
 var return_val = 0;
 if (ip) {
  var parts = ip.split('.');
  //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!

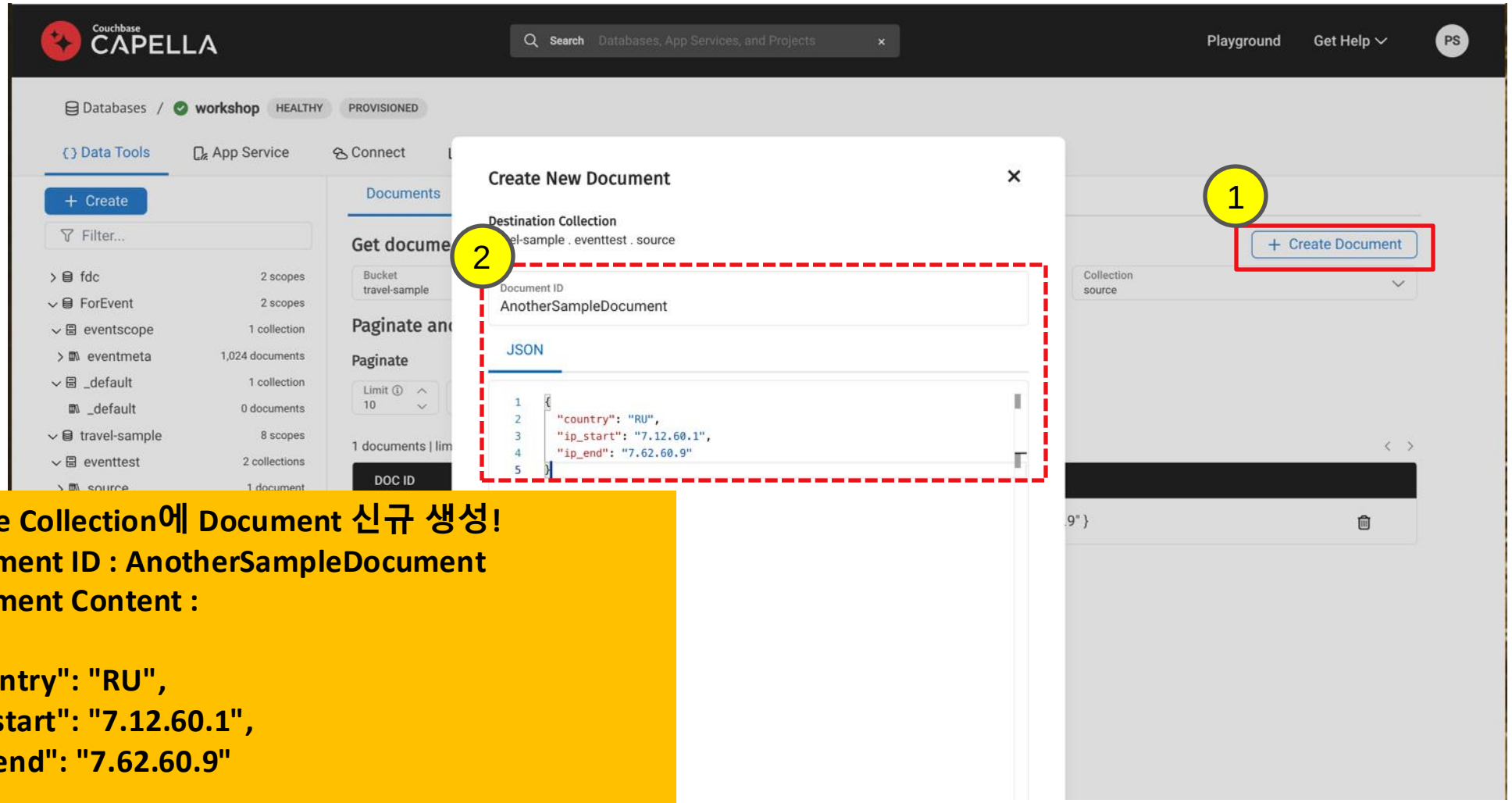
Eventing Exercise : Eventing Function

Eventing Function 생성 확인 및 구동(Deploy)



