

At this point server s3 is booted and running, has joined the group and caught up with the other servers in the group. Consulting the `performance_schema.replication_group_members` table again confirms this is the case.

```
mysql> SELECT * FROM performance_schema.replication_group_members;
```

CHANNEL_NAME	MEMBER_ID	MEMBER_HOST	MEMBER_PORT	MEMBER_STATE
group_replication_applier	395409e1-6dfa-11e6-970b-00212844f856	s1	3306	ONLINE
group_replication_applier	7eb217ff-6df3-11e6-966c-00212844f856	s3	3306	ONLINE
group_replication_applier	ac39f1e6-6dfa-11e6-a69d-00212844f856	s2	3306	ONLINE

Issuing this same query on server s2 or server s1 yields the same result. Also, you can verify that server s3 has caught up:

```
mysql> SHOW DATABASES LIKE 'test';
```

Database (test)
test

```
mysql> SELECT * FROM test.t1;
```

c1	c2
1	Luis

```
mysql> SHOW BINLOG EVENTS;
```

Log_name	Pos	Event_type	Server_id	End_log_pos	Info
binlog.000001	4	Format_desc	3	123	Server ver: 8.0.25-log, Binlog ver: 4
binlog.000001	123	Previous_gtids	3	150	
binlog.000001	150	Gtid	1	211	SET @@SESSION.GTID_NEXT= 'aaaaaaaa-aaaa-aa
binlog.000001	211	Query	1	270	BEGIN
binlog.000001	270	View_change	1	369	view_id=14724832985483517:1
binlog.000001	369	Query	1	434	COMMIT
binlog.000001	434	Gtid	1	495	SET @@SESSION.GTID_NEXT= 'aaaaaaaa-aaaa-aa
binlog.000001	495	Query	1	585	CREATE DATABASE test
binlog.000001	585	Gtid	1	646	SET @@SESSION.GTID_NEXT= 'aaaaaaaa-aaaa-aa
binlog.000001	646	Query	1	770	use `test`; CREATE TABLE t1 (c1 INT PRIMAR
binlog.000001	770	Gtid	1	831	SET @@SESSION.GTID_NEXT= 'aaaaaaaa-aaaa-aa
binlog.000001	831	Query	1	890	BEGIN
binlog.000001	890	Table_map	1	933	table_id: 108 (test.t1)
binlog.000001	933	Write_rows	1	975	table_id: 108 flags: STMT_END_F
binlog.000001	975	Xid	1	1002	COMMIT /* xid=29 */
binlog.000001	1002	Gtid	1	1063	SET @@SESSION.GTID_NEXT= 'aaaaaaaa-aaaa-aa
binlog.000001	1063	Query	1	1122	BEGIN
binlog.000001	1122	View_change	1	1261	view_id=14724832985483517:2
binlog.000001	1261	Query	1	1326	COMMIT
binlog.000001	1326	Gtid	1	1387	SET @@SESSION.GTID_NEXT= 'aaaaaaaa-aaaa-aa
binlog.000001	1387	Query	1	1446	BEGIN
binlog.000001	1446	View_change	1	1585	view_id=14724832985483517:3
binlog.000001	1585	Query	1	1650	COMMIT

## 18.2.2 Deploying Group Replication Locally

The most common way to deploy Group Replication is using multiple server instances, to provide high availability. It is also possible to deploy Group Replication locally, for example for testing purposes. This section explains how you can deploy Group Replication locally.