

## Apache Kafka 3.0.0 稳定版发布，有哪些值得关心的变化？

Apache Kafka 3.0 于2021年9月21日正式发布。本文将介绍这个版本的新功能。以下文章翻译自 [《What's New in Apache Kafka 3.0.0》](#)。



如果想及时了解Spark、Hadoop或者HBase相关的文章，欢迎关注微信公众号：过往记忆大数据

我很高兴地代表 Apache Kafka® 社区宣布 Apache Kafka 3.0 的发布。Apache Kafka 3.0 是一个大版本，其引入了各种新功能、API 发生重大变化以及对 KRaft 的改进——Apache Kafka 的内置共识机制将取代 Apache ZooKeeper™。

虽然 KRaft 还不推荐在生产中使用（可以参见[这里](#)），但我们对 KRaft 元数据和 API 进行了许多改进。支持 Exactly-once 和分区重分配值得强调。我们推荐您查看 KRaft 的新功能并在开发环境中试用它。

从 Apache Kafka 3.0 开始，Producer 默认启用最强的交付保证（acks=all，enable.idempotence=true）。这意味着默认情况下消息将有序并且持久性。

此外，不要错过 Kafka Connect 任务重启增强、Kafka Streams 基于时间戳同步的改进以及 MirrorMaker2 更灵活的配置选项。

要完整的特性和功能增强，可以到 [release notes](#) 里面了解。您还可以观看[发布视频](#)，了解 Apache Kafka 3.0.0 中新增功能的简要介绍。

## 普通变化

### KIP-750 (Part I): Deprecate support for Java 8 in Kafka

Apache Kafka 项目的所有组件在 3.0 中都将 Java 8 标记为 deprecated。这给用户足够的时间在下一个主要版本（4.0）之前进行调整，那时候 4.0 将不再支持 Java 8。KIP-750 参见[这里](#)

### KIP-751 (Part I): Deprecate support for Scala 2.12 in Kafka

Apache Kafka 项目的所有组件在 3.0 中都将 Scala 2.12 标记为 deprecated。这给用户足够的时间在下一个主要版本（4.0）之前进行调整，那时候 4.0 将不再支持 Scala 2.12。KIP-751 参见[这里](#)

## Kafka Broker, Producer, Consumer 和 AdminClient 方面的改进

### KIP-630: Kafka Raft Snapshot

我们在 3.0 中引入的一个主要功能是 KRaft Controllers 和 KRaft Brokers，能够为元数据主题 `_cluster_metadata` 的分区生成、复制和加载快照。Kafka 集群使用这个主题来存储和复制有关集群的元数据信息，例如 Broker 配置、topic 分区分配、leadership 等。随着此状态的增长，Kafka Raft Snapshot 提供了一种有效的方式来存储、加载和复制（replicate）这些信息。

### KIP-746: Revise KRaft Metadata Records

#### Kafka Raft Controller

第一版的经验和持续开发表明，我们需要修改一些元数据记录类型（metadata record types），这些元数据记录类型在 Kafka 配置不使用 ZooKeeper 的时候需要使用。

### KIP-730: Producer ID generation in KRaft mode

在 KIP-730 中，Kafka Controller 现在完全接管生成 Kafka Producer ID 的功能。Controller 在 ZK 和 KRaft 模式下都是这么做的。这让我们离桥式发布（bridge release）更近了一步，这将允许用户从使用 ZK 的 Kafka 部署过渡到使用 KRaft 的新部署。

### KIP-679: Producer will enable the strongest delivery guarantee by default

#### 从 3.0 开始，Kafka Producer

默认打开幂等性（`enable.idempotence=true`）以及所有副本的交付都需要确认（`acks=all`）。这使得默认情况下记录的交付更加可靠。

### KIP-735: Increase default consumer session timeout

Kafka Consumer 的配置属性 `session.timeout.ms` 的默认值从 10 秒增加到 45 秒。这将允许消费者在默认情况下更好地适应暂时的网络故障，并避免当消费者只是暂时离开组时的连续重新平衡（consecutive rebalances）。

## KIP-709: Extend OffsetFetch requests to accept multiple group ids

支持请求 Kafka 消费者组的当前偏移量的功能已经存在一段时间了。但是获取多个消费者组的偏移量需要对每个组进行单独的请求。在 KIP-709 中，`fetch` 和 `AdminClient` API 被扩展为支持在单个请求/响应中同时读取多个消费者组的偏移量。

