Persist events in online DB during network outages

Abstract

The Open Distributed Infrastructure Management (ODIM) framework aggregates events from the managed servers, storage and network on behalf of subscribers. The subscribers only receive events they have subscribed to with ODIMRA. The BMCs themselves only have to forward events to ODIM rather than multiple subscribers. ODIMRA will try to forward the events up to a configurable number of times with a wait period in between. If ODIMRA is unable to deliver the event within the number of times configured the event is dropped. This could happen if ODIMRA and subscribers are spread across remote sites and there is a sustained network outage.

Solution

The current code reads from a custom message queue implemented in go language. This message queue itself is populated by messages retrieved from the event message bus which is Kafka at the moment. The proposal is to use redis database to store the events that could not be received by the subscriber. We will also need to store the forwarding info including destination info. The event service will then periodically check the DB for pending events and attempt to send the same. The interval shall be added as configuration to ODIMRA. Alternatively a successful transmission can also trigger the forwarding of events pending in DB.

Alternates considered

- We could use a file store to store events. However the overheads of file store management including sync over multiple pods, handling DB operations in file store etc. makes this more complicated.
- Redis streams works allows a consumer/consumer group reads a message, process the same and acknowledges the same. It is also durable as compared to pub/sub. If consumer fails to acknowledge a message (event for us) then the message will be consumed by another event dispatcher thread. However this would be available to another thread immediately. Whereas the idea is re-transmission is done after a longer wait period.

References

1. Redis pub/sub
2. Redis Streams