

# ¡Más servicios!

//Más PaaS

IT BOARDING

BOOTCAMP



# // Roadmap



## Qué vimos hasta ahora

- Servicio de comunicación asíncrona: **BigQueue** (AKA: BQ)
- Servicio de bases de datos clave valor: **Key Value Store** (AKA: KVS)
- Servicio de bases de datos documentales y motor de búsqueda: **Document Search** (AKA: DS)



## Qué vamos a ver hoy

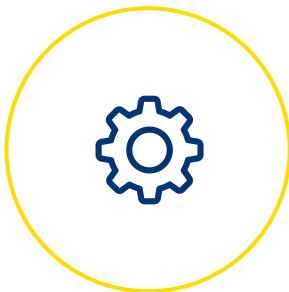
- Tareas programadas: **Jobs**
- Storage de propósito general: **Object Storage**
- Trazabilidad y persistencia de eventos: **Audits**
- Data streaming: **Streams**
- ¡Aún hay más!:
  - **Bazooka**
  - **Cache**
  - **Configs**
  - **Locks**
  - **Sequences**
  - **Sessions**

# // Jobs



## < QUÉ >

Permite definir tareas programadas, posiblemente de larga duración



## < SIMPLE >

Solo hace falta definir el schedule, y el endpoint de tu aplicación que iniciará el proceso a ejecutar



## < MONITOREO >

Se garantiza que la instancia donde corre el job no será “decomisada”, se puede conocer el estado de una ejecución, alertas predeterminadas por ejecuciones fallidas

# // Jobs - Create

- *Criticality*: Criticalidad de tu job.
- *Name*: Sin comentarios :)
- *Process Name*: Permite agrupar jobs
- *Endpoint*: Endpoint de la aplicación que ejecutará la tarea.
- *Timerule*: La cron expression que define el calendario de ejecución.
- *Retries*: Reintentos para una ejecución, mínimo 0, máximo 10.
- *Retries Interval (in minutes)*: Intervalo entre reintentos.
- *Scope*: El scope de Jobs donde se debe ejecutar la tarea.
- *Expected run time*: Cuánto tiempo esperamos que tome la tarea.
- *Tags*: Core metric & Core metric site.

IT BOARDING

BOOTCAMP

Create job

Create

Cancel

## Criticality

Level

Low



## Service

Name

test-jobs-docs

Process Name

jobs-docs



End Point

/test-jobs-docs



Time Rule

0 2,5,8 \*\*\*

At 02:00 AM, 05:00 AM and 08:00 AM. [Learn more](#)

Retries

3



Retries Interval (in minutes)

5



Scope

enforcement-job



Expected Run Time

10

When the job keeps running longer than this value we will emit a point datadog, so you can use it to create alerts. In seconds

Tags



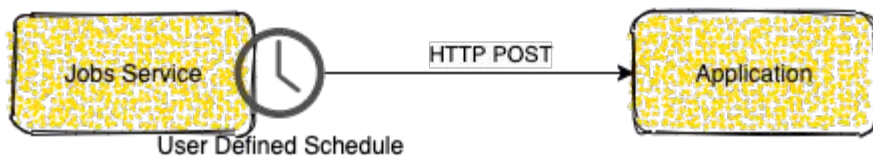
core-metric: not-apply

core-metric-site: not-apply

Create

Cancel

# // Jobs - Arquitectura



Sample APP: <https://web.furycloud.io/items-api-legacy/jobs>

# // Object Storage



## < QUÉ >

Permite guardar, obtener y gestionar cualquier tipo de objeto en un “bucket”



## < ACCESO >

Podemos tener objetos privados accesibles por URLs firmadas, u objetos públicos accesibles desde cualquier parte



## < USO >

Simple acceso a través de un SDK. Métricas, controles y alertas predefinidas

# // Object Storage - Create

- *Name*: nombre del servicio.
- *Description*: Detalles de lo que contendrá.
- *Test*: Sí o No.
- *Confidentiality*: Public, Internal, Confidential, Restricted.
- *Integrity*: Low, Moderate, High.
- *Availability*: Low, Moderate, High.
- *Write Throughput*: Estimado de tráfico de escritura (si se supera, el servicio devuelve 429).
- *Read Throughput*: Estimado de tráfico de lectura (si se supera, el servicio devuelve 429).