## KIP-699: Update FindCoordinator to resolve multiple Coordinators at a time

支持能够同时高效地应用于多个消费者组的操作，在很大程度上取决于客户有效地发现这些组的协调者的能力。这可以通过 KIP-699

实现，它增加了对通过一个请求发现多个组的协调器的支持。Kafka 客户端已经更新在与新的支持此请求的 Kafka broker 交互时使用此优化。

## KIP-724: Drop support for message formats v0 and v1

自 2017 年 6 月随 Kafka 0.11.0 推出四年以来，消息格式 v2 一直是默认消息格式。因此，由于已经有足够多的场景使用这个格式，3.0 的主要发行版为我们提供了一个很好的机会来弃用旧的消息格式——即 v0 和 v1，这些格式现在很少使用。在 3.0 中，如果用户将 Broker 配置为使用消息格式 v0 或 v1，则会收到警告。这个选项将在 Kafka 4.0 中被移除（有关 v0 和 v1 消息格式的弃用的细节和含义，请参阅 [KIP-724](#)）。

## KIP-707: The future of KafkaFuture

当 `KafkaFuture` 类型被引入以方便 Kafka `AdminClient` 的实现时，Java 8 之前的版本仍在广泛使用，并且 Kafka 正式支持 Java 7。几年后，Kafka 运行在支持 `CompletionStage` 和 `CompletableFuture` 类类型的 Java 版本上。通过 KIP-707，`KafkaFuture` 增加了一个方法来返回 `CompletionStage` 对象，以向后兼容的方式增强了 `KafkaFuture` 的可用性。

## KIP-466: Add support for List serialization and deserialization

KIP-466 为 generic lists 的序列化和反序列化添加了新的类和方法——这一特性对 Kafka 客户端和 Kafka Streams 都非常有用。

## KIP-734: Improve AdminClient.listOffsets to return timestamp and offset for the record with the largest timestamp

用户列出 Kafka 主题/分区偏移量的能力已得到扩展。使用 KIP-734，用户现在可以要求 `AdminClient` 返回 topic/分区中具有最高时间戳的记录的偏移量和时间戳。（不要将此与 `AdminClient` 已经返回的最新偏移量混淆，后者是要写入 topic/分区中的下一个记录的偏移量。）

对现有 ListOffsets API

的这个扩展允许用户通过询问最近写入的记录偏移量以及它的时间戳来探测分区的活跃度。

## Kafka Connect

### KIP-745: Connect API to restart connector and tasks

在 Kafka Connect 中，connector 在运行时被表示为一组 Connector 类实例和一个或多个 Task 类实例，通过 Connect REST API 对 connectors

的大多数操作都可以应用到整个组中。一个值得注意的例外是 Connector 和 Task 实例的重启端点（restart endpoints）。

要重新启动整个连接器，用户必须单独调用以重新启动连接器实例和任务实例。在 3.0 中，KIP-745 使用户能够通过一次调用重新启动所有或仅失败的连接器的 Connector 和 Task 实例。此功能是一项附加功能，重启 REST API 的先前行为保持不变。

### KIP-738: Removal of Connect's internal converter properties

在之前的主要版本（Apache Kafka 2.0）中弃用之后，`internal.key.converter` 和 `internal.value.converter` 作为配置属性和前缀在 Connect worker 的配置中被删除。接下来，内部 Connect 主题将专门使用 `JsonConverter` 来存储没有嵌入模式的记录。任何使用不同转换器的现有 Connect 集群都必须将其内部主题移植到新格式（有关升级路径的详细信息，请参阅 KIP-738）。

### KIP-722: Enable connector client overrides by default

从 Apache Kafka 2.3.0 开始，Connect worker 可以配置为允许连接器配置覆盖连接器使用的 Kafka 客户端属性。这是一个广泛使用的特性，现在随着主要版本的发布，默认情况下启用了覆盖连接器客户端属性的功能（`connector.client.config.override.policy` 默认为 All）。

### KIP-721: Enable connector log contexts in Connect Log4j configuration

另一个在 2.3.0 中引入但到目前为止尚未默认启用的功能是连接器日志上下文（connector log contexts）。这在 3.0 中发生了变化，连接器上下文默认添加到 Connect 工作器的 log4j 日志 pattern 中。从以前的版本升级到 3.0 将通过添加连接器上下文来更改 log4j 导出的日志行格式。

## Kafka Streams

### KIP-695: Further improve Kafka Streams timestamp synchronization

KIP-695 增强了 Streams 任务如何选择获取记录的语义，并扩展了配置属性 `max.task.idle.ms` 的含义和可用值。此更改需要 Kafka Consumer API 中名为 `currentLag` 的新方法；如果本地已知，它能够返回一个特定分区的消费者延迟，并且不需要和 Kafka Broker 交互。

### KIP-715: Expose committed offset in streams

从 3.0 开始，TaskMetadata 接口增加了三个新方法：committedOffsets、endOffsets 和 timeCurrentIdlingStarted。这些方法允许 Streams 应用程序跟踪任务的进度和运行状况。

