



Aurora MySQL 发布说明

Amazon Aurora



Amazon Aurora: Aurora MySQL 发布说明

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商标和商业外观不得用于任何非 Amazon 的商品或服务，也不得以任何可能引起客户混淆、贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其他商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

Table of Contents

Aurora MySQL 发布说明	1
Aurora MySQL 版本 3	2
Aurora MySQL 更新：2024-03-07 (版本 3.06.0 , 兼容 MySQL 8.0.34)	3
新功能	3
改进	4
集成了 MySQL 社区版本错误修复	6
Aurora MySQL 更新：2024-01-31 (版本 3.05.2 , 兼容 MySQL 8.0.32)	7
改进	7
集成了 MySQL 社区版本错误修复	8
Aurora MySQL 更新：2023-11-21 (版本 3.05.1 , 兼容 MySQL 8.0.32)	8
改进	9
集成了 MySQL 社区版本错误修复	9
Aurora MySQL 更新：2023-10-30 (版本 3.05.0.1 , 与 MySQL 8.0.32 兼容) 测试版	9
改进	10
Aurora MySQL 更新：2023-10-25 (版本 3.05.0 , 与 MySQL 8.0.32 兼容)	10
改进	11
集成了 MySQL 社区版本错误修复	14
Aurora MySQL 更新：2024-03-15 (版本 3.04.2 , 兼容 MySQL 8.0.28)	14
改进	15
集成了 MySQL 社区版本错误修复	17
Aurora MySQL 更新：2023-11-13 (版本 3.04.1 , 兼容 MySQL 8.0.28) 默认	17
改进	18
集成了 MySQL 社区版本错误修复	19
Aurora MySQL 更新：2023-07-31 (版本 3.04.0 , 兼容 MySQL 8.0.28)	19
改进	20
集成了 MySQL 社区版本错误修复	25
Aurora MySQL 更新：2023-12-08 (版本 3.03.3 , 与 MySQL 8.0.26 兼容)	25
改进	26
集成了 MySQL 社区版本错误修复	27
Aurora MySQL 更新：2023-08-29 (版本 3.03.2 , 与 MySQL 8.0.26 兼容)	27
改进	28
集成了 MySQL 社区版本错误修复	29
Aurora MySQL 更新：2023-05-11 (版本 3.03.1 , 与 MySQL 8.0.26 兼容)	29
改进	30

集成了 MySQL 社区版本错误修复	31
Aurora MySQL 更新：2023-03-01 (版本 3.03.0 , 与 MySQL 8.0.26 兼容) 不支持升级到此版本。	31
改进	32
集成了 MySQL 社区版本错误修复	34
Aurora MySQL 更新：2023-04-17 (版本 3.02.3 , 与 MySQL 8.0.23 兼容) 标准支持将于 2024 年 1 月 15 日结束。	35
改进	36
Aurora MySQL 更新：2022-11-18 (版本 3.02.2 , 与 MySQL 8.0.23 兼容) 标准支持将于 2024 年 1 月 15 日结束	36
改进	37
集成了 MySQL 社区版本错误修复	34
Aurora MySQL 更新：2022-09-07 (版本 3.02.1 , 与 MySQL 8.0.23 兼容) 标准支持将于 2024 年 1 月 15 日结束。不支持升级到此版本。	39
改进	40
Aurora MySQL 更新：2022-04-20 (版本 3.02.0 , 与 MySQL 8.0.23 兼容) 标准支持将于 2024 年 1 月 15 日结束。不支持升级到此版本。	41
改进	41
集成了 MySQL 社区版本错误修复	43
Aurora MySQL 更新：2022-04-15 (版本 3.01.1 , 与 MySQL 8.0.23 兼容) 标准支持将于 2024 年 1 月 15 日结束。不支持升级到此版本。	43
改进	44
集成了 MySQL 社区版本错误修复	46
Aurora MySQL 更新：2021-11-18 (版本 3.01.0 , 与 MySQL 8.0.23 兼容) 标准支持将于 2024 年 1 月 15 日结束。不支持升级到此版本。	46
改进	47
Aurora MySQL 版本 2	48
Aurora MySQL 更新：2024-03-19 (版本 2.12.2 , 兼容 MySQL 5.7.44)	50
改进	50
集成了 MySQL 社区版本错误修复	51
Aurora MySQL 版本 2 中不支持的功能	51
MySQL 5.7 兼容性	52
Aurora MySQL 更新：2023-12-28 (版本 2.12.1 , 与 MySQL 5.7.40 兼容)	52
改进	53
集成了 MySQL 社区版本错误修复	55
Aurora MySQL 版本 2 中不支持的功能	55

MySQL 5.7 兼容性	55
Aurora MySQL 更新：2023-10-25 (版本 2.12.0.1 , 与 MySQL 5.7.40 兼容) 测试版	56
改进	56
Aurora MySQL 更新：2023-07-25 (版本 2.12.0 , 与 MySQL 5.7.40 兼容)	57
改进	57
集成了 MySQL 社区版本错误修复	60
Aurora MySQL 版本 2 中不支持的功能	60
MySQL 5.7 兼容性	60
Aurora MySQL 更新：2024-03-26 (版本 2.11.5 , 兼容 MySQL 5.7.12)	61
改进	62
Aurora MySQL 版本 2 中不支持的功能	63
MySQL 5.7 兼容性	63
Aurora MySQL 更新：2023-10-17 (版本 2.11.4 , 与 MySQL 5.7.12 兼容)	64
改进	64
集成了 MySQL 社区版本错误修复	66
Aurora MySQL 版本 2 中不支持的功能	66
MySQL 5.7 兼容性	66
Aurora MySQL 更新：2023-06-09 (版本 2.11.3 , 与 MySQL 5.7.12 兼容)	67
改进	68
Aurora MySQL 版本 2 中不支持的功能	69
MySQL 5.7 兼容性	69
Aurora MySQL 更新：2023-03-24 (版本 2.11.2 , 兼容 MySQL 5.7.12)	69
改进	70
Aurora MySQL 版本 2 中不支持的功能	71
MySQL 5.7 兼容性	71
Aurora MySQL 更新：2023-02-14 (版本 2.11.1 , 与 MySQL 5.7.12 兼容)	72
改进	73
与 Aurora MySQL 版本 1 进行比较	73
MySQL 5.7 兼容性	74
Aurora MySQL 更新：2022-10-25 (版本 2.11.0 , 与 MySQL 5.7.12 兼容) 此版本不适用于新创建。	74
改进	75
集成了 MySQL 社区版本错误修复	78
与 Aurora MySQL 版本 1 进行比较	79
MySQL 5.7 兼容性	80
Aurora MySQL 更新：2022-11-01 (版本 2.10.3) (已弃用)	80

改进	81
集成了 MySQL 社区版本错误修复	82
与 Aurora MySQL 版本 1 进行比较	82
MySQL 5.7 兼容性	83
Aurora MySQL 更新：2022-01-26 (版本 2.10.2) (已弃用)	83
改进	84
集成了 MySQL 社区版本错误修复	34
与 Aurora MySQL 版本 1 进行比较	87
MySQL 5.7 兼容性	87
Aurora MySQL 更新：2021-10-21 (版本 2.10.1) (已弃用)	88
改进	88
集成了 MySQL 社区版本错误修复	89
与 Aurora MySQL 版本 1 进行比较	89
MySQL 5.7 兼容性	90
Aurora MySQL 更新：2021-05-25 (版本 2.10.0) (已弃用)	90
改进	91
集成了 MySQL 社区版本错误修复	93
与 Aurora MySQL 版本 1 进行比较	96
MySQL 5.7 兼容性	96
Aurora MySQL 更新：2021-11-12 (版本 2.09.3) (已弃用)	97
改进	97
集成了 MySQL 社区版本错误修复	99
与 Aurora MySQL 版本 1 进行比较	100
MySQL 5.7 兼容性	100
Aurora MySQL 更新：2021-02-26 (版本 2.09.2) (已弃用)	101
改进	101
与 Aurora MySQL 版本 1 进行比较	102
MySQL 5.7 兼容性	103
Aurora MySQL 更新：2020-12-11 (版本 2.09.1) (已弃用)	103
改进	104
集成了 MySQL 社区版本错误修复	105
与 Aurora MySQL 版本 1 进行比较	105
MySQL 5.7 兼容性	106
Aurora MySQL 更新：2020-09-17 (版本 2.09.0) (已弃用)	106
改进	107
集成了 MySQL 社区版本错误修复	111

与 Aurora MySQL 版本 1 进行比较	113
MySQL 5.7 兼容性	113
Aurora MySQL 更新：2022-01-06 (版本 2.08.4) (已弃用)	114
改进	114
与 Aurora MySQL 版本 1 进行比较	115
MySQL 5.7 兼容性	115
Aurora MySQL 更新：2020-11-12 (版本 2.08.3) (已弃用)	116
改进	116
集成了 MySQL 社区版本错误修复	117
与 Aurora MySQL 版本 1 进行比较	117
MySQL 5.7 兼容性	118
Aurora MySQL 更新：2020-08-28 (版本 2.08.2) (已弃用)	118
改进	119
与 Aurora MySQL 版本 1 进行比较	119
MySQL 5.7 兼容性	120
Aurora MySQL 更新：2020-06-18 (版本 2.08.1) (已弃用)	120
改进	121
与 Aurora MySQL 版本 1 进行比较	121
MySQL 5.7 兼容性	122
Aurora MySQL 更新：2020-06-02 (版本 2.08.0) (已弃用)	122
改进	123
集成了 MySQL 社区版本错误修复	125
与 Aurora MySQL 版本 1 进行比较	126
MySQL 5.7 兼容性	126
Aurora MySQL 更新：2023-08-15 (版本 2.07.10, 与 MySQL 5.7.12 兼容)	127
改进	128
Aurora MySQL 版本 2 中不支持的功能	128
MySQL 5.7 兼容性	128
Aurora MySQL 更新：2023-05-04 (版本 2.07.9, 与 MySQL 5.7.12 兼容)	129
改进	130
Aurora MySQL 版本 2 中不支持的功能	130
MySQL 5.7 兼容性	131
Aurora MySQL 更新：2022-06-16 (版本 2.07.8) (已弃用)	131
改进	132
集成了 MySQL 社区版本错误修复	132
与 Aurora MySQL 版本 1 进行比较	133

MySQL 5.7 兼容性	133
Aurora MySQL 更新：2021-11-24 (版本 2.07.7) (已弃用)	134
改进	134
与 Aurora MySQL 版本 1 进行比较	135
MySQL 5.7 兼容性	136
Aurora MySQL 更新：2021-09-02 (版本 2.07.6) (已弃用)	136
集成了 MySQL 社区版本错误修复	137
与 Aurora MySQL 版本 1 进行比较	137
MySQL 5.7 兼容性	137
Aurora MySQL 更新：2021-07-06 (版本 2.07.5) (已弃用)	138
改进	138
与 Aurora MySQL 版本 1 进行比较	139
MySQL 5.7 兼容性	139
Aurora MySQL 更新：2021-03-04 (版本 2.07.4) (已弃用)	140
改进	140
集成了 MySQL 社区版本错误修复	141
与 Aurora MySQL 版本 1 进行比较	141
MySQL 5.7 兼容性	142
Aurora MySQL 更新：2020-11-10 (版本 2.07.3) (已弃用)	142
改进	143
集成了 MySQL 社区版本错误修复	144
与 Aurora MySQL 版本 1 进行比较	145
MySQL 5.7 兼容性	146
Aurora MySQL 更新：2020-04-17 (版本 2.07.2) (已弃用)	146
改进	147
集成了 MySQL 社区版本错误修复	148
与 Aurora MySQL 版本 1 进行比较	148
MySQL 5.7 兼容性	148
Aurora MySQL 更新：2019-12-23 (版本 2.07.1) (已弃用)	149
改进	150
与 Aurora MySQL 版本 1 进行比较	150
MySQL 5.7 兼容性	150
Aurora MySQL 更新：2019-11-25 (版本 2.07.0) (已弃用)	151
改进	152
集成了 MySQL 社区版本错误修复	153
与 Aurora MySQL 版本 1 进行比较	153

MySQL 5.7 兼容性	154
Aurora MySQL 更新：2019-11-22 (版本 2.06.0) (已弃用)	154
改进	155
与 Aurora MySQL 版本 1 进行比较	157
MySQL 5.7 兼容性	157
Aurora MySQL 更新：2019-11-11 (版本 2.05.0) (已弃用)	158
改进	159
MySQL 错误修复汇总	159
与 Aurora MySQL 版本 1 进行比较	160
MySQL 5.7 兼容性	160
Aurora MySQL 更新：2020-08-14 (版本 2.04.9) (已弃用)	161
改进	161
MySQL 错误修复汇总	163
与 Aurora MySQL 版本 1 进行比较	164
MySQL 5.7 兼容性	165
Aurora MySQL 更新：2019-11-20 (版本 2.04.8) (已弃用)	165
改进	166
与 Aurora MySQL 版本 1 进行比较	167
MySQL 5.7 兼容性	167
Aurora MySQL 更新：2019-11-14 (版本 2.04.7) (已弃用)	168
改进	168
与 Aurora MySQL 版本 1 进行比较	169
MySQL 5.7 兼容性	170
Aurora MySQL 更新：2019-09-19 (版本 2.04.6) (已弃用)	170
改进	171
MySQL 错误修复汇总	171
与 Aurora MySQL 版本 1 进行比较	171
MySQL 5.7 兼容性	172
Aurora MySQL 更新：2019-07-08 (版本 2.04.5) (已弃用)	172
改进	173
与 Aurora MySQL 版本 1 进行比较	174
MySQL 5.7 兼容性	174
Aurora MySQL 更新：2019-05-29 (版本 2.04.4) (已弃用)	175
改进	175
与 Aurora MySQL 版本 1 进行比较	176
MySQL 5.7 兼容性	176

Aurora MySQL 更新：2019-05-09 (版本 2.04.3) (已弃用)	177
改进	178
与 Aurora MySQL 版本 1 进行比较	178
MySQL 5.7 兼容性	178
Aurora MySQL 更新：2019-05-02 (版本 2.04.2) (已弃用)	179
改进	180
MySQL 错误修复汇总	180
与 Aurora MySQL 版本 1 进行比较	180
MySQL 5.7 兼容性	181
Aurora MySQL 更新：2019-03-25 (版本 2.04.1) (已弃用)	181
改进	182
与 Aurora MySQL 版本 1 进行比较	182
MySQL 5.7 兼容性	183
Aurora MySQL 更新：2019-03-25 (版本 2.04.0) (已弃用)	183
改进	184
MySQL 错误修复汇总	184
与 Aurora MySQL 版本 1 进行比较	184
MySQL 5.7 兼容性	185
Aurora MySQL 更新：2019-02-07 (版本 2.03.4) (已弃用)	185
改进	186
与 Aurora MySQL 版本 1 进行比较	186
MySQL 5.7 兼容性	187
Aurora MySQL 更新：2019-01-18 (版本 2.03.3) (已弃用)	187
改进	188
MySQL 错误修复汇总	189
与 Aurora MySQL 版本 1 进行比较	189
MySQL 5.7 兼容性	190
Aurora MySQL 更新：2019-01-09 (版本 2.03.2) (已弃用)	190
改进	191
与 Aurora MySQL 版本 1 进行比较	191
MySQL 5.7 兼容性	192
Aurora MySQL 更新：2018-10-24 (版本 2.03.1) (已弃用)	192
改进	193
与 Aurora MySQL 版本 1 进行比较	193
MySQL 5.7 兼容性	194
Aurora MySQL 更新：2018-10-11 (版本 2.03) (已弃用)	194

改进	195
集成了 MySQL 社区版本错误修复	195
与 Aurora MySQL 版本 1 进行比较	195
MySQL 5.7 兼容性	196
Aurora MySQL 更新：2018-10-08 (版本 2.02.5) (已弃用)	197
改进	197
与 Aurora MySQL 版本 1 进行比较	197
MySQL 5.7 兼容性	198
Aurora MySQL 更新：2018-09-21 (版本 2.02.4) (已弃用)	199
改进	199
集成了 MySQL 社区版本错误修复	199
与 Aurora MySQL 版本 1 进行比较	200
MySQL 5.7 兼容性	201
Aurora MySQL 更新：2018-08-23 (版本 2.02.3) (已弃用)	201
与 Aurora MySQL 版本 1 进行比较	202
MySQL 5.7 兼容性	203
Aurora MySQL 2.x 和 Aurora MySQL 1.x 之间的 CLI 差异	203
改进	203
Aurora MySQL 更新：2018-06-04 (版本 2.02.2) (已弃用)	204
改进	204
与 Aurora MySQL 5.6 进行比较	205
MySQL 5.7 兼容性	205
Aurora MySQL 2.x 和 Aurora MySQL 1.x 之间的 CLI 差异	203
改进	204
Aurora MySQL 更新：2018-05-03 (版本 2.02) (已弃用)	206
与 Aurora MySQL 5.6 进行比较	207
MySQL 5.7 兼容性	207
Aurora MySQL 2.x 和 Aurora MySQL 1.x 之间的 CLI 差异	203
改进	208
MySQL 错误修复汇总	209
Aurora MySQL 更新：2018-03-13 (版本 2.01.1) (已弃用)	209
与 Aurora MySQL 5.6 进行比较	209
MySQL 5.7 兼容性	210
Aurora MySQL 2.x 和 Aurora MySQL 1.x 之间的 CLI 差异	211
改进	211
Aurora MySQL 更新：2018-02-06 (版本 2.01) (已弃用)	211

与 Aurora MySQL 5.6 进行比较	212
MySQL 5.7 兼容性	212
Aurora MySQL 2.x 和 Aurora MySQL 1.x 之间的 CLI 差异	203
Aurora MySQL 版本 1 (已弃用)	214
Aurora MySQL 更新：2021-09-30 (版本 1.23.4) (已弃用)	215
改进	216
Aurora MySQL 更新：2021-06-28 (版本 1.23.3) (已弃用)	216
改进	217
Aurora MySQL 更新：2021-03-18 (版本 1.23.2) (已弃用)	217
改进	218
集成了 MySQL 社区版本错误修复	219
Aurora MySQL 更新：2020-11-24 (版本 1.23.1) (已弃用)	219
改进	220
Aurora MySQL 更新：2020-09-02 (版本 1.23.0) (已弃用)	220
改进	221
集成了 MySQL 社区版本错误修复	224
Aurora MySQL 更新：2021-06-03 (版本 1.22.5) (已弃用)	225
改进	226
Aurora MySQL 更新：2021-03-04 (版本 1.22.4) (已弃用)	226
改进	227
Aurora MySQL 更新：2020-11-09 (版本 1.22.3) (已弃用)	227
改进	228
集成了 MySQL 社区版本错误修复	229
Aurora MySQL 更新：2020-03-05 (版本 1.22.2) (已弃用)	229
改进	230
Aurora MySQL 更新：2019-12-23 (版本 1.22.1) (已弃用)	230
改进	231
Aurora MySQL 更新：2019-11-25 (版本 1.22.0) (已弃用)	231
改进	232
集成了 MySQL 社区版本错误修复	235
Aurora MySQL 更新：2019-11-25 (版本 1.21.0) (已弃用)	235
改进	236
集成了 MySQL 社区版本错误修复	237
Aurora MySQL 更新：2020-03-05 (版本 1.20.1) (已弃用)	238
改进	238
Aurora MySQL 更新：2019-11-11 (版本 1.20.0) (已弃用)	239

改进	239
集成了 MySQL 社区版本错误修复	240
Aurora MySQL 更新：2020-03-05 (版本 1.19.6) (已弃用)	241
改进	241
Aurora MySQL 更新：2019-09-19 (版本 1.19.5) (已弃用)	241
改进	242
集成了 MySQL 社区版本错误修复	243
Aurora MySQL 更新：2019-06-05 (版本 1.19.2) (已弃用)	243
改进	244
Aurora MySQL 更新：2019-05-09 (版本 1.19.1) (已弃用)	244
改进	245
Aurora MySQL 更新：2019-02-07 (版本 1.19.0) (已弃用)	245
特征	246
改进	246
集成了 MySQL 社区版本错误修复	247
Aurora MySQL 更新：2018-09-20 (版本 1.18.0) (已弃用)	247
特征	248
Aurora MySQL 更新：2020-03-05 (版本 1.17.9) (已弃用)	248
改进	249
Aurora MySQL 更新：2019-01-17 (版本 1.17.8) (已弃用)	249
改进	250
集成了 MySQL 社区版本错误修复	250
Aurora MySQL 更新：2018-10-08 (版本 1.17.7) (已弃用)	250
改进	251
集成了 MySQL 社区版本错误修复	251
Aurora MySQL 更新：2018-09-06 (版本 1.17.6) (已弃用)	251
改进	252
集成了 MySQL 社区版本错误修复	252
Aurora MySQL 更新：2018-08-14 (版本 1.17.5) (已弃用)	252
改进	253
Aurora MySQL 更新：2018-08-07 (版本 1.17.4) (已弃用)	253
改进	254
Aurora MySQL 更新：2018-06-05 (版本 1.17.3) (已弃用)	254
改进	255
Aurora MySQL 更新：2018-04-27 (版本 1.17.2) (已弃用)	255
改进	256

Aurora MySQL 更新：2018-03-23 (版本 1.17.1) (已弃用)	256
改进	256
Aurora MySQL 更新：2018-03-13 (版本 1.17) (已弃用)	257
零停机时间修补	257
新特征	257
改进	257
MySQL 错误修复汇总	258
Aurora MySQL 更新：2017-12-11 (版本 1.16) (已弃用)	258
零停机时间修补	258
新特征	259
改进	259
MySQL 错误修复汇总	259
Aurora MySQL 更新：2017-11-20 (版本 1.15.1) (已弃用)	259
零停机时间修补	260
改进	260
MySQL 错误修复汇总	260
Aurora MySQL 更新：2017-10-24 (版本 1.15) (已弃用)	260
零停机时间修补	261
新特征	261
改进	261
MySQL 错误修复汇总	260
Aurora MySQL 更新：2018-03-13 (版本 1.14.4) (已弃用)	263
零停机时间修补	263
新特征	263
改进	263
MySQL 错误修复汇总	263
Aurora MySQL 更新：2017-09-22 (版本 1.14.1) (已弃用)	264
改进	264
Aurora MySQL 更新：2017-08-07 (版本 1.14) (已弃用)	264
零停机时间修补	265
改进	265
MySQL 错误修复汇总	266
Aurora MySQL 更新：2017-05-15 (版本 1.13) (已弃用)	266
零停机时间修补	266
新功能	266
改进	267

MySQL 错误修复汇总	267
Aurora MySQL 更新：2017-04-05 (版本 1.12) (已弃用)	268
新特征	268
改进	268
MySQL 错误修复汇总	269
Aurora MySQL 更新：2017-02-23 (版本 1.11) (已弃用)	269
新功能	270
改进	270
MySQL 错误修复汇总	271
Aurora MySQL 更新：2017-01-12 (版本 1.10.1) (已弃用)	272
新功能	272
改进	272
Aurora MySQL 更新：2016-12-14 (版本 1.10) (已弃用)	272
新功能	272
改进	274
MySQL 错误修复汇总	274
Aurora MySQL 更新：2016-11-10 (版本 1.9.0、1.9.1) (已弃用)	274
新功能	274
改进	275
Aurora MySQL 更新：2016-10-26 (版本 1.8.1) (已弃用)	275
改进	275
MySQL 错误修复汇总	276
Aurora MySQL 更新：2016-10-18 (版本 1.8) (已弃用)	276
新功能	276
改进	276
MySQL 错误修复汇总	277
Aurora MySQL 更新：2016-09-20 (版本 1.7.1) (已弃用)	277
改进	277
Aurora MySQL 更新：2016-08-30 (版本 1.7.0) (已弃用)	278
新功能	278
改进	278
MySQL 错误修复汇总	278
Aurora MySQL 更新：2016-06-01 (版本 1.6.5) (已弃用)	279
新功能	279
改进	279
MySQL 错误修复汇总	280

Aurora MySQL 更新：2016-04-06 (版本 1.6) (已弃用)	280
新功能	280
改进	281
MySQL 错误修复汇总	282
Aurora MySQL 更新：2016-01-11 (版本 1.5) (已弃用)	282
改进	282
MySQL 错误修复汇总	283
Aurora MySQL 更新：2015-12-03 (版本 1.4) (已弃用)	283
新功能	283
改进	283
MySQL 错误修复汇总	284
Aurora MySQL 更新：2015-10-16 (版本 1.2、1.3) (已弃用)	284
修复	285
改进	285
MySQL 错误修复汇总	285
Aurora MySQL 更新：2015-08-24 (版本 1.1) (已弃用)	287
由 Aurora MySQL 更新修复的 MySQL 错误	289
由 Aurora MySQL 3.x 更新修复的 MySQL 错误	289
由 Aurora MySQL 2.x 更新修复的 MySQL 错误	299
由 Aurora MySQL 1.x 更新修复的 MySQL 错误	316
Aurora MySQL 中修复的安全漏洞	332
文档历史记录	339
.....	cccli

Amazon Aurora MySQL 兼容版发布说明

Amazon Aurora MySQL 兼容版会定期发布更新。更新将在系统维护时段内应用于 Aurora MySQL 数据库集群。应用更新的时间取决于数据库集群的 AWS 区域 和维护时段设置以及更新的类型。

Amazon Aurora MySQL 版本在多天时间里陆续对所有 AWS 区域发布。某些区域可能会暂时显示另一区域尚不可用的引擎版本。

更新将同时应用于数据库集群中的所有实例。更新需要在数据库集群中的所有实例上重新启动数据库。因此会出现 20 到 30 秒钟的停机，之后，您可以继续使用数据库集群。您可以从 [AWS Management Console](#) 查看或更改维护时段设置。

主题

- [Amazon Aurora MySQL 版本 3 的数据库引擎更新](#)
- [Amazon Aurora MySQL 2 的数据库引擎更新](#)
- [Amazon Aurora MySQL 版本 1 的数据库引擎更新 \(已弃用 \)](#)
- [由 Aurora MySQL 数据库引擎更新修复的 MySQL 错误](#)
- [Aurora MySQL 中修复的安全漏洞](#)

Amazon Aurora MySQL 版本 3 的数据库引擎更新

以下是 Amazon Aurora MySQL 版本 3 的数据库引擎更新。

主题

- [Aurora MySQL 数据库引擎更新 2024-03-07 \(版本 3.06.0, 兼容 MySQL 8.0.34\)](#)
- [Aurora MySQL 数据库引擎更新 2024-01-31 \(版本 3.05.2, 兼容 MySQL 8.0.32\)](#)
- [Aurora MySQL 数据库引擎更新 2023-11-21 \(版本 3.05.1, 兼容 MySQL 8.0.32\)](#)
- [Aurora MySQL 数据库引擎更新 2023-10-30 \(版本 3.05.0.1, 与 MySQL 8.0.32 兼容\) 测试版](#)
- [Aurora MySQL 数据库引擎更新 2023-10-25 \(版本 3.05.0, 与 MySQL 8.0.32 兼容\)](#)
- [Aurora MySQL 数据库引擎更新 2024-03-15 \(版本 3.04.2, 兼容 MySQL 8.0.28\)](#)
- [Aurora MySQL 数据库引擎更新 2023-11-13 \(版本 3.04.1, 兼容 MySQL 8.0.28\) 默认](#)
- [Aurora MySQL 数据库引擎更新 2023-07-31 \(版本 3.04.0, 兼容 MySQL 8.0.28\)](#)
- [Aurora MySQL 数据库引擎更新 2023-12-08 \(版本 3.03.3, 与 MySQL 8.0.26 兼容\)](#)
- [Aurora MySQL 数据库引擎更新 2023-08-29 \(版本 3.03.2, 与 MySQL 8.0.26 兼容\)](#)
- [Aurora MySQL 数据库引擎更新 2023-05-11 \(版本 3.03.1, 与 MySQL 8.0.26 兼容\)](#)
- [Aurora MySQL 数据库引擎更新 2023-03-01 \(版本 3.03.0, 与 MySQL 8.0.26 兼容\) 不支持升级到此版本。](#)
- [Aurora MySQL 数据库引擎更新 2023-04-17 \(版本 3.02.3, 与 MySQL 8.0.23 兼容\) 标准支持将于 2024 年 1 月 15 日结束。](#)
- [Aurora MySQL 数据库引擎更新 2022-11-18 \(版本 3.02.2, 与 MySQL 8.0.23 兼容\) 标准支持将于 2024 年 1 月 15 日结束。](#)
- [Aurora MySQL 数据库引擎更新 2022-09-07 \(版本 3.02.1, 与 MySQL 8.0.23 兼容\) 标准支持将于 2024 年 1 月 15 日结束。不支持升级到此版本。](#)
- [Aurora MySQL 数据库引擎更新 2022-04-20 \(版本 3.02.0, 与 MySQL 8.0.23 兼容\) 标准支持将于 2024 年 1 月 15 日结束。不支持升级到此版本。](#)
- [Aurora MySQL 数据库引擎更新 2022-04-15 \(版本 3.01.1, 与 MySQL 8.0.23 兼容\) 标准支持将于 2024 年 1 月 15 日结束。不支持升级到此版本。](#)
- [Aurora MySQL 数据库引擎更新 2021-11-18 \(版本 3.01.0, 与 MySQL 8.0.23 兼容\) 标准支持将于 2024 年 1 月 15 日结束。不支持升级到此版本。](#)

Aurora MySQL 数据库引擎更新 2024-03-07 (版本 3.06.0 , 兼容 MySQL 8.0.34)

版本 : 3.06.0

Aurora MySQL 3.06.0 已正式上市。Aurora MySQL 3.06 版本与 MySQL 8.0.34 兼容。有关已发生的社群变更的更多信息，请参阅 [MySQL 8.0 Release Notes](#)。

有关 Aurora MySQL 版本 3 中的新功能的详细信息，请参阅[与 MySQL 8.0 兼容的 Aurora MySQL 版本 3](#)。有关 Aurora MySQL 版本 3 和 Aurora MySQL 版本 2 之间的区别，请参阅[比较 Aurora MySQL 版本 2 和 Aurora MySQL 版本 3](#)。有关 Aurora MySQL 版本 3 和 MySQL 8.0 社区版的[比较](#)，请参阅[亚马逊 Aurora 用户指南中的 Aurora MySQL 版本 3 和 MySQL 8.0 社区版的比较](#)。

目前支持的 Aurora MySQL 版本有 2.07.9、2.07.10、2.11.*、2.12.*、3.03.*、3.04.*、3.04.*、3.05.* 和 3.06.*。

您可以使用 A [mazon RDS 蓝/绿部署](#) 执行就地升级、还原快照或启动托管蓝/绿升级，从当前支持的任何 Aurora MySQL 版本 2 集群升级到 Aurora MySQL 3.06.0 版集群。

有关计划升级到 Aurora MySQL 版本 3 的信息，请参阅 A [urora MySQL 版本 3 的升级计划](#)。有关 Aurora MySQL 升级的一般信息，请参阅《Amazon Aurora 用户指南》中的[升级 Amazon Aurora MySQL 数据库集群](#)。

有关疑难解答信息，请参阅 A [mazon Aurora 用户指南中的 Aurora MySQL 版本 3 升级问题疑难解答](#)。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

新功能

- Aurora MySQL 3.06.0 版本支持 Amazon Bedrock 集成，并引入了新的保留关键字 `accept`、`aws_bedrock_invoke_model`、`aws_sagemaker_invoke_endpoint` 和 `content_type_timeout_ms`。升级到版本 3.06.0 之前，请检查对象定义中是否使用了新的保留关键字。要缓解与新的保留关键字的冲突，请引用对象定义中使用的保留关键字。有关 Amazon Bedrock 集成和处理保留关键词的更多信息，请参阅[什么是亚马逊 Bedrock](#)？在《亚马逊 Aurora 用户指南》中。有关更多信息，请参阅 MySQL 文档中的[关键字和保留字](#)、[INFORMATION_SCHEMA 关键字表](#)和[架构对象名称](#)。

- 在为具有多个二级索引的大型表复制事务时，提高了二进制日志副本的性能。此功能引入了一个线程池，用于在二进制日志副本上并行应用二级索引更改。该功能由[aurora_binlog_replication_sec_index_parallel_workers](#)数据库集群参数控制，该参数控制可用于应用二级索引更改的并行线程总数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[优化二进制日志复制](#)。
- 添加了一个新的存储过程[mysql.rds_set_read_only](#)，允许更改 Aurora MySQL 集群中数据库实例[read_only](#)上的全局系统变量的值。有关更多信息，请参阅 Amazon Aurora 用户指南中的[复制](#)。
- 添加了一个新的存储过程[mysql.rds_set_binlog_source_ssl](#)，允许通过为指定值来设置二进制日志副本的加密SOURCE_SSL。有关更多信息，请参阅 Amazon Aurora 用户指南中的[复制](#)。
- [Amazon Aurora Machine Learning](#) 是 Aurora MySQL 数据库和 AWS 机器学习 (ML) 服务之间的优化集成。现在支持 [Amazon Bedrock](#)，允许您使用 SQL 直接从 Aurora MySQL 数据库集群中调用 Amazon Bedrock 中的机器学习模型。有关将 Amazon Bedrock 与 Aurora MySQL 数据库集群配合使用的更多信息，请参阅[亚马逊 Aurora 用户指南中的将亚马逊 Aurora 机器学习与 Aurora MySQL 配合使用](#)。
- Aurora MySQL 3.06 版本增加了对[自动撤消表空间](#)截断的支持。此优化允许您在清除撤消日志后回收撤消表空间中未使用的空间。

改进

修复了安全问题和 CVE：

此版本中包含以下 CVE 修复程序：

- [CVE-2020-11104](#)
- [CVE-2020-11105](#)
- [CVE-2023-38545](#)
- [CVE-2023-38546](#)
- [CVE-2023-39975](#)

可用性改进：

- 修复了当写入器数据库实例的工作负载较高时，只读副本数据库实例无法成功启动的问题。
- 修复了 Aurora MySQL 写入器数据库实例可能由于与 Aurora 存储的通信缺陷而发生故障的问题。该缺陷是由于在 Aurora 存储实例进行软件更新后，数据库实例与底层存储之间的通信中断所致。
- 修复了在 InnoDB 分区表上处理INSERT查询时可能导致实例中可用内存逐渐减少的问题。

- 修复了在运行查询时使用哈希连接时，由于可用内存减少而导致 Aurora MySQL 数据库实例重启或故障转移的问题。
- 修复了同时运行 [SHOW STATUS](#) 和 [PURGE BINARY LOGS](#) 语句时可能导致数据库实例重新启动的问题。PURGE BINARY LOGS 是一个托管语句，用于遵守用户配置的二进制日志保留期。
- 修复了在使用或语句对非虚拟列进行重新排序的表上运行数据操纵语言 (DML) 语句后可能导致服务器意外关闭的问题。MODIFY COLUMN CHANGE COLUMN
- 修复了在数据库实例重启期间可能导致再次重启的问题。
- 修复了在表上定义级联 UPDATE 或 DELETE 外键约束时可能导致数据库重新启动的问题，该问题涉及虚拟列作为外键约束中的列或被引用表的成员。
- 在 Aurora MySQL 2.10 中，我们增加了对重启具有读取可用性的 Aurora 数据库集群的支持。此功能允许读取器数据库实例在写入器数据库实例重启时保持在线状态。Aurora MySQL 全局数据库的辅助数据库现在支持此功能，从而确保 AWS 区域在主集群上重启写入器实例期间，您仍然可以处理读取请求。以前，当写入器实例重新启动时，Aurora MySQL 辅助集群中的所有读取器实例也会重新启动。在此版本中，辅助集群读取器实例在写入器实例重启期间继续处理读取请求，从而提高了集群中的读取可用性。有关更多信息，请参阅[重启具有读取可用性的 Aurora 集群](#)。
- 修复了在运行涉及 AUTO_INCREMENT 列的繁重插入操作时重新启动时可能会中断数据库恢复的问题。

常规改进：

- 修复了从 Aurora 集群卷读取数据时，由于临时网络问题可能导致并行查询失败的问题。
- 修复了用户无法中断任何查询或为查询设置会话超时 performance_schema 的问题。
- 修复了在复制实例进行主机更换时，配置为使用自定义 SSL 证书 ([mysql.rds_import_binlog_ssl_material](#)) 的二进制日志 (binlog) 复制可能会失败的问题。
- 现在，当数据库实例面临内存压力时，内存小于或等于 4 GiB 的小型数据库实例会自动调整缓冲池并关闭占用内存最多的连接。有关更多信息，请参阅[亚马逊 Aurora 用户指南中的亚马逊 Aurora MySQL out-of-memory 问题](#)。
- 将内存超过 4 GiB 的所有数据库实例类的默认响应从空更改为。aurora_oom_response print 有关更多信息，请参阅[亚马逊 Aurora 用户指南中的亚马逊 Aurora MySQL out-of-memory 问题](#)。
- 修复了与审计日志文件管理有关的问题，该问题可能导致无法访问日志文件进行下载或轮换，在某些情况下还会增加 CPU 使用率。
- 优化了 AUTO_INCREMENT 密钥恢复，以缩短还原快照、执行 point-in-time 恢复和克隆数据库中包含大量表的数据数据库集群的完成时间。
- [修复了性能架构等待事件摘要表中未显示 wait/io/redo_log_flush 事件的问题。](#)

