

第八节：分布式恢复---进阶篇

1.概述：

每当一个MySQL server新加入或者重新加入一个MGR集群时，它都必须追平集群内相差的事务，保证这个节点的数据和集群内最新的数据是同步的。这个新加入集群的节点在追平集群中的数据或者重新加入集群的节点追平它脱离集群后到现在这段时间内相差的事务数据的过程称为分布式恢复。

申请加入集群的节点首先检查group_replication_applier通道中的中继日志，检查该节点目前尚未从集群中同步过来的事务数据。如果是重新加入集群的节点，则该节点会找到在离开集群后到现在和集群最新数据中未回放的事务数据，在这种情况下，该节点首先会应用这些未同步的事务。对于新加入集群的节点，直接从一个引导节点上进行全量数据恢复即可。

此后，新加入的节点和集群中现有的online状态的节点（引导节点）建立连接进行状态转移。新加入的节点从集群中的引导节点中同步加入集群之前或者离开集群后到现在未同步过来的数据，这些相差的事务由集群中的引导节点提供。接下来，新加入的节点应用从集群中的引导节点同步过来的未进行应用的事务。此时申请加入集群的节点将应用在状态传输过程中集群内新事务写入的数据。（此时集群内新事物写入的数据暂时存放在缓存队列中，并未将数据写入磁盘中）完成此过程后，新加入的节点的数据和整个集群的数据相比，处于一个追平的状态，并且该节点设置为online状态。

注意：新加入集群的节点，不论是之前有没有在此集群中，都会先随机选一个online节点先同步该节点和集群相差的事务。

组复制在分布式恢复期间用下述方法实现状态传输：

使用克隆插件的功能进行远程克隆操作，该插件可从MySQL 8.0.17开始支持。要使用这种方法，必须在引导节点和新加入的节点上提前安装克隆插件。组复制会自动配置所需的克隆插件设置，并管理远程克隆操作。

从引导节点的二进制日志复制数据并在新加入的节点上应用这些事务。此方法需要在引导节点和加入节点之间建立的名为group_replication_recovery的标准异步复制通道。

在加入节点上执行START GROUP_REPLICATION后，组复制会自动选择上述方法的最佳组合进行状态转移。为此，组复制将会检查集群中哪些现有节点适合用作引导节点，加入节点需要引导节点传输多少事务，以及是否有事务不再存在于集群中任意节点的二进制日志文件中。如果加入节点与引导节点之间的事务差距很大，或者如果某些要求的事务不在引导节点的二进制日志文件中，则组复制将通过远程克隆操作开始分布式恢复。如果没有较大的事务间隙，或者未安装克隆插件，则组复制将直接从引导节点的二进制日志进行状态转移。

在远程克隆操作期间，将删除加入节点上的现有数据，并用引导节点数据的副本替换。当远程克隆操作完成并且新加入节点已重新启动时，将继续执行来自引导节点二进制日志来进行状态转移，以获取在进行远程克隆操作时集群所写入的增量数据。

在从引导节点的二进制日志进行状态转移期间，新加入节点从引导节点的二进制日志中复制并应用所需的事务，并在收到事务时应用事务，直到二进制日志记录新加入节点加入了集群。（当加入节点成功加入集群时，二进制日志中会记录对应的视图更改event）在此过程中，加入节点将缓冲该集群应用的新事务数据。从二进制日志的状态传输完成后，新加入节点将应用缓冲的事务。

当加入节点与该集群的所有事务保持最新时，该节点将在设置为online状态并可以作为普通节点加入集群，并且分布式恢复已完成。

ps：从二进制日志进行状态转移是组复制进行分布式恢复的基本机制，并且如果未将复制组中的引导节点和加入节点设置为支持克隆。由于从二进制日志进行状态转移是基于经典的异步复制，因此，如果加入该集群的MySQL server根本没有该集群的数据，或者从非常旧的备份中获取了数据，则可能要花费很长时间来恢复最新数据。因此，在这种情况下，建议在将MySQL server添加到集群之前，则应通过传输