### KIP-740: Clean up public API in TaskId

KIP-740 中对 TaskId 类进行了重大更新。一些方法和所有内部字段都已弃用，新的 subtopology() 和 partition() getters 将取代旧的 topicGroupId 和分区字段(另请参阅 KIP-744 以了解相关更改和对 KIP-740 的修正)。

### KIP-744: Migrate TaskMetadata and ThreadMetadata to an interface with internal implementation

KIP-744 对 KIP-740 中提出的更改做了进一步修改，并将实现与许多类的公共 API 分开。为了实现这一点，引入了新的接口 TaskMetadata、ThreadMetadata 和 StreamsMetadata，同时弃用了具有相同名称的现有类。

### KIP-666: Add Instant-based methods to ReadOnlySessionStore

交互式查询 API 使用 ReadOnlySessionStore 和 SessionStore 接口中的一组新方法进行了扩展，这些方法接受 Instant 数据类型的参数。此更改将影响需要实现新方法的任何自定义只读交互式查询会话存储实现。

### KIP-622: Add currentSystemTimeMs and currentStreamTimeMs to ProcessorContext

ProcessorContext 在 3.0 中新增了两个新方法：currentSystemTimeMs 和 currentStreamTimeMs。新方法使用户能够分别查询缓存的系统时间和流时间（streams time），并且可以在生产和测试代码中以统一的方式使用它们。

### KIP-743: Remove config value 0.10.0-2.4 of Streams built-in metrics version config

3.0 中取消了对 Streams 内置指标中遗留指标结构（legacy metrics structure）的支持。KIP-743 正在从配置属性 built.in.metrics.version 中删除 0.10.0-2.4 的值。这使得 latest 成为该属性目前唯一有效值（从 2.5 开始就是默认值）。

### KIP-741: Change default SerDe to be null

删除了默认 SerDe 属性之前的默认值，Streams 之前的默认值为 ByteArraySerde。从 3.0 开始，没有默认值，并且用户需要在 API 中根据需要设置他们的 SerDe，或者在其 Streams 配置中通过 DEFAULT\_KEY\_SERDE\_CLASS\_CONFIG 和 DEFAULT\_VALUE\_SERDE\_CLASS\_CONFIG 设置默认值。先前的默认值几乎总是不适用于实际应用程序，造成更多的混乱而不是方便。

### KIP-733: Change Kafka Streams default replication factor config

随着主要版本的发布，Streams 配置属性 `replication.factor` 的默认值从 1 更改为 -1。这将允许新的 Streams 应用使用在 Kafka Broker 中定义的默认复制因子，因此当它们迁移到生产环境时，不需要设置这个配置值。注意，新的默认值需要 Kafka Brokers 2.5 或更高版本。

## KIP-732: Deprecate eos-alpha and replace eos-beta with eos-v2

另一个在 3.0 中被弃用的 Streams 配置值是 `exactly_once` 作为 `processing.guarantee` 属性的值。`exactly_once` 值对应的是精确一次语义 (Exactly Once Semantics, EOS) 的原始实现，任何连接到 Kafka 集群版本 0.11.0 或更新版本的 Streams 应用程序都可以使用。EOS 的第一个实现已经被 Streams 中的 EOS 的第二个实现所取代，后者由 `processing.guarantee` 属性中的 `exact_once_beta` 配置表示。接下来，名称 `exactly_once_beta` 也被弃用，并被新的 `exactly_once_v2` 取代。在下一个主要版本 (4.0) 中，`exactly_once` 和 `exactly_once_beta` 都将被删除，仅留下 `exactly_once_v2` 作为 EOS 交付保证的唯一选项。

## KIP-725: Streamlining configurations for WindowedSerializer and WindowedDeserializer

配置属性 `default.windowed.key.serde.inner` 和 `default.windowed.value.serde.inner` 被弃用，取而代之的是一个新的属性 `windowed.inner.class.serde`，供 Kafka 消费者使用。Kafka Streams 用户被推荐配置他们的窗口 SerDe，通过传递这个到 SerDe 构造器，然后在所有的地方使用这个 SerDe。

## KIP-633: Deprecate 24 hour default for the grace period in Streams

在 Kafka Streams 中，允许窗口操作根据一个称为宽限期的配置属性处理窗口外的记录。以前，这个配置是可选的，很容易被忽略，导致默认为 24 小时。这经常使 Suppression 操作符的用户感到困惑，因为它会缓冲记录，直到宽限期结束，因此会增加 24 小时的延迟。在 3.0 中，Windows 类通过工厂方法得到增强，这些工厂方法要求使用自定义宽限期或根本没有宽限期来构造它们。应用了默认的 24 小时宽限期的旧工厂方法已经被弃用，相应的 `grace()` api 与已经设置了此配置的新工厂方法不兼容。