- 添加了Aurora_lockmgr_memory_used和Aurora_lockmgr_buffer_pool_memory_used指标以跟踪锁定管理器的内存使用情况。有关更多信息，请参阅[亚马逊 Aurora 用户指南中的 Aurora MySQL 全局状态变量](#)。
- 修复了从低于 2.11.* 的 Aurora MySQL 版本升级后，小型只读副本实例的复制延迟可能会增加的问题。
- 修复了在快照还原、回溯或数据库克隆操作后可能导致使用降序索引的AUTO_INCREMENT列出现重复键错误的问题。
- 修复了以下问题：当 Aurora 读取器实例上至少有一个全文搜索 (FTS) 索引并且正在在 Aurora 写入器数据库实例上运行TRUNCATE语句时，该表上不存在错误表，导致SELECT查询失败。
- 修复了在使用哈希联接算法和 parallel LEFT JOIN | RIGHT JOIN 查询时可能导致结果集不完整的问题。

升级和迁移：

- 修复了如果表架构中存在用户定义的FTS_DOC_ID列，则可能导致主要版本升级失败的问题。
- 修复了在处理 InnoDB 表空间时由于同步问题而可能导致从 Aurora MySQL Version 2 升级到 Aurora MySQL 版本 3 失败的问题。
- 修复了可能导致 Aurora MySQL 版本 3 的主要版本升级失败的问题，因为在 Aurora MySQL 版本 2 中，InnoDB 系统表中存在已删除的表空间的孤立条目。

集成了 MySQL 社区版本错误修复

除以下内容外，此版本还包括8.0.34之前的所有社区错误修复。有关更多信息，请参阅 [Aurora MySQL 3.x 数据库引擎更新和之前的所有 MySQL 错误修复](#)。

- 修复了在基于 Graviton 的实例上可能无法正确计算缓存行值，从而导致数据库重启失败的问题。（社区错误修复 #35479763）
- 修复了存储例程中的某些子查询实例并非总是能正确处理的问题。（社区错误修复 #35377192）
- 修复了由于后台 TLS 证书轮换而可能导致更高 CPU 使用率的问题（社区错误修复 #34284186）。
- 修复了在Aurora MySQL版本低于3.05的版本中，InnoDB允许向MySQL系统架构中的表添加INSTANT列的问题，这可能会导致服务器在升级到Aurora MySQL版本3.05.0后意外关闭（数据库实例重新启动）。（社区错误修复 #35625510）。

Aurora MySQL 数据库引擎更新 2024-01-31 (版本 3.05.2 , 兼容 MySQL 8.0.32)

版本 : 3.05.2

Aurora MySQL 3.05.2 现已正式上市。Aurora MySQL 3.05 版本与 MySQL 8.0.32 兼容。有关已发生的社群变更的更多信息，请参阅 [MySQL 8.0 Release Notes](#)。

有关 Aurora MySQL 版本 3 中的新功能的详细信息，请参阅[与 MySQL 8.0 兼容的 Aurora MySQL 版本 3](#)。有关 Aurora MySQL 版本 3 和 Aurora MySQL 版本 2 之间的区别，请参阅[比较 Aurora MySQL 版本 2 和 Aurora MySQL 版本 3](#)。有关 Aurora MySQL 版本 3 和 MySQL 8.0 社区版的[比较](#)，请参阅[亚马逊 Aurora 用户指南中的 Aurora MySQL 版本 3 和 MySQL 8.0 社区版的比较](#)。

当前支持的 Aurora MySQL 版本有 2.07.9、2.07.10、2.11.*、2.12.*、3.03.*、3.04.* 和 3.05.*。

您可以使用 A [mazon RDS 蓝/绿部署](#) 执行就地升级、还原快照或启动托管蓝/绿升级，从当前支持的任何 Aurora MySQL 版本 2 集群升级到 Aurora MySQL 3.05.2 版集群。

有关计划升级到 Aurora MySQL 版本 3 的信息，请参阅 A [urora MySQL 版本 3 的升级计划](#)。有关 Aurora MySQL 升级的一般信息，请参阅《Amazon Aurora 用户指南》中的[升级 Amazon Aurora MySQL 数据库集群](#)。

有关疑难解答信息，请参阅 A mazon Aurora 用户指南中的 [Aurora MySQL 版本 3 升级问题疑难解答](#)。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

改进

修复了安全问题和 CVE :

此版本中包含以下 CVE 修复程序 :

- [CVE-2020-11104](#)
- [CVE-2020-11105](#)
- [CVE-2023-38545](#)
- [CVE-2023-39975](#)

可用性改进 :

- 修复了在InnoDB分区表上处理INSERT查询会导致实例中可用内存逐渐减少的问题。
- 修复了同时运行 `SHOW STATUS` 和 `PURGE BINARY LOGS` 语句时可能导致数据库实例重新启动的问题。PURGE BINARY LOGS是一条托管语句，运行该语句是为了遵守用户配置的二进制日志保留期而运行的。
- 修复了在使用或语句对非虚拟列进行重新排序的表上运行数据操纵语言 (DML) 语句后可能导致服务器意外关闭的问题。MODIFY COLUMN CHANGE COLUMN
- 修复了在数据库实例重启期间可能导致再次重启的问题。

常规改进：

- 修复了用户无法中断任何查询或为查询设置会话超时performance_schema的问题。
- 修复了在复制实例进行主机更换时，使用自定义 SSL 证书 (`mysql.rds_import_binlog_ssl_material`) 的二进制日志 (binlog) 复制设置可能会失败的问题。
- 修复了与审核日志文件管理有关的问题，该问题可能导致无法访问日志文件进行下载或轮换，在某些情况下还会增加 CPU 使用率。

升级和迁移：

- 修复了当表架构中存在用户定义的FTS_DOC_ID列时，可能导致从 Aurora MySQL 版本 2 升级到 Aurora MySQL 版本 3 失败的问题。
- 修复了在处理 InnoDB 表空间时由于同步问题而可能导致从 Aurora MySQL 版本 2 升级到 Aurora MySQL 版本 3 失败的问题。
- 修复了可能导致 Aurora MySQL 版本 3 的主要版本升级失败的问题，因为在 Aurora MySQL 版本 2 中，InnoDB 系统表中存在已删除的表空间的孤立条目。

集成了 MySQL 社区版本错误修复

此版本包括 8.0.32 之前的所有社区错误修复。有关更多信息，请参阅 [Aurora MySQL 3.x 数据库引擎更新和之前的所有 MySQL 错误修复](#)。

Aurora MySQL 数据库引擎更新 2023-11-21 (版本 3.05.1 , 兼容 MySQL 8.0.32)

版本：3.05.1

Aurora MySQL 3.05.1 现已正式上市。Aurora MySQL 3.05 版本与 MySQL 8.0.32 兼容。有关更多信息，请参阅 [MySQL 8.0 版本说明](#)。

当前支持的 Aurora MySQL 版本有 2.07.*、2.11.*、2.12.*、3.01.*、3.02.*、3.03.*、3.04.* 和 3.05.*。

你可以将现有的 Aurora MySQL 3.* 数据库集群升级到 Aurora MySQL 3.05.1。你也可以将当前支持的任何 Aurora MySQL 版本中的快照还原到 Aurora MySQL 3.05.1。

如果您将 Aurora MySQL Global Database 升级到版本 3.05.*，则必须将主数据库集群和辅助数据库集群升级到完全相同的版本（包括补丁级别）。有关升级 Aurora Global Database 次要版本的更多信息，请参阅 [次要版本升级](#)。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得 [AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

改进

修复了下面列出的安全问题和 CVE：

此版本包括 MySQL 8.0.32 及之前的所有社区 CVE 修复。

- [CVE-2023-38545](#)

集成了 MySQL 社区版本错误修复

除以下内容外，此版本还包括 8.0.32 及之前的所有社区错误修复。有关更多信息，请参阅 [Aurora MySQL 3.x 数据库引擎更新和之前的所有 MySQL 错误修复](#)。

- 修复了 InnoDB 中的一个问题，即在 Aurora MySQL 版本 3.01 至 Aurora MySQL 版本 3.04 之间，如果系统架构中的 MySQL 表添加了 INSTANT ADD 列，在 Aurora MySQL 升级到 3.05.0 版本之后，这些表上的 DML 将导致服务器意外关闭。（社区错误修复 35625510）

Aurora MySQL 数据库引擎更新 2023-10-30（版本 3.05.0.1，与 MySQL 8.0.32 兼容）测试版

版本：3.05.0.1

Aurora MySQL 3.05.0.1 现已在以下地区推出：美国东部（弗吉尼亚北部）、美国东部（俄亥俄州）、美国西部（加利福尼亚北部）、美国西部（俄勒冈）、AWS GovCloud（美国东部）和 AWS

GovCloud（美国西部）。这是一个仅限安全修复的早期版本。随着下一个补丁版本 3.05.1 的发布，这些修复将在所有地区更广泛地部署。Aurora MySQL 3.05 版本与 MySQL 8.0.32 兼容。有关已发生的社群变更的更多信息，请参阅 [MySQL 8.0 Release Notes](#)。

当前支持的 Aurora MySQL 版本有 2.07.*、2.11.*、2.12.*、3.01.*、3.02.*、3.03.*、3.04.* 和 3.05.*。

您可以将现有的 Aurora MySQL 3.* 数据库集群升级到 Aurora MySQL 3.05.0.1。您也可以将快照从当前支持的任何 Aurora MySQL 版本还原到 Aurora MySQL 3.05.0.1。

如果您将 Aurora MySQL Global Database 升级到版本 3.05.*，则必须将主数据库集群和辅助数据库集群升级到完全相同的版本（包括补丁级别）。有关升级 Aurora Global Database 次要版本的更多信息，请参阅 [次要版本升级](#)。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得 [AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

改进

修复了下面列出的安全问题和 CVE：

此版本包括 MySQL 8.0.32 及之前的所有社区 CVE 修复。

- [CVE-2023-38545](#)

Aurora MySQL 数据库引擎更新 2023-10-25（版本 3.05.0，与 MySQL 8.0.32 兼容）

版本：3.05.0

Aurora MySQL 3.05.0 已正式发布。Aurora MySQL 3.05 版本与 MySQL 8.0.32 兼容。有关已发生的社群变更的更多信息，请参阅 [MySQL 8.0 Release Notes](#)。

有关 Aurora MySQL 版本 3 中的新功能的详细信息，请参阅 [与 MySQL 8.0 兼容的 Aurora MySQL 版本 3](#)。有关 Aurora MySQL 版本 3 和 Aurora MySQL 版本 2 之间的区别，请参阅 [比较 Aurora MySQL 版本 2 和 Aurora MySQL 版本 3](#)。有关 Aurora MySQL 版本 3 和 MySQL 8.0 社群版的比较，请参阅 [比较 Aurora MySQL 版本 3 和 MySQL 8.0 社群版](#)。

当前支持的 Aurora MySQL 版本有 2.07.9、2.07.10、2.11.*、2.12.*、3.03.*、3.04.* 和 3.05.*。

您可以执行就地升级、还原快照或使用 [Amazon RDS 蓝绿部署](#) 启动托管蓝绿升级，从当前支持的任何 Aurora MySQL 版本 2 集群升级到 Aurora MySQL 版本 3.05.0 集群。

有关计划升级到 Aurora MySQL 版本 3 的信息，请参阅《Amazon Aurora 用户指南》中的 [Aurora MySQL 版本 3 的升级计划](#)。有关 Aurora MySQL 升级的一般信息，请参阅《Amazon Aurora 用户指南》中的 [升级 Amazon Aurora MySQL 数据库集群](#)。

有关问题排查信息，请参阅 [排查 Aurora MySQL 版本 3 升级问题](#)。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得 [AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

改进

新功能：

- 增加了对将数据从 Aurora MySQL 数据库集群保存到 Amazon S3 存储桶中使用 KMS 密钥加密 (SSE-KMS) 的文本文件的支持。有关更多信息，请参阅 [将数据从 Amazon Aurora MySQL 数据库集群保存到 Amazon S3 存储桶中的文本文件](#)。
- 引入了新的全局状态变量 `aurora_tmz_version` 来表示引擎使用的时区 (TZ) 信息的当前版本。这些值遵循 IANA 时区数据库版本，格式为“YYYYsuffix”，例如 2022a 和 2023c。有关更多信息，请参阅 [Aurora MySQL 全局状态变量](#)。

修复了下面列出的安全问题和 CVE：

对托管环境中的处理进行微调的修复和其他增强功能。其他 CVE 修复如下：

- [CVE-2022-37434](#)

可用性改进：

- 修复了使用并行查询的 Aurora MySQL 数据库实例在运行大量并发并行查询时可能会遇到数据库重启的问题。
- 修复了由审计日志记录线程导致的锁争用问题，该问题最终会导致 CPU 使用率过高和客户端应用程序超时。
- 修复了当任何二进制日志源将 `gtid_mode` 设置为 ON 或 ON_PERMISSIVE 时，在启用了增强型二进制日志的二进制日志 (binlog) 副本集群上可能导致已执行的 GTID 集无法正确恢复的问题。此问题可能会导致副本集群的写入器实例在恢复期间多重启一次，或者在查询已执行的 GTID 集时导致结果不正确。
- 修复了启用增强型二进制日志后，由于可用内存减少而可能导致 Aurora MySQL 数据库实例重启或失效转移的内存管理问题。

- 修复了在尝试读取属于已删除表的数据库页面时可能导致数据库实例重启的问题。
- 修复了当写入器实例将数据库卷增大为 160 GB 的倍数时，可能会导致读取器实例重启的问题。
- 修复了启用了增强型二进制日志功能的 Aurora MySQL 数据库实例在执行二进制日志恢复过程时可能在数据库实例启动期间卡住的问题。
- 修复了在初始化大型回滚段时，Aurora MySQL 数据库实例在实例启动期间可能会经历多次重启的问题。
- 修复了零停机时间修补期间的一个问题，该问题会导致实例重启，从而导致数据库连接意外关闭。
- 修复了在同时运行 [SHOW STATUS](#) 和 [PURGE BINARY LOGS](#) 语句时，由于死锁而可能导致数据库实例重启的问题。PURGE BINARY LOGS 是一种托管语句，执行该语句是为了遵守用户配置的二进制日志保留期。
- 修复了当数据库在内部系统表上创建或删除触发器时，如果写入器实例重启，则可能导致数据库集群不可用的问题。
- 修复了在具有 Aurora 副本的集群上使用增强型二进制日志功能时，由于长时间等待信号灯而可能导致数据库实例重启的问题。
- 修复了在执行引用聚合函数的查询时可能导致数据库实例重启的问题。
- 修复了在极少数情况下，Aurora Serverless v2 错误地尝试在扩展时更新表缓存，而可能导致数据库实例重启的问题。
- 修复了在实现中间临时表时考虑公用表表达式 (CTE) 不支持的索引扫描访问方法的问题，这可能会导致不良行为，包括数据库重启或查询结果不正确。我们通过避免使用 TempTable 存储引擎对表使用此类不支持的索引扫描访问方法来修复此问题。

常规改进：

- 修复了在 Aurora MySQL 3.04.0 上运行的 Aurora Serverless v2 数据库集群上启用增强型二进制日志时，可能导致数据库不可用的问题。
- 现在，在启用了增强型二进制日志功能时，在写入 Aurora 存储之前会删除未使用的存储元数据。这样可以避免在某些情况下，由于通过网络传输的字节数增加而导致写入延迟增加，从而进一步导致数据库重启或失效转移。
- 在 performance_schema 中添加 malloc_stats 和 malloc_stats_totals 表后，添加了三个高级系统变量来控制内部内存分配器 Jemalloc 的行为：
 - aurora_jemalloc_background_thread.
 - aurora_jemalloc_dirty_decay_ms.
 - aurora_jemalloc_tcache_enabled.

- 修复了升级或迁移时未创建 Aurora 特定性能架构表的问题。
- 添加了一个新的系统变量 `aurora_use_vector_instructions`。启用此参数后，Aurora MySQL 将使用优化的向量处理指令来提高 I/O 繁重的工作负载的性能。原定设置情况下，该设置在 Aurora MySQL 3.05 及更高版本中设置为 `ON`。有关更多详细信息，请参阅 [Aurora MySQL 配置参数](#)。
- 修复了启用增强型二进制日志时可能导致开启的 `NumBinaryLogFiles` 指标显示错误结果的问题。
CloudWatch
- 向 Amazon SageMaker 发出 [Aurora MySQL Machine Learning](#) 操作的请求超时已从 3 秒延长到 30 秒。这有助于解决这样一个问题，即在使用较大的批处理大小时，客户可能会看到从 Aurora MySQL Machine Learning 向 Amazon SageMaker 发出的请求的重试次数或失败次数增加。
- 在 `performance_schema` 数据库中添加了对 `malloc_stats` 和 `malloc_stats_totals` 表的支持。
- 将 `LOAD DATA FROM S3` 命令中的 `FROM` 关键字更新为可选关键字。有关更多信息，请参阅 [将数据从 Amazon S3 存储桶中的文本文件加载到 Amazon Aurora MySQL 数据库集群](#)。
- 增加了对参数 `innodb_aurora_instant_alter_column_allowed` 的支持，该参数控制 `INSTANT` 算法是否可用于 `ALTER COLUMN` 运算。有关更多信息，请参阅 [集群级别的参数](#)。
- 修复了在启用了写入转发时，可能无法与数据库建立新的客户端连接的问题。
- 修复了可能导致 `table_open_cache` 数据库参数的修改在数据库实例重启后才生效的问题。
- 修复了在快照还原、回溯或数据库克隆操作后，可能导致使用降序索引的 `AUTO_INCREMENT` 列出现重复键错误的问题。
- 修复了一个涉及索引扫描的问题，即在执行 `SELECT` 查询时，如果使用 `GROUP BY` 子句且 `aurora_parallel_query` 参数设置为 `ON`，则可能会返回不准确的结果。
- 修复了在对 `INFORMATION_SCHEMA INNODB_TABLESPACES` 表执行查询时可能导致可用内存耗尽的问题。
- 修复了读取器实例无法打开表并显示 `ERROR 1146` 的问题。在写入器实例上使用 `INPLACE` 算法并执行某些类型的在线数据定义语言 (DDL) 时，会出现此问题。
- 修复了一个问题，在内部监控流程错误地提交重复的 Aurora Serverless v2 扩展请求时，可避免在扩展期间重启实例。
- 修复了当连接的二进制日志 (binlog) 使用者使用重复的二进制日志复制服务器 ID 时，可能导致数据库重启的问题。
- 为 Aurora MySQL 管理的二进制日志副本引入了内存 [中继日志](#) 缓存。这一改进有助于将二进制日志复制吞吐量提高达 40%。使用单线程二进制日志复制时，或者在启用 [GTID 自动定位](#) 的情况下使用多线程复制时，会自动启用此增强功能。

升级和迁移：

- 从 MySQL 5.7 升级到 MySQL 8.0 时，如果单个数据库中有大量表，则会导致服务器消耗过多内存。经调查发现，我们在检查表是否可以升级的过程中，预先获取了所有数据字典 Table 对象，处理了每个对象并获取其名称，然后对列表执行了[检查版本兼容性](#)。在这种情况下，无需事先获取所有对象，这会极大增加内存消耗。为了更正这个问题，在这种情况下，我们现在一次获取一个 Table 对象，执行所有必需的检查，获取其名称并释放该对象，然后再继续处理下一个对象。（错误 #34526001）
- 通过使用数据库实例上所有可用的 vCPU 并行执行表空间检查，提高了从 Aurora MySQL 版本 2 到版本 3 的主要版本升级的性能。

集成了 MySQL 社区版本错误修复

除以下内容外，此版本还包括 8.0.32 及之前的所有社区错误修复。有关更多信息，请参阅 [Aurora MySQL 3.x 数据库引擎更新和之前的所有 MySQL 错误修复](#)。

- 修复了由于后台 TLS 证书轮换可能导致更高 CPU 使用率的问题。（社区错误修复 #34284186）

Aurora MySQL 数据库引擎更新 2024-03-15 (版本 3.04.2 , 兼容 MySQL 8.0.28)

版本：3.04.2

Aurora MySQL 3.04.2 现已正式上市。Aurora MySQL 3.04 版本与 MySQL 8.0.28 兼容。有关已发生的社区变更的更多信息，请参阅 [MySQL 8.0 发行说明](#)。

有关 Aurora MySQL 版本 3 中的新功能的详细信息，请参阅[与 MySQL 8.0 兼容的 Aurora MySQL 版本 3](#)。有关 Aurora MySQL 版本 3 和 Aurora MySQL 版本 2 之间的区别，请参阅[比较 Aurora MySQL 版本 2 和 Aurora MySQL 版本 3](#)。有关 Aurora MySQL 版本 3 和 MySQL 8.0 社群版的比较，请参阅[比较 Aurora MySQL 版本 3 和 MySQL 8.0 社群版](#)。

Note

此版本被指定为长期支持 (LTS) 版本。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [Aurora MySQL 长期支持 \(LTS\) 版本](#)。

目前支持的 Aurora MySQL 版本有 2.07.9、2.7.10、2.11.*、2.12.*、3.03.*、3.04.*、3.04.*、3.05.* 和 3.06.*。

您可以使用 [Amazon RDS Blue/Green 部署](#) 从任何当前可用的 [Aurora MySQL 版本 2 集群](#) 执行就地升级、还原快照或启动托管蓝/绿升级，从 Aurora MySQL 版本 2 集群升级到 Aurora MySQL 版本 3.04.2 集群。

有关计划升级到 Aurora MySQL 版本 3 的信息，请参阅《Amazon Aurora 用户指南》中的 [Aurora MySQL 版本 3 的升级计划](#)。有关 Aurora MySQL 升级的一般信息，请参阅《Amazon Aurora 用户指南》中的 [升级 Amazon Aurora MySQL 数据库集群](#)。

有关问题排查信息，请参阅 [排查 Aurora MySQL 版本 3 升级问题](#)。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得 [AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

改进

修复了安全问题和 CVE：

此版本中包含以下 CVE 修复程序：

- [CVE-2020-11104](#)
- [CVE-2020-11105](#)
- [CVE-2023-38545](#)
- [CVE-2023-38546](#)
- [CVE-2023-39975](#)

可用性改进：

- 修复了当写入器数据库实例的工作负载较高时，只读副本数据库实例无法成功启动的问题。
- 修复了由于与 Aurora 存储通信的组件存在缺陷而导致 Aurora MySQL 写入器数据库实例可能发生故障的问题。该缺陷是由于软件更新后数据库实例与底层存储之间的通信中断所致。
- 修复了同时运行 [SHOW STATUS](#) 和 [PURGE BINARY LOGS](#) 语句时可能导致数据库实例重新启动的问题。PURGE BINARY LOGS 是一个托管语句，用于遵守用户配置的二进制日志保留期。
- 修复了在数据库实例重启期间可能导致再次重启的问题。
- 修复了由审计日志记录线程导致的锁争用问题，该问题会导致 CPU 占用率过高和客户端应用程序超时。

- 修复了在初始化大型回滚段时，Aurora MySQL 数据库实例在实例启动期间可能会经历多次重启的问题。
- 修复了在运行引用聚合函数的查询时可能导致数据库实例重新启动的问题。

常规改进：

- 修复了从 Aurora 数据库集群卷读取数据时可能由于临时网络问题导致并行查询失败的问题
- 修复了用户无法中断任何查询或为查询设置会话超时 `performance_schema` 的问题。
- 修复了在复制实例进行主机更换时，配置为使用自定义 SSL 证书 (`mysql.rds_import_import_binlog_ssl_material`) 的二进制日志 (`binlog`) 复制可能会失败的问题。
- 修复了与审计日志文件管理有关的问题，该问题可能导致无法访问日志文件进行下载或轮换，在某些情况下还会增加 CPU 使用率。
- 优化了 `AUTO_INCREMENT` 密钥恢复，以缩短还原快照、执行 `point-in-time` 恢复和克隆数据库中包含大量表的数据数据库集群的完成时间。
- 修复了从社区 MySQL 迁移到 Aurora MySQL 版本 3.04.0 和 3.04.1 后，由于这些表丢失而引用某些 `performance_schema` 表的 SQL 语句可能会返回错误的问题。
- 修复了从低于 2.11.* 的 Aurora MySQL 版本升级后，小型只读副本实例的复制延迟可能会增加的问题。
- 修复了在快照还原、回溯或数据库克隆操作后可能导致使用降序索引的 `AUTO_INCREMENT` 列出现重复键错误的问题。
- 修复了可能导致 `table_open_cache` 数据库参数的修改在数据库实例重新启动后才生效的问题。
- 修复了读取器数据库实例无法打开表的问题，错误为 1146。在写入器数据库实例上使用 `INPLACE` 算法时运行某些类型的在线数据定义语言 (DDL) 语句时，会出现此问题。
- 修复了一个问题，在内部监控流程错误地提交重复的 Aurora Serverless v2 扩展请求时，可避免在扩展期间重启实例。
- 修复了连接的二进制日志 (binlog) 使用者使用重复的二进制日志复制服务器 ID 时可能导致数据库重启的问题。

升级和迁移：

- 修复了可能导致 Aurora MySQL 版本 3 的主要版本升级失败的问题，因为在 Aurora MySQL 版本 2 中，InnoDB 系统表中存在已删除的表空间的孤立条目。

集成了 MySQL 社区版本错误修复

除以下内容外，此版本还包括8.0.28之前的所有社区错误修复。有关更多信息，请参阅 [Aurora MySQL 3.x 数据库引擎更新和之前的所有 MySQL 错误修复](#)。

- 修复了在基于 Graviton 的实例上可能无法正确计算缓存行值，从而导致数据库重启失败的问题。（社区错误修复 #35479763）
- 重复运行存储的例程，将包含多个、或XOR条件的 SELECT 语句作为子查询 ANDOR，会导致过度消耗虚拟内存，并可能最终耗尽虚拟内存。（社区错误修复 #33852530）

Aurora MySQL 数据库引擎更新 2023-11-13 (版本 3.04.1 , 兼容 MySQL 8.0.28) 默认

版本 : 3.04.1

Aurora MySQL 3.04.1 现已正式上市。Aurora MySQL 3.04 版本与 MySQL 8.0.28 兼容。有关已发生的社区变更的更多信息，请参阅 [MySQL 8.0 发行说明](#)。

Note

此版本被指定为长期支持 (LTS) 版本。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [Aurora MySQL 长期支持 \(LTS \) 版本](#)。

有关 Aurora MySQL 版本 3 中的新功能的详细信息，请参阅[与 MySQL 8.0 兼容的 Aurora MySQL 版本 3](#)。有关 Aurora MySQL 版本 3 和 Aurora MySQL 版本 2 之间的区别，请参阅[比较 Aurora MySQL 版本 2 和 Aurora MySQL 版本 3](#)。有关 Aurora MySQL 版本 3 和 MySQL 8.0 社群版的比较，请参阅[比较 Aurora MySQL 版本 3 和 MySQL 8.0 社群版](#)。

目前支持的 Aurora MySQL 版本有

2.07.9、2.7.10、2.11.*、2.12.*、3.01.*、3.02.*、3.03.*、3.03.*、3.04.*、3.04.* 和 3.05.*。

您可以使用 A [mazon RDS Blue/Green 部署从任何当前可用的 Aurora MySQL 版本 2 集群执行就地升级、还原快照或启动托管蓝/绿升级](#)，从 Aurora MySQL 版本 2 集群升级到 Aurora MySQL 版本 3.04.1 集群。

有关计划升级到 Aurora MySQL 版本 3 的信息，请参阅《Amazon Aurora 用户指南》中的 [Aurora MySQL 版本 3 的升级计划](#)。有关 Aurora MySQL 升级的一般信息，请参阅《Amazon Aurora 用户指南》中的 [升级 Amazon Aurora MySQL 数据库集群](#)。

有关问题排查信息，请参阅[排查 Aurora MySQL 版本 3 升级问题](#)。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

改进

可用性改进：

- 修复了使用并行查询的 Aurora MySQL 数据库实例在运行大量并发并行查询时可能会遇到数据库重启的问题。
- 修复了当任何二进制日志源设置为或时，在启用增强型二进制日志的二进制日志 (binlog) 副本集群上可能导致已执行的 GTID 集无法正确恢复的问题。gtid_mode ON ON_PERMISSIVE此问题可能会导致副本集群的写入器实例在恢复期间多重启一次，或者在查询已执行的 GTID 集时导致结果不正确。
- 修复了启用增强型二进制日志后，由于可用内存减少而可能导致 Aurora MySQL 数据库实例重启或失效转移的内存管理问题。
- 修复了当写入器实例将数据库卷增大为 160 GB 的倍数时，可能会导致读取器实例重启的问题。
- 修复了启用了增强型二进制日志功能的 Aurora MySQL 数据库实例在执行二进制日志恢复过程时可能在数据库实例启动期间卡住的问题。
- 修复了在同时运行 [SHOW STATUS](#) 和 [PURGE BINARY LOGS](#) 语句时，由于死锁而可能导致数据库实例重启的问题。PURGE BINARY LOGS 是一种托管语句，执行该语句是为了遵守用户配置的二进制日志保留期。
- 修复了当数据库在内部系统表上创建或删除触发器时，如果写入器实例重启，则可能导致数据库集群不可用的问题。
- 修复了在具有 Aurora 副本的集群上使用增强型二进制日志功能时，由于长时间等待信号灯而可能导致数据库实例重启的问题。

常规改进：

- 修复了在 Aurora MySQL 3.04.0 上运行的 Aurora Serverless v2 数据库集群上启用增强型二进制日志时，可能导致数据库不可用的问题。

- 启用增强型二进制日志功能后，在写入 Aurora 存储之前删除了未使用的存储元数据。这样可以避免在某些情况下，由于通过网络传输的字节数增加而导致写入延迟增加，从而进一步导致数据库重启或失效转移。
- 修复了升级或迁移时未创建 Aurora 特定性能架构表的问题。
- 修复了启用增强型二进制日志时可能导致开启的NumBinaryLogFiles指标显示错误结果的问题。
CloudWatch

升级和迁移：

- 从 MySQL 5.7 升级到 MySQL 8.0 时，如果单个数据库中有大量表，则会导致服务器消耗过多内存。研究发现，在检查表是否可以升级的过程中，我们预先获取了所有数据字典Table对象，处理了每个对象并获取其名称，然后在列表[CHECK TABLE ... FOR UPGRADE](#)上执行。在这种情况下，无需事先获取所有对象，这会极大增加内存消耗。为了更正这个问题，在这种情况下，我们现在一次获取一个 Table 对象，执行所有必需的检查，获取其名称并释放该对象，然后再继续处理下一个对象。（错误 #34526001）

集成了 MySQL 社区版本错误修复

除以下内容外，此版本还包括 8.0.28 及之前的所有社区错误修复。有关更多信息，请参阅 [Aurora MySQL 3.x 数据库引擎更新和之前的所有 MySQL 错误修复](#)。

- 修复了因后台 TLS 证书轮换而可能导致更高 CPU 使用率的问题（社区错误修复 #34284186）

Aurora MySQL 数据库引擎更新 2023-07-31（版本 3.04.0，兼容 MySQL 8.0.28）

版本：3.04.0

Aurora MySQL 3.04.0 已正式发布。Aurora MySQL 3.04 版与 MySQL 8.0.28 兼容，Aurora MySQL 3.03 版与 MySQL 8.0.26 兼容，Aurora MySQL 3.02 版与 MySQL 8.0.23 兼容。有关从 8.0.23 到 8.0.28 发生的社群变更的更多信息，请参阅 [MySQL 8.0 Release Notes](#)。

Note

此版本被指定为长期支持 (LTS) 版本。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [Aurora MySQL 长期支持 \(LTS\) 版本](#)。

有关 Aurora MySQL 版本 3 中的新功能的详细信息，请参阅[与 MySQL 8.0 兼容的 Aurora MySQL 版本 3](#)。有关 Aurora MySQL 版本 3 和 Aurora MySQL 版本 2 之间的区别，请参阅[比较 Aurora MySQL 版本 2 和 Aurora MySQL 版本 3](#)。有关 Aurora MySQL 版本 3 和 MySQL 8.0 社群版的比较，请参阅[比较 Aurora MySQL 版本 3 和 MySQL 8.0 社群版](#)。

当前支持的 Aurora MySQL 版本有 2.07.9、2.11.1、2.11.2、3.01.*、3.02.*、3.03.* 和 3.04.0。

您可以执行就地升级、还原快照或使用 [Amazon RDS 蓝绿部署](#) 启动托管蓝绿升级，从当前支持的任何 Aurora MySQL 版本 2 集群升级到 Aurora MySQL 版本 3.04.0 集群。

有关计划升级到 Aurora MySQL 版本 3 的信息，请参阅《Amazon Aurora 用户指南》中的 [Aurora MySQL 版本 3 的升级计划](#)。有关 Aurora MySQL 升级的一般信息，请参阅《Amazon Aurora 用户指南》中的 [升级 Amazon Aurora MySQL 数据库集群](#)。

有关问题排查信息，请参阅[排查 Aurora MySQL 版本 3 升级问题](#)。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

Aurora MySQL 版本 3.04.0 上的 Aurora Serverless v2 数据库实例目前不支持 Aurora MySQL 增强型二进制日志 (binlog)。启用此功能可能会导致数据库不可用。如果您需要在 Aurora MySQL 版本 3.04.0 上使用增强型二进制日志，建议使用[非无服务器数据库实例类](#)或将 Serverless v2 数据库实例的最小和最大 ACU 设置为相同的值。

有关 Aurora MySQL 中增强型二进制日志记录的更多信息，请参阅 [Aurora 用户指南](#)。

改进

新功能：

- 提高了使用 InnoDB 全文索引在[自然语言模式](#)下搜索短语的查询性能。有关在 MySQL 中进行全文搜索的更多信息，请参阅 [Full-Text Search Functions](#)。
- Amazon Aurora MySQL 支持本地（集群内）写入转发。现在，您可以将写入操作从读取器数据库实例转发到 Aurora MySQL 数据库集群中的写入器数据库实例。有关更多信息，请参阅[在 Aurora MySQL 数据库集群中使用本地写入转发](#)。

- 添加了在禁用了 autocommit 的会话中更改在 [Amazon Aurora Global Database 中使用写入转发功能](#)的 aurora_replica_read_consistency 参数值的功能。有关更多信息，请参阅[写入转发的配置参数](#)。
- 从 Aurora MySQL 3.04 开始，对于[全局数据库写入转发](#)功能，您现在可以通过数据库集群和数据库实例参数组设置 aurora_replica_read_consistency 参数值。在 Aurora MySQL 版本 3.04 之前，只能在会话级别配置此参数的值。

修复了安全问题和 CVE：

- 将 SSL/TLS 提供程序从 OpenSSL 更改为 [AWS-LC](#)。这带来了许多变化，包括但不限于以下内容：
 - 从 Aurora MySQL 版本 3.04.0 升级到更高版本时，现在可以通过零停机重启和零停机修补来恢复使用 SSL 的数据库连接。
 - 支持 TLSv1.3，包括支持 TLS_AES_128_GCM_SHA256、TLS_AES_256_GCM_SHA384 和 TLS_CHACHA20_POLY1305_SHA256 SSL 密码。
 - 取消对不太安全的 DHE-RSA-* 密码的支持。

有关更多信息，请参阅[将 TLS 与 Aurora MySQL 数据库集群结合使用](#)。

- 向 rds_superuser_role 中添加了动态权限 SHOW_ROUTINE，允许访问所有存储例程的定义和属性，例如存储过程和函数。有关更多详细信息，请参阅 [SHOW_ROUTINE](#)。
- 修复了在审计日志文件轮换期间可能导致审计日志错过事件的问题。
- 启用了安全且高性能的传输层安全性协议 (TLS) 1.3 版的支持，同时保持与 TLS 1.2 版的兼容性。
- TLS 版本 TLSv1 和 TLSv1.1 在 MySQL 8.0.26 社群版中已弃用，在 Aurora MySQL 3.03 中也已相应地弃用。这些协议现已从 MySQL 8.0.28 社群版中删除，也已相应地从 Aurora MySQL 3.04 中删除。默认情况下，任何无法通过 TLS 1.2 或更高版本进行通信的安全客户端都将被拒绝。有关使用 TLS 连接数据库实例的更多信息，请参阅[使用 Amazon Aurora MySQL 实现高安全性](#)。

此版本中包含以下 CVE 修复程序：

- [CVE-2023-21963](#)
- [CVE-2023-21912](#)
- [CVE-2023-0215](#)
- [CVE-2022-43551](#)
- [CVE-2022-37434](#)

- [CVE-2022-21635](#)
- [CVE-2022-21556](#)
- [CVE-2022-21352](#)
- [CVE-2021-35630](#)
- [CVE-2021-35624](#)

