KIP-916: MM2 distributed mode flow log context

Status

Current state: *Under Discussion*

Discussion thread: *here*

JIRA: *here*

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

Motivation

MirrorMaker2 distributed mode is capable of running multiple Connect workers in a single process, possibly connecting to a set of different Kafka clusters. This simplifies the operation of async replication compared to operating separate Kafka Connect clusters.

At the same time, diagnosing issues in MM2 distributed mode can be quite challenging. Each flow has a dedicated Connect group, and for each flow, the Connectors have the same name. This causes the following monitoring and diagnostic issues with MM2:

1. The Connector log context provided by Connect contains the same Connector name/Task ID in each flow. In the MM2 logs, lines coming from inside the context cannot be distinguished easily/at all.
2. The internal Connect thread names only contain the Connector name/Task ID, and those cannot be distinguished easily/at all.

MM2 distributed mode should also provide the replication flow in the the log context and the thread names to address this problem.

Public Interfaces

MM2 dedicated mode logging change

- New MDC context key: flow.context
- flow.context will contain the replication flow using the following pattern: [<SOURCE>-><TARGET>] e.g. "[primarybackup]".
- flow.context will be exported in all locations where connector context information is currently exported.
- The connect-log4j.properties file will be changed by adding a comment about the flow.context key, and its usage in MM2:

```
# The `%X{connector.context}` parameter in the layout includes connector-specific and task-specific information
# in the log messages, where appropriate. This makes it easier to identify those log messages that apply to a
# specific connector.
#
# The `%X{flow.context}` parameter can be used in the layout in MM2 dedicated mode. flow.context includes flow-
specific information
# in the log messages, where appropriate. This makes it easier to identify those log messages that apply to a
# specific replication flow.
#
connect.log.pattern=[%d] %p %X{connector.context}%m (%c:%L)%n
```

Thread name changes

The Connect internal thread names will contain a |A->B suffix in MM2 mode. The following Connect internal thread names will change:

Only in MM2 dedicated mode:

- SourceTaskOffsetCommitter,commitExecutorService (current: SourceTaskOffsetCommitter-0, proposed: SourceTaskOffsetCommitter-0|primarybackup)
- WorkerTask thread name (current: task-thread-MyConnector-0, proposed: task-thread-MyConnector-0|primarybackup)
- KafkaStatusBackingStore.sendRetryExecutor (current: status-store-retry-mm2-status.primary.internal, proposed: status-store-retry-mm2-status.primary.internal|primarybackup)

- KafkaBasedLog.thread (current: KafkaBasedLog Work Thread - mm2-configs.primary.internal, proposed: KafkaBasedLog Work Thread - mm2-configs.primary.internal|primary->backup)

Both in Connect and in MM2 dedicated mode:

- AbstractHerder.connectorExecutor (current: unnamed, proposed for Connect: connector-executor, proposed for MM2: connector-executor|primarybackup)
- Worker.executor (current: unnamed, proposed for Connect: worker-executor, proposed for MM2: worker-executor|primary->backup)

Sidenote: the following Connect internal thread names will NOT change:

- DistributedHerder internal threads - their names already contain the client.id, which contains the flow in MM2 dedicated mode.

Proposed Changes

	Before	After (change is in bold)
Log pattern	[%d] %p [%t] %X{connector.context}%m (%c:%L)%n	[%d] %p [%t] **%X{flow.context}**%X{connector.context}%m (%c:%L)%n
MirrorSourceTask startup	[2023-06-12 11:07:42,246] INFO [task-thread-MirrorSourceConnector-0] [MirrorSourceConnector|task-0] Starting with 2 previously uncommitted partitions. (org.apache.kafka.connect.mirror.MirrorSourceTask:108)	[2023-06-12 11:07:42,246] INFO [task-thread-MirrorSourceConnector-0**|A->B**] **[A->B]** [MirrorSourceConnector|task-0] Starting with 2 previously uncommitted partitions. (org.apache.kafka.connect.mirror.MirrorSourceTask:108)
Connector logs	[2023-06-12 11:10:51,829] INFO [connector-thread-MirrorHeartbeatConnector] [MirrorHeartbeatConnector|worker] MirrorHeartbeatConfig values:	[2023-06-12 11:10:51,829] INFO [connector-thread-MirrorHeartbeatConnector**|A->B**] **[A->B]** [MirrorHeartbeatConnector|worker] MirrorHeartbeatConfig values:
Framework internal logs	[2023-06-12 11:10:51,975] INFO [connector-thread-MirrorSourceConnector] [MirrorSourceConnector|worker] Completed shutdown for WorkerConnector{id=MirrorSourceConnector} (org.apache.kafka.connect.runtime.WorkerConnector:287)	[2023-06-12 11:10:51,975] INFO [connector-thread-MirrorSourceConnector**|A->B**] **[A->B]** [MirrorSourceConnector|worker] Completed shutdown for WorkerConnector{id=MirrorSourceConnector} (org.apache.kafka.connect.runtime.WorkerConnector:287)

As seen in the example log lines, the **Before** lines do not provide the full context, the lines cannot be differentiated among the different flows. Lines after the change will contain the flow information in both the flow.context and the thread names.

There is overlap between the thread names and the flow.context. This shouldn't be a problem - many use-cases do not use the thread name in logging, for those cases, the flow.context is still useful. The thread name change also improves diagnostics in other areas, such as when generating thread dumps, or debugging a process.

Compatibility, Deprecation, and Migration Plan

- The impact on Connect is minimal, only internal thread names are changed.
- MM2 distributed mode impact is minimal
    - Connect internal thread names will be changed.
    - The flow context is only added if the user opts-in by referring to the flow.context MDC key in their logging configuration.

Test Plan

Unit tests focusing on the new MDC key flow.context.

Thread names are not tested, as they are not part of any contract (user-facing or programmatic).

Rejected Alternatives

Updating the existing connector.context MDC value with the flow information

Instead of exporting a separate flow.context MDC value, the existing connector.context can be updated (prefixed/suffixed with the flow). This would require extra configurations to allow users to opt-in into the new feature. Additionally, it is less flexible, as it would be tied to the connectors/tasks, while the flow information can be used in broader context (e.g. in Connect worker internal logging).