## KIP-623: Add "internal-topics" option to streams application reset tool

通过添加新的命令行参数：`--internal-topics`，使用应用程序重置工具 `kafka-streams-application-reset` 的 Streams 变得更加灵活。新参数接受一个以逗号分隔的主题名称列表，这些主题名称对应于可以计划使用此应用程序工具删除的内部主题。将此新参数与现有参数 `--dry-run` 结合使用，允许用户在实际执行删除操作之前确认将删除哪些主题并在必要时指定其中的一个子集。

## MirrorMaker

### KIP-720: Deprecate MirrorMaker v1

在 3.0 版本中，不推荐使用 MirrorMaker 的第一个版本。今后，新功能和主要改进将集中在

MirrorMaker2 (MM2)上。

## KIP-716: Allow configuring the location of the offset-syncs topic with MirrorMaker2

在 3.0 中，用户现在可以配置 MirrorMaker2 创建和存储用于转换消费者组偏移量的内部主题的位置。这将允许 MirrorMaker2 的用户将源 Kafka 集群维护为严格只读的集群，并使用不同的 Kafka 集群来存储偏移记录（即目标 Kafka 集群，甚至是源和目标集群之外的第三个集群）。

## 总结

Apache Kafka 3.0 是 Apache Kafka 项目向前迈出的重要一步。了解更多：

- 查阅[release notes](#) 完整的更改列表；
- 观看[发布视频](#)了解更多信息；
- 下载 [Apache Kafka 3.0.0](#) 并使用其中的新功能。

这是一项巨大的社区努力，因此感谢为此版本做出贡献的所有人，包括我们的所有用户以及我们的 141 位作者和审阅者：

A. Sophie Blee-Goldman, Adil Houmadi, Akhilesh Dubey, Alec Thomas, Alexander Iskuskov, Almog Gavra, Alok Nikhil, Alok Thatikunta, Andrew Lee, Bill Bejeck, Boyang Chen, Bruno Cadonna, CHUN-HAO TANG, Cao Manh Dat, Cheng Tan, Chia-Ping Tsai, Chris Egerton, Colin P. McCabe, Cong Ding, Daniel Urban, Daniyar Yeralin, David Arthur, David Christle, David Jacot, David Mao, David Osvath, Davor Poldrugo, Dejan Stojadinovi, Dhruvil Shah, Diego Erdody, Dong Lin, Dongjoon Hyun, Dániel Urbán, Edoardo Comar, Edwin Hobor, Eric Beaudet, Ewen Cheslack-Postava, Gardner Vickers, Gasparina Damien, Geordie, Greg Harris, Gunnar Morling, Guozhang Wang, Gwen (Chen) Shapira, Ignacio Acu, a Frías, Igor Soares, Ismael Juma, Israel Ekpo, Ivan Ponomarev, Ivan Yurchenko, Jason Gustafson, Jeff Kim, Jim Galasyn, Jim Hurne, JoelWee, John Gray, John Roesler, Jorge Esteban Quilcate Otoy, Josep Prat, José Armando García Sancio, Juan Gonzalez-Zurita, Jun Rao, Justin Mclean, Justine Olshan, Kahn Cheny, Kalpesh Patel, Kamal Chandraprakash, Konstantine Karantasis, Kowshik Prakasam, Leah Thomas, Lee Dongjin, Lev Zemlyanov, Liu Qiang, Lucas Bradstreet, Luke Chen, Manikumar Reddy, Marco Aurelio Lotz, Matthew de Detrich, Matthias J. Sax, Michael G. Noll, Michael Noll, Mickael Maison, Nathan Lincoln, Niket Goel, Nikhil Bhatia, Omnia G H Ibrahim, Peng Lei, Phil Hardwick, Rajini Sivaram, Randall Hauch, Rohan Desai, Rohit Deshpande, Rohit Sachan, Ron Dagostino, Ryan Dielhenn, Ryanne Dolan, Sanjana Kaundinya, Sarwar Bhuiyan, Satish Duggana, Scott Hendricks, Sergio Pe, a, Shao Yang Hong, Shay Elkin, Stanislav Vodetskyi, Sven Erik Knop, Tom Bentley, UnityLung, Uwe Eisele, Vahid Hashemian, Valery Kokorev, Victoria Xia, Viktor Somogyi-Vass, Viswanathan Ranganathan, Vito Jeng, Walker Carlson, Warren Zhu, Xavier Léauté, YiDing-Duke, Zara Lim, Zhao Haiyuan, bmaidics, cyc, dengziming, feyman2016, high.lee, iamgd67, iczellion, ketulgupta1995, lamberken, loboya~, nicolasguyomar, prince-mahajan, runom, shenwenbing, thomaskwscott, tinawenqiao, vamossagar12, wenbingshen, wyccccc, xjin-Confluent, zhaohaidao

本文翻译自：[What's New in Apache Kafka 3.0.0](#)

本博客文章除特别声明，全部都是原创！  
原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。  
本文链接：[【】（）](#)