可用性改进：

- 修复了在长时间事务恢复期间可能导致数据库重新启动的问题。
- 修复了数据库活动流事件加密中可能导致数据库重新启动的问题。
- 修复了在启动期间或在 Aurora Serverless v2 中扩展期间，因初始化 InnoDB 缓冲池时出现内存不足错误而导致的内存管理问题。此问题可能导致数据库实例重启或性能下降，包括吞吐量降低或延迟增加。
- 修复了在执行使用 Aurora MySQL 并行查询执行计划的查询时可能导致 Aurora MySQL 阅读器实例重启的问题。
- 修复了在某些情况下可能导致 Aurora 阅读器实例在测距期间重启的问题。
- 修复了在执行涉及自动增量列的繁重插入操作时重新启动时可能会中断数据库恢复的问题。
- 修复了 Aurora 高级审计中的以下问题：当服务器变量 `server_audit_events` 设置为 ALL 或 QUERY 时，会导致将过多信息性消息记录到 Aurora MySQL 错误日志中。此问题可能会导致数据库实例重新启动。
- 修复了启用并行查询后，在回滚 INSERT 语句期间可能导致数据库重启的问题。
- 修复了在 EXTRA 信息列中返回输出的 `all select tables were optimized away` 查询上运行 EXPLAIN ANALYZE 分析工具时可能导致数据库实例重新启动的问题。有关更多信息，请参阅关于 [EXPLAIN 输出格式](#) 的 MySQL 文档。
- 修复了在转发的 [隐式提交语句](#) 遇到错误时，可能导致使用全局写入转发的 Aurora 全局数据库辅助区域读取器实例重新启动的问题。
- 修复了使用来自 Aurora 全球数据库辅助区域的全局写入转发执行 SELECT FOR UPDATE 查询时，可能导致 Aurora 全球数据库主区域中的写入器实例重新启动的问题。

常规改进：

- 添加了新的存储过程 `mysql.rds_gtid_purged`，允许客户设置 `GTID_PURGED` 系统变量。有关更多信息，请参阅 [mysql.rds_gtid_purged](#)。

- 增加了两个新的存储过程 `mysql.rds_start_replication_until` 和 `mysql.rds_start_replication_until_gtid`，允许客户配置停止二进制日志复制的位置。有关在 Aurora MySQL 中为二进制日志复制配置停止位置的更多信息，请参阅 [mysql.rds_start_replication_until](#)。
- 修复了从禁用了自动提交模式的会话调用时，[Aurora MySQL 复制控制存储过程](#)无法修改 `sql_log_bin` 变量的问题。
- 增加了对以下数据控制语言 (DCL) 语句的逻辑复制支持：GRANT/REVOKE 和 CREATE/DROP/ALTER/RENAME USER。
- 修复了阻止 InnoDB 统计信息过时的的问题，这有时会生成次优的查询执行计划，从而可能导致查询执行时间延长。
- 增加了两个新的系统视图，`information_schema.aurora_global_db_instance_status` 和 `information_schema.aurora_global_db_status`。这些视图可用于显示 Aurora MySQL Global Database 集群中主要资源和次要资源的状态和拓扑。有关这两个系统视图的详细信息，请参阅 [Aurora MySQL 特定的 information_schema 表](#)。
- 修复了用户在执行带有转义通配符的 SET ROLE 语句后，无法访问名称中包含通配符的数据库的问题。
- 修复了在处理审计日志轮换时报告的事件可能未写入审计日志的问题。
- 修复了通过 TRIGGER 执行创建内部临时表可能导致写入器数据库实例重启的问题。
- 添加了一个新的系统变量 `innodb_aurora_max_partitions_for_range`。在某些无法获得持久统计数据的情况下，可以使用此参数来改善分区表的行计数估计执行时间。有关更多信息，请参阅文档 [Aurora MySQL 配置参数](#)。
- 修复了在创建分区表时，错误地允许客户将 ROW_FORMAT 设置为 COMPRESSED 的问题。表将被隐式转换为 COMPACT 格式，并发出警告，告知 Aurora MySQL 不支持压缩表。
- 修复了当 `replica_parallel_type` 变量设置为 LOGICAL_CLOCK 且 `replica_preserve_commit_order` 变量设置为 ON 时，可能导致多线程二进制日志复制停止的问题。当在源上执行大于 500MB 的事务时，可能会出现此问题。
- 修复了启用了[全局数据库写入转发](#)功能时出现的问题，该问题可能导致辅助区域中读取器实例的 `performance_schema` 配置更改无意中转发到主区域中的写入器实例。
- 修复了从 Aurora 存储文件系统读取数据页面后，可能无法更新服务器状态变量 `innodb_buffer_pool_reads` 的问题。
- 选择 Aurora I/O-Optimized 集群配置时，不支持 Aurora MySQL 并行查询。有关更多信息，请参阅 [Amazon Aurora MySQL 并行查询限制](#)。
- 修复了启用了并行查询时导致查询计划优化程序为某些受益于主索引或二级索引的 SELECT 查询选择效率低下的执行计划的问题。

- 已将时区定义升级到 IANA 2023c 版本。
- 在二进制日志副本上引入了文件管理性能优化，以帮助减少写入中继日志文件时的争用。
- 修复了无论用户工作量如何，`information_schema.aurora_global_db_status`表和 AuroraGlobalDBRPOLag CloudWatch 指标中的 `RPO_LAG_IN_MILLISECONDS` 列始终显示为零的问题。
- 引入了一个新参数 `aurora_tmptable_enable_per_table_limit`。启用此参数后，该 `tmp_table_size` 变量将定义 TempTable 存储引擎创建的单个内存内部临时表的最大大小。有关其他详细信息，请参阅[内部（隐式）临时表的存储引擎](#)。
- 修复了启用了[全局数据库写入转发](#)功能时，会创建其他连接的问题。当读取器实例上的只读事务错误地将隐式提交转发给写入器时，就会出现此问题。
- 修复了使用[全局数据库写入转发](#)功能的连接在主区域的写入器上未填充 `performance_schema.threads` 表中的 `PROCESLIST_USER` 和 `PROCESLIST_HOST` 字段的问题。有关此表和性能架构的更多信息，请参阅 MySQL 参考手册、[The threads Table](#) 和《Amazon Aurora 用户指南》中的[性能架构概览](#)。
- 修复了使用[全局数据库写入转发](#)功能时，CommitLatency Cloudwatch 指标为辅助区域的读取器实例显示的值不正确的问题。要监控辅助数据库集群上转发的 DML 语句延迟，建议使用 `ForwardingReplicaDMLLatency` 和 `ForwardingWriterDMLLatency` 指标。也可以使用主区域的写入器实例上的 `CommitLatency` 指标来观察提交延迟。有关更多信息，请参阅 Aurora 用户指南、[亚马逊写入转发 CloudWatch 指标](#)。
- 修复了通过将 `replica_parallel_workers` 变量的值设置为大于 0 来配置多线程二进制日志复制时，用于管理和配置二进制日志复制的 [Aurora MySQL 复制控制存储过程](#) 错误地报错的问题。

升级和迁移：

- 要执行将 Aurora Global Database 从 Aurora MySQL 版本 3.01、3.02 或 3.03 升级到 Aurora MySQL 版本 3.04 或更高版本的次要版本升级，请参阅[通过修改引擎版本升级 Aurora MySQL](#)。
- 修复了从 Aurora MySQL 2 升级到 Aurora MySQL 3 时，由于报告了 `mysql.general_log_backup`、`mysql.general_log`、`mysql.slow_log_backup` 和 `mysql.slow_log` 表的架构不一致错误，而可能导致升级预检查失败的问题。有关升级问题排查的更多信息，请参阅[Aurora MySQL 版本 3 升级问题排查](#)。
- 修复了如果触发器定义包含不在引号内的保留关键字，则升级到 Aurora MySQL 3 时可能导致主要版本升级失败的问题。

集成了 MySQL 社区版本错误修复

除以下内容外，此版本还包括 8.0.28 及之前的所有社区错误修复。有关更多信息，请参阅 [Aurora MySQL 3.x 数据库引擎更新和之前的所有 MySQL 错误修复](#)。

- 修复了在页面遍历期间重新定位包含内部临时表页面的缓冲块，从而导致断言失败的问题（错误 #33715694）
- InnoDB：阻止在线 DDL 操作访问 out-of-bounds 内存（错误# 34750489，错误# 108925）
- 修复了在处理由多个嵌套的公用表表达式（CTE）组成的复杂 SQL 语句时，有时会生成错误查询结果的问题（错误 #34572040、错误 #34634469、错误 #33856374）

Aurora MySQL 数据库引擎更新 2023-12-08（版本 3.03.3，与 MySQL 8.0.26 兼容）

版本：3.03.3

Aurora MySQL 3.03.3 已正式发布。Aurora MySQL 3.03 版本与 MySQL 8.0.26 兼容。有关从 8.0.23 到 8.0.28 发生的社群变更的更多信息，请参阅 [MySQL 8.0 Release Notes](#)。

有关 Aurora MySQL 版本 3 中的新功能的详细信息，请参阅[与 MySQL 8.0 兼容的 Aurora MySQL 版本 3](#)。有关 Aurora MySQL 版本 3 和 Aurora MySQL 版本 2 之间的区别，请参阅[比较 Aurora MySQL 版本 2 和 Aurora MySQL 版本 3](#)。有关 Aurora MySQL 版本 3 和 MySQL 8.0 社群版的比较，请参阅[比较 Aurora MySQL 版本 3 和 MySQL 8.0 社群版](#)。

目前可用的 Aurora MySQL 版本有 2.07.9、2.07.10、2.11.*、2.12.*、3.01.*、3.02.*、3.03.*、3.04.* 和 3.05.*。

您可以执行就地升级、还原快照或使用 [Amazon RDS 蓝绿部署](#) 启动托管蓝绿升级，从当前可用的任何 Aurora MySQL 版本 2 集群升级到 Aurora MySQL 版本 3.03.3 集群。

有关计划升级到 Aurora MySQL 版本 3 的信息，请参阅《Amazon Aurora 用户指南》中的 [Aurora MySQL 版本 3 的升级计划](#)。有关 Aurora MySQL 升级的一般信息，请参阅《Amazon Aurora 用户指南》中的[升级 Amazon Aurora MySQL 数据库集群](#)。

有关问题排查信息，请参阅[排查 Aurora MySQL 版本 3 升级问题](#)。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

改进

修复了下面列出的安全问题和 CVE：

对托管环境中的处理进行微调的修复和其他增强功能。其他 CVE 修复如下：

- [CVE-2023-38545](#)

可用性改进：

- 修复了使用并行查询的 Aurora MySQL 数据库实例在运行大量并发并行查询时可能会遇到数据库重启的问题。
- 修复了当任何二进制日志源将 `gtid_mode` 设置为 `ON` 或 `ON_PERMISSIVE` 时，在启用了增强型二进制日志的二进制日志 (binlog) 副本集群上可能导致已执行的 GTID 集无法正确恢复的问题。此问题可能会导致副本集群的写入器实例在恢复期间多重启一次，或者在查询已执行的 GTID 集时导致结果不正确。
- 修复了启用增强型二进制日志后，由于可用内存减少而可能导致 Aurora MySQL 数据库实例重启或失效转移的内存管理问题。
- 修复了当写入器实例将数据库卷增大为 160 GB 的倍数时，可能会导致读取器实例重启的问题。
- 修复了启用了增强型二进制日志功能的 Aurora MySQL 数据库实例在执行二进制日志恢复过程时可能在数据库实例启动期间卡住的问题。
- 修复了零停机时间修补期间的一个问题，该问题会导致实例重启，从而导致数据库连接意外关闭。
- 修复了在同时运行 [SHOW STATUS](#) 和 [PURGE BINARY LOGS](#) 语句时，由于死锁而可能导致数据库实例重启的问题。PURGE BINARY LOGS 是一种托管语句，执行该语句是为了遵守用户配置的二进制日志保留期。
- 修复了在具有 Aurora 副本的集群上使用增强型二进制日志功能时，由于长时间等待信号灯而可能导致数据库实例重启的问题。

常规改进：

- 现在，在启用了增强型二进制日志功能时，在写入 Aurora 存储之前会删除未使用的存储元数据。这样可以避免在某些情况下，由于通过网络传输的字节数增加而导致写入延迟增加，从而进一步导致数据库重启或失效转移。
 - 修复了启用增强型二进制日志时可能导致开启的 `NumBinaryLogFiles` 指标显示错误结果的问题。
- CloudWatch

- 修复了可能导致 `table_open_cache` 数据库参数的修改在数据库实例重启后才生效的问题。
- 修复了当连接的二进制日志 (binlog) 使用者使用重复的二进制日志复制服务器 ID 时，可能导致数据库重启的问题。

集成了 MySQL 社区版本错误修复

除以下内容外，此版本还包括 8.0.26 及之前的所有社区错误修复。有关更多信息，请参阅 [Aurora MySQL 3.x 数据库引擎更新和之前的所有 MySQL 错误修复](#)。

- 修复了因后台 TLS 证书轮换而可能导致更高 CPU 使用率的问题 (社区错误修复 #34284186)

Aurora MySQL 数据库引擎更新 2023-08-29 (版本 3.03.2 ，与 MySQL 8.0.26 兼容)

版本：3.03.2

Aurora MySQL 3.03.2 已正式发布。Aurora MySQL 3.04 版与 MySQL 8.0.28 兼容，Aurora MySQL 3.03 版与 MySQL 8.0.26 兼容，Aurora MySQL 3.02 版与 MySQL 8.0.23 兼容。有关从 8.0.23 到 8.0.28 发生的社群变更的更多信息，请参阅 [MySQL 8.0 Release Notes](#)。

有关 Aurora MySQL 版本 3 中的新功能的信息，请参阅[与 MySQL 8.0 兼容的 Aurora MySQL 版本 3](#)。有关 Aurora MySQL 版本 3 和 Aurora MySQL 版本 2 之间的区别，请参阅[比较 Aurora MySQL 版本 2 和 Aurora MySQL 版本 3](#)。有关 Aurora MySQL 版本 3 和 MySQL 8.0 社群版的比较，请参阅[比较 Aurora MySQL 版本 3 和 MySQL 8.0 社群版](#)。

目前可用的 Aurora MySQL 版本有 2.07.9、2.07.10、2.11.*、3.01.*、3.02.*、3.03.* 和 3.04.*。

您可以执行就地升级、还原快照或使用 [Amazon RDS 蓝绿部署](#) 启动托管蓝绿升级，从当前可用的任何 Aurora MySQL 版本 2 集群升级到 Aurora MySQL 版本 3.03.2 集群。

有关计划升级到 Aurora MySQL 版本 3 的信息，请参阅《Amazon Aurora 用户指南》中的 [Aurora MySQL 版本 3 的升级计划](#)。有关 Aurora MySQL 升级的一般信息，请参阅《Amazon Aurora 用户指南》中的 [升级 Amazon Aurora MySQL 数据库集群](#)。

有关问题排查信息，请参阅[排查 Aurora MySQL 版本 3 升级问题](#)。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

改进

修复了安全问题和 CVE：

- 修复了在审核日志文件轮换期间可能导致审核日志错过事件的问题。

此版本中包含以下 CVE 修复程序：

- [CVE-2023-21963](#)
- [CVE-2023-21912](#)
- [CVE-2023-0215](#)
- [CVE-2022-43551](#)
- [CVE-2022-37434](#)

可用性改进：

- 修复了在长时间事务恢复期间可能导致数据库重新启动的问题。
- 修复了在数据库在内部系统表上创建或删除触发器时，当写入器实例重新启动时，可能会导致数据库集群不可用的问题。
- 修复了在执行引用聚合函数的查询时可能导致数据库实例重新启动的问题。
- 修复了启用并行查询后，在回滚INSERT语句期间可能导致数据库重启的问题。
- 在 Aurora MySQL 3.03.2 及更高版本中，仅对常规 InnoDB 表启用快速插入。这种优化不适用于 InnoDB 临时表。有关快速插入优化的更多信息，请参阅 [Amazon Aurora MySQL 性能增强](#)。

常规改进：

- 修复了读取器实例无法打开表并显示 ERROR 1146 的问题。在写入器实例上使用 INPLACE 算法并执行某些类型的在线数据定义语言 (DDL) 时，会出现此问题。
- 在二进制日志副本上引入了文件管理性能优化，以帮助减少写入中继日志文件时的争用。
- 修复了启用了并行查询时导致查询计划优化程序为某些受益于主索引或二级索引的 SELECT 查询选择效率低下的执行计划的问题。
- 增加了对以下数据控制语言 (DCL) 语句的逻辑复制支持：GRANT/REVOKE 和 CREATE/DROP/ALTER/RENAME USER。

- 选择 Aurora I/O-Optimized 集群配置时，不支持 Amazon Aurora MySQL 并行查询。有关更多信息，请参阅 [Aurora MySQL 并行查询的限制](#)。

升级和迁移：

- 要执行将 Aurora Global Database 从 Aurora MySQL 版本 3.01 或 3.02 升级到 Aurora MySQL 版本 3.03 或更高版本的次要版本升级，请参阅[通过修改引擎版本升级 Aurora MySQL](#)。
- 修复了如果触发器定义包含不在引号内的保留关键字，则升级到 Aurora MySQL 版本 3 时可能导致主要版本升级失败的问题。

集成了 MySQL 社区版本错误修复

除以下内容外，此版本还包括 8.0.26 及之前的所有社区错误修复。有关更多信息，请参阅 [Aurora MySQL 3.x 数据库引擎更新和之前的所有 MySQL 错误修复](#)。

- 修复了在处理由多个嵌套的公用表表达式 (CTE) 组成的复杂 SQL 语句时，有时会生成错误查询结果的问题。(错误 #34572040、错误 #34634469、错误 #33856374)
- InnoDB：尝试取消初始化和初始化同一个表的统计信息的线程之间的争用条件会导致断言失败。(错误 #33135425)
- InnoDB：防止在线 DDL 操作访问内存。out-of-bounds (错误 #34750489、错误 #108925)

Aurora MySQL 数据库引擎更新 2023-05-11 (版本 3.03.1 ，与 MySQL 8.0.26 兼容)

版本：3.03.1

Aurora MySQL 3.03.1 已正式发布。Aurora MySQL 3.03 版与 MySQL 8.0.26 兼容，Aurora MySQL 3.02 版与 MySQL 8.0.23 兼容。有关从 8.0.23 到 8.0.26 发生的社群变更的更多信息，请参阅 [MySQL 8.0 Release Notes](#)。

有关 Aurora MySQL 版本 3 中的新功能的信息，请参阅[与 MySQL 8.0 兼容的 Aurora MySQL 版本 3](#)。有关 Aurora MySQL 版本 3 和 Aurora MySQL 版本 2 之间的区别，请参阅[比较 Aurora MySQL 版本 2 和 Aurora MySQL 版本 3](#)。有关 Aurora MySQL 版本 3 和 MySQL 8.0 社群版的比较，请参阅[比较 Aurora MySQL 版本 3 和 MySQL 8.0 社群版](#)。

当前支持的 Aurora MySQL 版本有 2.07.9、2.11.1、2.11.2、3.01.*、3.02.* 和 3.03.*。

您可以执行就地升级，或将快照从当前支持的任何 Aurora MySQL 版本 2 集群还原到 Aurora MySQL 3.03.1。

有关计划升级到 Aurora MySQL 版本 3 的信息，请参阅《Amazon Aurora 用户指南》中的 [Aurora MySQL 版本 3 的升级计划](#)。有关 Aurora MySQL 升级的一般信息，请参阅《Amazon Aurora 用户指南》中的 [升级 Amazon Aurora MySQL 数据库集群](#)。

有关问题排查信息，请参阅 [排查 Aurora MySQL 版本 3 升级问题](#)。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得 [AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

改进

新功能：

- Aurora I/O-Optimized 存储配置从版本 3.03.1 开始可用。有关更多信息，请参阅 [Amazon Aurora 数据库集群的存储配置](#)。
- 添加了一个新的系统变量 `innodb_aurora_max_partitions_for_range`。在某些无法获得持久统计数据的情况下，可以使用此参数来改善分区表的行计数估计执行时间。有关更多信息，请参阅文档 [Aurora MySQL 配置参数](#)。

可用性改进：

- 修复了在提交事务后立即关闭连接的情况下，可能由于错误访问无效内存而导致数据库实例重启的问题。
- 修复了 Aurora 高级审计中的以下问题：当服务器变量 `server_audit_events` 设置为 ALL 或 QUERY 时，会导致将过多信息性消息记录到 Aurora MySQL 错误日志中。此问题可能会导致数据库实例重启。
- 修复了在某些情况下，在范围估计期间尝试读取无法再访问的页面时，可能导致 Aurora 读取器实例重启的问题。
- 修复了在执行使用 Aurora 并行查询执行计划的查询时可能导致 Aurora MySQL 读取器实例重启的问题。
- 修复了在连接了多个二进制日志复制使用者时，使用二进制日志复制的数据库实例可能会遇到 CPU 使用率增加和连接失败的问题。
- 修复了在实现中间临时表时考虑公用表表达式 (CTE) 不支持的索引扫描访问方法的问题，这可能会导致不良行为，包括数据库重启或查询结果不正确。我们通过避免使用 TempTable 存储引擎对表使用此类不支持的索引扫描访问方法来解决此问题。

- 在此 Aurora MySQL 版本中未启用快速插入，这是因为在运行 INSERT INTO SELECT、和等查询时可能会出现不一致的问题。FROM 有关快速插入优化的更多信息，请参阅 [Amazon Aurora MySQL 性能增强](#)。

常规改进：

- 修复了可能导致 SHOW BINARY LOGS 语句执行时间超过预期的问题。这可能导致数据库的提交吞吐量下降。
- 修复了可能导致使用 Instant ADD COLUMN 功能添加了列的用户表在并行导出时失败的问题。
- 修复了在处理审计日志轮换时报告的事件可能未写入审计日志的问题。
- 修复了在对 INFORMATION_SCHEMA INNODB_TABLESPACES 表执行查询时可能导致可用内存耗尽的问题。
- 修复了创建分区表时，错误地允许客户将 ROW_FORMAT 设置为 COMPRESSED 的问题。表将被隐式转换为 COMPACT 格式，并发出警告，告知 Aurora MySQL 不支持压缩表。

升级和迁移：

- 要执行将 Aurora Global Database 从 Aurora MySQL 版本 3.01 或 3.02 升级到 Aurora MySQL 版本 3.03 或更高版本的次要版本升级，请参阅 [通过修改引擎版本升级 Aurora MySQL](#)。
- 修复了从 Aurora MySQL 2 升级到 Aurora MySQL 3 时，由于报告了 mysql.general_log_backup、mysql.general_log、mysql.slow_log_backup 和 mysql.slow_log 表的架构不一致错误，而可能导致升级预检查失败的问题。有关升级问题排查的更多信息，请参阅 [Aurora MySQL 版本 3 升级问题排查](#)。

集成了 MySQL 社区版本错误修复

除以下内容外，此版本还包括 8.0.26 及之前的所有社区错误修复。有关更多信息，请参阅 [Aurora MySQL 3.x 数据库引擎更新和之前的所有 MySQL 错误修复](#)。

- 修复了在页面遍历期间重新定位包含内部临时表页面的缓冲块，从而导致断言失败的问题。（错误 #33715694）

Aurora MySQL 数据库引擎更新 2023-03-01（版本 3.03.0，与 MySQL 8.0.26 兼容）不支持升级到此版本。

版本：3.03.0

Aurora MySQL 3.03.0 已正式发布。Aurora MySQL 3.03 版与 MySQL 8.0.26 兼容，Aurora MySQL 3.02 版与 MySQL 8.0.23 兼容。有关从 8.0.23 到 8.0.26 发生的社群变更的更多信息，请参阅 [MySQL 8.0 Release Notes](#)。

有关 Aurora MySQL 版本 3 中的新功能的详细信息，请参阅[与 MySQL 8.0 兼容的 Aurora MySQL 版本 3](#)。有关 Aurora MySQL 版本 3 和 Aurora MySQL 版本 2 之间的区别，请参阅[比较 Aurora MySQL 版本 2 和 Aurora MySQL 版本 3](#)。有关 Aurora MySQL 版本 3 和 MySQL 8.0 社群版的比较，请参阅[比较 Aurora MySQL 版本 3 和 MySQL 8.0 社群版](#)。

当前支持的 Aurora MySQL 版本有 2.07.*、2.11.*、3.01.*、3.02.* 和 3.03.*。

您可以执行就地升级，或将快照从当前支持的任何 Aurora MySQL 版本 2 集群还原到 Aurora MySQL 3.03.0。

有关计划升级到 Aurora MySQL 版本 3 的信息，请参阅《Amazon Aurora 用户指南》中的 [Aurora MySQL 版本 3 的升级计划](#)。有关 Aurora MySQL 升级的一般信息，请参阅《Amazon Aurora 用户指南》中的[升级 Amazon Aurora MySQL 数据库集群](#)。

有关问题排查信息，请参阅[排查 Aurora MySQL 版本 3 升级问题](#)。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

改进

修复了下面列出的安全问题和 CVE：

对托管环境中的处理进行微调的修复和其他增强功能。其他 CVE 修复如下：

- [CVE-2022-32221](#)
- [CVE-2022-21451](#)
- [CVE-2022-21444](#)

可用性改进：

- 修复了以下问题：由于缓冲池初始化用时比预期更长而导致较大的数据库实例类在重启期间可能遇到问题。

- 修复了在启用了二进制日志记录时，数据库实例可能会在数据库恢复过程中重启的问题。
- 修复了在执行数据控制语言 (DCL) 语句 (例如 GRANT 和 REVOKE) 或在写入器实例上建立新连接时，可能导致读取器实例连接失败的问题。
- 修复了对数据操作语言 (DML) 操作 (例如目前不支持的 DELETE 和 UPDATE 语句) 错误使用并行查询，从而导致数据库实例重启的问题。有关并行查询支持的操作的更多信息，请参阅 [Aurora MySQL 并行查询限制](#)。
- 修复了在极少数情况下，在写入器实例上同时执行大型更新操作或数据定义语言 (DDL) 工作负载并在 Aurora 副本上对同一组表进行读取操作时，可能导致 Aurora 副本重启的问题。
- 修复了 Aurora Serverless v2 读取器实例缩减操作的一个问题，该问题可能导致该读取器实例重启，在极少数情况下还会导致数据不一致。
- 修复了关闭与数据库实例的连接时，由于错误访问无效内存位置而可能导致数据库实例重启的问题。
- 修复了在极少数情况下，如果处理的查询带有将十进制列截断为零小数位的 GROUP BY 子句，则可能导致数据库实例重启的问题。
- 修复了使用空间索引执行范围查询时，由于错误访问记录而可能导致数据库实例重启的问题。
- 修复了当内部临时表超过默认的或客户配置的内存或 mmap 值时，可能会导致 Aurora MySQL 副本实例上的数据库重启的问题。
- 修复了一个问题，即，高级审计日志轮换可能导致内存管理问题。
- 在此 Aurora MySQL 版本中未启用快速插入，这是因为在运行 INSERT INTO SELECT、和等查询时可能会出现不一致的问题。FROM 有关快速插入优化的更多信息，请参阅 [Amazon Aurora MySQL 性能增强](#)。

常规改进：

- 使用 GLOBAL 读取一致性设置改善了全局数据库写入转发会话的读取查询延迟。
- 修复了在客户端会话执行 reset_connection 或 change_user 命令后，不采用 wait_timeout 参数值的问题。
- 修复了当数据库实例的传入连接突然增加时，应用程序在连接到数据库实例时可能会遇到延迟增加的问题。引入了两个新 CloudWatch 指标，AuroraSlowHandshakeCount 即和 AuroraSlowConnectionHandleCount，以帮助排除 Aurora MySQL 数据库实例的连接建立延迟问题。有关这些指标的更多信息，请参阅 Aurora CloudWatch 指标定义文档《[亚马逊 Aurora 的亚马逊 CloudWatch 指标](#)》。
- temptable_use_mmap 参数已被弃用，预计将在将来的 MySQL 版本中删除对它的支持。有关更多信息，请参阅 [内部 \(隐式 \) 临时表的存储引擎](#)。

- 修复了可能导致 SHOW BINARY LOGS 语句执行时间超过预期的问题。这可能导致数据库的提交吞吐量下降。

升级和迁移：

- 要执行将 Aurora Global Database 从 Aurora MySQL 版本 3.01 或 3.02 升级到 Aurora MySQL 版本 3.03 或更高版本的次要版本升级，请参阅[通过修改引擎版本升级 Aurora MySQL](#)。
- 修复了当集群中有大量表（超过 75 万个）时，可能导致从 Aurora MySQL 版本 2 到 Aurora MySQL 版本 3 的主要版本升级失败的问题。
- 修复了由于迁移 mysql.innodb_table_stats 和 mysql.innodb_index_stats 表的用时比预期时间更长，从而可能导致从 Aurora MySQL 版本 2 到 Aurora MySQL 版本 3 的主要版本升级失败的问题。此问题主要影响了拥有数百万个表的数据库集群。
- 修复了由于架构不一致错误，因此在从 Aurora MySQL 版本 2 升级到 Aurora MySQL 版本 3 期间可能导致失败的问题。这些错误是 mysql.general_log_template 和 mysql.slow_log_template 表的升级预检查器报告的。有关升级问题排查的更多信息，请参阅[Aurora MySQL 版本 3 升级问题排查](#)。
- 修复了由于 schemaInconsistencyCheck 错误而导致从 Aurora MySQL 版本 2 升级到 Aurora MySQL 版本 3 时失败的问题。如 upgrade-prechecks.log 报告所言，此错误是由于 mysql.table_migration_index_info 表中的架构不一致导致的。有关 Aurora MySQL 版本 3 升级问题排查的更多信息，请参阅[Aurora MySQL 版本 3 升级问题排查](#)。

集成了 MySQL 社区版本错误修复

除以下内容外，此版本还包括 8.0.26 及之前的所有社区错误修复。有关更多信息，请参阅[Aurora MySQL 3.x 数据库引擎更新和之前的所有 MySQL 错误修复](#)。

- 修复了以下问题：排序缓冲区的大小必须最少是排序中最大行的 15 倍，否则某些列类型（包括 JSON 和 TEXT）的排序有时会耗尽排序缓冲区。现在，排序缓冲区的大小只需是最大排序键的 15 倍。（错误 #103325、错误 #105532、错误 #32738705、错误 #33501541）
- 修复了 InnoDB 并不总能正确处理表分区的某些合法名称的问题。（错误 #32208630）
- 修复了在执行具有 OR 条件的查询时，由于空性属性的计算不准确，在某些条件下可能会返回错误结果的问题。（错误 #34060289）
- 修复了在某些情况下，满足以下两个条件时可能返回错误结果的问题：
 - 将一个派生表合并到了外部查询块中
 - 查询包括一个左联接和一个 IN 子查询

(错误 #34060289)

- 修复了在超过最大整数列值时生成不正确的 AUTO_INCREMENT 值的问题。该错误由于未考虑最大列值所致。在这种情况下，应返回上一个有效的 AUTO_INCREMENT 值，而没有这样做就导致了键重复错误。(错误 #87926、错误 #26906787)
- 修复了无法撤销性能架构上的 DROP 权限的问题。(错误 #33578113)
- 修复了包含使用 EXISTS 的 IF 语句的存储过程无法正确执行的问题，该语句对在两次执行之间删除并重新创建的一个或多个表执行操作，但在第一次执行之后的后续调用中无法正确执行。(错误 #32855634)。
- 修复了引用子查询中的视图和外部查询块的查询可能导致意外重启的问题。(错误 #32324234)

Aurora MySQL 数据库引擎更新 2023-04-17 (版本 3.02.3 ，与 MySQL 8.0.23 兼容) 标准支持将于 2024 年 1 月 15 日结束。

版本 : 3.02.3

Aurora MySQL 3.02.3 已正式发布。Aurora MySQL 3.02 版与 MySQL 8.0.23 兼容，Aurora MySQL 2.x 版与 MySQL 5.7 兼容。

有关 Aurora MySQL 版本 3 中的新功能以及 Aurora MySQL 版本 3 和 Aurora MySQL 版本 2 或 MySQL 8.0 社群版之间的区别的详细信息，请参阅《Amazon Aurora 用户指南》中的[比较 Aurora MySQL 版本 2 和 Aurora MySQL 版本 3](#)。

当前支持的 Aurora MySQL 版本有 2.07.*、2.11.1、2.11.2、3.01.*、3.02.* 和 3.03.*。

您可以执行就地升级，或将快照从当前支持的任何 Aurora MySQL 版本 2 集群还原到 Aurora MySQL 3.02.3。

有关计划升级到 Aurora MySQL 版本 3 的信息，请参阅《Amazon Aurora 用户指南》中的[Aurora MySQL 版本 3 的升级计划](#)。有关升级过程本身，请参阅《Amazon Aurora 用户指南》中的[升级到 Aurora MySQL 版本 3](#)。有关 Aurora MySQL 升级的一般信息，请参阅《Amazon Aurora 用户指南》中的[升级 Amazon Aurora MySQL 数据库集群](#)。

有关问题排查信息，请参阅[排查 Aurora MySQL 版本 3 升级问题](#)。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

改进

可用性改进：

- 修复了在提交事务后立即关闭连接的情况下，可能由于错误访问无效内存而导致数据库实例重启的问题。
- 在此 Aurora MySQL 版本中未启用快速插入，这是因为在运行 INSERT INTO SELECT、和等查询时可能会出现不一致的问题。FROM 有关快速插入优化的更多信息，请参阅 [Amazon Aurora MySQL 性能增强](#)。

常规改进：

- 修复了为实现中间临时表时考虑公用表表达式 (CTE) 不支持的索引扫描访问方法的问题，这可能会导致不良行为，包括数据库重启或查询结果不正确。已通过避免使用 TempTable 存储引擎对表使用此类不支持的索引扫描访问方法来修复此问题。
- 修复了在极少数情况下，访问在 Aurora MySQL 写入器实例上同时运行大型更新或数据定义语言 (DDL) 操作的表时，可能会导致 Aurora MySQL 读取器实例重启的问题。
- 修复了在某些情况下，在范围估计期间尝试读取无法再访问的页面时，可能导致 Aurora MySQL 读取器实例重启的问题。
- 修复了在连接了多个二进制日志复制使用者时，使用二进制日志复制的数据库实例可能会遇到 CPU 使用率增加和连接失败的问题。
- 修复了在执行使用 Aurora 并行查询执行计划的查询时可能导致 Aurora MySQL 读取器实例重启的问题。

Aurora MySQL 数据库引擎更新 2022-11-18 (版本 3.02.2 ，与 MySQL 8.0.23 兼容) 标准支持将于 2024 年 1 月 15 日结束。

版本：3.02.2

Aurora MySQL 3.02.2 已正式发布。Aurora MySQL 3.02 版与 MySQL 8.0.23 兼容，Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

有关 Aurora MySQL 版本 3 中的新功能以及 Aurora MySQL 版本 3 和 Aurora MySQL 版本 2 或 MySQL 8.0 社群版之间的区别的详细信息，请参阅《Amazon Aurora 用户指南》中的 [比较 Aurora MySQL 版本 2 和 Aurora MySQL 版本 3](#)。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

您可以将快照从当前支持的任何 Aurora MySQL 版本 2 集群还原到 Aurora MySQL 3.02.2。

有关计划升级到 Aurora MySQL 版本 3 的信息，请参阅《Amazon Aurora 用户指南》中的 [Aurora MySQL 版本 3 的升级计划](#)。有关升级过程本身，请参阅《Amazon Aurora 用户指南》中的 [升级到 Aurora MySQL 版本 3](#)。有关 Aurora MySQL 升级的一般信息，请参阅《Amazon Aurora 用户指南》中的 [升级 Amazon Aurora MySQL 数据库集群](#)。

有关问题排查信息，请参阅 [排查 Aurora MySQL 版本 3 升级问题](#)。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得 [AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

改进

Aurora MySQL 版本 3.02.2 已正式发布并一般与 MySQL 8.0.23 社群版兼容。

修复了下面列出的安全问题和 CVE：

对托管环境中的处理进行微调的修复和其他增强功能。其他 CVE 修复如下：

- [CVE-2022-21451](#)
- [CVE-2021-36222](#)
- [CVE-2021-22926](#)

- [CVE-2022-21444](#)

可用性改进：

- 修复了显式或隐式关闭与数据库实例的连接时，由于错误访问无效内存而可能导致数据库实例重启的问题。
- 修复了由于缓冲池初始化用时比预期更长而可能导致在较大的实例类上重复中断数据库启动的问题。
- 修复了在极少数情况下，Aurora Serverless v2 错误地尝试在扩展时更新表缓存，而可能导致数据库实例重启的问题。
- 修复了在极少数情况下，如果处理的查询带有将十进制列截断为零小数位的 GROUP BY 子句，则可能导致数据库重启的问题。

- 在此 Aurora MySQL 版本中未启用快速插入，这是因为在运行 INSERT INTO SELECT、和等查询时可能会出现不一致的问题。FROM 有关快速插入优化的更多信息，请参阅 [Amazon Aurora MySQL 性能增强](#)。