IT BOARDING

BOOTCAMP

## Create Object Storage

Create

Cancel

## Service

## Name

## Description

0 / 100

## Criticality

## Level



## Tags



## Compliance

## Availability



## Integrity



## Confidentiality



## Traffic

## Write throughput



## Read throughput



These fields will be used for throttling

Create

Cancel

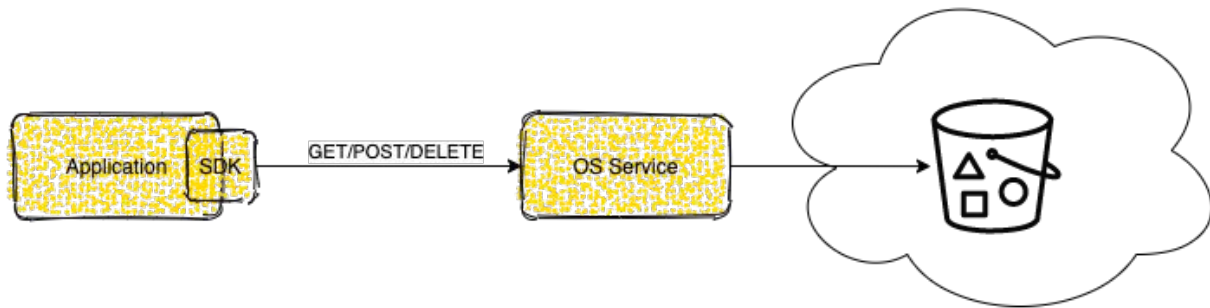
## // Object Storage - Uso (Java)

```
ObjectStorage client = new ObjectStorageService("my_container");  
byte[] data = client.get("my_resource");  
  
OutputStream os = ...;  
  
client.get("my_resource", os);  
  
client.delete("my_resource");  
  
ListOutput data = client.list();  
  
List<PathInfo> files = data.getPaths();
```

**Sample APP:** <https://web.furycloud.io/emkt-communication-manager/summary>



# // Object Storage - Arquitectura



# // Audits



## < QUÉ >

Permite registrar acciones específicas llevadas a cabo por la aplicación



## < USO >

SDK para registro, un Front End para consulta



## < MONITOREO >

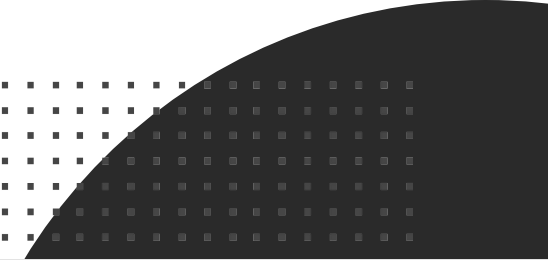
Provee métricas y alertas predefinidas

# // Audits - Create

- *Name*: Nombre del recurso.
- *Test*: Sí o No.
- *Description*: Detalle del uso que tendrá el recurso.
- *Write Throughput*: Estimación de registros a escribir por minuto.
- *Lifecycle*: Mínimo de meses de vida de la información.
- *Confidentiality*: Sí o No.
- *Visibility*: Quién puede ver la información.
- *Resource Types*: Qué entidades del ecosistema estamos registrando.