集群中已有节点的相当近期的快照来使用集群的数据对其进行设置。这可以最大程度地减少分布式恢复所需的时间，并减少对引导节点的影响，因为引导节点必须保留和传输较少的二进制日志文件。

2. 分布式恢复的连接：

当加入节点连接到现有节点中的引导节点进行分布式恢复期间的状态转移时，加入节点充当客户端，而引导节点充当服务端。当通过此连接（使用异步复制通道group_replication_recovery）从引导节点的二进制日志进行状态转移时，加入节点充当副本，引导节点充当源端。通过此连接进行远程克隆操作时，新加入节点充当全量数据接收者，引导节点充当全量数据提供者。应用于组复制上下文之外的角色的配置设置也可以应用于组复制，除非它们被特定于组复制的配置设置或行为所覆盖。

现有节点提供给新加入节点以进行分布式恢复的连接与组复制用于集群内节点之间的通信的连接是不同的。

组通信引擎用于组复制（XCom，Paxos变体），用于远程XCom实例之间的TCP通信的连接由group_replication_local_address系统变量指定。此连接用于集群内online节点之间的TCP / IP消息传递。与本地实例的通信是通过使用内存内共享的传输通道进行的。

对于分布式恢复，直到MySQL 8.0.20为止，集群内的节点都将其标准SQL客户端连接提供给加入节点，这由MySQL Server的主机名和端口系统变量指定。如果report_port系统变量指定了备用端口号，则改用该端口号。

从MySQL 8.0.21开始，组成员可以将分布式恢复端点的替代列表作为加入成员的专用客户端连接，从而使使得独立于成员的常规客户端用户的连接可以用来控制分布式恢复。可以使用group_replication_advertise_recovery_endpoints系统变量来指定此列表，并且成员在加入组时将其分布式恢复端点的列表传输到该组。默认值为成员继续提供与早期版本相同的标准SQL客户端连接。

PS：

如果加入节点无法使用MySQL Server的系统变量定义的主机名正确识别其他节点，则分布式恢复可能会失败。建议运行MySQL的操作系统使用DNS或本地设置具有正确配置的唯一主机名。可以在“performance_schema”库下的Replication_group_members表的Member_host列中验证server用于SQL客户端连接的主机名。如果多个组成员将操作系统设置的默认主机名外部化，则加入节点有可能无法将其解析为正确的地址，并且无法连接以进行分布式恢复。在这种情况下，可以使用MySQL Server的report_host系统变量来配置由每个server外部化的唯一主机名。

加入节点为分布式恢复建立连接的步骤如下：

当节点加入集群时，它会使用group_replication_group_seeds系统变量中列表中包含的一个种子节点进行连接，最初使用该列表中指定的group_replication_local_address连接。种子节点可能是集群数据的子集。

通过此连接，种子节点使用组复制的成员资格服务以视图的形式向加入的节点提供集群中所有online节点的列表。成员资格信息包括每个成员为分布式恢复提供的分布式恢复端点或标准SQL客户端连接的详细信息。

加入节点从此列表中选择合适的online节点作为其引导节点进行分布式恢复

加入节点尝试使用引导节点的分布式恢复端点来连接到引导节点，并按列表中指定的顺序依次尝试连接每个端点。如果引导节点没有提供端点，则加入节点将尝试使用引导节点的标准SQL客户端连接进行连接。连接的SSL要求由group_replication_recovery_ssl_*选项指定。

如果加入节点无法连接到指定的引导节点，则它将与其它合适的引导节点重试连接。如果加入节点在没有建立连接的情况下耗尽了端点的广播列表，则它不会回退到引导节点的标准SQL客户端连接，而是切换到另一个引导节点尝试重新建立连接。

加入节点与引导节点建立分布式恢复连接时，它将使用该连接进行状态转移，加入节点的日志中显示了所使用的连接的主机和端口。如果使用远程克隆操作，则在操作结束时重新启动加入节点时，它将与新的引导节点建立连接，从引导节点的二进制日志进行状态转移。这可能是与用于远程克隆操作的引导节点不同的连接，也可能是与引导节点建立相同的连接。无论如何，分布式恢复将以相同的方式与引导节点建立连接。