常规改进：

- 修复了由于 mysql.host 表中的元数据不一致，可能导致从 Aurora MySQL 版本 2 (与 MySQL 5.7 兼容) 到 Aurora MySQL 版本 3 (与 MySQL 8.0 兼容) 的升级失败的问题。
- 增加了性能改进，缩短了从 Aurora MySQL 版本 2 (与 MySQL 5.7 兼容) 升级到 Aurora MySQL 版本 3 (与 MySQL 8.0 兼容) 所用的时间。通过并行执行某些升级步骤，使用更大的实例类 (例如 db.r6g.16xlarge 或 db.r5.24xlarge) 时，升级时间进一步缩短。
- 增加了在从 Aurora MySQL 版本 2 (与 MySQL 5.7 兼容) 升级到 Aurora MySQL 版本 3 (与 MySQL 8.0 兼容) 时，显示所有错误的支持，而之前的版本仅限于显示 50 个错误。
- 修复了在极少数情况下，从 Aurora MySQL 版本 2 (与 MySQL 5.7 兼容) 到 Aurora MySQL 版本 3 (与 MySQL 8.0 兼容) 的主要版本升级之后，可能导致自动增量计数器不正确的问题。
- 修复了由于迁移“mysql.innodb_table_stats”和“mysql.innodb_index_stats”表的用时比预期时间更长，从而可能导致从 Aurora MySQL 版本 2 到 Aurora MySQL 版本 3 的主要版本升级失败的问题。此问题主要影响具有大量表 (>150 万个) 的数据库集群。
- 修复了由于 AMS 8.0 引擎升级工作流存在缺陷而可能导致从 Aurora MySQL 版本 2 到 Aurora MySQL 版本 3 的主要版本升级失败的问题，该缺陷会导致日志记录累积在 Aurora 存储集群卷上并停止正常写入操作。此问题主要影响具有大量表 (大约 >75 万个) 的数据库集群。
- 修复了由于 MySQL 清除线程未正确保持活动状态而导致 Aurora MySQL Serverless v2 空闲实例无法缩减到 0.5 个 ACU 的问题。
- 修复了当数据库实例的传入连接突然增加时，应用程序在连接到数据库实例时可能会遇到延迟增加的问题。
- 引入了两个新的亚马逊 CloudWatch 指标，以帮助解决 Aurora MySQL 数据库实例的连接建立延迟问题。有关 AuroraSlowHandshakeCount 和 AuroraSlowConnectionHandleCount 指标的更多信息，请参阅 [Aurora CloudWatch 指标定义](#)。

集成了 MySQL 社区版本错误修复

除以下内容外，此版本还包括 8.0.23 及之前的所有社区错误修复。有关更多信息，请参阅 [Aurora MySQL 3.x 数据库引擎更新和之前的所有 MySQL 错误修复](#)。

- 修复了在执行具有 OR 条件的查询时，由于空性属性的计算不准确，在某些条件下可能会返回错误结果的问题。（错误 #34060289）
- 修复了在某些情况下，满足以下两个条件时可能返回错误结果的问题：
 - 将一个派生表合并到了外部查询块中。
 - 查询包括一个左联接和一个 IN 子查询。（错误 #34060289）
- 修复了无法撤销性能架构上的 DROP 权限的问题。（错误 #33578113）
- 修复了包含使用 EXISTS 的 IF 语句的存储过程无法正确执行的问题，该语句对在两次执行之间删除并重新创建的一个或多个表执行操作，但在第一次执行之后的后续调用中无法正确执行。（MySQL 错误 #32855634）。
- 当超过最大整数列值时，生成的 AUTO_INCREMENT 值不正确。该错误由于未考虑最大列值所致。在这种情况下，应返回上一个有效的 AUTO_INCREMENT 值，而没有这样做就导致了键重复错误。（错误 #87926、错误 #26906787）
- 修复了在升级 Aurora MySQL 版本 1（与 MySQL 5.6 兼容）数据库集群时，如果该集群包含用户创建的带有特定表 ID 的表，则可能导致失败的问题。从 Aurora MySQL 版本 2（与 MySQL 5.7 兼容）升级到 Aurora MySQL 版本 3（与 MySQL 8.0 兼容）时，分配这些表 ID 可能会导致数据字典表 ID 冲突（错误 #33919635）

Aurora MySQL 数据库引擎更新 2022-09-07（版本 3.02.1，与 MySQL 8.0.23 兼容）标准支持将于 2024 年 1 月 15 日结束。不支持升级到此版本。

版本：3.02.1

Aurora MySQL 3.02.1 已正式发布。Aurora MySQL 3.02 版与 MySQL 8.0.23 兼容，Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

有关 Aurora MySQL 版本 3 中的新功能以及 Aurora MySQL 版本 3 和 Aurora MySQL 版本 2 或 MySQL 8.0 社群版之间的区别的详细信息，请参阅《Amazon Aurora 用户指南》中的[比较 Aurora MySQL 版本 2 和 Aurora MySQL 版本 3](#)。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

有关计划升级到 Aurora MySQL 版本 3 的信息，请参阅《Amazon Aurora 用户指南》中的[Aurora MySQL 版本 3 的升级计划](#)。有关升级过程本身，请参阅《Amazon Aurora 用户指南》中的[升级到](#)

[Aurora MySQL 版本 3](#)。有关 Aurora MySQL 升级的一般信息，请参阅《Amazon Aurora 用户指南》中的[升级 Amazon Aurora MySQL 数据库集群](#)。

有关问题排查信息，请参阅[排查 Aurora MySQL 版本 3 升级问题](#)。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

改进

Aurora MySQL 版本 3.02.1 已正式发布并一般与 MySQL 8.0.23 社群版兼容。

修复了下面列出的安全问题和 CVE：

对托管环境中的处理进行微调的修复和其他增强功能。其他 CVE 修复如下：

- [CVE-2022-0778](#)

可用性改进：

- 修复了将多个 MySQL 二进制日志 (binlog) 副本连接到 Aurora 写入器节点时，或同时发生大量长时间运行的并发查询和新连接请求激增时，可能导致连接失败和高延迟的问题。
- 修复了在开启 CONNECT 事件高级审计时导致数据库重启的问题。
- 修复了当内部临时表耗尽内存中分配的大小并且 mmap 文件设置为客户配置或默认值时，可能会导致 Aurora MySQL 只读副本实例上的数据库重启的问题。
- 修复了在存储过程上执行并发 DDL 操作期间可能导致只读副本反复重启的问题。
- 在此 Aurora MySQL 版本中未启用快速插入，这是因为在运行 INSERT INTO SELECT、和等查询时可能会出现不一致的问题。FROM 有关快速插入优化的更多信息，请参阅 [Amazon Aurora MySQL 性能增强](#)。

常规改进：

- 增加了对 R6i 实例的支持。

附加信息：

- Aurora MySQL 版本 3.02.1 不支持直接从 Aurora MySQL 版本 2 (与 MySQL 5.7 兼容) 进行主要版本升级。要对此版本执行主要版本升级, 请先执行到 Aurora MySQL 版本 3.02.0 的主要版本升级, 然后执行到 Aurora MySQL 版本 3.02.1 的就地次要版本升级。

Aurora MySQL 数据库引擎更新 2022-04-20 (版本 3.02.0, 与 MySQL 8.0.23 兼容) 标准支持将于 2024 年 1 月 15 日结束。不支持升级到此版本。

版本 : 3.02.0

Aurora MySQL 3.02.0 已正式发布。Aurora MySQL 3.02 版与 MySQL 8.0.23 兼容, Aurora MySQL 2.x 版与 MySQL 5.7 兼容, Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

有关 Aurora MySQL 版本 3 中的新功能以及 Aurora MySQL 版本 3 和 Aurora MySQL 版本 2 或 MySQL 8.0 社群版之间的区别的详细信息, 请参阅《Amazon Aurora 用户指南》中的[比较 Aurora MySQL 版本 2 和 Aurora MySQL 版本 3](#)。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

您可以将快照从当前支持的任何 Aurora MySQL 版本 2 集群还原到 Aurora MySQL 3.02.0。

有关计划升级到 Aurora MySQL 版本 3 的信息, 请参阅《Amazon Aurora 用户指南》中的[Aurora MySQL 版本 3 的升级计划](#)。有关升级过程本身, 请参阅《Amazon Aurora 用户指南》中的[升级到 Aurora MySQL 版本 3](#)。有关 Aurora MySQL 升级的一般信息, 请参阅《Amazon Aurora 用户指南》中的[升级 Amazon Aurora MySQL 数据库集群](#)。

有关问题排查信息, 请参阅[排查 Aurora MySQL 版本 3 升级问题](#)。

如果您有任何疑问或疑虑, 可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

改进

Aurora MySQL 版本 3.02.0 已正式发布并一般与 MySQL 8.0.23 社群版兼容。

修复了下面列出的安全问题和 CVE :

对托管环境中的处理进行微调的修复和其他增强功能。其他 CVE 修复如下：

- [CVE-2021-22946](#)

新功能：

- Amazon Aurora Serverless v2 已正式发布。有关更多信息，请参阅 [Amazon Aurora Serverless 概述](#)、[博客](#) 和 [使用 Aurora Serverless v2](#) 文档。立即开始使用 AWS Management Console 中的很少的几个步骤创建 Aurora Serverless v2 数据库。

可用性改进：

- 修复了在删除记录或包含两个或更多可变长度列 (VARCHAR、VARBINARY、BLOB 和 TEXT 类型) 的表时，可能导致服务器进入重启循环且不可用的问题。有关列类型的更多详细信息，请参阅 [innodb-row-format](#)。
- 修复了以下问题：在开启了二进制日志且至少连接了一个二进制日志使用者的集群上，由于会导致应用程序和使用者之间的资源争用，因此现有连接会超时且无法建立新连接。
- 可用内存由 FreeableMemory CloudWatch 指标指示。有关更多信息，请参阅 [亚马逊 Aurora 的亚马逊 CloudWatch 指标](#)。
 - 修复了启用二进制日志复制后，由于可用内存减少而可能导致数据库实例重启或失效转移的问题。
 - 修复了在设置会话变量时，由于可用内存减少而可能导致数据库实例重启或失效转移的问题。
 - 修复了在数据库进程打开现有文件时，由于可用内存减少而可能导致数据库实例重启或失效转移的问题。
- 修复了在极少数情况下，如果在从快照还原的集群上，向包含 AUTO_INCREMENT 列的表中插入新行，则可能导致重复项错误的问题。
- 在此 Aurora MySQL 版本中未启用快速插入，这是因为在运行 INSERT INTO SELECT、和等查询时可能会出现不一致的问题。FROM 有关快速插入优化的更多信息，请参阅 [Amazon Aurora MySQL 性能增强](#)。

常规改进：

- 修复了使用 SHOW VOLUME STATUS 命令时未显示卷状态的问题。有关更多信息，请参阅 [AuroraMySQL.Managing. VolumeStatus](#)。
- 修复了导致对 [mysql_rds_import_binlog_ssl_material](#) 的调用失败并显示 [MySQL server ERROR 3512](#) 的问题。

- 修复了针对已删除的 Aurora 读取器实例不正确地报告 Aurora 副本延迟的问题。

升级/迁移：

- 修复了由于将 ibdata 文件和表空间复制到 Aurora 存储时出现问题，而可能导致 MySQL 8.0.x 数据库到 Aurora MySQL 版本 3 的迁移失败的问题。
- 修复了当数据库表包含大量数据时，可能导致集群从 Aurora MySQL 版本 2 到 Aurora MySQL 版本 3 的升级失败的问题。
- 修复了由于无法为表创建[序列化数据字典信息](#) (SDI)，可能导致集群从 Aurora MySQL 版本 2 到 Aurora MySQL 版本 3 的恢复失败的问题。
- 修复了由于 RDS 系统表的升级预检查报告了架构不一致错误，可能导致从 Aurora MySQL 版本 2 到 Aurora MySQL 版本 3 的升级失败的问题。
- 修复了由于 RDS 托管存储过程中的语法无效，可能导致从 RDS for MySQL 8.0 或 Aurora MySQL 版本 2 到 Aurora MySQL 版本 3 数据库的迁移或恢复失败的问题。
- 修复了由于 [general log](#) 和 [slow log](#) 表的升级预检查报告了架构不一致错误，可能导致从 Aurora MySQL 2 到 Aurora MySQL 3 的升级失败的问题。

集成了 MySQL 社区版本错误修复

除以下内容外，此版本还包括 8.0.23 及之前的所有社区错误修复。有关更多信息，请参阅 [Aurora MySQL 3.x 数据库引擎更新和之前的所有 MySQL 错误修复](#)。

- 修复了存储过程中用于游标的临时表处理不当的问题，这可能会导致服务器出现意外行为，[mysqld-8-0-24-bug](#)。（错误 #32416811）

Aurora MySQL 数据库引擎更新 2022-04-15 (版本 3.01.1 ，与 MySQL 8.0.23 兼容) 标准支持将于 2024 年 1 月 15 日结束。不支持升级到此版本。

版本：3.01.1

Aurora MySQL 3.01.1 已正式发布。Aurora MySQL 3.01 版与 MySQL 8.0.23 兼容，Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

有关 Aurora MySQL 版本 3 中的新功能以及 Aurora MySQL 版本 3 和 Aurora MySQL 版本 2 或 MySQL 8.0 社群版之间的区别的详细信息，请参阅《Amazon Aurora 用户指南》中的[比较 Aurora MySQL 版本 2 和 Aurora MySQL 版本 3](#)。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

您可以将快照从当前支持的任何 Aurora MySQL 版本 2 集群还原到 Aurora MySQL 3.01.1。

有关计划升级到 Aurora MySQL 版本 3 的信息，请参阅《Amazon Aurora 用户指南》中的[Aurora MySQL 版本 3 的升级计划](#)。有关升级过程本身，请参阅《Amazon Aurora 用户指南》中的[升级到 Aurora MySQL 版本 3](#)。有关 Aurora MySQL 升级的一般信息，请参阅《Amazon Aurora 用户指南》中的[升级 Amazon Aurora MySQL 数据库集群](#)。

有关问题排查信息，请参阅[排查 Aurora MySQL 版本 3 升级问题](#)。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

改进

Aurora MySQL 版本 3.01.1 已正式发布并一般与 MySQL 8.0.23 社群版兼容。

建议使用 Aurora MySQL 版本 3.01.1 升级和迁移与 Aurora MySQL 8.0 兼容的 Aurora 数据库。

修复了下面列出的安全问题和 CVE：

对托管环境中的处理进行微调的修复和其他增强功能。其他 CVE 修复如下：

- [CVE-2021-36222](#)
- [CVE-2021-22946](#)
- [CVE-2021-22926](#)

可用性改进：

- 可用内存由FreeableMemory CloudWatch 指标指示。有关更多信息，请参阅[亚马逊 Aurora 的亚马逊 CloudWatch 指标](#)。
 - 修复了启用二进制日志复制后，由于可用内存减少而可能导致数据库实例重启或失效转移的问题。
 - 修复了在设置会话变量时，由于可用内存减少而可能导致数据库实例重启或失效转移的问题。

- 修复了在数据库进程打开现有文件时，由于可用内存减少而可能导致数据库实例重启或失效转移的问题。
- 修复了在极少数情况下，如果在从快照还原的集群上，向包含 AUTO_INCREMENT 列的表中插入新行，则可能导致重复项错误的问题。
- 在此 Aurora MySQL 版本中未启用快速插入，这是因为在运行 INSERT INTO SELECT、和等查询时可能会出现不一致的问题。FROM 有关快速插入优化的更多信息，请参阅 [Amazon Aurora MySQL 性能增强](#)。

常规改进：

- 修复了使用 SHOW VOLUME STATUS 命令时未显示卷状态的问题。有关更多信息，请参阅 [Aurora MySQL.Managing. VolumeStatus](#)。
- 修复了导致对 [mysql_rds_import_binlog_ssl_material](#) 的调用失败并显示 [MySQL server ERROR 3512](#) 的问题。
- 修复了针对已删除的 Aurora 读取器实例不正确地报告 Aurora 副本延迟的问题。

升级/迁移：

- 修复了由于将 ibdata 文件和表空间复制到 Aurora 存储时出现问题，而可能导致 MySQL 8.0.x 数据库到 Aurora MySQL 版本 3 的迁移失败的问题。
- 修复了当数据库表包含大量数据时，可能导致集群从 Aurora MySQL 版本 2 到 Aurora MySQL 版本 3 的升级失败的问题。
- 修复了由于无法为表创建 [序列化数据字典信息](#) (SDI)，可能导致集群从 Aurora MySQL 版本 2 到 Aurora MySQL 版本 3 的恢复失败的问题。
- 修复了由于 RDS 系统表的升级预检查报告了架构不一致错误，可能导致从 Aurora MySQL 版本 2 到 Aurora MySQL 版本 3 的升级失败的问题。
- 修复了由于 RDS 托管存储过程中的语法无效，可能导致从 RDS for MySQL 8.0 或 Aurora MySQL 版本 2 到 Aurora MySQL 版本 3 数据库的迁移或恢复失败的问题。
- 修复了由于 [general log](#) 和 [slow log](#) 表的升级预检查报告了架构不一致错误，可能导致从 Aurora MySQL 2 到 Aurora MySQL 3 的升级失败的问题。

集成了 MySQL 社区版本错误修复

除以下内容外，此版本还包括 8.0.23 及之前的所有社区错误修复。有关更多信息，请参阅 [Aurora MySQL 3.x 数据库引擎更新和之前的所有 MySQL 错误修复](#)。

- 修复了存储过程中用于游标的临时表处理不当的问题，这可能会导致服务器出现意外行为，[mysqld-8-0-24-bug](#)。（错误 #32416811）

Aurora MySQL 数据库引擎更新 2021-11-18 (版本 3.01.0，与 MySQL 8.0.23 兼容) 标准支持将于 2024 年 1 月 15 日结束。不支持升级到此版本。

版本：3.01.0

Aurora MySQL 3.01.0 已正式发布。Aurora MySQL 3.01 版与 MySQL 8.0.23 兼容，Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

有关 Aurora MySQL 版本 3 中的新功能以及 Aurora MySQL 版本 3 和 Aurora MySQL 版本 2 或 MySQL 8.0 社群版之间的区别的详细信息，请参阅《Amazon Aurora 用户指南》中的[比较 Aurora MySQL 版本 2 和 Aurora MySQL 版本 3](#)。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

您可以将快照从当前支持的任何 Aurora MySQL 版本 2 集群还原到 Aurora MySQL 3.01.0。

有关计划升级到 Aurora MySQL 版本 3 的信息，请参阅《Amazon Aurora 用户指南》中的[Aurora MySQL 版本 3 的升级计划](#)。有关升级过程本身，请参阅《Amazon Aurora 用户指南》中的[升级到 Aurora MySQL 版本 3](#)。有关 Aurora MySQL 升级的一般信息，请参阅《Amazon Aurora 用户指南》中的[升级 Amazon Aurora MySQL 数据库集群](#)。

有关问题排查信息，请参阅[排查 Aurora MySQL 版本 3 升级问题](#)。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

改进

Aurora MySQL 版本 3.01.0 一般与社群 MySQL 8.0.23 兼容。此版本包括自社群 MySQL 8.0.23 起发生的常见漏洞和风险 (CVE) 问题的安全修复。

Aurora MySQL 版本 3.01.0 包含了通过 Aurora MySQL 版本 2.10.0 进行的所有特定于 Aurora 的错误修复。

有关 Aurora MySQL 版本 3 中的新功能的详细信息，请参阅《Amazon Aurora 用户指南》中的 [MySQL 8.0 社群版中的功能](#)和[新的并行查询优化](#)。

可用性改进：

- 在此 Aurora MySQL 版本中未启用快速插入，这是因为在运行 INSERT INTO SELECT、和等查询时可能会出现不一致的问题。FROM 有关快速插入优化的更多信息，请参阅 [Amazon Aurora MySQL 性能增强](#)。

Amazon Aurora MySQL 2 的数据库引擎更新

以下是 Amazon Aurora MySQL 版本 2 的数据库引擎更新。

- [Aurora MySQL 数据库引擎更新 2024-03-19 \(版本 2.12.2, 兼容 MySQL 5.7.44\)](#)
- [Aurora MySQL 数据库引擎更新 2023-12-28 \(版本 2.12.1, 与 MySQL 5.7.40 兼容\)](#)
- [Aurora MySQL 数据库引擎更新 2023-10-25 \(版本 2.12.0.1, 与 MySQL 5.7.40 兼容\) 测试版](#)
- [Aurora MySQL 数据库引擎更新 2023-07-25 \(版本 2.12.0, 与 MySQL 5.7.40 兼容\)](#)
- [Aurora MySQL 数据库引擎更新 2024-03-26 \(版本 2.11.5, 兼容 MySQL 5.7.12\)](#)
- [Aurora MySQL 数据库引擎更新 2023-10-17 \(版本 2.11.4, 与 MySQL 5.7.12 兼容\)](#)
- [Aurora MySQL 数据库引擎更新 2023-06-09 \(版本 2.11.3, 与 MySQL 5.7.12 兼容\)](#)
- [Aurora MySQL 数据库引擎更新 2023-03-24 \(版本 2.11.2, 兼容 MySQL 5.7.12\)](#)
- [Aurora MySQL 数据库引擎更新 2023-02-14 \(版本 2.11.1, 与 MySQL 5.7.12 兼容\)](#)
- [Aurora MySQL 数据库引擎更新 2022-10-25 \(版本 2.11.0, 与 MySQL 5.7.12 兼容\) 此版本不适用于新创建。](#)
- [Aurora MySQL 数据库引擎更新 2022-11-01 \(版本 2.10.3\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2022-01-26 \(版本 2.10.2\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2021-10-21 \(版本 2.10.1\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2021-05-25 \(版本 2.10.0\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2021-11-12 \(版本 2.09.3\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2021-02-26 \(版本 2.09.2\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2020-12-11 \(版本 2.09.1\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2020-09-17 \(版本 2.09.0\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2022-01-06 \(版本 2.08.4\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2020-11-12 \(版本 2.08.3\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2020-08-28 \(版本 2.08.2\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2020-06-18 \(版本 2.08.1\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2020-06-02 \(版本 2.08.0\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2023-08-15 \(版本 2.07.10, 与 MySQL 5.7.12 兼容\)](#)

- [Aurora MySQL 数据库引擎更新 2023-05-04 \(版本 2.07.9, 与 MySQL 5.7.12 兼容\)](#)
- [Aurora MySQL 数据库引擎更新 2022-06-16 \(版本 2.07.8\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2021-11-24 \(版本 2.07.7\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2021-09-02 \(版本 2.07.6\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2021-07-06 \(版本 2.07.5\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2021-03-04 \(版本 2.07.4\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2020-11-10 \(版本 2.07.3\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2020-04-17 \(版本 2.07.2\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2019-12-23 \(版本 2.07.1\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2019-11-25 \(版本 2.07.0\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2019-11-22 \(版本 2.06.0\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2019-11-11 \(版本 2.05.0\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2020-08-14 \(版本 2.04.9\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2019-11-20 \(版本 2.04.8\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2019-11-14 \(版本 2.04.7\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2019-09-19 \(版本 2.04.6\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2019-07-08 \(版本 2.04.5\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2019-05-29 \(版本 2.04.4\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2019-05-09 \(版本 2.04.3\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2019-05-02 \(版本 2.04.2\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2019-03-25 \(版本 2.04.1\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2019-03-25 \(版本 2.04.0\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2019-02-07 \(版本 2.03.4\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2019-01-18 \(版本 2.03.3\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2019-01-09 \(版本 2.03.2\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2018-10-24 \(版本 2.03.1\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2018-10-11 \(版本 2.03\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2018-10-08 \(版本 2.02.5\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2018-09-21 \(版本 2.02.4\) \(已弃用\)](#)

- [Aurora MySQL 数据库引擎更新 2018-08-23 \(版本 2.02.3\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2018-06-04 \(版本 2.02.2\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2018-05-03 \(版本 2.02\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2018-03-13 \(版本 2.01.1\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2018-02-06 \(版本 2.01\) \(已弃用\)](#)

Aurora MySQL 数据库引擎更新 2024-03-19 (版本 2.12.2 , 兼容 MySQL 5.7.44)

版本 : 2.12 .2

Aurora MySQL 2.12.2 已正式上市。Aurora MySQL 2.12 版本与 MySQL 5.7.44 之前兼容。有关社区变更的更多信息，请参阅 [MySQL 5.7.44 中的更改 \(2022-10-11\)](#)，正式发布。

目前支持的 Aurora MySQL 版本有 2.07.9、2.07.10、2.11.*、2.12.*、3.03.*、3.04.*、3.04.*、3.05.* 和 3.06.*。

你可以将现有的 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.12.2。你也可以将当前支持的任何 Aurora MySQL 版本中的快照还原到 Aurora MySQL 2.12.2。

如果您将 Aurora MySQL Global Database 升级到版本 2.12.* 并且开启了写入转发功能，则必须将主数据库集群和辅助数据库集群升级到完全相同的版本（包括补丁级别），才能继续使用写入转发功能。有关升级 Aurora Global Database 次要版本的更多信息，请参阅[次要版本升级](#)。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

修复了下面列出的安全问题和 CVE：

此版本包括 MySQL 5.7.44 之前的所有社区 CVE 修复。包括以下 CVE 修复程序：

- [CVE-2024-20963](#)
- [CVE-2023-39975](#)
- [CVE-2023-38545](#)

安全问题：

- 添加了一个修复程序，确保如果源支持加密连接，则无论设置如何，二进制日志副本都默认使用 SSL/TLS。MASTER_SSL

可用性改进：

- 修复了在写入器实例工作负载较高时可能导致只读副本实例无法成功启动的问题。
- 修复了由于与 Aurora 存储通信的组件存在缺陷而可能导致 Aurora MySQL 数据库写入器实例故障转移的问题。出现该缺陷的原因是 Aurora 存储实例软件更新后，数据库实例与底层存储之间的通信中断。
- 修复了在极少数情况下可能导致读取器实例重新启动的问题。
- 修复了特权用户可以修改与用户 `rds admin` 关联的[资源限制](#)的问题。如果设置不正确，这些资源限制可能会阻碍 RDS 监控代理监控数据库实例运行状况的能力，从而导致数据库不可用。

升级和迁移：

- 修复了尝试为从 Amazon RDS MySQL 5.7 迁移且包含不支持的存储过程的 Aurora MySQL 集群启动二进制日志复制时出现的问题。
- 在主要版本升级到 Aurora MySQL 版本 3 期间禁用了数据库事件调度程序。此更新有助于避免在主版本升级过程中因事件执行而对数据库进行任何更改。

集成了 MySQL 社区版本错误修复

此版本包括 5.7.44 之前的所有社区错误修复。有关更多信息，请参阅 [Aurora MySQL 2.x 数据库引擎更新和之前的所有 MySQL 错误修复](#)。

Aurora MySQL 版本 2 中不支持的功能

以下功能目前在 Aurora MySQL 版本 2 (与 MySQL 5.7 兼容) 中不受支持。

- 扫描批处理

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能：

- CREATE TABLESPACE SQL 语句
- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- X 协议

Aurora MySQL 数据库引擎更新 2023-12-28 (版本 2.12.1 ， 与 MySQL 5.7.40 兼容)

版本：2.12.1

Aurora MySQL 2.12.1 已正式发布。Aurora MySQL 2.12 版本与 MySQL 5.7.40 之前的版本兼容。有关社群变更的更多信息，请参阅 [Changes in MySQL 5.7.40 \(2022-10-11, General Availability\)](#)。

当前支持的 Aurora MySQL 版本有 2.07.*、2.11.*、2.12.*、3.01.*、3.02.*、3.03.*、3.04.* 和 3.05.*。

您可以将现有的 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.12.1。您也可以将当前支持的任何 Aurora MySQL 版本的快照还原到 Aurora MySQL 2.12.1。

如果您将 Aurora MySQL Global Database 升级到版本 2.12.* 并且开启了写入转发功能，则必须将主数据库集群和辅助数据库集群升级到完全相同的版本（包括补丁级别），才能继续使用写入转发功能。有关升级 Aurora Global Database 次要版本的更多信息，请参阅[次要版本升级](#)。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

修复了下面列出的安全问题和 CVE：

此版本包括 MySQL 5.7.44 之前的所有社区 CVE 修复。

对托管环境中的处理进行微调的修复和其他增强功能。其他 CVE 修复如下：

- [CVE-2023-38546](#)
- [CVE-2023-38545](#)
- [CVE-2023-22053](#)
- [CVE-2023-22028](#)
- [CVE-2023-22026](#)
- [CVE-2023-22015](#)
- [CVE-2022-24407](#)
- [CVE-2020-11105](#)
- [CVE-2020-11104](#)
- 修复了全文搜索 (FTS, Full-Text Search) 解析器插件对单字符令牌的处理（错误 35432973）
- 修复了在处理审核日志轮换时报告的事件可能无法写入审核日志的问题

新功能：

- 增加了对多线程二进制日志 (binlog) 复制的支持，即 binlog 副本上的 SQL 线程将尽可能并行应用二进制日志事件。查看 [Aurora 用户指南](#)，详细了解有助于优化多线程复制的配置选项。

可用性改进：

- 修复了使用并行查询的 Aurora MySQL 数据库实例在运行大量并发并行查询时可能会遇到数据库重启的问题。
- 修复了由审计日志记录线程导致的锁争用问题，该问题会导致 CPU 占用率过高和客户端应用程序超时。
- 修复了在尝试读取属于已删除表的数据库页面时可能导致数据库实例重启的问题。
- 修复了当写入器实例将数据库卷增大为 160 GB 的倍数时，可能会导致读取器实例重启的问题。
- 修复了锁定管理器中的问题，在处理两阶段提交时，如果隔离级别设置为 READ_COMMITTED 或 READ_UNCOMMITTED，并且使用 XA 事务或启用了二进制日志 (binlog)，该问题可能导致重启或失效转移。
- 修复了当数据库在内部系统表上创建或删除触发器时，如果写入器实例重启，则可能导致数据库集群不可用的问题。
- 修复了在数据库连接数量接近 max_connections 参数所设置的值时，可能导致数据库实例重新启动的问题。
- 修复了在包含全文索引的表上执行数据操作语言 (DML, Data Manipulation Language) 查询时，可能导致 Aurora 读取器实例重启的问题。
- 在此 Aurora MySQL 版本中未启用快速插入，这是因为在运行 INSERT INTO SELECT、和等查询时可能会出现不一致的问题。FROM 有关快速插入优化的更多信息，请参阅 [Amazon Aurora MySQL 性能增强](#)。

常规改进：

- 修复了在从 Aurora 集群卷读取数据时，可能由于暂时性网络问题导致并行查询失败的问题。
- 修复了与审计日志文件管理有关的问题，该问题可能导致无法访问日志文件以进行下载或轮换，在某些情况下还会增加 CPU 占用率。
- 修复了从低于 2.11.* 的版本升级后，小型只读副本实例的复制延迟可能会增加的问题。
- 修复了在查阅 [procs_priv 授权表](#) 以验证涉及存储例程的请求时，可能导致日志消息过多的问题。
- 修复了内存管理问题，该问题可能导致数据库实例在使用哈希联接优化执行查询时使用过多内存。
- 修复了使用写入转发时，可能会在 information_schema 和 performance_schema 全局状态表中生成错误的 Threads_running 变量值的问题。
- 修复了执行带分区表（在支持旧 ha_partition 分区处理程序的 MySQL 版本中创建）的 SELECT 语句，且查询计划程序选择了并行查询时，会导致数据库重启的问题。

- 修复了在启用了写入转发时，可能无法与数据库建立新的客户端连接的问题。
- 减少了二进制日志 (binlog) 复制在以下条件下的延迟：在没有 USE 命令定义的默认数据库的情况下，Aurora MySQL 二进制日志副本向源二进制日志文件执行 QUERY 事件写入。
- 修复了在innodb_flush_log_at_trx_commit参数未设置为 1 时可能导致错误报告CommitLatency CloudWatch 指标的问题。
- 修复了可能导致数据库连接在建立之前关闭的问题。此问题更有可能影响高速打开和关闭连接的数据库实例。
- 修复了当连接的二进制日志 (binlog) 使用者使用重复的二进制日志复制服务器 ID 时，可能导致数据库重启的问题。

集成了 MySQL 社区版本错误修复

除以下内容外，此版本还包括 5.7.40 及之前的所有社区错误修复。有关更多信息，请参阅 [Aurora MySQL 2.x 数据库引擎更新和之前的所有 MySQL 错误修复](#)。

- 修复了在有 SHOW PROCESSLIST 语句同时运行时，可能导致现有和新的远程连接停顿的问题（社区错误 34857411）
- 复制：某些二进制日志事件的处理并不总是正确的（错误 34617506）

Aurora MySQL 版本 2 中不支持的功能

以下功能目前在 Aurora MySQL 版本 2（与 MySQL 5.7 兼容）中不受支持。

- 扫描批处理

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能：

- CREATE TABLESPACE SQL 语句
- 组复制插件
- 增加的页面大小

- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- X 协议

Aurora MySQL 数据库引擎更新 2023-10-25 (版本 2.12.0.1 , 与 MySQL 5.7.40 兼容) 测试版

版本 : 2.12.0.1

Aurora MySQL 2.12.0.1 现已在以下地区推出 : 美国东部 (弗吉尼亚北部)、美国东部 (俄亥俄州)、美国西部 (加利福尼亚北部)、美国西部 (俄勒冈)、AWS GovCloud (美国东部) 和 AWS GovCloud (美国西部)。这是一个仅限安全修复的早期版本。随着下一个补丁版本 2.12.1 的发布 , 这些修复将在所有地区更广泛地部署。Aurora MySQL 2.12 版本与 MySQL 5.7.40 兼容。

当前支持的 Aurora MySQL 版本有 2.07.*、2.11.*、2.12.*、3.01.*、3.02.*、3.03.* 和 3.04.*。

您可以将现有的 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.12.0.1。您也可以将快照从当前支持的任何 Aurora MySQL 版本还原到 Aurora MySQL 2.12.0.1。

如果您将 Aurora MySQL Global Database 升级到版本 2.12.* , 则必须将主数据库集群和辅助数据库集群升级到完全相同的版本 (包括补丁级别)。有关升级 Aurora Global Database 次要版本的更多信息 , 请参阅[次要版本升级](#)。

如果您有任何疑问或疑虑 , 可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息 , 请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

改进

修复了下面列出的安全问题和 CVE :

此版本包括 MySQL 5.7.40 及之前的所有社区 CVE 修复。

- [CVE-2023-38545](#)

可用性改进：

- 在此 Aurora MySQL 版本中未启用快速插入，这是因为在运行 INSERT INTO SELECT、和等查询时可能会出现不一致的问题。FROM 有关快速插入优化的更多信息，请参阅 [Amazon Aurora MySQL 性能增强](#)。

Aurora MySQL 数据库引擎更新 2023-07-25 (版本 2.12.0 , 与 MySQL 5.7.40 兼容)

版本：2.12.0

Aurora MySQL 2.12.0 已正式发布。Aurora MySQL 2.12 版本与 MySQL 5.7.40 之前的版本兼容。有关社群变更的更多信息，请参阅 [Changes in MySQL 5.7.40 \(2022-10-11, General Availability\)](#)。

当前支持的 Aurora MySQL 版本有 2.07.*、2.11.*、2.12.*、3.01.*、3.02.* 和 3.03.*。

您可以将现有的 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.12.0。您也可以将快照从当前支持的任何 Aurora MySQL 版本还原到 Aurora MySQL 2.12.0。

如果您将 Aurora MySQL Global Database 升级到版本 2.12.* 并且开启了写入转发功能，则必须将主数据库集群和辅助数据库集群升级到完全相同的版本（包括补丁级别），才能继续使用写入转发功能。有关升级 Aurora Global Database 次要版本的更多信息，请参阅 [次要版本升级](#)。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得 [AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的 [升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

修复了下面列出的安全问题和 CVE：

此版本包括 MySQL 5.7.40 及之前的所有社区 CVE 修复。

- Aurora MySQL 使用的默认 SSL 密码已更新为将安全性较低的 DES-CBC3-SHA 值排除在了 [SSL_CIPHER](#) 数据库参数之外。如果您因移除 DES-CBC3-SHA 密码而遇到 SSL 连接问题，请使用以下列表中适用的安全密码：[配置密码套件以连接到 Aurora MySQL 数据库集群](#)。有关 MySQL 客户端[连接密码配置](#)的更多信息，请参阅 MySQL 文档。
- [CVE-2023-21963](#)
- [CVE-2023-21912](#)
- [CVE-2023-21840](#)
- [CVE-2023-0215](#)
- [CVE-2022-43551](#)
- [CVE-2022-37434](#)
- [CVE-2022-32221](#)
- [CVE-2021-36222](#)
- [CVE-2021-22926](#)
- [CVE-2021-2169](#)