IT BOARDING

BOOTCAMP



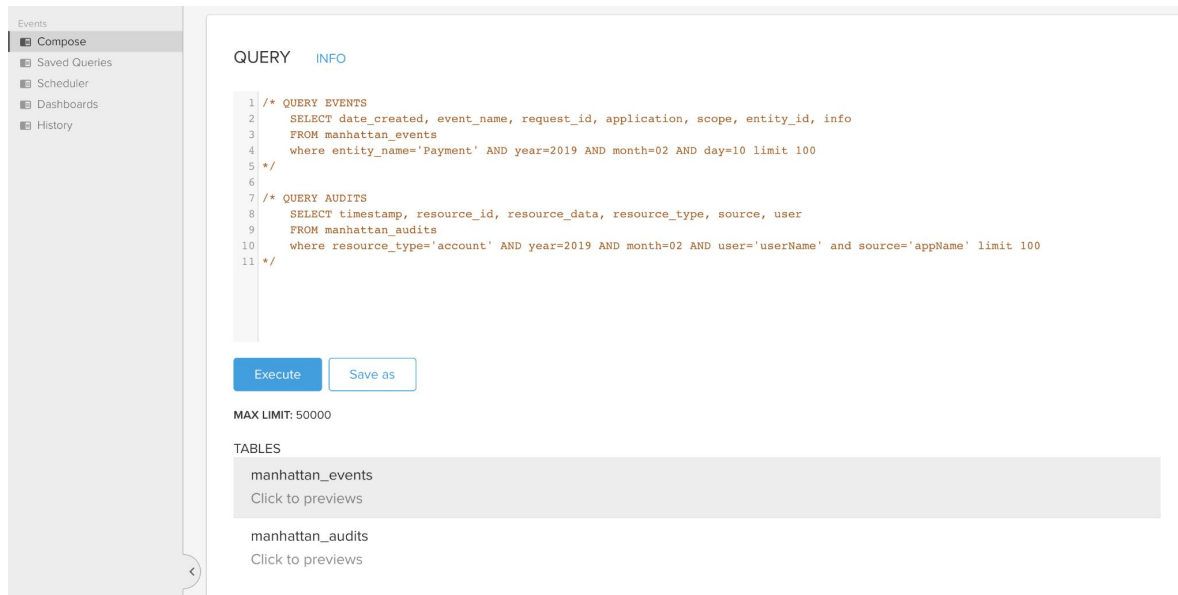
Name	<input type="text" value="Audit source's name"/>	
Test ⓘ	<input type="text" value="Select if this is a test source"/>	
Description	<input type="text" value="Description"/>	
Write Throughput ⓘ	<input type="text" value="100"/>	rpm
Lifecycle ⓘ	<input type="text" value="6"/>	months
Confidentiality	<input type="text" value="Select a confidentiality level"/>	
Visibility ⓘ	<input type="text" value="Select a visibility level"/>	
Monthly cost ⓘ	USD 5.19	
Min query available ⓘ	<input type="text" value="1 d"/>	
Resource Types ⓘ	<input type="text"/>	<input type="button" value="Add"/>

## // Audits - Uso (Java)

```
AuditApiConfiguration config =  
AuditApiConfiguration.builder().build();  
  
AuditClient client = new AuditClient(config, "audit_name");  
  
AuditRecord record = new AuditRecord(...);  
  
client.saveAuditSync(record);
```

Sample app: [https://github.com/mercadolibre/fury\\_sample-audits-java](https://github.com/mercadolibre/fury_sample-audits-java)

# // Audits - Uso (Front End)



The screenshot displays the Manhattan FE interface. On the left is a sidebar with a menu containing 'Compose', 'Saved Queries', 'Scheduler', 'Dashboards', and 'History'. The main area is titled 'QUERY INFO' and contains a SQL query editor with the following text:

```
1 /* QUERY EVENTS
2   SELECT date_created, event_name, request_id, application, scope, entity_id, info
3   FROM manhattan_events
4   where entity_name='Payment' AND year=2019 AND month=02 AND day=10 limit 100
5 */
6
7 /* QUERY AUDITS
8   SELECT timestamp, resource_id, resource_data, resource_type, source, user
9   FROM manhattan_audits
10  where resource_type='account' AND year=2019 AND month=02 AND user='userName' and source='appName' limit 100
11 */
```

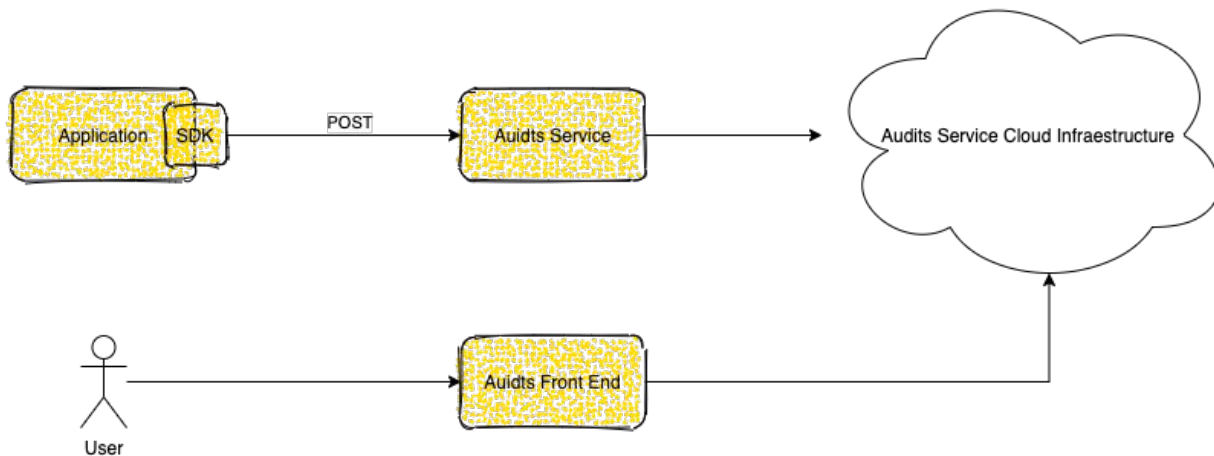
Below the query editor are two buttons: 'Execute' and 'Save as'. Underneath these buttons, it states 'MAX LIMIT: 50000'. A section titled 'TABLES' lists two tables: 'manhattan\_events' and 'manhattan\_audits', each with a 'Click to previews' link.

