

# Pulsar Catalog

## Background and Motivation

When discussing provides access to event streams served by Apache Pulsar in [FLIP-72](#), a more detailed design description on enabling Pulsar as a catalog is proposed. Make it a separate doc for ease of reading and checking.

### Map Pulsar topics to tables

[Apache Pulsar](#) has a built-in schema registry that enables clients to upload data schemas on a per-topic basis. Therefore, a topic in Pulsar could be regarded as a table in databases.

### Map `tenant/namespace` to databases

Pulsar was created from the ground up as a multi-tenant system. In Pulsar, the authentication provider is responsible for properly identifying clients and associating the clients with role tokens, and authorization is the process that determines what clients are able to do.

Generally, there are 3 kinds of roles for a Pulsar instance:

- **Superuser**, who has the most privileges and can check all tenants with all their namespaces and topics.
- **Tenant admin**, who has access to all the namespaces and topics inside the tenant.
- **Produce or Consume**, who can only see the topics its granted with.

Besides, users could be granted with multiple topics in several `tenant/namespace`.

Therefore, for all the users, a pulsar instance could be regarded as a multi-database system. With each visible pulsar topic as a table, and the `tenant/namespace` of each topic as the container database.

## Proposal

We propose a throughout Pulsar connector implementation in Flink.

## Scope

The Pulsar catalog supports the following features by extending `**AbstractCatalog**` API:

- Open and close the catalog
- Database operations: list, get, exists, create, drop (create and drop should first check if the user has sufficient permissions)

- Table operations: list, get, exist, create, drop (create and drop should first check if the user has sufficient permissions)

## Unsupported Catalog features

Operation in <b>AbstractCatalog</b> API	Why not this operation
alterDatabase	Pulsar doesn't support store extra metadata of a namespace
alterTable	Pulsar doesn't support store extra metadata of a topic
Partition related operations	Pulsar doesn't support layered tree-structured data storage, the partition in a topic has fixed naming style and couldn't be used to store column values ( <i>-partition-n</i> )
Function related operations	Pulsar doesn't provide key-based binary data load & store
Statistics related operations	AFAIK, Pulsar could only provide `rowCount` for a table, other kinds of statistics are not available at the moment

## Some Implementation Details

### HTTP Client

All catalog operations for the pulsar catalog are actually manipulating metadata of Pulsar. Therefore, we use **PulsarAdmin** when we implementing the **AbstractCatalog** API. PulsarAdmin is an HTTP client and talks to a Pulsar instance by specifying the `serviceHttpUrl`. Besides, the authentication and authorization are also done while we are setting up the connection. Please refer to [PulsarAdminBuilder](#) for all available options.

### Partitioned and Non-partitioned Topic

Both partitioned and non-partitioned topic is mapped to one table in the catalog. We could probably provide the `"__partition_id"` as one of the metadata columns in a table schema, just like the other metadata fields that could be provided by Pulsar: `"__topic"`, `"__key"`, `"__messageId"`, `"__publishTime"`, `"__eventTime"`.

## Operations

Catalog Operation	Actions in Pulsar
open/close	Create and close the PulsarAdmin client
listDB/getDB/createDB/dropDB	List/get/create/drop tenant/namespace one has permission
listTable/tableExists/dropTable	list/check/drop a topic if permitted
getTable	Get a topic and convert its schema to table schema in Flink (if permitted)
createTable	Create a topic, convert flink schema into its counterparts in Pulsar and upload it to Pulsar

## YAML Configuration

Option	Value	Default	Description
serviceUrl	A service URL of your Pulsar cluster		The Pulsar `serviceUrl` configuration.
adminUrl	A service HTTP URL of your Pulsar cluster		The Pulsar `serviceHttpUrl` configuration.
default-database	The default database name	public/default	A topic in Pulsar is treated as a table in Flink when using Pulsar catalog, therefore, `database` is another name for `tenant/namespace`. The database is the basic path for table lookup or creation.
startingOffsets	The following are	"latest"	`startingOffsets`

	<p>valid values: "earliest"(streaming and batch queries)</p> <p>"latest" (streaming query)</p>		option controls where a table reads data from.
tableDefaultPartitions	The default number of partitions when a table is created in Table API.	5	A table in Pulsar catalog is a topic in Pulsar, when creating table in Pulsar catalog, `table.partitions` controls the number of partitions when creating a topic.