可用性改进：

- 修复了数据库活动流事件加密中可能导致数据库重启的问题。
- 修复了在执行数据定义语言 (DDL) 查询时，如果发生数据库重启，则会导致重启失败的两个问题
- 修复了连接激增可能导致查询延迟增加或数据库实例重启的问题
- 修复了在极少数情况下，在写入器实例上同时执行大型更新操作或数据定义语言 (DDL) 工作负载并在 Aurora 副本上对同一组表进行读取操作时，可能导致 Aurora 副本重启的问题
- 修复了连接激增可能导致连接建立过程需要更长时间才能完成或因超时错误而失败的问题
- 修复了高级审计日志轮换可能会减少可用内存的问题，这可能会导致数据库实例重启
- 修复了在执行使用 Aurora 并行查询执行计划的查询时可能导致 Aurora MySQL 读取器实例重启的问题
- 修复了在使用全文搜索 (FTS) 索引的表上执行 OPTIMIZE TABLE 查询时，可能导致写入器实例重启的问题
- 修复了使用来自 Aurora 全局数据库辅助 AWS 区域的全局写入转发执行 SELECT FOR UPDATE 查询时，可能导致 Aurora 全球数据库主区域中的写入器实例重启的问题
- 修复了在转发的[隐式提交语句](#)遇到错误时可能导致使用全局写入转发的 Aurora 全局数据库辅助 AWS 区域读取器实例重启的问题

- 在此 Aurora MySQL 版本中未启用快速插入，这是因为在运行 INSERT INTO SELECT、和等查询时可能会出现不一致的问题。FROM 有关快速插入优化的更多信息，请参阅 [Amazon Aurora MySQL 性能增强](#)。

常规改进：

- 在二进制日志副本上引入了文件管理性能优化，以帮助减少写入中继日志文件时的争用
- 修复了可能导致在 information_schema 指标中无法正确报告 buffer_pool_read_requests 计数器的问题
- 修复了在执行 LOAD FROM S3 或 SELECT INTO S3 操作时可能导致本地存储空间填满的问题。该问题还可能导致更高的 CPU 使用率、由于内存不足而导致数据库重启，以及这些查询的延迟增加。
- 修复了在连接了多个二进制日志复制使用者时，使用二进制日志复制的数据库实例可能会遇到 CPU 使用率增加和连接失败的问题
- 修复了未填充 SSL 服务器状态变量的问题
- 修复了执行重复写入的数据操作语言 (DML) 语句可能导致错误日志记录过多和查询延迟增加的问题
- 已将时区定义升级到 IANA 2023c 版本
- 增加了对启用和禁用会话级二进制日志记录的支持。请参阅《Amazon Aurora 用户指南》中的 [存储过程 - 复制](#)
- 增加了对设置会话级二进制日志格式的支持。请参阅《Amazon Aurora 用户指南》中的 [存储过程 - 复制](#)
- 修复了将 aurora_disable_hash_join 参数设置为 1 或 ON 可能不会阻止优化程序使用哈希联接的问题
- 修复了一个涉及索引扫描的问题，即在执行 SELECT 查询时，如果使用 GROUP BY 子句且 aurora_parallel_query 参数设置为 ON，则可能会返回不准确的结果
- 修复了在极少数情况下，访问在写入器实例上同时运行大型更新或数据定义语言 (DDL) 操作的表时，可能会导致 Amazon Aurora 读取器实例重启的问题
- 修复了可能导致在 information_schema 指标中无法正确报告 buffer_pool_read_requests 计数器的问题
- 修复了在源系统变量 [server uuid](#) 缺失或值无效时可能导致二进制日志副本重启的问题
- 修复了阻止 InnoDB 统计信息过时的的问题，这有时会生成次优的查询执行计划，从而可能导致查询执行时间延长
- 修复了无论用户工作量如何，AuroraGlobalDBRPOLag CloudWatch 指标始终显示为零的问题

升级和迁移：

- 要执行将 Aurora Global Database 从 Aurora MySQL 版本 2.07 或 2.11 升级到 Aurora MySQL 版本 2.12 或更高版本的次要版本升级，请参阅[通过修改引擎版本升级 Aurora MySQL](#)。

集成了 MySQL 社区版本错误修复

除以下内容外，此版本还包括 5.7.40 及之前的所有社区错误修复。有关更多信息，请参阅 [Aurora MySQL 2.x 数据库引擎更新和之前的所有 MySQL 错误修复](#)。

- 修复了因后台 TLS 证书轮换而可能导致更高 CPU 使用率的问题（社区错误修复 #34284186）

Aurora MySQL 版本 2 中不支持的功能

以下功能目前在 Aurora MySQL 版本 2（与 MySQL 5.7 兼容）中不受支持。

- 扫描批处理。

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能：

- CREATE TABLESPACE SQL 语句
- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件

- 复制筛选
- X 协议

Aurora MySQL 数据库引擎更新 2024-03-26 (版本 2.11.5 , 兼容 MySQL 5.7.12)

版本 : 2.11 .5

Aurora MySQL 2.11.5 现已正式上市。Aurora MySQL 2.11 版本与 MySQL 5.7.12 兼容。有关社群变更的更多信息，请参阅 [Changes in MySQL 5.7.12 \(2016-04-11, General Availability\)](#)。

目前支持的 Aurora MySQL 版本是

2.07.9、2.07.10、2.11.*、2.12.*、3.01.*、3.02.*、3.02.*、3.03.*、3.04.*、3.04.*、3.04.*、3.05.* 和 3.06.*。

你可以将现有的 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.11.5。你也可以将当前支持的任何较低的 Aurora MySQL 版本 2 版本的快照还原到 Aurora MySQL 2.11.5。

如果您将 Aurora MySQL Global Database 升级到版本 2.11.*，则必须将主数据库集群和辅助数据库集群升级到完全相同的版本（包括补丁级别）。有关升级 Aurora Global Database 次要版本的更多信息，请参阅[次要版本升级](#)。

在执行到 Aurora MySQL 2.11.* 的就地引擎版本升级后，操作系统升级将立即自动应用于 db.r4、db.r5、db.t2 和 db.t3 数据库实例类上所有受影响的实例（如果这些实例是在旧的操作系统版本上运行的）。在多可用区数据库集群中，所有读取器实例首先应用操作系统升级。当第一个读取器实例的操作系统升级完成后，将进行失效转移，并升级前一个写入器实例。

Note

在主要版本升级期间，操作系统升级不会自动应用于 Aurora Global Database。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

改进

修复了安全问题和 CVE：

此版本中包含以下 CVE 修复程序：

- [CVE-2020-11104](#)
- [CVE-2020-11105](#)
- [CVE-2023-22015](#)
- [CVE-2023-22026](#)
- [CVE-2023-22028](#)
- [CVE-2023-22084](#)
- [CVE-2023-38545](#)
- [CVE-2023-38546](#)
- [CVE-2024-20963](#)

可用性改进：

- 修复了由于与 Aurora 存储通信的组件存在缺陷而导致 Aurora MySQL 写入器数据库实例可能发生故障的问题。该缺陷是由于软件更新后数据库实例与底层存储之间的通信中断所致。
- 修复了在极少数情况下可能导致读取器数据库实例重新启动的问题。
- 修复了由审计日志记录线程导致的锁争用问题，该问题会导致 CPU 占用率过高和客户端应用程序超时。

常规改进：

- 修复了从 Aurora 数据库集群卷读取数据时，由于临时网络问题可能导致并行查询失败的问题。
- 修复了与审核日志文件管理有关的问题，该问题可能导致无法访问日志文件进行下载或轮换，在某些情况下还会增加 CPU 使用率。
- 修复了使用写入转发时，可能会在 `information_schema` 和 `performance_schema` 全局状态表中生成错误 `Threads_running` 变量值的问题。

升级和迁移：

- 修复了无法在从 RDS for MySQL 5.7 迁移的 Aurora MySQL 数据库集群上启动二进制日志复制的问题。
- 在 Aurora MySQL 版本 3 的主要版本升级期间禁用了数据库事件调度程序。这有助于避免在主版本升级过程中通过事件执行对数据库进行任何更改。

Aurora MySQL 版本 2 中不支持的功能

以下功能目前在 Aurora MySQL 版本 2 (与 MySQL 5.7 兼容) 中不受支持。

- 扫描批处理。有关更多信息，请参阅 [Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16 \) \(已弃用 \)](#)。

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2023-10-17 (版本 2.11.4 , 与 MySQL 5.7.12 兼容)

版本 : 2.11.4

Aurora MySQL 2.11.4 已正式发布。Aurora MySQL 2.11 版本与 MySQL 5.7.12 兼容。有关社群变更的更多信息，请参阅 [Changes in MySQL 5.7.12 \(2016-04-11, General Availability\)](#)。

当前支持的 Aurora MySQL 版本有 2.07.9、2.07.10、2.11.*、2.12.*、3.01.*、3.02.*、3.03.* 和 3.04.*。

您可以将现有的 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.11.4。您也可以将快照从当前支持的任何 Aurora MySQL 版本还原到 Aurora MySQL 2.11.4。

如果您将 Aurora MySQL Global Database 升级到版本 2.11.*，则必须将主数据库集群和辅助数据库集群升级到完全相同的版本（包括补丁级别）。有关升级 Aurora Global Database 次要版本的更多信息，请参阅[次要版本升级](#)。

在执行到 Aurora MySQL 2.11.* 的就地引擎版本升级后，操作系统升级将立即自动应用于 db.r4、db.r5、db.t2 和 db.t3 数据库实例类上所有受影响的实例（如果这些实例是在旧的操作系统版本上运行的）。在多可用区数据库集群中，所有读取器实例首先应用操作系统升级。当第一个读取器实例的操作系统升级完成后，将进行失效转移，并升级前一个写入器实例。

Note

在主要版本升级期间，操作系统升级不会自动应用于 Aurora Global Database。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

改进

修复了下面列出的安全问题和 CVE：

- 修复了在处理审计日志轮换时报告的事件可能未写入审计日志的问题。
- [CVE-2022-24407](#)

可用性改进：

- 修复了使用并行查询的 Aurora MySQL 数据库实例在运行大量并发并行查询时可能会遇到数据库重启的问题。
- 修复了在执行 I/O 密集型读取工作负载时，可能导致数据库实例重启的问题。
- 修复了在尝试读取属于已删除表的数据库页面时可能导致数据库实例重启的问题。
- 修复了当写入器实例将数据库卷增大为 160 GB 的倍数时，可能会导致读取器实例重启的问题。
- 修复了当数据库在内部系统表上创建或删除触发器时，如果写入器实例重启，则可能导致数据库集群不可用的问题。
- 修复了在包含全文索引的表上执行数据操作语言 (DML) 查询时，可能导致读取器实例重启的问题。
- 修复了在执行使用 Aurora 并行查询执行计划的查询时可能导致读取器实例重启的问题。
- 修复了在使用全文搜索 (FTS) 索引的表上执行 OPTIMIZE TABLE 查询时，可能导致写入器实例重启的问题。
- 在此 Aurora MySQL 版本中未启用快速插入，这是因为在运行 INSERT INTO SELECT、和等查询时可能会出现不一致的问题。FROM 有关快速插入优化的更多信息，请参阅 [Amazon Aurora MySQL 性能增强](#)。

常规改进：

- 修复了从低于 2.11.* 的版本升级后，小型只读副本实例的复制延迟可能会增加的问题。
- 修复了在查阅 [procs_priv 授权表](#) 以验证涉及存储例程的请求时，可能导致日志消息过多的问题。
- 修复了内存管理问题，该问题可能导致数据库实例在使用哈希联接优化执行查询时使用过多内存。
- 修复了执行带分区表 (在支持旧 ha_partition 分区处理程序的 MySQL 版本中创建) 的 SELECT 语句，且查询计划程序选择了并行查询时，会导致数据库重启的问题。
- 修复了在启用了写入转发时，可能无法与数据库建立新的客户端连接的问题。
- 减少了二进制日志 (binlog) 复制在以下条件下的延迟：在没有 USE 命令定义的默认数据库的情况下，Aurora MySQL 二进制日志副本向源二进制日志文件执行 QUERY 事件写入。
- 修复了一个涉及索引扫描的问题，即在执行 SELECT 查询时，如果使用 GROUP BY 子句且 aurora_parallel_query 参数设置为 ON，则可能会返回不准确的结果。

- 增加了对启用和禁用会话级二进制日志记录的支持。请参阅《Amazon Aurora 用户指南》中的[存储过程 - 复制](#)。
- 修复了在源系统变量 `server_uuid` 缺失或值无效时可能导致二进制日志副本重启的问题。
- 增加了对设置会话级二进制日志格式的支持。请参阅《Amazon Aurora 用户指南》中的[存储过程 - 复制](#)。
- 修复了在 `innodb_flush_log_at_trx_commit` 参数未设置为 1 时可能导致错误报告 CommitLatency CloudWatch 指标的问题。
- 修复了阻止 InnoDB 统计信息过时的的问题，这有时会生成次优的查询执行计划，从而可能导致查询执行时间延长。
- 修复了当连接的二进制日志 (binlog) 使用者使用重复的二进制日志复制服务器 ID 时，可能导致数据库重启的问题。

集成了 MySQL 社区版本错误修复

除以下内容外，此版本还包括 5.7.12 及之前的所有社区错误修复。有关更多信息，请参阅 [Aurora MySQL 2.x 数据库引擎更新和之前的所有 MySQL 错误修复](#)。

- 复制：某些二进制日志事件的处理并不总是正确的。（错误 #34617506）
- 修复了因后台 TLS 证书轮换而可能导致更高 CPU 使用率的问题（社区错误修复 #34284186）。
- 在预编译语句中，某些类型的子查询可能会导致服务器退出。（错误 #33100586）

Aurora MySQL 版本 2 中不支持的功能

以下功能目前在 Aurora MySQL 版本 2 (与 MySQL 5.7 兼容) 中不受支持。

- 扫描批处理。有关更多信息，请参阅 [Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16 \) \(已弃用 \)](#)。

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2023-06-09 (版本 2.11.3 , 与 MySQL 5.7.12 兼容)

版本 : 2.11.3

Aurora MySQL 2.11.3 已正式发布。Aurora MySQL 2.11 版本与 MySQL 5.7.12 兼容。有关社群变更的更多信息，请参阅 [Changes in MySQL 5.7.12 \(2016-04-11, General Availability\)](#)。

当前支持的 Aurora MySQL 版本有 2.07.*、2.11.*、3.01.*、3.02.* 和 3.03.*。

您可以将现有的 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.11.3。您也可以将快照从当前支持的任何 Aurora MySQL 版本还原到 Aurora MySQL 2.11.3。

如果您将 Aurora MySQL Global Database 升级到版本 2.11.*，则必须将主数据库集群和辅助数据库集群升级到完全相同的版本（包括补丁级别）。有关升级 Aurora Global Database 次要版本的更多信息，请参阅[次要版本升级](#)。

在执行到 Aurora MySQL 2.11.* 的就地引擎版本升级后，操作系统升级将立即自动应用于 db.r4、db.r5、db.t2 和 db.t3 数据库实例类上所有受影响的实例（如果这些实例是在旧的操作系统版本上运行的）。在多可用区数据库集群中，所有读取器实例首先应用操作系统升级。当第一个读取器实例的操作系统升级完成后，将进行失效转移，并升级前一个写入器实例。

Note

在主要版本升级期间，操作系统升级不会自动应用于 Aurora Global Database。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

修复了下面列出的安全问题和 CVE：

- 更新了 Aurora MySQL 使用的默认 SSL 密码，将安全性较低的 DES-CBC3-SHA 值排除在了 [SSL_CIPHER](#) 数据库参数之外。如果您因移除 DES-CBC3-SHA 密码而遇到 SSL 连接问题，请使用此列表中适用的安全密码。[ConfiguringCipherSuites](#) 有关 MySQL 客户端 [连接密码配置](#) 的更多信息，请参阅 MySQL 文档。
- [CVE-2023-21963](#)
- [CVE-2023-21912](#)
- [CVE-2023-0215](#)
- [CVE-2022-43551](#)
- [CVE-2022-37434](#)

可用性改进：

- 修复了数据库活动流 (DAS) 事件加密中可能导致数据库重启的问题。
- 修复了在执行数据定义语言 (DDL) 查询时，如果发生数据库重启，则会导致重启失败的两个问题。
- 在此 Aurora MySQL 版本中未启用快速插入，这是因为在运行 INSERT INTO SELECT、和等查询时可能会出现不一致的问题。FROM 有关快速插入优化的更多信息，请参阅 [Amazon Aurora MySQL 性能增强](#)。

常规改进：

- 在二进制日志副本上引入了文件管理性能优化，以帮助减少写入中继日志文件时的争用。
- 修复了将 aurora_disable_hash_join 参数设置为 1 或 ON 可能不会阻止优化程序使用哈希联接的问题。

- 修复了可能导致在 `information_schema` 指标中无法正确报告 `buffer_pool_read_requests` 计数器的问题。
- 修复了在执行 `LOAD FROM S3` 或 `SELECT INTO S3` 操作时可能导致本地存储空间填满的问题。该问题还可能导致更高的 CPU 使用率、由于内存不足而导致数据库重启，以及这些查询的延迟增加。

Aurora MySQL 版本 2 中不支持的功能

以下功能目前在 Aurora MySQL 版本 2 (与 MySQL 5.7 兼容) 中不受支持。

- 扫描批处理。有关更多信息，请参阅[Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)](#)。

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- `CREATE TABLESPACE` SQL 语句

Aurora MySQL 数据库引擎更新 2023-03-24 (版本 2.11.2，兼容 MySQL 5.7.12)

版本：2.11.2

Aurora MySQL 2.11.2 已正式发布。Aurora MySQL 2.11 版本与 MySQL 5.7.12 兼容。有关社群变更的更多信息，请参阅 [Changes in MySQL 5.7.12 \(2016-04-11, General Availability\)](#)。

当前支持的 Aurora MySQL 版本有 2.07.*、2.11.*、3.01.*、3.02.* 和 3.03.*。

您可以将现有的 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.11.2。您也可以将快照从当前支持的任何 Aurora MySQL 版本还原到 Aurora MySQL 2.11.2。

如果您将 Aurora MySQL Global Database 升级到版本 2.11.*，则必须将主数据库集群和辅助数据库集群升级到完全相同的版本（包括补丁级别）。有关升级 Aurora Global Database 次要版本的更多信息，请参阅[次要版本升级](#)。

在执行到 Aurora MySQL 2.11.* 的就地引擎版本升级后，操作系统升级将立即自动应用于 db.r4、db.r5、db.t2 和 db.t3 数据库实例类上所有受影响的实例（如果这些实例是在旧的操作系统版本上运行的）。在多可用区数据库集群中，所有读取器实例首先应用操作系统升级。当第一个读取器实例的操作系统升级完成后，将进行失效转移，并升级前一个写入器实例。

Note

在主要版本升级期间，操作系统升级不会自动应用于 Aurora Global Database。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

常规改进：

- 修复了在连接了多个二进制日志复制使用者时，使用二进制日志复制的数据库实例可能会遇到 CPU 使用率增加和连接失败的问题。

- 修复了在升级到 Aurora MySQL 版本 2.11 后，如果主数据库写入器使用的是 Aurora MySQL 版本 2.10，则可能导致全局数据库辅助区域中的读取器实例不同步的问题。

可用性改进：

- 在此 Aurora MySQL 版本中未启用快速插入，这是因为在运行 INSERT INTO SELECT、和等查询时可能会出现不一致的问题。FROM 有关快速插入优化的更多信息，请参阅 [Amazon Aurora MySQL 性能增强](#)。

Aurora MySQL 版本 2 中不支持的功能

以下功能目前在 Aurora MySQL 版本 2 (与 MySQL 5.7 兼容) 中不受支持。

- 扫描批处理。有关更多信息，请参阅 [Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)](#)。

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2023-02-14 (版本 2.11.1 , 与 MySQL 5.7.12 兼容)

版本 : 2.11.1

Aurora MySQL 2.11.1 已正式发布。Aurora MySQL 2.11 版本与 MySQL 5.7.12 兼容。有关社群变更的更多信息，请参阅 [Changes in MySQL 5.7.12 \(2016-04-11, General Availability\)](#)。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.07.*、2.09.*、2.10.*、2.11.*、3.01.* 和 3.02.*。

您可以将现有的 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.11.1。对于运行 Aurora MySQL 版本 1 的集群，您可以直接将现有 Aurora MySQL 1.23 或更高版本的集群升级到 2.11.1。您也可以将快照从当前支持的任何 Aurora MySQL 版本还原到 Aurora MySQL 2.11.1。

如果您将 Aurora MySQL Global Database 升级到版本 2.11.* 并且开启了写入转发功能，则必须将主数据库集群和辅助数据库集群升级到完全相同的版本（包括补丁级别），才能继续使用写入转发功能。有关升级 Aurora Global Database 次要版本的更多信息，请参阅[次要版本升级](#)。

在执行到 Aurora MySQL 2.11.* 的就地引擎版本升级后，操作系统升级将立即自动应用于 db.r4、db.r5、db.t2 和 db.t3 数据库实例类上所有受影响的实例（如果这些实例是在旧的操作系统版本上运行的）。在多可用区数据库集群中，所有读取器实例首先应用操作系统升级。当第一个读取器实例的操作系统升级完成后，将进行失效转移，并升级前一个写入器实例。

Note

在主要版本升级期间，操作系统升级不会自动应用于 Aurora Global Database。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

修复了下面列出的安全问题和 CVE：

对托管环境中的处理进行微调的修复和其他增强功能。其他 CVE 修复如下：

- [CVE-2022-32221](#)
- [CVE-2021-36222](#)
- [CVE-2021-22926](#)
- [CVE-2021-2169](#)

可用性改进：

- 修复了连接激增可能导致查询延迟增加或数据库实例重启的问题。
- 修复了在极少数情况下，在写入器实例上同时执行大型更新操作或数据定义语言 (DDL) 工作负载并在 Aurora 副本上对同一组表进行读取操作时，可能导致 Aurora 副本重启的问题。
- 修复了连接激增可能导致连接建立过程需要更长时间才能完成或因超时错误而失败的问题。
- 修复了高级审计日志轮换可能会减少可用内存的问题，这可能会导致数据库实例重启。
- 在此 Aurora MySQL 版本中未启用快速插入，这是因为在运行 INSERT INTO SELECT、和等查询时可能会出现不一致的问题。FROM 有关快速插入优化的更多信息，请参阅 [Amazon Aurora MySQL 性能增强](#)。

常规改进：

- 修复了未填充 [SSL 服务器状态变量](#) 的问题。
- 修复了执行重复写入的数据操作语言 (DML) 语句可能导致错误日志记录过多和查询延迟增加的问题。

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持，但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [使用哈希联接优化大型 Aurora MySQL 联接查询](#)。

- 用于同步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息，请参阅“[Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)](#)”。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2022-10-25 (版本 2.11.0，与 MySQL 5.7.12 兼容) 此版本不适用于新创建。

版本：2.11.0

Aurora MySQL 2.11.0 已正式发布。Aurora MySQL 2.x 版本与 MySQL 5.7.12 兼容。有关社群变更的更多信息，请参阅 [Changes in MySQL 5.7.12 \(2016-04-11, General Availability\)](#)。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、2.11.*、3.01.* 和 3.02.*。

您可以将现有的 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.11.0。对于运行 Aurora MySQL 版本 1 的集群，您可以直接将现有 Aurora MySQL 1.23 或更高版本的集群升级到 2.11.0。您也可以将快照从当前支持的任何 Aurora MySQL 版本还原到 Aurora MySQL 2.11.0。

如果您将 Aurora MySQL Global Database 升级到版本 2.11.* 并且开启了写入转发功能，则必须将主数据库集群和辅助数据库集群升级到完全相同的版本（包括补丁级别），才能继续使用写入转发功能。有关升级 Aurora Global Database 次要版本的更多信息，请参阅[次要版本升级](#)。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

修复了下面列出的安全问题和 CVE：

对托管环境中的处理进行微调的修复和其他增强功能。其他 CVE 修复如下：

- [CVE-2022-21460](#)
- [CVE-2022-21451](#)
- [CVE-2022-21444](#)
- [CVE-2022-21417](#)
- [CVE-2022-21304](#)
- [CVE-2022-21303](#)
- [CVE-2022-21245](#)
- [CVE-2021-36222](#)
- [CVE-2021-28196](#)
- [CVE-2021-23841](#)

- [CVE-2021-22926](#)
- [CVE-2021-3449](#)
- [CVE-2021-2307](#)
- [CVE-2021-2226](#)
- [CVE-2021-2202](#)
- [CVE-2021-2194](#)
- [CVE-2021-2179](#)
- [CVE-2021-2178](#)
- [CVE-2021-2174](#)
- [CVE-2021-2171](#)
- [CVE-2021-2169](#)
- [CVE-2021-2166](#)
- [CVE-2021-2160](#)
- [CVE-2021-2154](#)

新功能：

- 随着 Aurora MySQL 版本 2.11 的发布，还提供了新的操作系统升级。建议您在升级到版本 2.11 后，将此待处理的操作系统更新应用于您的所有 Aurora MySQL 数据库实例。有关更多信息，请参阅[使用操作系统更新](#)。
- 新的动态配置选项 `innodb_deadlock_detect` 可用于禁用死锁检测。在高并发系统中，当许多线程等待同一个锁时，死锁检测可能会导致速度下降。有时，禁用死锁检测并在发生死锁时依赖 `innodb_lock_wait_timeout` 设置进行事务回滚可能更高效。（错误 #23477773）有关 InnoDB 死锁检测的更多信息，请参阅[MySQL 文档](#)。
- 已添加来自 MySQL 8.0 的 `UUID_TO_BIN`、`BIN_TO_UUID` 和 `IS_UUID` 函数。有关使用这些函数的更多信息，请参阅[MySQL Miscellaneous function](#)。
- 增加了对优化程序提示的支持，允许用户在每个表或每个查询的基础上启用或禁用 Aurora MySQL 并行查询。
 - [使用 Amazon Aurora MySQL 的并行查询](#)
 - [Aurora MySQL 提示](#)
- 移除了 R3 实例类型支持。
- 增加了对 R6i 实例的支持。

可用性改进：

- 修复了由于错误日志中写入的二进制日志文件和位置不正确而可能导致无法在数据库集群中进行跨区域逻辑复制的问题。在运行 DDL 语句后重启引擎时，可能会出现此问题。
- 修复了在极少数情况下，在写入器实例上运行访问控制列表 (ACL) 语句 (例如 GRANT 和 FLUSH) 时，可能会导致 Aurora 读取器实例重启的问题。此问题更有可能影响具有大量用户和 ACL 操作 (例如权限更改) 的读取器实例。
- 修复了在极少数情况下，当事务访问被另一个事务删除的行时，可能会导致写入器实例重启或失效转移的问题。
- 提高了全文短语搜索性能，显著缩短了带有全文索引的表中搜索短语所用的时间。
- 修复了写入器实例在重启后会因恢复缓慢而卡住并随后再次重启的问题。如果初次重启时，数据库中有大量未提交的行，就会发生此问题。
- 修复了在极少数情况下，当[死锁检测器线程](#)卡住时，由于信号灯等待时间过长而导致数据库服务器重启的问题。
- 修复了在极少数情况下，当 I/O 线程出现死锁时，由于信号灯等待时间过长而导致数据库重启的问题。
- 在此 Aurora MySQL 版本中未启用快速插入，这是因为在运行 INSERT INTO SELECT、和等查询时可能会出现不一致的问题。FROM 有关快速插入优化的更多信息，请参阅 [Amazon Aurora MySQL 性能增强](#)。

常规改进：

- 修复了在满足以下所有条件时可能导致数据库服务器重启的问题：
 - 在 SQL MODE 中禁用了 ALLOW_INVALID_DATES。
 - 数据库服务器正在处理的 INSERT、UPDATE、DELETE 或 SELECT 语句带有 DATETIME 类型的无效值 (月份不在 1 到 12 之间)。
- 修复了将 log-bin 设置为 OFF 时不遵守二进制日志保留期的问题，这会导致存储利用率高于预期。修复后，将根据您的保留期清除二进制日志。有关如何配置二进制日志保留期的更多信息，请参阅 [Aurora MySQL 用户指南](#)。
- 修复了在数据库实例上运行某些数据控制语言 (DCL) SQL 语句 (例如 GRANT、FLUSH PRIVILEGES 等) 时，可能会导致该实例上的可用内存减少的问题。频繁使用此类语句会导致可用内存不断减少，并可能导致数据库实例因 out-of-memory 问题而重新启动。在写入器实例上使用此类语句也可能导致读取器实例上的可用内存减少。

- 为从中继日志执行的读取引入了更大的读取缓冲区大小，以最大限度地减少读取 I/O 操作的数量，从而减少了 I/O 和 SQL 线程之间的争用。
- 修复了可能导致 `mysql.rds_rotate_slow_log` 存储过程失败并显示错误消息“Table 'mysql.slow_log_backup' doesn't exist”的问题。
- 修复了以下问题：由于只读副本必须从磁盘而不是查询缓存读取数据，因此查询缓存失效过多，从而导致只读副本的 CPU 使用率和延迟高于预期。
- 修复了允许用户在读取器实例上运行 `INSTALL PLUGIN` 和 `UNINSTALL PLUGIN` 命令的问题，该问题可能会导致 `LOCK_plugin`、`LOCK_system_variables_hash`、`LOCK_global_system_variables` 出现死锁。这些语句现在只能在数据库集群中的写入器实例上执行。
- 修复了在启用了二进制日志记录时，集群可能会遇到高于预期的提交延迟的问题。这会影响所有生成大型二进制日志事件（大小超过 500MB）的事务。
- 修复了可能导致 `INFORMATION_SCHEMA.INNODB_METRICS` 表中的 `trx_active_transactions` 指标值不正确的问题。
- 修复了在为大型事务执行回滚到保存点时，由于二进制日志文件变得不一致而导致逻辑复制停止的问题。
- 默认情况下，使用一致的掩码密钥在常规日志和审计日志中屏蔽凭证哈希。 `slow-query-log` 这可以通过 `aurora_mask_password_hashes_type` 参数进行配置。
- 修复了在客户观察到的事件中，零停机时间重启（ZDR）持续时间报告有误的问题。
- 修复了可能导致 [mysql_rds_import_binlog_ssl_material](#) 调用失败并显示 [MySQL server ERROR 1457](#) 的问题。
- 修复了转储线程初始化可能与清除二进制日志的线程发生死锁的问题。这可能会让活动的二进制日志文件停止轮换而继续增长，或者导致新的二进制日志副本连接出现问题。
- 修复了查询缓存可能会在 Aurora 只读副本上返回过时结果的问题。

集成了 MySQL 社区版本错误修复

除以下内容外，此版本还包括 5.7 及之前的所有社区错误修复。有关更多信息，请参阅 [Aurora MySQL 2.x 数据库引擎更新和之前的所有 MySQL 错误修复](#)。

- 修复了从性能架构语句事件表（例如，`events_statements_current`）中读取字符集信息的代码无法阻止同时写入该字符集信息的问题。因此，SQL 查询文本字符集可能无效，从而可能导致服务器退出。通过此修复，无效的字符集会导致 `SQL_TEXT` 列被截断并阻止服务器退出。（错误 #23540008）

- InnoDB：社区错误 #25189192、错误 #84038 修复的向后移植。修复了在将表移至其他架构的 RENAME TABLE 操作后，InnoDB 无法更新 INNODB_SYS_DATAFILES 数据字典表的问题。这会导致重启时出错，表明它找不到表空间数据文件。
- InnoDB：修复了以下问题：在添加新索引时，服务器会丢弃内部定义的外键索引，并试图使用在虚拟生成的列上定义的二级索引作为外键索引，从而导致服务器退出。InnoDB 现在允许外键约束引用在虚拟生成的列上定义的二级索引。（错误 23533396）
- 修复了两个会话同时执行 INSERT ... ON DUPLICATE KEY UPDATE 操作时，会生成死锁的问题。在元组的部分回滚期间，另一个会话可能会对其进行更新。此错误的修复撤销了对错误 #11758237、错误 #17604730 和错误 #20040791 的修复。（错误 #25966845）
- 社区错误 #27407480 修复的向后移植：修复了即使启用了 automatic_sp_privileges，也无法正确向例程创建者授予 EXECUTE 和 ALTER ROUTINE 权限的问题。
- 社区错误 #24671968 修复的向后移植：修复了以下问题：如果 WHERE 子句包含依赖子查询，表在选择列表中的列上有二级索引（后跟子查询中的列），且 GROUP BY 或 DISTINCT 允许查询使用松散索引扫描，则查询可能会产生错误的结果。
- 修复了如果对多个带外键的表发出多表删除语句，则复制会中断的问题。（错误 #80821）
- 修复了在特殊情况下，即使启用了 [slave_skip_errors](#)，某些从属服务器错误也不会被忽略的问题。如果打开和锁定表失败，或者在运行基于行的复制的服务器上进行字段转换失败，则该错误被视为严重错误，[slave_skip_errors](#) 状态将被忽略。该修复可确保在启用 [slave_skip_errors](#) 的情况下，应用事务期间报告的所有错误都得到正确处理。（错误 #70640、错误 #17653275）
- 修复了在 [SET PASSWORD](#) 语句从 MySQL 5.6 主服务器复制到 MySQL 5.7 从属服务器，或者从 [log_builtin_as_identified_by_password](#) 系统变量设置为 ON 的 MySQL 5.7 主服务器复制到 MySQL 5.7 从属服务器时，密码哈希本身在存储在从属服务器上之前也经过哈希处理的问题。该问题现已得到修复，复制的密码哈希按最初传递给从属服务器的原样存储。（错误 #24687073）
- 修复了由包裹在多级 JSON 数组和/或对象中的大型子文档组成的 JSON 值的序列化有时需要过长时间才能完成的问题。（错误 #23031146）
- 无法解析的语句（例如，由于语法错误）将不再写入慢速查询日志。（错误 #33732907）

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1（兼容 MySQL 5.6）中受支持，但这些功能目前在 Aurora MySQL 版本 2（兼容 MySQL 5.7）中不受支持。

- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。

- 用于同步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息，请参阅“[Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)](#)”。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2022-11-01 (版本 2.10.3) (已弃用)

版本：2.10.3

Aurora MySQL 2.10.3 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、2.11.*、3.01.* 和 3.02.*。

您可以将现有的 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.10.3。对于运行 Aurora MySQL 版本 1 的集群，您可以直接将现有 Aurora MySQL 1.23 或更高版本的集群升级到 2.10.3。您也可以将快照从当前支持的任何 Aurora MySQL 版本还原到 Aurora MySQL 2.10.3。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的 [升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

修复了下面列出的安全问题和 CVE：

对托管环境中的处理进行微调的修复和其他增强功能。其他 CVE 修复如下：

- [CVE-2022-21444](#)
- [CVE-2022-21344](#)
- [CVE-2022-21304](#)
- [CVE-2022-21245](#)
- [CVE-2021-36222](#)
- [CVE-2021-22926](#)

常规改进：

- 修复了在极少数情况下，当 [死锁检测器线程](#) 卡住时，由于信号灯等待时间过长而导致数据库服务器重启的问题。
- 修复了在数据库实例上运行某些数据控制语言 (DCL) SQL 语句 (例如 GRANT、FLUSH PRIVILEGES 等) 时，可能会导致该实例上的可用内存减少的问题。频繁使用此类语句会导致可用内存不断减少，并可能导致数据库实例因内存不足问题而重启。在写入器实例上使用此类语句也可能导致读取器实例上的可用内存减少。

- 修复了以下问题：当数据库实例的负载较重且启用了“wait/io/aurora_respond_to_client”performance_schema 等待事件时，对“performance_schema.events_waits_summary_global_by_event_name”表的查询速度可能变慢。
- 修复了在极少数情况下，由于二级索引上的约束违规而使事务部分回滚时，可能导致数据库服务器停止运行并重启的问题。
- 修复了在极少数情况下，当事务访问被另一个事务删除的行时，可能会导致写入器实例重启或失效转移的问题。
- 修复了在极少数情况下，当 I/O 线程出现死锁时，由于信号灯等待时间过长而可能导致数据库重启的问题。
- 修复了在极少数情况下，使用 Unix 套接字锁定文件时，可能导致只读副本在失效转移期间重启的问题。
- 修复了以下问题：由于只读副本必须从磁盘而不是查询缓存读取数据，因此查询缓存失效过多，从而导致只读副本的 CPU 使用率和延迟高于预期。

集成了 MySQL 社区版本错误修复

除以下内容外，此版本还包括 5.7 及之前的所有社区错误修复。有关更多信息，请参阅 [Aurora MySQL 2.x 数据库引擎更新和之前的所有 MySQL 错误修复](#)。

- 修复了从性能架构语句事件表（例如，events_statements_current）中读取字符集信息的代码无法阻止同时写入该字符集信息的问题。因此，SQL 查询文本字符集可能无效，从而可能导致服务器退出。通过此修复，无效的字符集会​​导致 SQL_TEXT 列被截断并阻止服务器退出。（错误 #23540008）
- 修复了当 UPDATE 需要一个主键大于 1024 个字节的临时表，而该表是使用 InnoDB 创建的时，服务器可能会退出的问题。（错误 #25153670）
- 修复了两个会话同时执行 INSERT ... ON DUPLICATE KEY UPDATE 操作时，会生成死锁的问题。在元组的部分回滚期间，另一个会话可能会对其进行更新。此错误的修复撤销了对错误 #11758237、错误 #17604730 和错误 #20040791 的修复。（错误 #25966845）