2.1 分布式恢复端地址的查找：

`group_replication_advertise_recovery_endpoints`系统变量作为分布式恢复端提供的IP地址，不必为MySQL Server配置（也就是说，不必由`admin_address`系统变量或在`bind_address`系统变量的列表中指定）。

为MySQL Server配置为分布式恢复端提供的端口，必须由`port`，`report_port`或`admin_port`系统变量指定。必须在这些端口上侦听TCP / IP连接。如果指定`admin_port`，则用于分布式恢复的复制用户需要`SERVICE_CONNECTION_ADMIN`权限才能连接。选择`admin_port`可使分布式恢复连接与常规MySQL客户端连接分开。

加入节点按列表中指定的顺序依次尝试每个端点。如果将`group_replication_advertise_recovery_endpoints`设置为`DEFAULT`而不是端点列表，则将提供标准SQL客户端连接。标准SQL客户端连接不会自动包含在分布式恢复端点列表中，并且如果引导节点的端点列表在没有连接的情况下被用尽，则不会将其作为备用。如果要提供标准SQL客户端连接作为多个分布式恢复端点之一，则必须将其显式包括在`group_replication_advertise_recovery_endpoints`指定的列表中。可以将其放在最后，以便作为连接的最后手段。

无需将组成员的分布式恢复终点（或标准SQL客户端连接，如果未提供终点）添加到`group_replication_ip_allowlist`（来自MySQL 8.0.22）或`group_replication_ip_whitelist`系统变量指定的组复制允许列表中。许可列表仅适用于由`group_replication_local_address`为每个节点指定的地址。加入节点必须具有与允许列表允许的集群的初始连接，以便检索一个或多个地址进行分布式恢复。

设置系统变量和执行`START GROUP_REPLICATION`语句后，将验证列出的分布式恢复端点。如果无法正确解析列表，或者由于服务未在侦听列表而无法在主机上访问任何端点，则组复制将记录错误并且无法启动。

2.2 分布式恢复压缩

在MySQL 8.0.18中，可以选择使用引导节点二进制日志中的状态转移方法为分布式恢复配置压缩。在网络带宽有限的情况下，压缩可以使分布式恢复受益，而引导节点必须将许多事务传输给加入节点。

`group_replication_recovery_compression_algorithm`和`group_replication_recovery_zstd_compression_level`系统变量配置允许的压缩算法以及zstd压缩级别，这些级别用于从引导节点的二进制日志执行状态转移时使用。

这些压缩设置不适用于远程克隆操作。当远程克隆操作用于分布式恢复时，将应用克隆插件的`clone_enable_compression`设置。

2.3 分布式恢复的用户

分布式恢复需要具有正确权限的复制用户，以便组复制可以建立直接的节点到节点的复制通道。复制用户还必须具有正确的权限，如果该复制用户还同充当远程克隆操作中的克隆用户，则在引导节点中该复制用户还必须具有远程克隆相关的权限（`BACKUP_ADMIN`权限）才能充当引导节点上的克隆用户以进行远程克隆操作。除此之外，必须将同一复制用户用于集群内每个节点上的分布式恢复。

2.4分布式恢复和SSL认证

用于分布式恢复的SSL与用于普通组通信的SSL分开配置，这由server的SSL设置和group_replication_ssl_mode系统变量确定。对于分布式恢复连接，可以使用专用的组复制分布式恢复SSL系统变量来配置专门用于分布式恢复的证书和密钥的使用。

默认情况下，SSL不用于分布式恢复连接。设置group_replication_recovery_use_ssl = ON启用，然后配置组复制分布式恢复SSL系统变量，将复制用户设置为使用SSL。

将分布式恢复配置为使用SSL时，组复制会将此设置应用于远程克隆操作以及从引导节点的二进制日志进行状态转移。组复制会自动配置克隆SSL选项（clone_ssl_ca, clone_ssl_cert和clone_ssl_key），以匹配相应组复制分布式恢复选项（group_replication_recovery_ssl_ca, group_replication_recovery_ssl_cert和group_replication_recovery_ssl_key）的设置。

