⌂ Reference Terminology

Pulsar Terminology

Here is a glossary of terms related to Apache Pulsar:

Concepts

Pulsar

Pulsar is a distributed messaging system originally created by Yahoo but now under the stewardship of the Apache Software Foundation.

Message

Messages are the basic unit of Pulsar. They're what producers publish to topics and what consumers then consume from topics.

Topic

A named channel used to pass messages published by producers to consumers who process those messages.

Partitioned Topic

A topic that is served by multiple Pulsar brokers, which enables higher throughput.

Namespace

A grouping mechanism for related topics.

Namespace Bundle

A virtual group of topics that belong to the same namespace. A namespace bundle is defined as a range between two 32-bit hashes, such as 0x00000000 and 0xffffffff.

Tenant

An administrative unit for allocating capacity and enforcing an authentication/authorization scheme.

Subscription

A lease on a topic established by a group of consumers. Pulsar has four subscription modes (exclusive, shared, failover and key_shared).

Pub-Sub

A messaging pattern in which producer processes publish messages on topics that are then consumed (processed) by consumer processes.

Producer

A process that publishes messages to a Pulsar topic.

Consumer

A process that establishes a subscription to a Pulsar topic and processes messages published to that topic by producers.

Reader

Pulsar readers are message processors much like Pulsar consumers but with two crucial differences:

- you can specify *where* on a topic readers begin processing messages (consumers always begin with the latest available unacked message);
- readers don't retain data or acknowledge messages.

Cursor

The subscription position for a consumer.

Acknowledgment (ack)

A message sent to a Pulsar broker by a consumer that a message has been successfully processed. An acknowledgment (ack) is Pulsar's way of knowing that the message can be deleted from the system; if no acknowledgment, then the message will be retained until it's processed.

Negative Acknowledgment (nack)

When an application fails to process a particular message, it can send a "negative ack" to Pulsar to signal that the message should be replayed at a later timer. (By default, failed messages are replayed after a 1-minute delay). Be aware that negative acknowledgment on ordered subscription types, such as Exclusive, Failover and Key_Shared, can cause failed messages to arrive to consumers out of the original order.

Unacknowledged

A message that has been delivered to a consumer for processing but not yet confirmed as processed by the consumer.

Retention Policy

Size and time limits that you can set on a namespace to configure retention of messages that have already been acknowledged.

Multi-Tenancy

The ability to isolate namespaces, specify quotas, and configure authentication and authorization on a per-tenant basis.

Failure Domain

A logical domain under a Pulsar cluster. Each logical domain contains a pre-configured list of brokers.

Anti-affinity Namespaces

A group of namespaces that have anti-affinity to each other.

Architecture

Standalone

A lightweight Pulsar broker in which all components run in a single Java Virtual Machine (JVM) process. Standalone clusters can be run on a single machine and are useful for development purposes.

Cluster

A set of Pulsar brokers and BookKeeper servers (aka bookies). Clusters can reside in different geographical regions and replicate messages to one another in a process called geo-replication.

Instance

A group of Pulsar clusters that act together as a single unit.

Geo-Replication

Replication of messages across Pulsar clusters, potentially in different datacenters or geographical regions.

Configuration Store

Pulsar's configuration store (previously known as configuration store) is a ZooKeeper quorum that is used for configuration-specific tasks. A multi-cluster Pulsar installation requires just one configuration store across all clusters.

Topic Lookup

A service provided by Pulsar brokers that enables connecting clients to automatically determine which Pulsar cluster is responsible for a topic (and thus where message traffic for the topic needs to be routed).

Service Discovery

A mechanism provided by Pulsar that enables connecting clients to use just a single URL to interact with all the brokers in a cluster.

Broker

A stateless component of Pulsar clusters that runs two other components: an HTTP server exposing a REST interface for administration and topic lookup and a dispatcher that handles all message transfers. Pulsar clusters typically consist of multiple brokers.

Dispatcher

An asynchronous TCP server used for all data transfers in and out of a Pulsar broker](#broker). The Pulsar dispatcher uses a custom binary protocol for all communications.

Storage

BookKeeper

Apache BookKeeper is a scalable, low-latency persistent log storage service that Pulsar uses to store data.

Bookie

Bookie is the name of an individual BookKeeper server. It is effectively the storage server of Pulsar.

Ledger

An append-only data structure in BookKeeper that is used to persistently store messages in Pulsar topics.

Functions

Pulsar Functions are lightweight functions that can consume messages from Pulsar topics, apply custom processing logic, and, if desired, publish results to topics.