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1（兼容 MySQL 5.6）中受支持，但这些功能目前在 Aurora MySQL 版本 2（兼容 MySQL 5.7）中不受支持。

- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于异步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息，请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)”](#)。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2022-01-26 (版本 2.10.2) (已弃用)

版本：2.10.2

Aurora MySQL 2.10.2 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

您可以将现有的 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.10.0。对于运行 Aurora MySQL 版本 1 的集群，您可以直接将现有 Aurora MySQL 1.23 或更高版本的集群升级到 2.10.0。您也可以将快照从当前支持的任何 Aurora MySQL 版本还原到 Aurora MySQL 2.10.0。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

修复了下面列出的安全问题和 CVE：

对托管环境中的处理进行微调的修复和其他增强功能。其他 CVE 修复如下：

- [CVE-2021-36222](#)
- [CVE-2021-35624](#)
- [CVE-2021-35604](#)
- [CVE-2021-22926](#)
- [CVE-2021-2390](#)
- [CVE-2021-2389](#)
- [CVE-2021-2385](#)
- [CVE-2021-2356](#)
- [CVE-2019-17543](#)
- [CVE-2019-2960](#)

常规改进：

- 添加了性能优化，帮助缩短 24XL 实例类中的数据库 IO 延迟。
- 添加了对 ECDHE SSL 密码的支持。有关配置客户端使用这些 SSL 密码的更多信息，请参阅以下 MySQL 文档：[加密连接协议密码](#)
- 修复了 Aurora MySQL 与其他 AWS 服务（例如 Amazon S3、Amazon ML 和 AWS Lambda）集成的相关安全问题。
- 修复了当数据库具有大约超过 1GB 的用户和权限组合时，可能导致数据库实例重启失败的问题。
- 修复了并行查询的问题，该问题可能导致数据库在使用 GROUP BY 子句和包含范围谓词的 WHERE 子句执行查询时返回错误的分组或排序顺序。
- 修复了在将主要版本从 Aurora MySQL 1.x（与 MySQL 5.6 兼容）就地升级到 Aurora MySQL 2.x（与 MySQL 5.7 兼容）后导致 general_log 和 slow_log 表无法访问的问题。
- 修复了在极少数情况下，当数据库处于繁重工作负载期间查询 innodb_trx、innodb_locks 或 innodb_lockwaits 表时，可能导致数据库实例重新启动的问题。监控工具（例如 Performance Insights）可能会查询此类表格。
- 修复了在满足以下所有条件时将现有行的 TIMESTAMP 列的值更新为最新时间戳的问题：
 1. 该表存在触发器。
 2. 在具有 ON DUPLICATE KEY UPDATE 子句的表上执行 INSERT。
 3. 插入的行会导致 UNIQUE 索引或 PRIMARY KEY 中的重复值冲突。
 4. 一个或多个列为 TIMESTAMP 数据类型，原定设置值为 CURRENT_TIMESTAMP。
- 修复了在极少数情况下，可能阻止二进制日志副本连接到已启用二进制日志的实例的问题。
- 修复了在极少数情况下，在已启用二进制日志的实例上运行时，无法提交事务的问题。
- 修复了无法与已启用二进制日志的实例建立新连接的问题。
- 修复了在尝试零停机时间修补和重新启动导致本地存储填满时，可能导致内部日志记录过多的问题。
- 修复了在复制某些 DDL 和 DCL 语句时导致二进制日志副本停止并出现 HA_ERR_FOUND_DUPP_KEY 错误的问题。当源实例配置了 MIXED 二进制日志记录格式和 READ COMMITTED 或 READ UNCOMMITTED 隔离级别时，会出现此问题。
- 修复了启用多线程复制时二进制日志复制输入/输出线程无法跟上主实例的问题
- 修复了在极少数情况下，与数据库实例的大量活动连接可能导致错误报告 CloudWatch CommitLatency 指标的问题。
- 修复了执行 LOAD FROM S3 或 SELECT INTO S3 时，导致 Graviton 实例上的本地存储空间填满的问题。
- 修复了通过外键查询表且同时满足以下两个条件时，可能导致查询结果错误的问题：
 1. 已启用查询缓存

2. 已回滚对该表进行级联删除或更新的事务

- 修复了在极少数情况下可能导致 Aurora 读取器实例重新启动的问题。随着事务回滚次数增加，出现此问题的几率也会增加。
- 修复了打开和关闭会话时，性能架构中出现互斥锁“LOCK_epoch_id_master”的次数增加的问题。
- 修复了当大量事务同时对同一组行进行更新时，可能导致其工作负载出现越来越多的死锁的问题。
- 修复了在极少数情况下当数据库卷增加到 160 GB 的倍数时可能导致实例重新启动的问题。
- 修复了使用 LIMIT 子句执行 SQL 语句时可能导致数据库重新启动的并行查询问题。
- 修复了在极少数情况下，使用具有 READ COMMITTED 隔离级别的 XA 事务时可能导致数据库实例重新启动的问题。
- 修复了 Aurora 读取实例重启后，如果重启期间 DDL 工作负载繁重，实例可能会再次重启的问题。
- 修复了 Aurora 读取器复制滞后报告错误的问题。
- 修复了在极少数情况下会导致在内存中数据完整性检查失败时重新启动写入器实例的问题。
- 修复了在极少数情况下，即使会话已完成处理并处于空闲状态，Performance Insights (PI) 会话中的“数据库负载”图表也会错误显示为实际使用 CPU 的问题。
- 修复了在极少数情况下，使用并行查询处理查询时可能导致数据库服务器重新启动的问题。
- 修复了在极少数情况下，由于全局数据库复制期间出现争用情况，导致全局数据库主集群中的写入器实例重新启动的问题。
- 修复了数据库实例重启期间可能出现的会导致多次重启的问题。

集成了 MySQL 社区版本错误修复

- 修复了 InnoDB 中的一个问题：与表统计数据相关的代码错误导致 dict0stats.cc 源文件出现断言。（错误 #24585978）
- 修复了在线构建索引时，虚拟列上的二级索引损坏的问题。对于 [UPDATE](#) 语句，我们按以下方式修复此问题：如果索引记录的虚拟列值设置为 NULL，就从集群索引记录中生成此值。（错误 #30556595）
- 修复了 InnoDB 中删除标记行能够在部分回滚完成之前获取外部只读锁定的问题。外部只读锁定阻止了在部分回滚期间将隐式锁转换为显式锁，导致断言失败。（错误 #29195848）
- 修复了账户中的主机名为空时可能导致服务器行为异常的问题。（错误 #28653104）
- 修复了 InnoDB 中锁定等待期间查询中断导致错误的问题。（错误 #28068293）
- 修复了复制中当事务隔离级别设置为 [REPEATABLE READ](#) 时，交错事务有时可能会使从属应用程序死锁的问题。（错误 #25040331）

- 修复了锁定等待超时导致二进制日志副本停顿的问题。（错误 #27189701）

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1（兼容 MySQL 5.6）中受支持，但这些功能目前在 Aurora MySQL 版本 2（兼容 MySQL 5.7）中不受支持。

- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于异步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息，请参阅“[Aurora MySQL 数据库引擎更新 2017-12-11（版本 1.16）（已弃用）](#)”。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2021-10-21 (版本 2.10.1) (已弃用)

版本 : 2.10.1

Aurora MySQL 2.10.1 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

您可以将现有的 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.10.0。对于运行 Aurora MySQL 版本 1 的集群，您可以直接将现有 Aurora MySQL 1.23 或更高版本的集群升级到 2.10.0。您也可以将快照从当前支持的任何 Aurora MySQL 版本还原到 Aurora MySQL 2.10.0。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的 [升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

修复了下面列出的安全问题和 CVE：

对托管环境中的处理进行微调的修复和其他增强功能。其他 CVE 修复如下：

- [CVE-2021-2307](#)
- [CVE-2021-2226](#)
- [CVE-2021-2194](#)
- [CVE-2021-2174](#)
- [CVE-2021-2171](#)
- [CVE-2021-2169](#)
- [CVE-2021-2166](#)

- [CVE-2021-2160](#)
- [CVE-2021-2154](#)
- [CVE-2021-2032](#)
- [CVE-2021-2001](#)

可用性改进：

- 在未来的主要版本升级中添加了彻底关闭集群的能力。

常规改进：

- 修复了由于内部诊断日志文件中的信息性消息记录过多而导致读取器实例 CPU 占用过高的问题。
- 修复了在满足以下所有条件时将现有行的 TIMESTAMP 列的值更新为最新时间戳的问题：
 1. 该表存在触发器。
 2. 在具有 ON DUPLICATE KEY UPDATE 子句的表上执行 INSERT。
 3. 插入的行会导致 UNIQUE 索引或 PRIMARY KEY 中的重复值冲突。
 4. 一个或多个列为 TIMESTAMP 数据类型，原定设置值为 CURRENT_TIMESTAMP。
- 修复了版本 2.10.0 中引入的问题，该问题导致在某些情况下使用 json_merge 函数会引发错误代码。特别是，当 json_merge 函数在包含生成的列的 DDL 中使用，它可能会返回错误代码 1305。
- 修复了在极少数情况下，对只读副本上的事务读取视图验证大型对象的更新历史记录时，只读副本会重新启动的问题。
- 修复了在极少数情况下会导致在内存中数据完整性检查失败时重新启动写入器实例的问题。

集成了 MySQL 社区版本错误修复

- CURRENT_TIMESTAMP PRODUCES ZEROS IN TRIGGER。（错误 25209512）

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1（兼容 MySQL 5.6）中受支持，但这些功能目前在 Aurora MySQL 版本 2（兼容 MySQL 5.7）中不受支持。

- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。

- 用于异步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息，请参阅“[Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)](#)”。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2021-05-25 (版本 2.10.0) (已弃用)

版本：2.10.0

Aurora MySQL 2.10.0 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

您可以将现有的 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.10.0。对于运行 Aurora MySQL 版本 1 的集群，您可以直接将现有 Aurora MySQL 1.23 或更高版本的集群升级到 2.10.0。您也可以将快照从当前支持的任何 Aurora MySQL 版本还原到 Aurora MySQL 2.10.0。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

修复了下面列出的安全问题和 CVE：

对托管环境中的处理进行微调的修复和其他增强功能。其他 CVE 修复如下：

- [CVE-2021-23841](#)
- [CVE-2021-3449](#)
- [CVE-2020-28196](#)
- [CVE-2020-14790](#)
- [CVE-2020-14776](#)
- [CVE-2020-14567](#)
- [CVE-2020-14559](#)
- [CVE-2020-14553](#)
- [CVE-2020-14547](#)
- [CVE-2020-14540](#)
- [CVE-2020-14539](#)
- [CVE-2018-3251](#)
- [CVE-2018-3156](#)
- [CVE-2018-3143](#)

- [CVE-2016-5440](#)

新功能：

- Aurora MySQL 现在支持 db.t3.large 实例类。
- 二进制日志复制：
 - 引入了二进制日志 I/O 缓存，通过减少写入器线程和转储线程之间的争用来提高二进制日志性能。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[优化二进制日志复制](#)。
 - 在 [Aurora MySQL 版本 2.08](#) 中，我们引入了二进制日志 (binlog) 处理，以便在涉及非常大的事务时缩短崩溃恢复时间和提交时间延迟。对于启用了 GTID 的集群，现在支持这些改进。
- 改进了读取器实例的可用性：
 - 以前，写入器实例重新启动时，Aurora MySQL 集群中的所有读取器实例也会重新启动。从今天起，区域内读取器实例在写入器实例重启期间继续提供读取请求，从而提高了集群中的读取可用性。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[重启 Aurora MySQL 集群 \(版本 2.10 及更高版本\)](#)。

Important

升级到 Aurora MySQL 2.10 之后，重启写入器实例不会重启整个集群。如果您想重启整个集群，现在您可以在重启写入器实例后重启集群中的任何读取器实例。

- 改进了逻辑提前读取 (LRA) 技术请求的提前读取页面的读取性能。这是通过在发送到 Aurora 存储的单个请求中批处理多个页面读取来完成的。因此，使用 LRA 优化的查询的执行速度提高了 3 倍。
- 零停机时间重启和修补：
 - 改进了零停机时间重启 (ZDR) 和零停机时间修补 (ZDP)，以在更广泛的场景中启用 ZDR 和 ZDP，包括增加了对启用二进制日志记录的情况的支持。此外，还提高了对 ZDR 和 ZDP 事件的可见性。有关详细信息，请参阅《Amazon Aurora 用户指南》中的[Amazon Aurora MySQL 的零停机重启 \(ZDR\)](#) 和[使用零停机时间修补](#)。

可用性改进：

- 当数据库在先前中断的 DDL 活动期间创建了大量临时索引和表时，改进了启动速度。
- 修复了在中断的 DDL 操作崩溃恢复期间重复重启的多个相关问题，如 DROP TRIGGER、ALTER TABLE，特别是修改表中分区类型或分区数的 ALTER TABLE 问题。
- 修复了在数据库活动流 (DAS) 日志处理期间可能导致服务器重启的问题。

- 修复了处理系统表 ALTER 查询时打印错误消息的问题。

常规改进：

- 修复了查询缓存可能会在读取器实例上返回过时结果的问题。
- 修复了系统变量 `innodb_flush_log_at_trx_commit` 设置为 0 或 2 时某些 Aurora 提交指标未更新的问题。
- 修复了存储在查询缓存中的查询结果未被多语句事务刷新的问题。
- 修复了可能导致二进制日志文件上次修改的时间戳无法正确更新的问题。这可能会导致在达到客户配置的保留期之前提前清除二进制日志文件。
- 修复了崩溃恢复后从 InnoDB 报告错误的二进制日志文件名和位置的问题。
- 修复了如果 `binlog_checksum` 参数设置为 NONE，可能导致大型事务生成不正确的二进制日志事件的问题。
- 修复了在复制的事务包含 DDL 语句和大量行更改时导致二进制日志副本停止并显示错误消息的问题。
- 修复了删除表时导致读取器实例重启的问题。
- 修复了在尝试使用大型事务的二进制日志文件时导致开源连接器失败的问题。
- 修复了在具有大几何值的表上创建空间索引后，可能导致大几何列查询结果不正确的问题。
- 现在，数据库在重新启动期间重新创建临时表空间，这允许释放和回收关联的存储空间。
- 修复了无法在 Aurora 读取器实例上截断 `performance_schema` 表的问题。
- 修复了导致二进制日志副本停止并出现 `HA_ERR_KEY_NOT_FOUND` 错误的问题。
- 修复了在运行 `FLUSH TABLES WITH READ LOCK` 语句时导致数据库重新启动的问题。
- 修复了无法在 Aurora 读取器实例上使用用户级锁定功能的问题。

集成了 MySQL 社区版本错误修复

- 当事务隔离级别设置为 [REPEATABLE READ](#) 时，交错事务有时可能会使副本应用程序死锁。（错误 #25040331）
- 当存储过程包含引用一个视图的语句，而该语句又引用了另一个视图时，该过程不能多次成功调用。（错误 #87858、错误 #26864199）
- 对于具有多个 OR 条件的查询，优化程序现在的内存效率更高，并且超过 [range_optimizer_max_mem_size](#) 系统变量规定内存限制的可能性更小。此外，该变量的默认值已从 1,536,000 提高到 8,388,608。（错误 #79450、错误 #22283790）

- 复制：在由副本的 SQL 线程调用以从中继日志中读取下一个事件的 `next_event()` 函数中，SQL 线程在遇到错误（例如，由于中继日志关闭）时没有释放获取的 `relaylog.log_lock`，导致等待从中继日志中获取锁定的所有其他线程挂起。采用此修复方式后，锁定在这种情况下将于 SQL 线程离开函数之前释放。（错误 #21697821）
- 使用虚拟列修复 ALTER TABLE 内存损坏。（错误 #24961167、错误 #24960450）
- 复制：如果多线程副本需要处理大于该大小的事务，则无法使用 [slave_pending_jobs_size_max](#) 来配置较小的队列大小。任何大于 [slave_pending_jobs_size_max](#) 的数据包都会被拒绝并显示错误 ER_MTS_EVENT_BIGGER_PENDING_JOBS_SIZE_MAX，即使该数据包小于 [slave_max_allowed_packet](#) 设置的限制。采用此修复方式后，[slave_pending_jobs_size_max](#) 将成为软限制而不是硬限制。如果数据包的大小超过了 [slave_pending_jobs_size_max](#) 但小于 [slave_max_allowed_packet](#)，则该事务将保留直到所有副本工作线程都有空队列为止，然后进行处理。所有后续事务将一直保留到大型事务完成。因此，副本工作线程的队列大小可以受到限制，同时仍允许偶尔执行较大事务。（错误 #21280753、错误 #77406）
- 复制：使用多线程副本时，应用程序错误显示的工作线程 ID 数据与性能架构副本表中外部化的数据不一致。（错误 #25231367）
- 复制：在基于 GTID-[mode=on](#)、[-log-bin=off](#) 和使用 [-](#) 的复制副本上，当遇到应忽略的错误时 `slave-skip-errors`，未正确更新，导致与失去同步。Exec_Master_Log_Pos Exec_Master_Log_Pos Read_master_log_pos 如果未指定 GTID_NEXT，则从单语句事务回滚时，副本永远不会更新其 GTID 状态。Exec_Master_Log_Pos 不会更新，因为即使事务已完成，其 GTID 状态也会显示为其他状态。此修复消除了只在指定 GTID_NEXT 的情况下回滚事务时才更新 GTID 状态的限制。（错误 #22268777）
- 复制：禁用二进制日志记录时，部分失败的语句未正确使用自动生成或指定的 GTID。此修复可确保在禁用二进制日志记录时，部分失败的 [DROP TABLE](#)、部分失败的 [DROP USER](#) 或部分失败的 [DROP VIEW](#) 会分别使用相关的 GTID 并将其保存到 @@GLOBAL.GTID_EXECUTED 和 mysql.gtid_executed 表中。（错误 #21686749）
- 复制：由于检索不属于 MySQL 5.5 的 [server_uuid](#) 时出错，运行 MySQL 5.7 的副本无法连接到 MySQL 5.5 源。这是由于检索 `server_uuid` 的方法更改造成的。（错误 #22748612）
- 复制：以无提示方式跳过先前执行的 GTID 事务的 GTID 事务跳过机制对 XA 事务无效。（错误 #25041920）
- 因给出了错误的事务 ID 而失败的 [">XA ROLLBACK](#) 语句可以用正确的事务 ID 记录在二进制日志中，因此可以由复制副本执行操作。现在，在进行二进制日志记录之前会检查错误情况，不会记录失败的 XA ROLLBACK 语句。（错误 #26618925）
- 复制：如果使用未指定源日志文件名和源日志位置的 [CHANGE MASTER TO](#) 语句设置副本，则在发出 [START SLAVE](#) 之前将其关闭，然后使用选项 `set` 重新启动，则复制不会

- 开始。发生这种情况的原因是，在尝试恢复中继日志之前没有启动接收方线程，因此中继日志中没有可用的日志轮换事件来提供源日志文件名和源日志位置。在这种情况下，现在副本跳过中继日志恢复并记录警告，然后继续开始复制。（错误 #28996606、错误 #93397）
- 复制：在基于行的复制中，从含有 utf8mb3 列的表复制到使用 utf8mb4 字符集定义该列的具有相同定义的表时，会返回错误显示字段长度的消息。（错误 #25135304、错误 #83918）
 - 复制：当在使用 GTID 的复制副本上发出 [RESET SLAVE](#) 语句时，现有的中继日志文件将被清除，但是在清除该通道收到的 GTID 集之前，生成了替换的新中继日志文件。因此，以前的 GTID 集作为 PREVIOUS_GTIDS 事件写入新的中继日志文件，导致复制过程中出现致命错误，指出副本的 GTID 比源多，即使两个服务器的 gtid_executed 集都是空的。现在，当 RESET SLAVE 发出时，收到的 GTID 集将在生成新的中继日志文件之前清除，就不会发生这种情况了。（错误 #27411175）
 - 复制：当 GTID 用于复制时，那些包括导致解析错误 ([ER_PARSE_ERROR](#)) 语句的事务，无法通过注入具有相同 GTID 的空事务或替换事务的推荐方法手动跳过。此操作应导致副本将 GTID 识别为已使用，从而跳过共享其 GTID 的不需要的事务。但是，尽管无论如何都要跳过事务，在解析错误的情况下，由于语句在检查 GTID 以查看是否需要跳过之前解析了语句，因此复制应用程序线程由于解析错误而停止。采用此修复方式后，如果因为已经使用了 GTID 而需要跳过相关事务，则复制应用程序线程现在会忽略解析错误。请注意，此行为更改不适用于由 mysqlbinlog 生成的二进制日志输出组成的工作负载。在这种情况下，紧跟着跳过的事务之后出现解析错误的事务也会以无提示的方式跳过，因为该事务应引发错误。（错误 #27638268）
 - 复制：启用 SQL 线程以通过 GTID 跳过部分事务。（错误 #25800025）
 - 复制：向 WAIT_UNTIL_SQL_THREAD_AFTER_GTIDS() 提供负数或小数超时参数时，服务器出现意外行为。采用此修复方式：
 - 小数超时值按原样读取，没有舍入。
 - 如果服务器处于严格 SQL 模式，则拒绝负超时值并显示错误；如果服务器不处于严格 SQL 模式，则该值会使函数立即返回 NULL 而无需等待，然后发出警告。（错误 #24976304、错误 #83537）
 - 复制：如果 WAIT_FOR_EXECUTED_GTID_SET() 函数与包括小数部分 (例如 1.5) 的超时值一起使用，则转换逻辑中的错误意味着超时向下舍入到最接近的整秒，对于小于 1 秒的值 (例如，0.1)，则为零。转换逻辑现已被纠正，以便按照最初指定的方式应用超时值，而不舍入。感谢 Dirkjan Bussink 的贡献。（错误 #29324564、错误 #94247）
 - 启用 GTID 后，多语句事务中断开的 XA 事务上的 [XA COMMIT](#) 会引发断言。（错误 #22173903）
 - 复制：如果手动设置 [gtid_next](#) 值时为未知的事务标识符发出 [XA ROLLBACK](#) 语句，则在调试版本中引发了断言。如果 XA ROLLBACK 语句失败并出现错误，现在服务器不会尝试更新 GTID 状态。（错误 #27928837、错误 #90640）
 - 修复 CASE 子句中使用多个 ORDER BY 函数时排序顺序错误的问题（错误 #22810883）。

- 某些使用排序的查询可能会在优化期间访问未初始化的列并导致服务器退出。(错误 #27389294)

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持, 但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 哈希联接。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于同步调用 AWS Lambda 函数的本机函数。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息, 请参阅“[Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16 \) \(已弃用 \)](#)”。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容, 包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能, 使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上, 读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2021-11-12 (版本 2.09.3) (已弃用)

版本 : 2.09.3

Aurora MySQL 2.09.3 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

您可以将现有的 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.10.0。对于运行 Aurora MySQL 版本 1 的集群，您可以直接将现有 Aurora MySQL 1.23 或更高版本的集群升级到 2.10.0。您也可以将快照从当前支持的任何 Aurora MySQL 版本还原到 Aurora MySQL 2.10.0。

要使用较旧版本的 Aurora MySQL 创建集群，请通过 AWS Management Console、AWS CLI 或 Amazon RDS API 指定引擎版本。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的 [升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

安全修复：

对托管环境中的处理进行微调的修复和其他增强功能。其他 CVE 修复如下：

- [CVE-2021-23841](#)
- [CVE-2021-3712](#)
- [CVE-2021-3449](#)
- [CVE-2021-2307](#)
- [CVE-2021-2226](#)
- [CVE-2021-2174](#)

- [CVE-2021-2171](#)
- [CVE-2021-2169](#)
- [CVE-2021-2166](#)
- [CVE-2021-2154](#)
- [CVE-2021-2060](#)
- [CVE-2021-2032](#)
- [CVE-2021-2001](#)
- [CVE-2020-28196](#)
- [CVE-2020-14769](#)
- [CVE-2019-17543](#)
- [CVE-2019-2960](#)

可用性改进：

- 引入了一种优化，该优化可以减少对 `information_schema` 中的表执行的查询的争用。
- 增加对 ECDHE SSL 密码的支持。

常规改进：

- 修复了在极少数情况下会导致在内存中数据完整性检查失败时重新启动写入器实例的问题。
- 修复了在极少数情况下，在启用二进制日志记录的情况下，集群卷扩展时，可能导致数据库实例重新启动的问题。
- 修复了数据库实例重启期间的罕见争用情况，这种情况可能会导致多次重启。
- 修复了当数据库具有大量用户和权限组合时，可能导致数据库实例重启失败的问题。
- 修复了使用 LIMIT 子句执行 SQL 语句时可能导致数据库重新启动的并行查询问题。
- 修复了 aurora 复制滞后报告错误的问题。
- 修复了在将主要版本从 Aurora-MySQL 1.x (基于 MySQL 5.6) 就地升级到 Aurora-MySQL 2.x (基于 MySQL 5.7) 后导致 `general_log` 和 `slow_log` 表无法访问的问题。
- 修复了在极少数情况下，当数据库处于繁重工作负载期间查询 `innodb_trx`、`innodb_locks` 或 `innodb_lockwaits` 表时，可能会导致数据库实例重新启动的问题。监控工具和功能 (例如性能详情) 可能会查询此类表格。
- 修复了在执行“FLUSH TABLES WITH READ LOCK”SQL 语句时可能导致数据库实例重新启动的问题。

- 修复了在删除读取器实例期间 InnoDB 清除过程暂停导致历史记录列表长度暂时增加的问题。
- 修复了并行查询的问题，该问题可能会导致数据库在对包含虚拟列的表执行 SQL 语句时重新启动。
- 修复了并行查询的问题，该问题可能导致数据库在使用 GROUP BY 子句和包含范围谓词的 WHERE 子句执行查询时返回错误的分组或排序顺序。
- 修复了并行查询的问题，该问题在极少数情况下可能会导致数据库在使用 JSON 函数执行 SQL 语句时重新启动。
- 修复了在极少数情况下，由于全局数据库复制期间出现竞争条件，可能导致主全局数据库集群中的写入器实例重新启动的问题。
- 修复了在复制某些 DDL 和 DCL 语句时导致二进制日志副本停止并出现 HA_ERR_FOUND_DUPP_KEY 错误的问题。当源实例配置了 MIXED 二进制日志记录格式和 READ COMMITTED 或 READ UNCOMMITTED 隔离级别时，会出现此问题。
- 修复了以 READ COMMITTED 隔离级别使用 XA 事务时，在极少数情况下可能会导致数据库实例重新启动的问题。
- 修复了在满足以下所有条件时，现有行的 TIMESTAMP 列的值更新为最新时间戳的问题：1. 表存在触发条件；2. 对具有 ON DUPLICATE KEY UPDATE 子句的表执行 INSERT；3. 插入的行可能会导致 UNIQUE 索引或 PRIMARY KEY 中存在重复值冲突；以及 4. 一个或多个列为 TIMESTAMP 数据类型，且默认值为 CURRENT_TIMESTAMP。
- 修复了在极少数情况下，由于检查处理不正确，可能导致读取器实例重新启动的问题。
- 修复了当写入器实例增大数据库卷从而超出特定卷大小边界时，可能会导致读取器实例重新启动的问题。
- 修复了可能导致使用克隆的集群卷的数据库实例重启时间更长的的问题。
- 修复了在对写入器实例执行 TRUNCATE TABLE 操作后，数据库实例重启可能会失败一次或多次的问题。
- 修复了在极少数情况下可能会导致数据库实例重新启动的问题。
- 修复了在极少数情况下，当数据库卷增大为 160 GB 的倍数时，可能会导致写入器实例重新启动的问题。

集成了 MySQL 社区版本错误修复

- 错误 #23533396 - 添加新索引时，服务器丢弃了内部定义的外键索引，并试图使用在虚拟生成的列上定义的二级索引作为外键索引，从而导致服务器退出。InnoDB 现在允许外键约束引用在虚拟生成的列上定义的二级索引。

- 错误 #29550513 - 复制：WAIT_FOR_EXECUTED_GTID_SET() 函数中的锁定问题可能会导致服务器在某些情况下挂起。这个问题现在已得到纠正。

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持，但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 异步键预取 (AKP)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于异步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息，请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)”](#)。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件

- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2021-02-26 (版本 2.09.2) (已弃用)

版本 : 2.09.2

Aurora MySQL 2.09.2 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

您可以将现有的 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.09.2。对于运行 Aurora MySQL 版本 1 的集群，您可以直接将现有 Aurora MySQL 1.23 或更高版本的集群升级到 2.09.2。您也可以将快照从当前支持的任何 Aurora MySQL 版本还原到 Aurora MySQL 2.09.2。

要使用较旧版本的 Aurora MySQL 创建集群，请通过 AWS Management Console、AWS CLI 或 Amazon RDS API 指定引擎版本。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

新功能：

- Aurora MySQL 集群现在支持以下由基于 ARM 的 AWS Graviton2 处理器提供支持的 EC2 R6g 实例：r6g.large、r6g.xlarge、r6g.2xlarge、r6g.4xlarge、r6g.8xlarge、r6g.12xlarge、r6g.16xlarge。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[Aurora 数据库实例类](#)。

安全修复：

对托管环境中的处理进行微调的修复和其他增强功能。其他 CVE 修复如下：

- [CVE-2020-14775](#)
- [CVE-2020-14793](#)
- [CVE-2020-14765](#)
- [CVE-2020-14769](#)
- [CVE-2020-14812](#)
- [CVE-2020-14760](#)
- [CVE-2020-14672](#)
- [CVE-2020-14790](#)
- [CVE-2020-1971](#)

可用性改进：

- 修复了 2.09.0 中的问题，该问题可能导致集群存储卷扩展期间写入延迟更长。
- 修复了动态调整大小功能中可能导致 Aurora 只读副本重启的问题。
- 修复了在从 1.23.* 升级到 2.09.* 期间可能导致更长停机时间的问题。
- 修复了在页面预取请求期间 DDL 或 DML 可能导致引擎重启的问题。
- 修复了在复制的事务包含 DDL 语句和大量行更改时导致二进制日志副本停止并显示错误消息的问题。
- 修复了在 mysql time_zone 表上复制 DDL 事件时充当二进制日志副本的数据库可能重启的问题。
- 修复了如果 binlog_checksum 参数设置为 NONE，可能导致大型事务生成不正确的二进制日志事件的问题。
- 修复了导致二进制日志副本停止并出现 HA_ERR_KEY_NOT_FOUND 错误的问题。
- 提高了整体稳定性。

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持，但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 异步键预取 (AKP)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于异步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息，请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)”](#)。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2020-12-11 (版本 2.09.1) (已弃用)

版本：2.09.1

Aurora MySQL 2.09.1 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

您可以将现有的 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.09.1。对于运行 Aurora MySQL 版本 1 的集群，您可以直接将现有 Aurora MySQL 1.23 或更高版本的集群升级到 2.09.1。您也可以将快照从当前支持的任何 Aurora MySQL 版本还原到 Aurora MySQL 2.09.1。

要使用较旧版本的 Aurora MySQL 创建集群，请通过 AWS Management Console、AWS CLI 或 RDS API 指定引擎版本。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的 [升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

安全修复：

对托管环境中的处理进行微调的修复和其他增强功能。其他 CVE 修复如下：

- [CVE-2020-14567](#)
- [CVE-2020-14559](#)
- [CVE-2020-14553](#)
- [CVE-2020-14547](#)
- [CVE-2020-14540](#)
- [CVE-2020-2812](#)
- [CVE-2020-2806](#)
- [CVE-2020-2780](#)
- [CVE-2020-2765](#)
- [CVE-2020-2763](#)

- [CVE-2020-2760](#)
- [CVE-2020-2579](#)

不兼容的更改：

此版本新增了影响 `mysqldump` 命令行为的权限更改。用户必须具有访问 `PROCESS` 表的 `INFORMATION_SCHEMA.FILES` 权限。要在不进行任何更改的情况下运行 `mysqldump` 命令，请向 `PROCESS` 命令连接的数据库用户授予 `mysqldump` 权限。您也可以运行带 `mysqldump` 选项的 `--no-tablespaces` 命令。使用该选项，`mysqldump` 输出不包含任何 `CREATE LOGFILE GROUP` 或 `CREATE TABLESPACE` 语句。在这种情况下，`mysqldump` 命令不访问 `INFORMATION_SCHEMA.FILES` 表，您也无需授予 `PROCESS` 权限。

可用性改进：

- 修复了当数据库引擎在从网络读取或写入网络时遇到错误时可能导致客户端会话挂起的问题。
- 修复了 2.09.0 中引入的动态调整大小功能中的内存泄漏问题。

全局数据库：

- 修复了当主区域写入器使用较旧版本时，全局数据库辅助区域的副本可能会在升级到版本 2.09.0 时重新启动的多个问题。

集成了 MySQL 社区版本错误修复

- 复制：当事务隔离级别设置为 [REPEATABLE READ](#) 时，交错事务有时可能会使从属应用程序死锁。（错误 #25040331）
- 对于 [TIMESTAMP](#) 或 [DATETIME](#) 列默认值为 [CURRENT_TIMESTAMP](#) 的表，列可以在表具有 `BEFORE INSERT` 触发条件时初始化为 `0000-00-00 00:00:00`。（错误 #25209512、错误 #84077）
- 对于 `VALUES` 列表使用包含联接的子查询为第二行或后面的行生成值的 [INSERT](#) 语句，服务器可能会在解析所需的特权失败后退出。（错误 #23762382）

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1（兼容 MySQL 5.6）中受支持，但这些功能目前在 Aurora MySQL 版本 2（兼容 MySQL 5.7）中不受支持。

- 异步键预取 (AKP)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于异步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息，请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)”](#)。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2020-09-17 (版本 2.09.0) (已弃用)

版本：2.09.0

Aurora MySQL 2.09.0 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

您可以将快照从 Aurora MySQL 1.23.* 还原到 Aurora MySQL 2.09.0。您还可以选择将现有 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.09.0。不能将现有的 Aurora MySQL 1.23.* 集群直接升级到 2.09.0，但是可以将其快照还原到 Aurora MySQL 2.09.0。

Important

此版本中对 Aurora 存储的改进将可用升级路径限制为从 Aurora MySQL 1.* 到 Aurora MySQL 2.09.. 将 Aurora MySQL 1.* 集群升级到 2.09 时，您必须从 Aurora MySQL 1.23 升级。

要使用较旧版本的 Aurora MySQL 创建集群，请通过 AWS Management Console、AWS CLI 或 RDS API 指定引擎版本。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的 [升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

新功能：

- 通过此版本，您可以创建具有最多 128 TiB 存储空间的 Amazon Aurora MySQL 数据库实例。新的存储空间限制比之前的 64 TiB 有所增加。128 TiB 存储大小支持更大的数据库。小型实例大小（db.t2 或 db.t3）不支持此功能。由于 [InnoDB 具有页面大小为 16 KB 的限制](#)，因此单个表空间无法增长到 64 TiB 以上。

当集群卷大小接近 128 TiB 时，Aurora 发出警报，以便您可以在达到大小限制之前采取措施。警报显示在 mysql 日志和 AWS Management Console 的 RDS 事件中。

- 现在，您可以通过更改数据库集群参数 `aurora_parallel_query` 的值来为现有集群打开或关闭并行查询。创建集群时，不需要使用 `parallelquery` 参数的 `--engine-mode` 设置。

并行查询现在已扩展到 Aurora MySQL 可用的所有区域。

对于在 Aurora 集群中升级和启用并行查询的过程，还有许多其他功能增强和更改。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon Aurora MySQL 的并行查询](#)。

- Aurora 动态调整集群存储空间的大小。通过动态调整大小，当您从 Aurora 数据库集群中删除数据时，该数据库集群的存储空间会自动减少。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[存储扩展](#)。

Note

动态调整大小功能正在分阶段部署到 Aurora 可用的AWS区域。根据集群所在的区域，此功能可能尚不可用。有关更多信息，请参阅[新增功能公告](#)。

高优先级修复：

- 逆向移植社区错误 #27659490：SELECT USING DYNAMIC RANGE AND INDEX MERGE USE TOO MUCH MEMORY (OOM)
- 错误 #26881508：MYSQL #1: DISABLE_ABORT_ON_ERROR IN AUTH_COMMON.H
- 逆向移植社区错误 #24437124：POSSIBLE BUFFER OVERFLOW ON CREATE TABLE
- 逆向移植错误 #27158030：INNODB ONLINE ALTER CRASHES WITH CONCURRENT DML
- 错误 #29770705：SERVER CRASHED WHILE EXECUTING SELECT WITH SPECIFIC WHERE CLAUSE
- 逆向移植错误 #26502135：MYSQLD SEGFAULTS IN MDL_CONTEXT::TRY_ACQUIRE_LOCK_IMPL
- 逆向移植社区错误 #26935001：ALTER TABLE AUTO_INCREMENT TRIES TO READ INDEX FROM DISCARDED TABLESPACE
- 错误 #28491099：[FATAL] MEMORY BLOCK IS INVALID | INNODB: ASSERTION FAILURE: UT0UT.CC:670
- 错误 #30499288：GCC 9.2.1 REPORTS A NEW WARNING FOR OS_FILE_GET_PARENT_DIR
- 错误 #29952565，其中 MYSQLD GOT SIGNAL 11 WHILE EXECUTING A QUERY(UNION + ORDER BY + SUB-QUERY)
- 错误 #30628268：OUT OF MEMORY CRASH