如果未使用SSL进行分布式恢复（group_replication_recovery_use_ssl设置为OFF），并且组复制的复制用户帐户使用caching_sha2_password插件（MySQL 8.0中的默认设置）或sha256_password插件进行身份验证，则RSA密钥对为用于密码交换。在这种情况下，使用group_replication_recovery_public_key_path系统变量指定RSA公共密钥文件，或者使用group_replication_recovery_get_public_key系统变量请求公共密钥。否则整个分布式恢复会因为报错导致恢复失败。

3.利用克隆插件进行分布式恢复：

MySQL Server的克隆插件可从MySQL 8.0.17获得。如果要将远程克隆操作用于集群中的分布式恢复，则必须预先设置现有节点和加入节点才能支持此功能。如果不想在集群中使用此功能，请不要进行设置，在这种情况下，组复制仅使用二进制日志中的状态传输。

要使用克隆插件，必须预先设置至少一个现有的集群节点和加入节点支持远程克隆操作。至少，必须在引导节点和加入节点上安装克隆插件，将BACKUP_ADMIN权限授予复制用户以进行分布式恢复，并将group_replication_clone_threshold系统变量设置为适当的级别。（默认情况下为GTID序列允许的最大值，表示正常情况下，始终优先使用基于二进制日志的状态传输，除非joiner节点所请求的事务在组中任意成员中都不存在，这个时候，如果设置好了克隆功能，则无论该系统变量的值设置为多少，都会触发通过克隆的方式进行分布式恢复，例如：全新初始化的Server申请加入组时。如果不希望使用克隆功能，则不要对其进行安装与配置）为了确保引导节点的最大可用性，建议设置所有当前和将来的集群节点支持远程克隆操作。以便后续有Server加入集群时能够使用远程克隆操作来快速追赶集群中的最新数据。

在从引导节点向加入节点传输数据之前，远程克隆操作会删除加入节点中用户创建的表空间和数据。如果在中途意外终止操作，则加入节点可能只剩下部分数据或没有数据。可以通过重新执行组复制自动执行的远程克隆操作来修复此问题。

这里主要针对远程克隆时使用DATA DIRECTORY子选项指定了一个数据保存路径的情况，指定路径时，数据会保存在指定的目录下，即克隆之后的数据与操作克隆的实例没有关联，需要手动启动实例并指定datadir到保存克隆数据的目录进行启动，当然，MGR插件可以自动执行远程克隆的重试操作（需要保证克隆操作不指定DATA DIRECTORY子选项，在这种情况下，远程克隆数据会覆盖掉操作远程克隆的Server数据，完成远程克隆操作之后，操作远程克隆的Server会基于克隆数据自动重新启动）。另外，克隆插件虽然与组复制配合使用对组复制的管理维护来说更加自动化，但是，克隆插件不要求必须在集群中运行（但MGR插件必须要安装）。

3.1克隆的基本条件

对于组复制，使用克隆插件进行分布式恢复需要注意以下要点和区别：

现有集群节点和加入节点必须已安装克隆插件并处于激活状态。

引导节点和加入节点必须在相同的操作系统上运行，并且必须具有相同的MySQL Server版本（必须为MySQL 8.0.17或更高版本才能支持克隆插件）。因此，克隆不适用于成员运行不同MySQL版本的集群。

引导节点和加入节点必须已安装并激活了“组复制”插件，引导节点上所有激活的其他插件（例如，密钥环插件）也必须在加入节点上处于激活状态。

如果将分布式恢复配置为使用SSL（`group_replication_recovery_use_ssl = ON`），则组复制会将此设置应用于远程克隆操作。组复制会自动配置克隆SSL选项（`clone_ssl_ca`，`clone_ssl_cert`和`clone_ssl_key`）的设置，以匹配相应组复制分布式恢复选项（`group_replication_recovery_ssl_ca`，`group_replication_recovery_ssl_cert`和`group_replication_recovery_ssl_key`）的设置。