Manhattan FE: <https://manhattan.adminml.com/query>

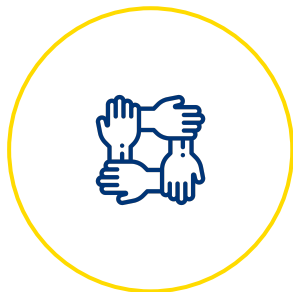
IT BOARDING

BOOTCAMP

# // Audits - Arquitectura



# // Streams



## < QUÉ >

Canal de transmisión continua de información con determinadas garantías (orden por partition key, at least once delivery)



## < USO >

Dado un stream al que tenemos acceso, podemos crear un **sink** HTTP para consumir sus mensajes en nuestra aplicación.



## < MONITOREO >

Provee métricas y alertas predefinidas

# // Streams Sink - Create

- *Name*: Nombre del sink.
- *Stream*: Stream sobre.
- *Type*: Por ahora solo HTTP (otros servicios en el futuro).
- *Scope*: Scope donde se ejecutará el sink.
- *URI*: Endpoint de la aplicación que recibirá los mensajes.
- *Criticality*: Low, Medium, High, Test.
- *Rate limit*: Máximas requests por minuto a ser ejecutadas sobre el sink

## Create Sink

Name

Service name

Stream

Stream

Type

HTTP

Scope

Scope

URI

Criticality

Level

Low

Traffic

Rate Limit

5000

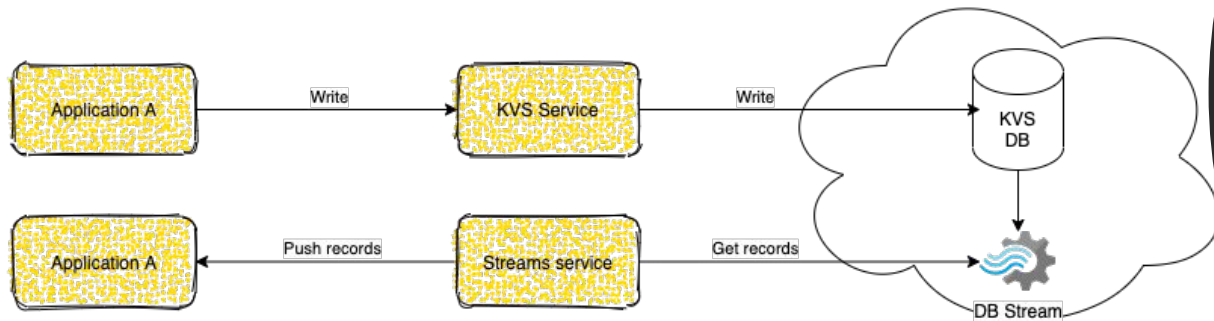
RPM

Create

Cancel



# // Streams - Arquitectura



Sample APP: <https://web.furycloud.io/sem-event-collector/streams>

# // ¡Aún hay más!



## Bazooka

Manages and runs stress/performance testing



## Cache

Distributed in-memory key-value cache.



## Configurations

Makes easier the task of managing app settings, from the definition to the deployment.



## Lock

Prevents concurrent access / operations over distributed accesible resources



## Sequence

Provides unique ranges of numeric IDs



“

*Comemos el pescado que vendemos*

”

IT BOARDING

**BOOTCAMP**





# Gracias.

IT BOARDING

**BOOTCAMP**