- 错误 #30441969 : 错误 #29723340 : MYSQL SERVER CRASH AFTER SQL QUERY WITH DATA ?AST
- 错误 #30569003 : 5.7 REPLICATION BREAKAGE WITH SYNTAX ERROR WITH GRANT MANAGEMENT
- 错误 #29915479 : RUNNING COM_REGISTER_SLAVE WITHOUT COM_BINLOG_DUMP CAN RESULTS IN SERVER EXIT
- 错误 #30569003 : 5.7 REPLICATION BREAKAGE WITH SYNTAX ERROR WITH GRANT MANAGEMENT
- 错误 #29915479 : RUNNING COM_REGISTER_SLAVE WITHOUT COM_BINLOG_DUMP CAN RESULTS IN SERVER EXIT
- 错误 #20712046 : SHOW PROCESSLIST AND PERFORMANCE_SCHEMA TABLES DO NOT MASK PASSWORD FROM QUERY
- 逆向移植错误 #18898433 : EXTREMELY SLOW PERFORMANCE WITH OUTER JOINS AND JOIN BUFFER (在 5.7.21 中已修复)。在使用联接缓冲 (例如 , 使用块嵌套循环算法) 的情况下 , 带有许多左联接的查询会很慢。 (错误 #18898433、错误 #72854)
- 逆向移植错误 #26402045 : MYSQLD CRASHES ON QUERY (在 MySQL 5.7.23 中已修复)。某些情况下的子查询具体化可能会导致服务器退出。这些查询现在会产生建议禁止具体化的错误。 (错误 #26402045)
- [从 MySQL 逆向移植] 不允许 rdsadmin 以外的用户更新读取器副本中的 pfs 表。
- 修复客户无法更新读取器副本中的 perfschema 的问题
- 错误 #26666274 : 性能架构缓冲区容器中的无限循环
- [错误 #26997096](#) : relay_log_space 值不以同步的方式更新 , 因此其值有时大大高于中继日志使用的实际磁盘空间。
- 错误 #25082593 : FOREIGN KEY VALIDATION DOESN'T NEED TO ACQUIRE GAP LOCK IN READ COMMITTED
- [CVE-2019-2731](#)
- [CVE-2018-2645](#)
- [CVE-2019-2581](#)
- [CVE-2018-2787](#)
- [CVE-2019-2482](#)
- [CVE-2018-2640](#)
- [CVE-2018-2784](#)

- [CVE-2019-2628](#)
- [CVE-2019-2911](#)
- [CVE-2019-2628](#)
- [CVE-2018-3284](#)
- [CVE-2018-3065](#)
- [CVE-2019-2537](#)
- [CVE-2019-2948](#)
- [CVE-2019-2434](#)
- [CVE-2019-2420](#)

可用性改进：

- 默认情况下启用锁定管理器 ABA 修复。
- 修复了锁管理器中的一个问题，其中，竞争条件可能导致两个事务共享一个锁，从而导致数据库重新启动。
- 修复了使用压缩行格式创建临时表时可能导致重新启动的问题。
- 修复 16XL 和 24XL 实例上 `table_open_cache` 的默认值，此值可能导致在大型实例类 (R4/R5-16XL、R5-12XL、R5-24XL) 上发生重复的故障转移和较高的 CPU 利用率。这影响了 2.07.x。
- 修复了以下问题：当 S3 备份不包含 `mysql.host` 表时，将集群从 Amazon S3 还原到 Aurora MySQL 版本 2.08.0 所需时间比预期时间长。
- 修复了由于更新具有二级索引的虚拟列而可能导致重复故障转移的问题。
- 修复了一个与事务锁内存管理 (即，长时间运行的写事务导致数据库重新启动) 相关的问题。
- 修复了在检查安全点以进行修补时，引擎可能会在零停机时间修补过程中崩溃的多个问题。
- 修复了跳过临时表的重做日志记录的问题，此问题之前导致崩溃。
- 修复了锁管理器中终止连接/查询与已终止会话之间的竞争条件。
- 修复了以下问题：如果数据库是二进制日志副本并通过 MySQL `time_zone` 表接收 DDL 事件，则此数据库可能会崩溃。

全局数据库：

- 辅助区域中的 MySQL `INFORMATION_SCHEMA.REPLICA_HOST_STATUS` 视图现在显示属于该区域的副本的条目。

- 修复了若干个意外查询失败：在主区域和辅助区域之间发生临时网络连接问题之后，导致在全局数据库辅助区域中可能发生这些意外查询失败。

-

并行查询：

- 修复了并行查询的 EXPLAIN 计划，这对于简单的单表查询是不正确的。
- 修复了启用并行查询时可能发生的自死锁。

常规改进：

- 导出到 S3 现在支持 ENCRYPTION 关键字。
- `aurora_binlog_replication_max_yield_seconds` 参数现在的最大值为 36,000。之前接受的最大值为 45。仅当参数 `aurora_binlog_use_large_read_buffer` 设置为 1 时，此参数才有效。
- 已将执行 MIXED 时的行为更改为将 `binlog_format ROW` 映射到 STATEMENT，而不是映射到 LOAD DATA FROM INFILE | S3。
- 修复了以下问题：当主服务器执行了 LOAD DATA FROM S3 且 `binlog_format` 设置为 STATEMENT 时，连接到 Aurora MySQL 二进制日志主服务器的二进制日志副本可能显示不完整的数据。
- 已将审计系统变量 `server_audit_incl_users` 和 `server_audit_excl_users` 允许的最大长度从 1024 字节增加到 2000 字节。
- 修复了以下问题：如果当前连接数大于设置的值，在降低参数组中的 `max_connections` 参数时，用户可能会失去对数据库的访问权限。
- 修复了数据活动流中单引号和反斜杠未正确转义的问题。

集成了 MySQL 社区版本错误修复

- 错误 #27659490：SELECT USING DYNAMIC RANGE AND INDEX MERGE USE TOO MUCH MEMORY(OOM)
- 错误 #26881508：MYSQL #1: DISABLE_ABORT_ON_ERROR IN AUTH_COMMON.H
- 错误 #24437124：POSSIBLE BUFFER OVERFLOW ON CREATE TABLE
- 错误 #27158030：INNODB ONLINE ALTER CRASHES WITH CONCURRENT DML

- 错误 #29770705 : SERVER CRASHED WHILE EXECUTING SELECT WITH SPECIFIC WHERE CLAUSE
- 错误 #26502135 : MYSQLD SEGFALTS IN MDL_CONTEXT::TRY_ACQUIRE_LOCK_IMPL
- 错误 #26935001 : ALTER TABLE AUTO_INCREMENT TRIES TO READ INDEX FROM DISCARDED TABLESPACE
- 错误 #28491099 : [FATAL] MEMORY BLOCK IS INVALID | INNODB: ASSERTION FAILURE: UT0UT.CC:670
- 错误 #30499288 : GCC 9.2.1 REPORTS A NEW WARNING FOR OS_FILE_GET_PARENT_DIR
- 错误 #29952565 : 其中 MYSQLD GOT SIGNAL 11 WHILE EXECUTING A QUERY(UNION + ORDER BY + SUB-QUERY)
- 错误 #30628268 : OUT OF MEMORY CRASH
- 错误 #30441969 : 错误 #29723340 : MYSQL SERVER CRASH AFTER SQL QUERY WITH DATA ?AST
- 错误 #30569003 : 5.7 REPLICATION BREAKAGE WITH SYNTAX ERROR WITH GRANT MANAGEMENT
- 错误 #29915479 : RUNNING COM_REGISTER_SLAVE WITHOUT COM_BINLOG_DUMP CAN RESULTS IN SERVER EXIT
- 错误 #30569003 : 5.7 REPLICATION BREAKAGE WITH SYNTAX ERROR WITH GRANT MANAGEMENT
- 错误 #29915479 : RUNNING COM_REGISTER_SLAVE WITHOUT COM_BINLOG_DUMP CAN RESULTS IN SERVER EXIT
- 错误 #20712046 : SHOW PROCESSLIST AND PERFORMANCE_SCHEMA TABLES DO NOT MASK PASSWORD FROM QUERY
- 错误 #18898433 : EXTREMELY SLOW PERFORMANCE WITH OUTER JOINS AND JOIN BUFFER (在 5.7.21 中已修复)
- 错误 #26402045 : MYSQLD CRASHES ON QUERY (在 MySQL 5.7.23 中已修复)
- 错误 #23103937 : PS_TRUNCATE_ALL_TABLES() 在 SUPER_READ_ONLY 模式下无法运行
- 错误 #26666274 : 性能架构缓冲区容器中的无限循环
- 错误 #26997096 : relay_log_space 值不以同步的方式更新，因此其值有时大大高于中继日志使用的实际磁盘空间。 (<https://github.com/mysql/mysql-server/commit/78f25d2809ad457e81f90342239c9bc32a36cdfa>)
- 错误 #25082593 : FOREIGN KEY VALIDATION DOESN'T NEED TO ACQUIRE GAP LOCK IN READ COMMITTED

- 错误 #24764800 : REPLICATION FAILING ON SLAVE WITH XAER_RMFAIL ERROR。
- 错误 #81441 : WARNING ABOUT LOCALHOST WHEN USING SKIP-NAME-RESOLVE。

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持, 但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 异步键预取 (AKP)。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于异步调用 AWS Lambda 函数的本机函数。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息, 请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16 \) \(已弃用 \)”](#)。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容, 包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能, 使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上, 读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能 :

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件

- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2022-01-06 (版本 2.08.4) (已弃用)

版本 : 2.08.4

Aurora MySQL 2.08.4 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

您可以将现有的 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.10.0。对于运行 Aurora MySQL 版本 1 的集群，您可以直接将现有 Aurora MySQL 1.23 或更高版本的集群升级到 2.10.0。您也可以将快照从当前支持的任何 Aurora MySQL 版本还原到 Aurora MySQL 2.10.0。

要使用较旧版本的 Aurora MySQL 创建集群，请通过 AWS Management Console、AWS CLI 或 Amazon RDS API 指定引擎版本。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的 [升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

安全修复和常规改进：

- 修复了 Aurora MySQL 与其他 AWS 服务（例如 Amazon S3、Amazon ML 和 AWS Lambda）集成的相关安全问题。

- 修复了在满足以下所有条件时，现有行的 TIMESTAMP 列的值更新为最新时间戳的问题：1. 表存在触发条件；2. 对具有 ON DUPLICATE KEY UPDATE 子句的表执行 INSERT；3. 插入的行可能会导致 UNIQUE 索引或 PRIMARY KEY 中存在重复值冲突；以及 4. 一个或多个列为 TIMESTAMP 数据类型，且默认值为 CURRENT_TIMESTAMP。
- 修复了在极少数情况下，内存中数据完整性检查失败时导致写入器实例重新启动的问题。
- 修复了使用 LIMIT 子句执行 SQL 语句时可能导致数据库重新启动的并行查询问题。

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持，但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 异步键预取 (AKP)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于异步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息，请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)”](#)。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件

- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2020-11-12 (版本 2.08.3) (已弃用)

版本 : 2.08.3

Aurora MySQL 2.08.3 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

您现在可以将现有的 Aurora MySQL 2.* 数据库集群直接升级到 Aurora MySQL 2.08.3。您现在可以将现有的 Aurora MySQL 1.* 集群直接升级到 2.07.3 或更高版本，然后直接升级到 2.08.3。

要使用较旧版本的 Aurora MySQL 创建集群，请通过 AWS Management Console、AWS CLI 或 RDS API 指定引擎版本。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的 [升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

安全修复：

对托管环境中的处理进行微调的修复和其他增强功能。其他 CVE 修复如下：

- [CVE-2020-14567](#)
- [CVE-2020-14559](#)
- [CVE-2020-14553](#)
- [CVE-2020-14547](#)
- [CVE-2020-14540](#)
- [CVE-2020-2812](#)
- [CVE-2020-2806](#)
- [CVE-2020-2780](#)
- [CVE-2020-2765](#)
- [CVE-2020-2763](#)
- [CVE-2020-2760](#)
- [CVE-2020-2579](#)

不兼容的更改：

此版本新增了影响 `mysqldump` 命令行为的权限更改。用户必须具有访问 `PROCESS` 表的 `INFORMATION_SCHEMA.FILES` 权限。要在不进行任何更改的情况下运行 `mysqldump` 命令，请向 `PROCESS` 命令连接的数据库用户授予 `mysqldump` 权限。您也可以运行带 `mysqldump` 选项的 `--no-tablespaces` 命令。使用该选项，`mysqldump` 输出不包含任何 `CREATE LOGFILE GROUP` 或 `CREATE TABLESPACE` 语句。在这种情况下，`mysqldump` 命令不访问 `INFORMATION_SCHEMA.FILES` 表，您也无需授予 `PROCESS` 权限。

集成了 MySQL 社区版本错误修复

- 错误 #23762382 - INSERT VALUES QUERY WITH JOIN IN A SELECT CAUSES INCORRECT BEHAVIOR。
- 错误 #25209512 - CURRENT_TIMESTAMP PRODUCES ZEROS IN TRIGGER。

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持，但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 异步键预取 (AKP)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于异步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息，请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)”](#)。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2020-08-28 (版本 2.08.2) (已弃用)

版本：2.08.2

Aurora MySQL 2.08.2 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

您可以将快照从当前支持的 Aurora MySQL 版本还原到 Aurora MySQL 2.08.2。您还可以选择将现有 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.08.2。不能将现有的 Aurora MySQL 1.* 集群直接升级到 2.08.2；但是可以将其快照还原到 Aurora MySQL 2.08.2。有关还原快照的更多信息，请参阅《Amazon Aurora 用户指南》中的[从数据库集群快照还原](#)。

要使用较旧版本的 Aurora MySQL 创建集群，请通过 AWS Management Console、AWS CLI 或 RDS API 指定引擎版本。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

关键修复：

- 修复了一个可能导致计划外停机并影响数据库可用性的问题。

可用性修复：

- 修复了以下这个问题：如果 Aurora MySQL 数据库是二进制日志副本并通过 `mysql time_zone` 表复制 DDL 事件，则此数据库可能会重新启动。

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持，但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 异步键预取 (AKP)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于异步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息，请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)”](#)。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2020-06-18 (版本 2.08.1) (已弃用)

版本：2.08.1

Aurora MySQL 2.08.1 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

您可以将快照从当前支持的 Aurora MySQL 版本还原到 Aurora MySQL 2.08.1。您还可以选择将现有 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.08.1。不能将现有的 Aurora MySQL 1.* 集群直接升级到 2.08.1；但是可以将其快照还原到 Aurora MySQL 2.08.1。

要使用较旧版本的 Aurora MySQL 创建集群，请通过 AWS Management Console、AWS CLI 或 RDS API 指定引擎版本。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的 [升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

新功能：

- 全局数据库写入转发。在 Aurora 全局数据库中，现在您可以在连接到辅助集群时执行某些写入操作，如 DML 语句。写入操作将转发到主集群，所有更改都会复制回辅助集群。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [在 Amazon Aurora Global Database 中使用写入转发](#)。

一般稳定性修复：

- 修复了以下问题：如果 S3 备份不包括 mysql.host 表，则将集群从 Amazon S3 还原到 Aurora MySQL 版本 2.08.0 所花的时间比预期时间长。

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持，但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 异步键预取 (AKP)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于异步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息，请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16 \) \(已弃用 \)”](#)。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2020-06-02 (版本 2.08.0) (已弃用)

版本：2.08.0

Aurora MySQL 2.08.0 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

您可以将快照从当前支持的 Aurora MySQL 版本还原到 Aurora MySQL 2.08.0。您还可以选择将现有 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.08.0。不能将现有的 Aurora MySQL 1.* 集群直接升级到 2.08.0；但是可以将其快照还原到 Aurora MySQL 2.08.0。

要使用较旧版本的 Aurora MySQL 创建集群，请通过 AWS Management Console、AWS CLI 或 RDS API 指定引擎版本。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的 [升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

新功能：

- 改进了二进制日志 (binlog) 处理，以便在涉及非常大的事务时缩短崩溃恢复时间和提交时间延迟。
- 启动 Aurora MySQL 的数据库活动流 (DAS) 功能。此功能可提供关系数据库中数据库活动的近实时数据流，以帮助您监视活动。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [使用数据库活动流监控 Amazon Aurora](#)。
- 更新了时区文件以支持最新的巴西时区更改。
- 在 SQL 中引入了新关键字，以便为特定表和/或内部表执行哈希联接功能：HASH_JOIN、HASH_JOIN_PROBING 和 HASH_JOIN_BUILDING。有关更多详细信息，请参阅《Amazon Aurora 用户指南》中的 [Aurora MySQL 参考](#)。
- 通过逆向移植 [MySQL 8.0 功能](#)，在 Aurora MySQL 5.7 中引入了联接顺序提示支持。新提示是 JOIN_FIXED_ORDER、JOIN_ORDER、JOIN_PREFIX 和 JOIN_SUFFIX。有关联接顺序提示支持的详细文档，请参阅 [WL#9158：联接顺序提示](#)。

- Aurora Machine Learning 现在支持将 MEDIUMINT 作为返回类型的用户定义的函数。
- `lambda_async()` 存储过程现在支持所有 MySQL utf8 字符。

高优先级修复：

- 修复了在写入器数据库实例上查询 `INFORMATION_SCHEMA.INNODB_SYS_TABLES` 表后，可能会导致读取器数据库实例为 FTS 查询返回不完整结果的问题。
- [CVE-2019-5443](#)
- [CVE-2019-3822](#)

可用性改进：

- 修复了在启用查询缓存的情况下执行访问多个表或数据库的多查询语句后导致数据库重新启动的问题。
- 修复了锁定管理器中导致事务回滚期间数据库重新启动或故障转移的争用条件。
- 修复了在多个连接尝试使用全文搜索索引更新同一表时触发数据库重新启动或故障转移的问题。
- 修复了在 `kill session` 命令期间可能会触发数据库重新启动或故障转移的问题。如果遇到此问题，请联系 AWS Support 以在您的实例上启用此修复程序。
- 修复了在具有多个 `SELECT` 语句的多语句事务和已启用 `AUTOCOMMIT` 的写入器数据库实例上的写入工作负载繁重期间导致读取器数据库实例重新启动的问题。
- 修复了在写入器数据库实例处于繁重 OLTP 写入工作负载下时执行长时间运行的查询后导致读取器数据库实例重新启动的问题。

常规改进：

- 缩短了启用二进制日志时长时间运行的事务的数据库恢复时间和提交延迟。
- 改进了算法，以生成更好的统计信息，用于估计索引列（包括具有偏斜数据分布的列）上的不同值计数。
- 减少了访问 MyISAM 临时表的联接查询的响应时间和 CPU 利用率，并且结果溢出到本地存储。
- 修复了阻止数据库或表名称包含空格的 Aurora MySQL 5.6 快照还原到新的 Aurora MySQL 5.7 集群的问题。
- 在 `show engine innodb status` 中解决死锁时包含牺牲品事务信息。
- 修复了在多个不同版本的客户端连接到同一数据库并访问查询缓存时引发连接卡住的问题。

- 修复了在数据库实例的整个生命周期内多次调用零停机补丁 (ZDP) 或零停机重新启动 (ZDR) 工作流程导致的内存泄漏。
- 修复了零停机补丁 (ZDP) 或零停机重新启动 (ZDR) 操作中的错误消息，该消息错误地说明如果已关闭自动提交标志，则最后一个事务已中止。
- 修复了零停机补丁 (ZDP) 操作中在新数据库进程中还原用户会话变量时可能会导致服务器故障错误消息的问题。
- 修复了零停机补丁 (ZDP) 操作中在修补期间存在长时间运行的查询时可能会引起间歇性数据库故障的问题。
- 修复了由于来自 Amazon SageMaker 和 Amazon Comprehend 等机器学习服务的错误响应未得到正确处理而导致包含 Aurora 机器学习功能的查询返回空错误消息的问题。
- 修复了内存不足监视功能中不支持 `table_definition_cache` 参数的自定义值的问题。
- 如果 Aurora Machine Learning 查询中断，则会返回错误消息“Query execution was interrupted (查询执行中断)”。以前，会返回常规消息“Internal error in processing ML request (处理 ML 请求时出现内部错误)”。
- 修复了在 `slave_net_timeout` 参数小于 `aurora_binlog_replication_max_yield_seconds` 参数且二进制日志主集群上的工作负载较低时、可能会引发二进制日志工作线程遇到连接超时的问题。
- 通过在错误日志中以每分钟一条消息的频率输出信息性消息，改进了对二进制日志恢复进度的监视。
- 修复了可能会引发 `SHOW ENGINE INNODB STATUS` 查询不报告活动事务的问题。

集成了 MySQL 社区版本错误修复

- [错误 #25289359](#)：如果全文缓存大小超过全文缓存大小限制，则未释放数据同步时执行的全文缓存锁定。
- [错误 #29138644](#)：在 MySQL 服务器运行时手动更改系统时间引发了页面清理器线程延迟。
- [错误 #25222337](#)：在填充受外键约束影响的虚拟列时进行字段名称比较期间，虚拟索引中的 NULL 虚拟列字段名称引发了服务器退出。
- [错误 #25053286](#)：执行包含访问视图的查询的存储过程可能会分配直到会话结束才释放的内存。
- [错误 #25586773](#)：执行包含从某些 `SELECT` 语句的内容创建表的语句的存储过程可能会导致内存泄漏。
- [错误 #28834208](#)：在日志应用程序期间，在 `OPTIMIZE TABLE` 操作之后，InnoDB 在检查虚拟列索引更新之前没有填充虚拟列。

- [错误 #26666274](#)：由于 32 位无符号整数溢出，性能架构缓冲区容器中出现无限循环。

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持，但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 异步键预取 (AKP)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于异步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息，请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)”](#)。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选

- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2023-08-15 (版本 2.07.10 , 与 MySQL 5.7.12 兼容)

版本 : 2.07.10

Aurora MySQL 2.07.10 已正式发布。Aurora MySQL 2.07 版本与 MySQL 5.7.12 兼容。有关社群变更的更多信息，请参阅 [Changes in MySQL 5.7.12 \(2016-04-11, General Availability\)](#)。

Note

此版本被指定为长期支持 (LTS) 版本。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [Aurora MySQL 长期支持 \(LTS\) 版本](#)。

当前支持的 Aurora MySQL 版本有 2.07.*、2.11.*、3.01.*、3.02.*、3.03.* 和 3.04.*。

您可以将快照从当前支持的 Aurora MySQL 版本还原到 Aurora MySQL 2.07.10。您还可以选择将现有 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.07.10。Aurora MySQL 1.* 集群可以就地升级到 Aurora MySQL 2.* (请参阅 [从 Aurora MySQL 1.x 升级到 2.x](#))。Aurora MySQL 2.* 集群也可以就地升级到 Aurora MySQL 3.* (请参阅 [从 Aurora MySQL 2.x 升级到 3.x](#))。

在执行到 Aurora MySQL 2.07.10 的就地引擎版本升级后，操作系统升级将立即自动应用于 db.r4、db.r5、db.t2 和 db.t3 数据库实例类上所有受影响的实例 (如果这些实例是在旧的操作系统版本上运行的)。在多可用区数据库集群中，所有读取器实例首先应用操作系统升级。当第一个读取器实例的操作系统升级完成后，将进行失效转移，并升级前一个写入器实例。

Note

在主要版本升级期间，操作系统升级不会自动应用于 Aurora Global Database。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的 [升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

修复了下面列出的安全问题和 CVE：

对托管环境中的处理进行微调的修复和其他增强功能。其他 CVE 修复如下：

- [CVE-2023-21963](#)
- [CVE-2023-21912](#)
- [CVE-2023-0215](#)
- [CVE-2022-43551](#)
- [CVE-2022-37434](#)
- 修复了在处理审计日志轮换时报告的事件可能未写入审计日志的问题。
- 更新了 Aurora MySQL 使用的默认 SSL 密码，将安全性较低的 DES-CBC3-SHA 值排除在了 [SSL_CIPHER](#) 数据库参数之外。如果您因移除 DES-CBC3-SHA 密码而遇到 SSL 连接问题，请使用以下信息中适用的安全密码：[配置密码套件以连接到 Aurora MySQL 数据库集群](#)。
- OpenSSL 已升级至版本 1.0.2zh。

常规改进：

- 增加了对使用较小密钥大小进行加密的 ECDHE-RSA SSL 密码的支持。
- 修复了使用哈希联接执行查询时的内存管理问题。

Aurora MySQL 版本 2 中不支持的功能

以下功能目前在 Aurora MySQL 版本 2 (与 MySQL 5.7 兼容) 中不受支持。

- 异步键预取 (AKP)。
- 扫描批处理。有关更多信息，请参阅[Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)](#)：

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能：

- CREATE TABLESPACE SQL 语句
- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- X 协议

Aurora MySQL 数据库引擎更新 2023-05-04 (版本 2.07.9 , 与 MySQL 5.7.12 兼容)

版本：2.07.9

Aurora MySQL 2.07.9 已正式发布。Aurora MySQL 2.07 版本与 MySQL 5.7.12 兼容。有关社群变更的更多信息，请参阅 [Changes in MySQL 5.7.12 \(2016-04-11, General Availability\)](#)。

Note

此版本被指定为长期支持 (LTS) 版本。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [Aurora MySQL 长期支持 \(LTS\) 版本](#)。

当前支持的 Aurora MySQL 版本有 2.07.*、2.11.*、3.01.*、3.02.* 和 3.03.*。

您可以将快照从当前支持的 Aurora MySQL 版本还原到 Aurora MySQL 2.07.9。您还可以选择将现有 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.07.9。Aurora MySQL 1.* 集群可以就地升级到 Aurora MySQL 2.* (请参阅 [从 Aurora MySQL 1.x 升级到 2.x](#))。Aurora MySQL 2.* 集群也可以就地升级到 Aurora MySQL 3.* (请参阅 [从 Aurora MySQL 2.x 升级到 3.x](#))。

在执行到 Aurora MySQL 2.07.9 的就地引擎版本升级后，操作系统升级将立即自动应用于 db.r4、db.r5、db.t2 和 db.t3 数据库实例类上所有受影响的实例（如果这些实例是在旧的操作系统版本上运行的）。在多可用区数据库集群中，所有读取器实例首先应用操作系统升级。当第一个读取器实例的操作系统升级完成后，将进行失效转移，并升级前一个写入器实例。

Note

在主要版本升级期间，操作系统升级不会自动应用于 Aurora Global Database。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的 [升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

修复了下面列出的安全问题和 CVE：

对托管环境中的处理进行微调的修复和其他增强功能。其他 CVE 修复如下：

- [CVE-2022-32221](#)

可用性改进：

- 修复了高级审计日志轮换可能会减少可用内存的问题，这可能会导致数据库实例重启。
- 修复了数据库重启期间可能出现的问题，该问题会导致数据库长时间无法成功启动。

常规改进：

- 修复了在极少数情况下当数据库卷增加到 160 GB 的倍数时可能导致实例重新启动的问题。

Aurora MySQL 版本 2 中不支持的功能

以下功能目前在 Aurora MySQL 版本 2（与 MySQL 5.7 兼容）中不受支持。

- 异步键预取 (AKP)。

- 扫描批处理。有关更多信息，请参阅[Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)](#)：

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2022-06-16 (版本 2.07.8) (已弃用)

版本：2.07.8

Aurora MySQL 2.07.8 已正式发布。Aurora MySQL 2.* 版与 MySQL 5.7 兼容，Aurora MySQL 1.* 版与 MySQL 5.6 兼容。

Note

此版本被指定为长期支持 (LTS) 版本。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [Aurora MySQL 长期支持 \(LTS\) 版本](#)。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

您可以将快照从当前支持的 Aurora MySQL 版本还原到 Aurora MySQL 2.07.8。您还可以选择将现有 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.07.8。Aurora MySQL 1.* 集群可以就地升级到 Aurora MySQL 2.* (请参阅[从 Aurora MySQL 1.x 升级到 2.x](#))。Aurora MySQL 2.* 集群也可以就地升级到 Aurora MySQL 3.* (请参阅[从 Aurora MySQL 2.x 升级到 3.x](#))。

要使用较旧版本的 Aurora MySQL 创建集群，请通过 AWS Management Console、AWS CLI 或 RDS API 指定引擎版本。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

安全修复：

对托管环境中的处理进行微调的修复和其他增强功能。其他 CVE 修复如下：

- [CVE-2022-21245](#)
- [CVE-2021-36222](#)
- [CVE-2021-22926](#)

常规改进：

- 修复了在极少数情况下，当死锁检测器线程因争用条件而卡住时，会导致数据库服务器重启的问题。

集成了 MySQL 社区版本错误修复

- 当 UPDATE 需要一个主键大于 1024 个字节的临时表，而该表是使用 InnoDB 创建的时，服务器可能会退出。(错误 #25153670)

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持, 但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 异步键预取 (AKP)。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于异步调用 AWS Lambda 函数的本机函数。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息, 请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)”](#)。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容, 包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能, 使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上, 读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能:

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2021-11-24 (版本 2.07.7) (已弃用)

版本 : 2.07.7

Aurora MySQL 2.07.7 已正式发布。Aurora MySQL 2.* 版与 MySQL 5.7 兼容，Aurora MySQL 1.* 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

您可以将快照从当前支持的 Aurora MySQL 版本还原到 Aurora MySQL 2.07.7。您还可以选择将现有 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.07.7。不能将现有的 Aurora MySQL 1.* 集群直接升级到 2.07.7，但是可以将其快照还原到 Aurora MySQL 2.07.7。

要使用较旧版本的 Aurora MySQL 创建集群，请通过 AWS Management Console、AWS CLI 或 RDS API 指定引擎版本。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的 [升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

安全修复：

对托管环境中的处理进行微调的修复和其他增强功能。其他 CVE 修复如下：

- [CVE-2019-17543](#)
- [CVE-2019-2960](#)

常规改进：

- 修复了 Aurora MySQL 与其他 AWS 服务 (例如 Amazon S3、Amazon ML、Lambda) 集成的相关安全问题。
- 修复了 Aurora 复制滞后报告错误的问题。
- 修复了当数据库具有大量用户和权限组合时, 可能导致数据库实例重启失败的问题。
- 修复了在将主要版本从 Aurora MySQL 1.x (基于 MySQL 5.6) 就地升级到 Aurora MySQL 2.x (基于 MySQL 5.7) 后导致 general_log 和 slow_log 表无法访问的问题。
- 修复了在极少数情况下, 由于检查处理不正确, 可能导致读取器实例重新启动的问题。
- 修复了即使会话已完成处理并处于空闲状态, 但在极少数情况下, 性能详情 (PI) 会话中的“数据库负载”图表也会显示为主动使用 CPU 的问题。
- 修复了使用 LIMIT 子句执行 SQL 语句时可能导致数据库重新启动的并行查询问题。
- 修复了在满足以下所有条件时将现有行的 TIMESTAMP 列的值更新为最新时间戳的问题: 1. 表存在触发器; 2. 对具有 ON DUPLICATE KEY UPDATE 子句的表执行 INSERT; 3. 插入的行可能会导致 UNIQUE 索引或 PRIMARY KEY 中的重复值冲突; 4. 一个或多个列为 TIMESTAMP 数据类型, 原定设置值为 CURRENT_TIMESTAMP。
- 修复了以 READ COMMITTED 隔离级别使用 XA 事务时, 在极少数情况下可能会导致数据库实例重新启动的问题。

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持, 但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 异步键预取 (AKP)。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于异步调用 AWS Lambda 函数的本机函数。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息, 请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16 \) \(已弃用 \)”](#)。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2021-09-02 (版本 2.07.6) (已弃用)

版本：2.07.6

Aurora MySQL 2.07.6 已正式发布。Aurora MySQL 2.* 版与 MySQL 5.7 兼容，Aurora MySQL 1.* 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

您可以将快照从当前支持的 Aurora MySQL 版本还原到 Aurora MySQL 2.07.6。您还可以选择将现有 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.07.6。不能将现有的 Aurora MySQL 1.* 集群直接升级到 2.07.6，但是可以将其快照还原到 Aurora MySQL 2.07.6。

要使用较旧版本的 Aurora MySQL 创建集群，请通过 AWS Management Console、AWS CLI 或 RDS API 指定引擎版本。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

集成了 MySQL 社区版本错误修复

- 插入 64K 大小的记录需要太多的时间。([错误 23031146](#))

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持，但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 异步键预取 (AKP)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于异步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息，请参阅“[Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)](#)”。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制

- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2021-07-06 (版本 2.07.5) (已弃用)

版本 : 2.07.5

Aurora MySQL 2.07.5 已正式发布。Aurora MySQL 2.* 版与 MySQL 5.7 兼容，Aurora MySQL 1.* 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

您可以将快照从当前支持的 Aurora MySQL 版本还原到 Aurora MySQL 2.07.5。您还可以选择将现有 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.07.5。不能将现有的 Aurora MySQL 1.* 集群直接升级到 2.07.5；但是可以将其快照还原到 Aurora MySQL 2.07.5。

要使用较旧版本的 Aurora MySQL 创建集群，请通过 AWS Management Console、AWS CLI 或 RDS API 指定引擎版本。

Note

此版本被指定为长期支持 (LTS) 版本。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [Aurora MySQL 长期支持 \(LTS\) 版本](#)。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

改进

可用性改进：

- 修复了 Aurora 副本中无法进行用户级锁定的问题。
- 修复了在 READ COMMITTED 隔离级别使用 XA 事务时可能导致数据库重启的问题。
- 将 `server_audit_incl_users` 和 `server_audit_excl_users` 全局参数所允许的最大长度增加到了 2000。

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持, 但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 异步键预取 (AKP)。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于异步调用 AWS Lambda 函数的本机函数。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息, 请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)”](#)。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容, 包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能, 使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上, 读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能:

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整

- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2021-03-04 (版本 2.07.4) (已弃用)

版本 : 2.07.4

Aurora MySQL 2.07.4 已正式发布。Aurora MySQL 2.* 版与 MySQL 5.7 兼容 , Aurora MySQL 1.* 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

您可以将快照从当前支持的 Aurora MySQL 版本还原到 Aurora MySQL 2.07.4。您还可以选择将现有 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.07.4。不能将现有的 Aurora MySQL 1.* 集群直接升级到 2.07.4 ; 但是可以将其快照还原到 Aurora MySQL 2.07.4。

要使用较旧版本的 Aurora MySQL 创建集群 , 请通过AWS Management Console、AWS CLI 或 RDS API 指定引擎版本。

Note

此版本被指定为长期支持 (LTS) 版本。有关更多信息 , 请参阅《Amazon Aurora 用户指南》中的 [Aurora MySQL 长期支持 \(LTS \) 版本](#)。

如果您有任何问题或疑问 , 可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息 , 请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

改进

安全修复 :

- [CVE-2020-14812](#)

- [CVE-2020-14793](#)
- [CVE-2020-14790](#)
- [CVE-2020-14775](#)
- [CVE-2020-14769](#)
- [CVE-2020-14765](#)
- [CVE-2020-14760](#)
- [CVE-2020-14672](#)
- [CVE-2020-1971](#)

可用性改进：

- 修复了在读取或写入网络数据包期间一旦出现网络错误可能导致客户端挂起的问题。
- 在某些情况下，DDL 中断后，引擎重启时间缩短。
- 修复了在页面预取请求期间 DDL 或 DML 可能导致引擎重启的问题。
- 修复了在 Aurora 只读副本上执行表/索引反向扫描时副本可能重启的问题。
- 修复了克隆集群操作中可能导致克隆花费更长时间的问题。
- 修复了在对地理空间列使用并行查询优化时可能导致数据库重启的问题。
- 修复了导致二进制日志副本停止并出现 HA_ERR_KEY_NOT_FOUND 错误的问题。

集成了 MySQL 社区版本错误修复

- 修复了在处理包含“ ”(空格)、“%”或“,”的令牌时，全文 ngram 解析器中的问题。如果使用 ngram 解析器，客户应重建其 FTS 索引。(错误 #25873310)
- 修复了在使用嵌套 SQL 视图执行查询期间可能导致引擎重启的问题。(错误 #27214153、错误 #26864199)

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持，但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 异步键预取 (AKP)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。

- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于异步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息，请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16 \) \(已弃用 \)”](#)。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2020-11-10 (版本 2.07.3) (已弃用)

版本：2.07.3

Aurora MySQL 2.07.3 已正式发布。Aurora MySQL 2.* 版与 MySQL 5.7 兼容，Aurora MySQL 1.* 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

您可以将快照从当前支持的 Aurora MySQL 版本还原到 Aurora MySQL 2.07.3。您还可以选择将现有 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.07.3。不能将现有的 Aurora MySQL 1.* 集群直接升级到 2.07.3，但是可以将其快照还原到 Aurora MySQL 2.07.3。

要使用较旧版本的 Aurora MySQL 创建集群，请通过 AWS Management Console、AWS CLI 或 RDS API 指定引擎版本。

Note

此版本被指定为长期支持 (LTS) 版本。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [Aurora MySQL 长期支持 \(LTS\) 版本](#)。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