不需要在加入节点上为加入集群而在`clone_valid_donor_list`系统变量中设置有效引导节点列表。MGR自动从现有的集群节点中选择引导节点后，组复制会自动配置此设置。注意，远程克隆操作使用server的SQL协议主机名和端口，而非集群成员之间内部通讯的地址和端口。

克隆插件具有许多系统变量，可管理远程克隆操作的网络负载和性能影响。组复制未配置这些设置，因此可以查看它们并根据需要进行设置，也可以将其设置为默认设置，当使用远程克隆操作进行分布式恢复时，克隆插件的`clone_enable_compression`设置将应用于该操作，而不会影响现有配置好的组复制压缩设置。

为了对加入节点调用远程克隆操作，组复制使用内部mysql.session用户，该用户已经具有CLONE_ADMIN特权，因此无需进行特别设置。

作为远程克隆操作的引导节点上的克隆用户，组复制使用为分布式恢复设置的复制用户。因此，必须在所有支持克隆的集群节点上将此复制用户赋予BACKUP_ADMIN特权。在为节点配置组复制时，如果有加入节点，还应向该节点上的复制用户授予此权限，因为加入节点加入集群后，他们可以充当其他加入节点的引导节点。同一复制用户用于每个集群节点上的分布式恢复。要将此权限授予现有节点上的复制用户，可以在禁用二进制日志记录的情况下在每个集群节点上单独执行此语句，或者在启用二进制日志记录的情况下在一个集群的primary节点上执行如下语句：

```
GRANT BACKUP_ADMIN ON *.* TO *rpl_user*@'%';
```

如果在使用START GROUP_REPLICATION以前在提供用户凭据的server上使用CHANGE MASTER TO指定复制用户凭据，请确保在进行任何远程克隆操作之前，从复制元数据存储库中删除该用户凭据。还要确保在加入成员上设置了`group_replication_start_on_boot = OFF`。如果未取消设置用户凭据，则在远程克隆操作期间会将它们转移到加入成员。然后，可能会在原始成员或从其克隆的成员上意外地使用存储的凭据启动`group_replication_recovery`通道。server启动时（包括在远程克隆操作之后）自动启动组复制将使用存储的用户凭据，并且如果未在START GROUP_REPLICATION命令上指定分布式恢复凭据，也将使用它们。

3.2克隆的阈值

设置组成员支持克隆后，`group_replication_clone_threshold`系统变量将指定一个阈值，表示为多少个事务，以便在分布式恢复中使用远程克隆操作。如果引导节点上的事务与加入节点上的事务之间的数量大于此数目，则在技术上可行时，将使用远程克隆操作将状态转移到加入节点。组复制基于现有组成员的`gtid_executed`集来计算是否已超过阈值。在事务间隙较大的情况下使用远程克隆操作，可以将新成员添加到集群中，而无需事先将集群的数据手动传输到服务器，还可以使落后的节点更有效地进行数据追赶。

`group_replication_clone_threshold`组复制系统变量的默认设置非常高（GTID中事务的最大允许序列号），因此，只要有可能从二进制日志转移状态，它都会有效地禁用克隆。要使组复制能够选择更适合状态传输的远程克隆操作，设置系统变量，以指定多少个事务作为要进行克隆的事务间隔。

PS：

不要在活跃的集群中为`group_replication_clone_threshold`使用较低的设置。如果在进行远程克隆操作的同时集群中发生了超过阈值的事务，则加入成员在重新启动后会再次触发远程克隆操作，并且可以无限期地继续进行。为避免这种情况，确保将阈值设置为一个可靠的数字，该阈值应大于在远程克隆操作所花费的时间段内集群中预期发生的事务数。

当无法从引导节点的二进制日志进行状态转移时，组复制尝试执行远程克隆操作，而不管此时的阈值如何，例如，因为加入成员所需的事务在任何现有组成员的二进制日志中均不可用。组复制基于现有组成员的gtid_purged集对此进行标识。当所需的事务在任何成员的二进制日志文件中不可用时，不能使用group_replication_clone_threshold系统变量来停用克隆，因为在这种情况下，克隆是手动将数据传输到加入节点的唯一选择。