Eventing Exercise : Eventing Function

Eventing Function 적용 확인

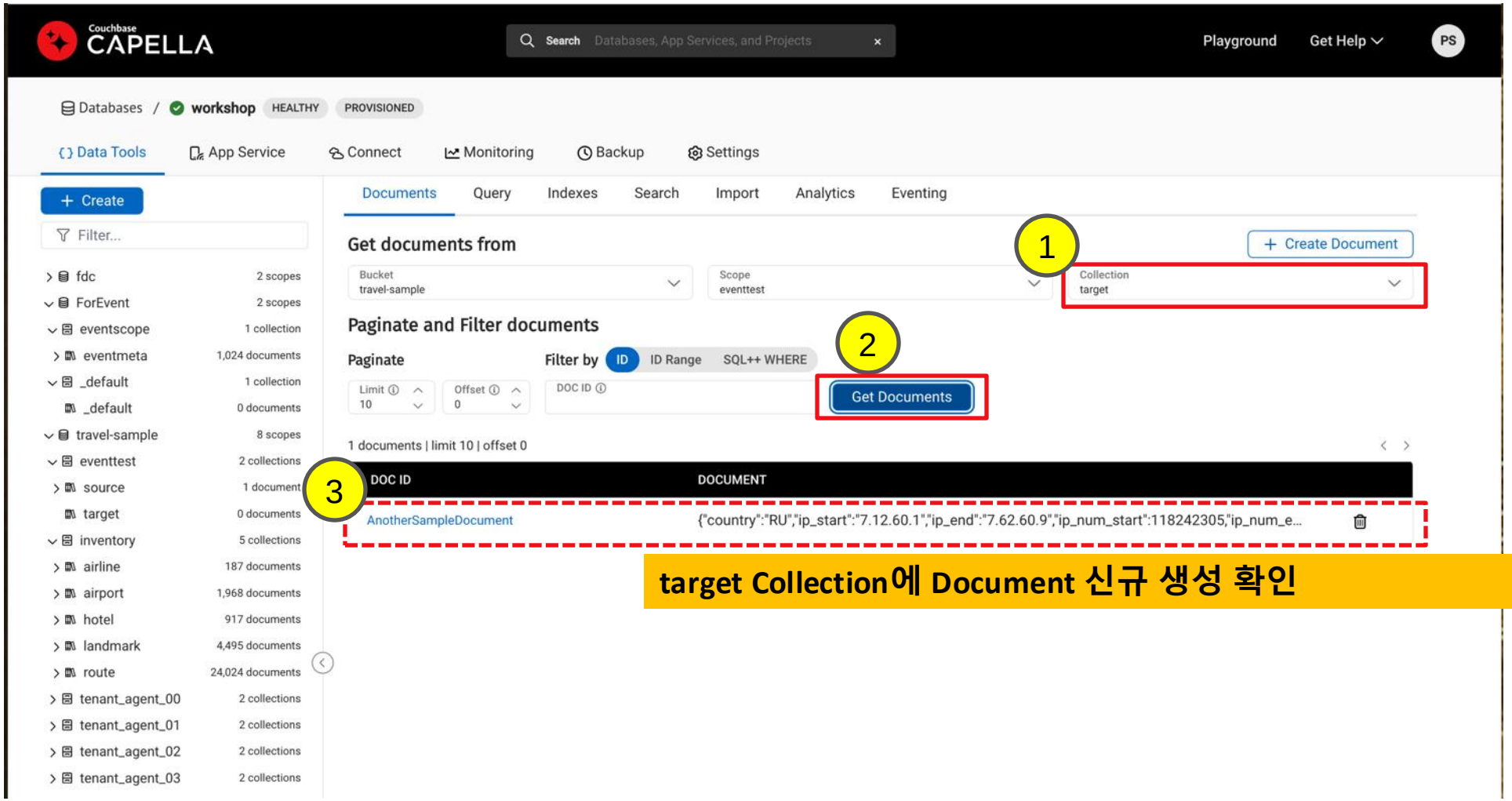


source Collection에 Document 신규 생성!
Document ID : AnotherSampleDocument
Document Content :

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

Eventing Exercise : Eventing Function

Eventing Function 적용 확인



Thank you!

Paul.Son@couchbase.com

www.couchbase.com

cloud.couchbase.com

Couchbase