KIP-808: Add support for different unix precisions in TimestampConverter SMT

Status

Current state: Accepted

Discussion thread: *here*

Vote thread: *here*

JIRA: *KAFKA-13511*

Please keep the discussion on the mailing list rather than commenting on the wiki (wiki discussions get unwieldy fast).

Motivation

Currently, the Kafka Connect SMT TimestampConverter can convert Timestamp from multiples sources types (String, Unix Long or Date) into different target types (String, Unix Long or Date).

The problem is that Unix Long as a source or as a target type must be with milliseconds precision.

In many cases, Unix time is represented with different precisions within external systems : seconds, microseconds, nanoseconds.

When such case arise, Kafka Connect can't do anything expect pass along the Unix Long and leave the conversion to another layer.

This issue was raised several times :

- KAFKA-12364
- KAFKA-10561

TimestampConverter should have a config to define which precision to use when converting from and to Long Unix timestamp.

Public Interfaces

name	description	type	default	valid values	importance
`unix.precision`	The desired unix precision for the timestamp. Used to generate the output when type=unix or used to parse the input if the input is a Long. Note: This SMT will cause precision loss during conversions from and to values with sub-milliseconds components.	String	`milliseconds`	`seconds`, `milliseconds`, `microseconds`, `nanoseconds`	low

Proposed Changes

New config property for TimestampConverter

Implementation details to be discussed :

- TimeUnit.MILLISECONDS.toMicros(unixMillis) and so on for the other conversions seems the easiest way.

Unix Long to Timestamp example:

```
"transforms.TimestampConverter.type": "org.apache.kafka.connect.transforms.TimestampConverter$Value",
"transforms.TimestampConverter.field": "event_date_long",
"transforms.TimestampConverter.unix.precision": "microseconds",
"transforms.TimestampConverter.target.type": "Timestamp"
```

String to Unix Long nanoseconds example:

```
"transforms.TimestampConverter.type": "org.apache.kafka.connect.transforms.TimestampConverter$Value",
"transforms.TimestampConverter.field": "event_date_str",
"transforms.TimestampConverter.format": "yyyy-MM-dd'T'HH:mm:ss.SSS",
"transforms.TimestampConverter.target.type": "unix",
"transforms.TimestampConverter.unix.precision": "nanoseconds"
```

java.util.Date and SimpleDateFormat limitations

Since these classes can only handle precisions down to the millisecond, it should be noted that:

- converting source Unix Long microseconds or nanos into any target type leads to a precision loss (truncation after millis)
- converting any source type into target Unix Long microseconds or nanos, the part after milliseconds will always be 0
- A KIP that address Date vs Instant may be more appropriate but it impacts so much of the code that I believe this is a good first step.

int32 and seconds

Systems that produces int32 into Kafka should willingly chain Cast SMT and then TimestampConverter SMT if they want to use this feature.

```
"transforms": "Cast,TimestampConverter",
"transforms.Cast.type": "org.apache.kafka.connect.transforms.Cast$Value",
"transforms.Cast.spec": "event_date_int:int64",
"transforms.TimestampConverter.type": "org.apache.kafka.connect.transforms.TimestampConverter$Value",
"transforms.TimestampConverter.field": "event_date_int",
"transforms.TimestampConverter.unix.precision": "seconds",
"transforms.TimestampConverter.target.type": "Timestamp"
```

Compatibility, Deprecation, and Migration Plan

The change will not break the compatibility.

Rejected Alternatives

If there are alternative ways of accomplishing the same thing, what were they? The purpose of this section is to motivate why the design is the way it is and not some other way.

Name the field `epoch.precision`

Since epoch is not a measure but rather a point in time, it can't be associated to a precision.

It makes more sense to name the field `unix.precision` for that reason.

Use `seconds`, `millis`, `micros`, `nanos` or `s,ms,us,ns` as values

seconds is a unit, but millis micros, nanos are really just prefixes. Mixing up doesn't work well.

s, ms, µs, ns are a valid SI symbols but the µs (or its accepted equivalent **us**) can be confusing.

For clarity, it was decided to use the plaintext naming convention.