3.3克隆操作

设置集群节点和加入节点进行克隆后，组复制将管理远程克隆操作。远程克隆操作需要一些时间才能完成，具体取决于数据的大小。

performance_schema.clone_progress表中记录了整个克隆操作的每一个阶段及其对应的阶段信息，每一个阶段会生成一行记录（注意，该表中只记录一次克隆操作的过程信息，下一次执行克隆操作时，上一次的信息会被覆盖）

```
select * from clone_progress;
+-----+-----+-----+-----+-----+-----+-----+-----+
-----+-----+-----+-----+-----+-----+-----+-----+
-----+
| ID | STAGE | STATE | BEGIN_TIME | END_TIME | THREADS | ESTIMATE | DATA |
NETWORK | DATA_SPEED | NETWORK_SPEED |
+-----+-----+-----+-----+-----+-----+-----+-----+
-----+-----+-----+-----+-----+-----+-----+-----+
-----+
| 1 | DROP DATA | Completed | 2019-10-08 16:46:58.757964 | 2019-10-08
16:46:59.128436 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | FILE COPY | Completed | 2019-10-08 16:46:59.128766 | 2019-10-08
16:47:16.857536 | 8 | 8429731840 | 8429731840 | 8430190882 | 0 | 0 |
| 1 | PAGE COPY | Completed | 2019-10-08 16:47:16.857737 | 2019-10-08
16:47:17.159531 | 8 | 0 | 0 | 785 | 0 | 0 |
| 1 | REDO COPY | Completed | 2019-10-08 16:47:17.159748 | 2019-10-08
16:47:17.460516 | 8 | 2560 | 2560 | 3717 | 0 | 0 |
| 1 | FILE SYNC | Completed | 2019-10-08 16:47:17.460788 | 2019-10-08
16:47:20.926184 | 8 | 0 | 0 | 0 | 0 | 0 |
| 1 | RESTART | Completed | 2019-10-08 16:47:20.926184 | 2019-10-08
16:47:28.623732 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | RECOVERY | Completed | 2019-10-08 16:47:28.623732 | 2019-10-08
16:47:34.898453 | 0 | 0 | 0 | 0 | 0 | 0 |
+-----+-----+-----+-----+-----+-----+-----+-----+
-----+-----+-----+-----+-----+-----+-----+-----+
-----+
7 rows in set (0.00 sec)
select * from clone_status\G
***** 1. row *****
ID: 1
PID: 0
STATE: Completed
BEGIN_TIME: 2019-10-08 16:46:58.758
END_TIME: 2019-10-08 16:47:34.898
SOURCE: 10.10.30.162:3306
DESTINATION: LOCAL INSTANCE
ERROR_NO: 0
ERROR_MESSAGE:
BINLOG_FILE: mysql-bin.000022
BINLOG_POSITION: 222104704
GTID_EXECUTED: 320675e6-de7b-11e9-b3a9-5254002a54f2:1-4,
aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa:1-2771494
```



```
1 row in set (0.01 sec)
```

PS:

状态转移完成后，组复制将重新启动加入节点以完成该过程。如果在加入节点上设置了 `group_replication_start_on_boot = OFF`，例如，因为在 `START GROUP_REPLICATION` 语句上指定了复制用户凭据，则在重新启动后必须再次手动发布 `START GROUP_REPLICATION`。如果在配置文件中或使用 `SET PERSIST` 语句设置了 `group_replication_start_on_boot = ON` 以及启动组复制所需的其他设置，则无需干预，该过程会自动继续使加入节点设置为 `online` 状态。

远程克隆操作会将引导节点的数据目录下的各种数据文件克隆到加入节点中（表中可能包含了一些配置信息及其用户数据等）。但保存在配置文件（如组复制本地地址配置等）中的组复制成员设置不会被克隆，也不会加入节点上做任何更改。即，组复制相关的配置需要自行配置好，不能跟集群中的现有成员冲突，远程克隆操作只负责克隆数据文件，不会克隆配置信息（当然，如果某些配置信息保存在表里，对于克隆操作来说，也会被当做数据进行克隆）。

