JetStream Per-Message TTL

Metadata	Value
Date	2024-07-11
Author	@ripienaar
Status	Implemented
Tags	jetstream, client, server, 2.11

Context and motivation

Streams support a one-size-fits-all approach to message TTL based on the MaxAge setting. This causes any message in the Stream to expire at that age.

There are numerous uses for a per-message version of this limit, some listed below:

- KV tombstones are a problem in that they forever clog up the buckets with noise, these could have a TTL to make them expire once not useful anymore
- Server-applied limits can result in tombstones with a short per message TTL so that consumers can be notified of limits being processed. Useful in KV watch scenarios being notified about TTL removals
- A stream may have a general MaxAge but some messages may have infinite retention, think a schema or type hints in a KV bucket that is forever while general keys have TTLs

Related issues #3268

Per-Message TTL

General Behavior

We will allow a message to supply a TTL using a header called `Nats-TTL` followed by the duration as seconds or as a Go duration string like `1h`.

The duration will be used by the server to calculate the deadline for removing the message based on its Stream timestamp and the stated duration.

Setting the header `Nats-TTL` to `never` will result in a message that will never be expired.

A TTL of zero will be ignored, any other unparsable value will result in a error reported in the Pub Ack and the message being discarded.

When a message with the `Nats-TTL` header is published to a stream with the feature disabled the message will be rejected with an error.

Limit Markers

Several scenarios for server-created markers can be imagined, the most often requested one though is when MaxAge removes last value (ie. the current value) for a Key.

In this case when the server removes a message and the message is the last in the subject it would place a message with a TTL matching the Stream configuration value. The following headers would be placed:

```
Nats-Marker-Reason: MaxAge
Nats-TTL: 1
```

This marker will also be placed for a message removed by the `Nats-TTL` timer.

This behaviour is off by default unless opted in on the `SubjectDeleteMarkerTTL` Stream Configuration.

Delete API Call Marker

> [!IMPORTANT] This feature will come either later in 2.11.x series or in 2.12.

When someone calls the delete message API of a stream the server will place a the following headers:

```
Nats-Marker-Reason: Remove
Nats-TTL: 1
```

Purge API Call Marker

> [!IMPORTANT] This feature will come either later in 2.11.x series or in 2.12.

When someone calls the purge subject API of a stream the server will place a the following headers:

```
Nats-Marker-Reason: Purge
Nats-TTL: 1
```

Sources and Mirrors

Sources and Mirrors will always accept and store messages with `Nats-TTL` header present, even if the `AllowMsgTTL` setting is disabled in the Stream settings.

If the `AllowMsgTTL` setting is enabled then processing continues as outlined in the General Behavior section with messages removed after the TTL. With the setting disabled the messages are just stored.

Sources may set the `SubjectDeleteMarkerTTL` option and processing of messages with the `Nats-TTL` will place tombstones, but, Mirrors may not enable `SubjectDeleteMarkerTTL` since it would insert new messages into the Stream it might make it impossible to match sequences from the Mirrored Stream.

Stream Configuration

Weather or not a stream support this behavior should be a configuration opt-in. We want clients to definitely know when this is supported which the opt-in approach with a boolean on the configuration would make clear.

We have to assume someone will want to create a replication topology where at some point in the topology these tombstone type messages are retained for an audit trail. So a Stream with this feature enabled can replicate to one with it disabled and all the messages that would have been TTLed will be retained.

```go
type StreamConfig struct {
    // AllowMsgTTL allows header initiated per-message TTLs
    AllowMsgTTL bool          `json:"allow_msg_ttl"`

    // Enables and sets a duration for adding server markers for delete, purge and
max age limits
    SubjectDeleteMarkerTTL time.Duration
`json:"subject_delete_marker_ttl,omitempty"`
}
```

Restrictions:

- The `AllowMsgTTL` field can be enabled on existing streams but not disabled.
- The `AllowMsgTTL` and `SubjectDeleteMarkerTTL` has a minimum value of 1 second.
- The `SubjectDeleteMarkerTTL` setting may not be set on a Mirror Stream.
- When `AllowMsgTTL` or `SubjectDeleteMarkerTTL` are set the Stream should require API level `1`.
- `AllowRollup` must be `true`, stream update and create should set this unless pedantic mode is enabled.
- `DenyPurge` must be `false`, stream update and create should set this unless pedantic mode is enabled.
- Unless `MaxMsgsPer` equals 1 the server should treat `SubjectDeleteMarkerTTL` as the minimum for `Nats-TTL` but not reject messages that do not satisfy that. This might be changed in 2.12 depending on some internal implementation fixes in the server.