如果远程克隆过程花费很长时间，则在MySQL 8.0.22之前的发行版中，在该时间段内为该集群累积的一组认证信息可能会变得太大而无法传输给加入成员。在这种情况下，加入成员会记录一条错误消息，并且不会加入该集群。从MySQL 8.0.22开始，组复制以不同的方式管理应用事务的垃圾收集过程，以避免发生这种情况。在早期版本中，如果确实看到此错误，则在远程克隆操作完成之后，请等待两分钟，以允许进行一轮垃圾收集，以减小集群的认证信息的大小。然后在加入成员上发出以下声明，以使其停止尝试应用先前的认证信息集：

```
RESET SLAVE FOR CHANNEL group_replication_recovery;  
  
RESET REPLICA FOR CHANNEL group_replication_recovery; (从8.0.22开始)
```

引导节点中用于组复制专用通道 `group_replication_recovery` 的用户凭证（复制用户和密码），在克隆操作完成之后，会被新成员使用，所以，该用户和密码及其权限必须在新成员中也有效。因此，所有组成员才能够使用相同的复制用户和密码通过远程克隆操作接收状态传输进行分布式恢复。但是，组复制会保留与使用SSL相关的组复制通道设置，这些设置对单个成员来说可以是惟一的（即，每个组成员使用不同的复制用户和密码）。如果使用了 `PRIVILEGE_CHECKS_USER` 帐户来帮助保护复制应用线程（从MySQL 8.0.18开始，可以创建一个具有特定权限的用户账号，然后将其指定为 `PRIVILEGE_CHECKS_USER` 帐户，这样可以防止将未经授权或意外将具有特权的账号用于组复制通道），则在克隆操作完成之后新加入成员不会使用该用户帐户作为组复制通道的用户。此时必须为组复制通道手工指定合适的复制用户。

如果引导节点用于 `group_replication_recovery` 复制通道的复制用户凭据已使用 `CHANGE MASTER TO` 语句存储在复制元数据存储库中，则在克隆后将它们转移到加入成员并由其使用，并且它们在此处必须有效。因此，使用存储的凭据，所有通过远程克隆操作接收状态转移的组成员都会自动接收复制用户和密码，进行分布式恢复。如果在 `START GROUP_REPLICATION` 语句上指定了复制用户凭据，则这些凭据将用于启动远程克隆操作，但是在克隆后它们不会传输到加入节点并由其使用。如果不希望将凭据转移到新的server上并记录在那里，确保在进行远程克隆操作之前取消设置它们，并使用 `START GROUP_REPLICATION` 代替提供它们。

ps:如果已使用 `PRIVILEGE_CHECKS_USER` 帐户来帮助保护复制应用程序，则从MySQL 8.0.19开始，会将 `PRIVILEGE_CHECKS_USER` 帐户以及来自引导节点的相关设置克隆出来。如果将加入节点设置为在启动时启动组复制，它将自动使用该帐户在相应的复制通道上进行权限检查。（在MySQL 8.0.18中，由于许多限制，建议不要将 `PRIVILEGE_CHECKS_USER` 帐户用于组复制通道。）

3.4克隆的其他用处

组复制启动并管理用于分布式恢复的克隆操作。设置为支持克隆的组成员也可以参与用户手动启动的克隆操作。例如，可能希望通过从组成员作为引导节点来进行克隆来创建新的MySQL实例，但是不希望新的服务器实例立即加入或可能永远不会加入该集群。

在所有支持克隆的发行版中，可以手动启动涉及停止了组复制的组成员的克隆操作。由于克隆要求引导节点和接收数据的节点上的克隆插件必须匹配，因此即使不希望该实例加入集群，也必须在另一个实例上安装并激活组复制插件。可以通过发出以下语句来安装插件：

```
INSTALL PLUGIN group_replication SONAME 'group_replication.so';
```

在MySQL 8.0.20之前的发行版中，如果操作涉及正在运行“组复制”的组成员，则无法手动启动克隆操作。从MySQL 8.0.20开始，只要克隆操作不会删除和替换接收者上的数据，就可以执行此操作。因此，如果正在运行组复制，则用于启动克隆操作的语句必须包含DATA DIRECTORY子句。