改进

安全修复：

对托管环境中的处理进行微调的修复和其他增强功能。

- [CVE-2021-2144](#)
- [CVE-2020-14567](#)
- [CVE-2020-14559](#)
- [CVE-2020-14553](#)
- [CVE-2020-14547](#)
- [CVE-2020-14540](#)
- [CVE-2020-2812](#)
- [CVE-2020-2806](#)
- [CVE-2020-2780](#)
- [CVE-2020-2765](#)

- [CVE-2020-2763](#)
- [CVE-2020-2760](#)
- [CVE-2020-2579](#)
- [CVE-2019-2740](#)

不兼容的更改：

此版本新增了影响 `mysqldump` 命令行为的权限更改。用户必须具有访问 `PROCESS` 表的 `INFORMATION_SCHEMA.FILES` 权限。要在不进行任何更改的情况下运行 `mysqldump` 命令，请向 `PROCESS` 命令连接的数据库用户授予 `mysqldump` 权限。您也可以运行带 `mysqldump` 选项的 `--no-tablespaces` 命令。使用该选项，`mysqldump` 输出不包含任何 `CREATE LOGFILE GROUP` 或 `CREATE TABLESPACE` 语句。在这种情况下，`mysqldump` 命令不访问 `INFORMATION_SCHEMA.FILES` 表，您也无需授予 `PROCESS` 权限。

可用性改进：

- 修复了锁定管理器在终止连接/查询和终止会话导致数据库重新启动之间的争用情况。
- 修复了在启用查询缓存的情况下执行访问多个表或数据库的多查询语句后导致数据库重新启动的问题。
- 修复了由于更新具有二级索引的虚拟列而可能导致重复重启的问题。

集成了 MySQL 社区版本错误修复

- InnoDB：在主节点上成功运行到 XA 准备阶段的并发 XA 事务在从属节点上重播时会发生冲突，导致应用程序线程中的锁定等待超时。冲突是由于在从属节点上连续重播事务时，GAP 锁定范围会有所不同。为了防止这种类型的冲突，当 XA 事务进入准备阶段时，XA 事务在 [READ COMMITTED](#) 隔离级别中采取的 GAP 锁定将被释放（而且不再继承）。（错误 #27189701、错误 #25866046）
- InnoDB：使用 [READ COMMITTED](#) 隔离级别时，在验证外键期间多余地使用了 gap 锁定。（错误 #25082593）
- 复制：使用 XA 事务时，如果复制从属节点上的应用程序 (SQL) 线程出现锁定等待超时或死锁，则自动重试不起作用。原因是，尽管 SQL 线程会进行回滚，但它不会回滚 XA 事务。这意味着，当重试交易时，第一个事件是 XA START，由于 XA 事务已在进行中，导致 XAER_RMFAIL 错误，该事件无效。（错误 #24764800）
- 复制：当事务隔离级别设置为 [REPEATABLE READ](#) 时，交错事务有时可能会使从属应用程序死锁。（错误 #25040331）

- 复制：所有现有中继日志文件 (Relay_Log_Space) 的总组合大小的 [SHOW SLAVE STATUS](#) 语句返回的值可能会比中继日志文件实际使用的磁盘空间大得多。I/O 线程在更新值时没有锁定变量，因此 SQL 线程可以在 I/O 线程完成值的更新之前自动删除中继日志文件并写入减少的值。然后，输入/输出线程写入其原始大小计算，从而忽略了 SQL 线程的更新，将删除文件的空间添加回来。Relay_Log_Space 值现已在更新期间被锁定，以防止并发更新，确保计算准确。（错误 #26997096，错误 #87832）
- 对于 VALUES 列表使用包含联接的子查询为第二行或后面的行生成值的 [INSERT](#) 语句，服务器可能会在解析所需的特权失败后退出。（错误 #23762382）
- 对于 [TIMESTAMP](#) 或 [DATETIME](#) 列默认值为 [CURRENT_TIMESTAMP](#) 的表，列可以在表具有 BEFORE INSERT 触发条件时初始化为 0000-00-00 00:00:00。（错误 #25209512、错误 #84077）
- 服务器退出可能是由于多个线程同时尝试注册和取消注册元数据性能架构对象造成的。（错误 #26502135）
- 执行包含从某些 [SELECT](#) 语句的内容创建表的语句的存储过程可能会导致内存泄漏。（错误 #25586773）
- 执行包含访问视图的查询的存储过程可能会分配直到会话结束才释放的内存。（错误 #25053286）
- 某些情况下的子查询具体化可能会导致服务器退出。这些查询现在会产生建议禁止具体化的错误。（错误 #26402045）
- 在使用联接缓冲（例如，使用块嵌套循环算法）的情况下，带有许多左联接的查询会很慢。（错误 #18898433、错误 #72854）
- 在对第二列执行带有 LIKE 子句的内部联接时，优化程序跳过了复合索引中的第二列。（错误 #28086754）

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1（兼容 MySQL 5.6）中受支持，但这些功能目前在 Aurora MySQL 版本 2（兼容 MySQL 5.7）中不受支持。

- 异步键预取 (AKP)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于异步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。

- 扫描批处理。有关更多信息，请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)”](#)。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2020-04-17 (版本 2.07.2) (已弃用)

版本：2.07.2

Aurora MySQL 2.07.2 已正式发布。Aurora MySQL 2.* 版与 MySQL 5.7 兼容，Aurora MySQL 1.* 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

您可以将快照从当前支持的 Aurora MySQL 版本还原到 Aurora MySQL 2.07.2。您还可以选择将现有 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.07.2。不能将现有的 Aurora MySQL 1.* 集群直接升级到 2.07.2；但是可以将其快照还原到 Aurora MySQL 2.07.2。

要使用旧版本的 Aurora MySQL 创建集群，请通过 AWS Management Console、AWS CLI、或 RDS API 指定引擎版本。

Note

此版本被指定为长期支持 (LTS) 版本。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [Aurora MySQL 长期支持 \(LTS\) 版本](#)。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得 [AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

改进

安全修复：

- [CVE-2016-8287](#)
- [CVE-2016-5634](#)

高优先级修复：

- 修复了在某些具有高写入负载的数据库集群上，导致克隆花费时间较长的问题。
- 修复了一个问题，在对具有执行计划（使用了二级索引）的读取器数据库实例进行查询时，该问题可能导致返回未提交的数据。该问题仅限于受修改主索引或二级索引键列的数据操作语言 (DML) 操作所影响的数据。

常规改进：

- 修复了一个问题，在将包含 FTS（全文搜索）索引的 Aurora 1.x 数据库集群还原到 Aurora 2.x 数据库集群时，该问题导致速度缓慢。
- 修复了一个问题，在将包含分区表且表名称具有特殊字符的 Aurora 1.x 数据库快照还原到 Aurora 2.x 数据库集群时，该问题导致速度缓慢。
- 修复了一个问题，该问题导致在查询读取器数据库实例的速度缓慢日志和常规日志时出错。

集成了 MySQL 社区版本错误修复

- 错误 #23104498：修复了性能架构中报告使用的总内存的问题。(<https://github.com/mysql/mysql-server/commit/20b6840df5452f47313c6f9a6ca075bfbc00a96b>)
- 错误 #22551677：修复了性能架构中可能导致数据库引擎在尝试脱机时崩溃的问题。(<https://github.com/mysql/mysql-server/commit/05e2386eccd32b6b444b900c9f8a87a1d8d531e9>)
- 错误 #23550835、错误 #23298025、错误 #81464：修复了性能架构中导致数据库引擎崩溃的问题（因为超出内部缓冲区的容量）。(<https://github.com/mysql/mysql-server/commit/b4287f93857bf2f99b18fd06f555bbe5b12debfcb>)

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1（兼容 MySQL 5.6）中受支持，但这些功能目前在 Aurora MySQL 版本 2（兼容 MySQL 5.7）中不受支持。

- 异步键预取 (AKP)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于同步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息，请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11（版本 1.16）（已弃用）”](#)。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小

- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2019-12-23 (版本 2.07.1) (已弃用)

版本 : 2.07.1

Aurora MySQL 2.07.1 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

您可以将快照从当前支持的 Aurora MySQL 版本还原到 Aurora MySQL 2.07.1。您还可以选择将现有 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.07.1。不能将现有的 Aurora MySQL 1.* 集群直接升级到 2.07.1；但是可以将其快照还原到 Aurora MySQL 2.07.1。

要使用旧版本的 Aurora MySQL 创建集群，请通过 AWS Management Console AWS CLI、或 RDS API 指定引擎版本。

Note

此版本目前不适用于以下 AWS 区域：AWS GovCloud (美国东部) [us-gov-east-1]、(美国西部) [-1]、中国 AWS GovCloud (宁夏) [cn-northwest-us-gov-west 1]、亚太地区 (香港) [ap-east-1] 和中东 (巴林) [me-south-1]。在提供后，将发布单独的公告。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

高优先级修复：

- 修复了 Aurora 特定的数据库跟踪和日志记录子系统上的缓慢内存泄漏，此问题会降低可释放内存。

一般稳定性修复：

- 修复了在执行复杂查询期间发生的崩溃，该查询涉及在内部使用中间表的多表联接和聚合。

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持，但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 异步键预取 (AKP)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 扫描批处理。有关更多信息，请参阅“[Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)](#)”。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能：

- 组复制插件

- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2019-11-25 (版本 2.07.0) (已弃用)

版本 : 2.07.0

Aurora MySQL 2.07.0 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

您可以将快照从当前支持的 Aurora MySQL 版本还原到 Aurora MySQL 2.07.0。您还可以选择将现有 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.07.0。不能将现有的 Aurora MySQL 1.* 集群直接升级到 2.07.0；但是可以将其快照还原到 Aurora MySQL 2.07.0。

要使用旧版本的 Aurora MySQL 创建集群，请通过 AWS Management Console AWS CLI、或 RDS API 指定引擎版本。

Note

此版本目前不适用于以下 AWS 区域：AWS GovCloud (美国东部) [us-gov-east-1]、(美国西部) [-1]、中国 AWS GovCloud (宁夏) [cn-northwest-us-gov-west 1]、亚太地区 (香港) [ap-east-1]、中东 (巴林) [me-south-1] 和南美洲 (圣保罗) [sa-east-1]。在提供后，将发布单独的公告。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

新功能：

- 全球数据库现在允许为部署在以下地区的数据库集群添加辅助只读副本区域：AWS 区域：美国东部（弗吉尼亚北部）[us-east-1]、美国东部（俄亥俄州）[us-east-2]、美国西部（加利福尼亚北部）[us-west-1]、美国西部（俄勒冈）[us-west-2]、欧洲（爱尔兰）[eu-west-1]、欧洲（伦敦）[eu-west-2]、欧洲（west-2）、欧洲（伦敦）[eu-west-2]、欧洲（west-2）巴黎）[eu-west-3]、亚太地区（东京）[ap-northeast-1]、亚太地区（首尔）[ap-northeast-2]、亚太地区（新加坡）[ap-southeast-1]、亚太地区（悉尼）[ap-southeast-2]、加拿大（中部）[ca-central-1]、欧洲（法兰克福）[eu-central-1] 和亚太地区（孟买）[ap-south-1]。
- Amazon Aurora 机器学习是 Aurora MySQL 数据库和 AWS 机器学习 (ML) 服务之间经过高度优化的集成。Aurora 机器学习允许开发人员使用他们已经用于数据库开发的熟悉的 SQL 编程语言调用 ML 模型，从而向其数据库应用程序中添加各种基于 ML 的预测，而无需构建自定义集成或另外学习工具。有关更多信息，请参阅[在 Amazon Aurora 中使用机器学习 \(ML\) 功能](#)。
- 在只读副本上增加了对 ANSI READ COMMITTED 隔离级别的支持。此隔离级别支持对只读副本执行长时间运行的查询，而不会影响写入器节点上写入的高吞吐量。有关更多信息，请参阅[Aurora MySQL 隔离级别](#)。

关键修复：

- [CVE-2019-2922](#)
- [CVE-2019-2923](#)
- [CVE-2019-2924](#)
- [CVE-2019-2910](#)

高优先级修复：

- 修复了 DDL 恢复中导致数据库停机时间延长的问题。在执行多表 drop 语句之后变得不可用的集群 (例如 DROP TABLE t1, t2, t3) 应该更新到这个版本。
- 修复了 DDL 恢复中导致数据库停机时间延长的问题。在执行 INPLACE ALTER TABLE DDL 语句之后变得不可用的集群应该更新到这个版本。

一般稳定性修复：

- 修复了在 information_schema.replica_host_status 表中生成不一致数据的问题。

集成了 MySQL 社区版本错误修复

- 错误 #26251621 : INCORRECT BEHAVIOR WITH TRIGGER AND GCOL
- 错误 #22574695 : ASSERTION `!TABLE || (!TABLE->READ_SET || BITMAP_IS_SET(TABLE->READ_SET, FIEL
- 错误 #25966845 : INSERT ON DUPLICATE KEY GENERATE A DEADLOCK
- 错误 #23070734 : CONCURRENT TRUNCATE TABLES CAUSE STALL
- 错误 #26191879 : FOREIGN KEY CASCADES USE EXCESSIVE MEMORY
- 错误 #20989615 : INNODB AUTO_INCREMENT PRODUCES SAME VALUE TWICE

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持，但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 异步键预取 (AKP)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 扫描批处理。有关更多信息，请参阅“[Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16 \) \(已弃用 \)](#)”。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

Aurora MySQL 2.07.0 与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

Aurora MySQL 2.07.0 当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2019-11-22 (版本 2.06.0) (已弃用)

版本：2.06.0

Aurora MySQL 2.06.0 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.14.*、1.15.*、1.16.*、1.17.*、1.18.*、1.19.*、2.01.*、2.02.*、2.03.*、2.04.*、2.05.* 和 2.06.*。

您可以将快照从当前支持的 Aurora MySQL 版本还原到 Aurora MySQL 2.06.0。您还可以选择将现有 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.06.0。不能将现有的 Aurora MySQL 1.* 集群直接升级到 2.06.0；但是可以将其快照还原到 Aurora MySQL 2.06.0。

要使用旧版本的 Aurora MySQL 创建集群，请通过 AWS Management Console AWS CLI、或 RDS API 指定引擎版本。

Note

此版本目前不适用于以下 AWS 区域：AWS GovCloud（美国东部）[us-gov-east-1]、（美国西部）[-1]、中国 AWS GovCloud（宁夏）[cn-northwest-us-gov-west 1]、亚太地区（香港）[ap-east-1] 和中东（巴林）[me-south-1]。在提供后，将发布单独的公告。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

新功能：

- Aurora MySQL 集群现在支持实例类型 db.r5.8xlarge、db.r5.16xlarge 和 db.r5.24xlarge。有关 Aurora MySQL 集群实例类型的更多信息，请参阅《Amazon Aurora 用户指南》中的[Aurora 数据库实例类](#)。
- 哈希联接功能现在正式推出，无需将 Aurora 实验室模式设置为 ON。在需要使用 equi-join 联接大量数据时，该功能可以提高查询性能。有关使用此功能的更多信息，请参阅《Amazon Aurora 用户指南》中的[使用适用于 Aurora Serverless 的 Data API](#)。
- 热行争用功能现在正式推出，无需将 Aurora 实验室模式设置为 ON。在很多事务争用同一页面上的行时，该功能大大提高了工作负载的吞吐量。
- Aurora MySQL 2.06 及更高版本支持将数据库集群“倒回”到特定时间，而无需从备份还原数据。此功能称为“回溯”，可以快速恢复用户的错误操作，例如错误删除表或行。即使对于大型数据库，回溯也可在几秒钟内完成。阅读[AWS 博客](#)了解概述，有关更多详细信息，请参阅 Amazon Aurora 用户指南中的回溯 Aurora [数据库集群](#)。
- Aurora 2.06 及更高版本支持通过 AWS Lambda 原生函数进行同步调用。lambda_sync() 还提供了本机函数 lambda_async()，它可以替代异步 Lambda 调用的现有存储过程。有关调用 Lambda 函数的信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。

关键修复：

无。

高优先级修复：

安全修复

- [CVE-2019-2805](#)
- [CVE-2019-2791](#)
- [CVE-2019-2778](#)
- [CVE-2019-2758](#)
- [CVE-2019-2739](#)
- [CVE-2019-2730](#)
- [CVE-2018-3064](#)
- [CVE-2018-3058](#)
- [CVE-2018-2786](#)
- [CVE-2017-3653](#)
- [CVE-2017-3465](#)
- [CVE-2017-3455](#)
- [CVE-2017-3244](#)
- [CVE-2016-5612](#)

连接处理

- 数据库可用性已得到改进，能够在执行一个或多个 DDL 时更好地处理客户端连接数量激增的情况。这是通过在需要时临时创建额外的线程来处理的。如果处理 DDL 期间连接数量激增后数据库没有响应，建议进行升级。

引擎重新启动

- 修复了重新启动引擎时长时间不可用的问题。这解决了缓冲池初始化中的一个问题。此问题很少发生，但可能会影响任何受支持的版本。
- 修复了导致配置为二进制日志 (binlog) 主实例的数据库在运行繁重写入工作负载时重启的问题。

一般稳定性修复：

- 在访问未缓存数据的查询可能比平常慢的地方进行了改进。如果客户在访问未缓存数据时遇到无法解释的读取延迟增加问题，则建议他们升级，以免再次遇到这个问题。
- 修复了无法从数据库快照还原分区表的问题。如果客户在数据库中访问已从 Aurora MySQL 1.* 数据库快照还原的分区表时遇到错误，建议使用该版本。
- 通过修复对写入器数据库实例进行 DDL 查询时，处理读取查询的线程与应用架构更改的线程之间的锁定争用问题，提高了 Aurora 副本的稳定性。
- 修复了与 DDL 操作触发的 `mysql.innodb_table_stats` 表更新有关的稳定性问题。
- 修复了针对 Aurora 副本上的临时表执行嵌套查询时错误报告 ERROR 1836 的问题。

性能增强：

- 通过防止在二进制日志工作线程上禁用查询缓存后对缓存进行不必要的 API 调用，提高了二进制日志复制的性能。

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持，但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 异步键预取 (AKP)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 扫描批处理。有关更多信息，请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)”](#)。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

Aurora MySQL 2.06.0 与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

Aurora MySQL 2.06.0 当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2019-11-11 (版本 2.05.0) (已弃用)

版本 : 2.05.0

Aurora MySQL 2.05.0 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.14.*、1.15.*、1.16.*、1.17.*、1.18.*、1.19.*、2.01.*、2.02.*、2.03.* 和 2.04.*。

您可以将快照从当前支持的 Aurora MySQL 版本还原到 Aurora MySQL 2.05.0。您还可以选择将现有 Aurora MySQL 2.* 数据库集群 (最多 2.04.6) 升级到 Aurora MySQL 2.05.0。不能将现有的 Aurora MySQL 1.* 集群直接升级到 2.05.0；但是可以将其快照还原到 Aurora MySQL 2.05.0。

要使用旧版本的 Aurora MySQL 创建集群，请通过 AWS Management Console AWS CLI、或 RDS API 指定引擎版本。

Note

此版本目前不适用于以下 AWS 区域：AWS GovCloud (美国东部) [us-gov-east-1]、(美国西部) [-1]、中国 AWS GovCloud (宁夏) [cn-northwest us-gov-west -1]、亚太地区 (香港) [ap-east-1]、欧洲 (斯德哥尔摩) [eu-north-1] 和中东 (巴林) [me-south-1]。在提供后，将发布单独的公告。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

关键修复：

- [CVE-2018-0734](#)
- [CVE-2019-2534](#)
- [CVE-2018-3155](#)
- [CVE-2018-2612](#)
- [CVE-2017-3599](#)
- [CVE-2018-3056](#)
- [CVE-2018-2562](#)
- [CVE-2017-3329](#)
- [CVE-2018-2696](#)
- 修复了以下问题：如果参数 `sync_binlog` 的值未设定为 1，则不在工作线程上复制主实例上的当前二进制日志文件中的事件。

高优先级修复：

- 强烈建议数据库大小接近 64 TiB 的客户升级到此版本，以避免由于稳定性错误影响接近 Aurora 存储限制的卷而导致停机。
- 参数 `aurora_binlog_replication_max_yield_seconds` 的默认值已更改为零，防止复制延迟增加，有利于提高二进制日志主实例的前台查询性能。

MySQL 错误修复汇总

- 错误 #23054591: PURGE BINARY LOGS TO 正在读取整个二进制日志文件并导致停滞 MySQL

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持, 但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 异步键预取 (AKP)。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于同步调用 AWS Lambda 函数的本机函数。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息, 请参阅“[Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)](#)”。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

Aurora MySQL 2.05.0 与 MySQL 5.7 数据兼容, 包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能, 使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上, 读取性能提高 10 倍以上。

Aurora MySQL 2.05.0 当前不支持以下 MySQL 5.7 功能:

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2020-08-14 (版本 2.04.9) (已弃用)

版本 : 2.04.9

Aurora MySQL 2.04.9 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

要使用旧版本的 Aurora MySQL 创建集群，请通过 AWS Management Console AWS CLI、或 RDS API 指定引擎版本。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

此版本目前不适用于以下 AWS 区域：AWS GovCloud (美国东部) [us-gov-east-1]、(美国西部) [-1]、亚太地区 AWS GovCloud (香港) [ap-east us-gov-west -1] 和中东 (巴林) [me-south-1] [me-south-1]。在提供后，将发布单独的公告。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

高优先级修复：

关键修复：

- [CVE-2020-2760](#)
- [CVE-2019-5443](#)
- [CVE-2019-3822](#)

- [CVE-2019-2924](#)
- [CVE-2019-2923](#)
- [CVE-2019-2922](#)
- [CVE-2019-2911](#)
- [CVE-2019-2910](#)
- [CVE-2019-2805](#)
- [CVE-2019-2791](#)
- [CVE-2019-2778](#)
- [CVE-2019-2758](#)
- [CVE-2019-2740](#)
- [CVE-2019-2739](#)
- [CVE-2019-2730](#)
- [CVE-2019-2628](#)
- [CVE-2018-3064](#)
- [CVE-2018-3058](#)
- [CVE-2018-2813](#)
- [CVE-2018-2786](#)
- [CVE-2017-3653](#)
- [CVE-2017-3465](#)
- [CVE-2017-3464](#)
- [CVE-2017-3455](#)
- [CVE-2017-3244](#)
- [CVE-2016-5612](#)
- [CVE-2016-5436](#)

可用性改进：

- 修复了执行 `kill session` 命令可能导致数据库重新启动或故障转移的问题。如果您遇到此问题，请联系 AWS 支持人员以在您的实例上启用此修复程序。
- 修复了在执行涉及内部使用中间表的多表联接和聚合的复杂查询时导致数据库重新启动的问题。
- 修复了对多个表执行 `DROP TABLE` 发生中断时导致数据库重新启动的问题。

- 修复了在数据库恢复期间导致数据库故障转移的问题。
- 修复了启用审核和慢速查询日志时由于错误报告 `threads_running` 而导致的数据库重启问题。
- 修复了 `kill query` 命令在执行过程中可能会卡住的问题。
- 修复了锁定管理器中导致事务回滚期间数据库重新启动或故障转移的争用条件。
- 修复了在多个连接尝试使用全文搜索索引更新同一表时触发数据库重新启动或故障转移的问题。
- 修复了清除索引导致故障转移或重新启动时可能导致死锁的问题。

常规改进：

- 修复了可能导致对只读副本的查询使用未提交事务中的数据的问题。此问题仅限于在数据库重启后立即启动的事务。
- 修复了对定义了触发器的表执行 `INPLACE ALTER TABLE` 的过程中以及 DDL 不包含 `RENAME` 子句时遇到的问题。
- 修复了导致某些具有高写入负载的数据库集群上的克隆时间较长的问题。
- 修复了名称中嵌入了空格的分区表在升级过程中遇到的问题。
- 修复了只读副本可能暂时只看到写入器上最近提交的事务的部分结果的问题。
- 修复了针对 FTS 表对只读副本进行的查询可能会产生过时结果的问题。只有在对只读副本的 FTS 查询紧跟在写入器上对同一 FTS 表的 `INFORMATION_SCHEMA.INNODB_SYS_TABLES` 查询之后时，才会发生这种情况。
- 修复了导致将包含 FTS (全文搜索) 索引的 Aurora 1.x 数据库集群还原到 Aurora 2.x 数据库集群速度缓慢的问题。
- 将 `server_audit_incl_users` 和 `server_audit_excl_users` 全局参数所允许的最大长度增加到了 2000。
- 修复了从 Aurora 1.x 还原到 Aurora 2.x 可能需要更长时间才能完成的问题。
- 修复了通过存储过程调用 `lambda_async` 不适用于 Unicode 的问题。
- 修复了空间索引无法正确处理未记录的几何列时遇到的问题。
- 修复了可能导致对读取器数据库实例的查询因 `InternalFailureException` 错误失败并显示“Operation terminated (internal error)”消息的问题。

MySQL 错误修复汇总

- 错误 #23070734、错误 #80060：并发 `TRUNCATE TABLE` 导致停机
- 错误 #23103937：`PS_TRUNCATE_ALL_TABLES()` 在 `SUPER_READ_ONLY` 模式下无法运行

- 错误 #22551677：使服务器脱机时，性能架构中的争用条件可能导致服务器退出。
- 错误 #27082268：无效的 FTS 同步。
- 错误 #12589870：修复了在启用查询缓存的情况下导致使用多查询语句重新启动的问题。
- 错误 #26402045：某些情况下的子查询具体化可能会导致服务器退出。这些查询现在会产生建议禁止具体化的错误。
- 错误 #18898433：在使用联接缓冲（例如，使用块嵌套循环算法）的情况下，带有许多左联接的查询会很慢。
- 错误 #25222337：在填充受外键约束影响的虚拟列的同时比较字段名称的过程中，虚拟索引中的 NULL 虚拟列字段名称引发了服务器退出。<https://github.com/mysql/mysql-server/commit/273d5c9d7072c63b6c47dbef6963d7dc491d5131>
- 错误 #25053286：执行包含访问视图的查询的存储过程可能会分配直到会话结束才释放的内存。<https://github.com/mysql/mysql-server/commit/d7b37d4d141a95f577916448650c429f0d6e193d>
- 错误 #25586773：执行包含从某些 SELECT <https://dev.mysql.com/doc/refman/5.7/en/select.html> 语句的内容创建表的语句的存储过程可能会导致内存泄漏。<https://github.com/mysql/mysql-server/commit/88301e5adab65f6750f66af284be410c4369d0c1>
- 错误 #26666274：性能架构缓冲区容器中的无限循环。
- 错误 #23550835、错误 #23298025、错误 #81464：当内部缓冲区已满时，对性能架构表执行 SELECT 可能会导致服务器退出。

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1（兼容 MySQL 5.6）中受支持，但这些功能目前在 Aurora MySQL 版本 2（兼容 MySQL 5.7）中不受支持。

- 异步键预取 (AKP)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于同步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息，请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11（版本 1.16）（已弃用）”](#)。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

Aurora MySQL 2.04.9 与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

Aurora MySQL 2.04.9 当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2019-11-20 (版本 2.04.8) (已弃用)

版本：2.04.8

Aurora MySQL 2.04.8 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

您可以将任何 2.* Aurora MySQL 版本的快照还原到 Aurora MySQL 2.04.8。您还可以选择将现有 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.04.8。

要使用旧版本的 Aurora MySQL 创建集群，请通过 AWS Management Console AWS CLI、或 RDS API 指定引擎版本。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

此版本目前不适用于以下 AWS 区域：AWS GovCloud（美国东部）[us-gov-east-1]、（美国西部）[-1]、亚太地区 AWS GovCloud（香港）[ap-east us-gov-west -1] 和中东（巴林）[me-south-1] [me-south-1]。在提供后，将发布单独的公告。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

新功能：

• 只读副本改进：

- 通过有效地将数据传输到 Aurora 数据库集群中的读取器实例，减少了来自写入器实例的网络流量。默认情况下启用这项改进，因为这有助于防止副本滞后并重新启动。此功能的参数为 `aurora_enable_repl_bin_log_filtering`。
- 使用压缩功能减少了 Aurora 数据库集群中从写入器实例到读取器实例的网络流量。默认情况下，只对 8xlarge 和 16xlarge 实例类启用这项改进，因为这些实例可以承受额外的 CPU 压缩开销。此功能的参数为 `aurora_enable_replica_log_compression`。

高优先级修复：

- 改进了 Aurora 写入器实例中的内存管理，可防止 Aurora 数据库集群中存在读取器实例的情况下由于繁重工作负载期间内存不足而导致写入器重新启动。
- 在计划程序中，修复了并发访问性能架构对象时导致引擎重新启动的非确定性条件。

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持, 但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 异步键预取 (AKP)。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于同步调用 AWS Lambda 函数的本机函数。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息, 请参阅“[Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)](#)”。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

Aurora MySQL 2.04.8 与 MySQL 5.7 数据兼容, 包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能, 使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上, 读取性能提高 10 倍以上。

Aurora MySQL 2.04.8 当前不支持以下 MySQL 5.7 功能:

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2019-11-14 (版本 2.04.7) (已弃用)

版本 : 2.04.7

Aurora MySQL 2.04.7 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

您可以将快照从当前支持的 Aurora MySQL 版本还原到 Aurora MySQL 2.04.7。您还可以选择将现有 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.04.7。不能将现有的 Aurora MySQL 1.* 集群直接升级到 2.04.7；但是可以将其快照还原到 Aurora MySQL 2.04.7。

要使用旧版本的 Aurora MySQL 创建集群，请通过 AWS Management Console AWS CLI、或 RDS API 指定引擎版本。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

此版本目前不适用于以下 AWS 区域：AWS GovCloud (美国东部) [us-gov-east-1]、(美国西部) [-1]、亚太地区 AWS GovCloud (香港) [ap-east us-gov-west -1] 和中东 (巴林) [me-south-1] [me-south-1]。在提供后，将发布单独的公告。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

高优先级修复：

连接处理

- 数据库可用性已得到改进，能够在执行一个或多个 DDL 时更好地处理客户端连接数量激增的情况。这是通过在需要时临时创建额外的线程来处理的。如果处理 DDL 期间连接数量激增后数据库没有响应，建议进行升级。
- 修复了导致 `Threads_running` 全局状态变量值不正确的问题。

引擎重新启动

- 修复了重新启动引擎时长时间不可用的问题。这解决了缓冲池初始化中的一个问题。此问题很少发生，但可能会影响任何受支持的版本。

一般稳定性修复：

- 在访问未缓存数据的查询可能比平常慢的地方进行了改进。如果客户在访问未缓存数据时遇到无法解释的读取延迟增加问题，则建议他们升级，以免再次遇到这个问题。

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持，但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 异步键预取 (AKP)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于同步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息，请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)”](#)。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

Aurora MySQL 2.04.7 与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

Aurora MySQL 2.04.7 当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2019-09-19 (版本 2.04.6) (已弃用)

版本：2.04.6

Aurora MySQL 2.04.6 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

您可以选择将现有 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.04.6。不允许就地升级 Aurora MySQL 1.* 集群。此限制将在稍后的 Aurora MySQL 2.* 版本中解除。您可以将 Aurora MySQL 1.14.*、1.15.*、1.16.*、1.17.*、1.18.*、1.19.*、2.01.*、2.02.*、2.03.* 和 2.04.* 的快照还原到 Aurora MySQL 2.04.6。

要使用旧版本的 Aurora MySQL，您可以通过、或 Amazon RDS API 指定引擎版本来创建新的数据库集群。AWS Management Console AWS CLI

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

此版本目前在以下 AWS 地区不可用：欧洲（伦敦）[eu-west-2]、（美国东部）[-1]、AWS GovCloud（美国西部）[us-gov-east-1]、中国 AWS GovCloud（宁夏）[cn-northwest us-gov-west -1] 和亚太地区（香港）[ap-east-1]。在提供后，将发布单独的公告。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

- 修复了以下问题：如果参数 `sync_binlog` 的值未设定为 1，则不在工作线程上复制主实例上的当前二进制日志文件中的事件。
- 参数 `aurora_binlog_replication_max_yield_seconds` 的默认值已更改为零，防止复制延迟增加，有利于提高二进制日志主实例的前台查询性能。

MySQL 错误修复汇总

- 错误 #23054591: PURGE BINARY LOGS TO 正在读取整个二进制日志文件并导致停滞 MySQL

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1（兼容 MySQL 5.6）中受支持，但这些功能目前在 Aurora MySQL 版本 2（兼容 MySQL 5.7）中不受支持。

- 异步键预取 (AKP)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。

- 用于同步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息，请参阅“[Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)](#)”。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

Aurora MySQL 2.04.6 与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

Aurora MySQL 2.04.6 当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2019-07-08 (版本 2.04.5) (已弃用)

版本：2.04.5

Aurora MySQL 2.04.5 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

您可以选择将现有 Aurora MySQL 2.* 数据库集群升级到 Aurora MySQL 2.04.5。不允许就地升级 Aurora MySQL 1.* 集群。此限制将在稍后的 Aurora MySQL 2.* 版本中解除。您可以将 Aurora MySQL 1.14.*、1.15.*、1.16.*、1.17.*、1.18.*、1.19.*、2.01.*、2.02.*、2.03.* 和 2.04.* 的快照还原到 Aurora MySQL 2.04.5。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

安全修复：

- [CVE-2016-3518](#)

常规修复：

- 修复了存储卷增长期间导致数据库重新启动的竞争条件。
- 修复了卷打开期间导致数据库重新启动的内部通信故障。
- 添加了针对分区表上的 ALTER TABLE ALGORITHM=INPLACE 的 DDL 恢复支持。
- 修复了导致数据库重新启动的 ALTER TABLE ALGORITHM=COPY 的 DDL 恢复。
- 提高了写入器上的重度删除工作负载下的 Aurora 副本稳定性。
- 修复了执行全文搜索索引同步的线程与从字典缓存执行全文搜索表逐出的线程之间的死锁导致的数据库重新启动问题。
- 修复了在二进制日志主实例连接不稳定的情况下、复制 DDL 期间在二进制日志工作线程上出现的稳定性问题。
- out-of-memory 修复了全文搜索代码中导致数据库重新启动的问题。
- 修复了在使用整个 64 TiB 卷时导致 Aurora 写入器重新启动的问题。
- 修复了性能架构功能中导致数据库重新启动的竞争条件。

- 修复了在处理网络协议管理错误时导致连接中止的问题。

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持, 但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 异步键预取 (AKP)。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于同步调用 AWS Lambda 函数的本机函数。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息, 请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)”](#)。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

Aurora MySQL 2.04.5 与 MySQL 5.7 数据兼容, 包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能, 使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上, 读取性能提高 10 倍以上。

Aurora MySQL 2.04.5 当前不支持以下 MySQL 5.7 功能:

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选

- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2019-05-29 (版本 2.04.4) (已弃用)

版本 : 2.04.4

Aurora MySQL 2.04.4 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

在创建新的 Aurora MySQL 数据库集群 (包括还原快照) 时，您可以选择与 MySQL 5.7 或 MySQL 5.6 兼容。我们不允许就地升级 Aurora MySQL 1.* 集群或通过 Amazon S3 备份将 Aurora MySQL 1.* 集群还原到 Aurora MySQL 2.04.4。我们计划在以后的 Aurora MySQL 2.* 版本中删除这些限制。

您可以将 Aurora MySQL 1.14.*、1.15.*、1.16.*、1.17.*、1.18.*、1.19.*、2.01.*、2.02.*、2.03.* 和 2.04.* 的快照还原到 Aurora MySQL 2.04.4。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

此版本目前不在 AWS GovCloud (美国西部) [us-gov-west-1]、欧洲 (斯德哥尔摩) [eu-north-1]、中国 (宁夏) [cn-northwest-1] 和亚太地区 (香港) [ap-east-1] [ap-east-1] 地区推出。AWS 在提供后，将发布单独的公告。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

- 修复了在将数据从 S3 加载到 Aurora 时可能造成故障的问题。

- 修复了在将数据从 Aurora 上传到 S3 时可能造成故障的问题。
- 修复了在处理网络协议管理错误时导致连接中止的问题。
- 修复了在处理分区表时可能导致崩溃的问题。
- 修复了 Performance Insights 功能在某些区域不可用的问题。

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持, 但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 异步键预取 (AKP)。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于同步调用 AWS Lambda 函数的本机函数。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息, 请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)”](#)。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

Aurora MySQL 2.04.4 与 MySQL 5.7 数据兼容, 包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能, 使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上, 读取性能提高 10 倍以上。

Aurora MySQL 2.04.4 当前不支持以下 MySQL 5.7 功能:

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整

- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2019-05-09 (版本 2.04.3) (已弃用)

版本 : 2.04.3

Aurora MySQL 2.04.3 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

在创建新的 Aurora MySQL 数据库集群 (包括还原快照) 时，您可以选择与 MySQL 5.7 或 MySQL 5.6 兼容。我们不允许就地升级 Aurora MySQL 1.* 集群或通过 Amazon S3 备份将 Aurora MySQL 1.* 集群还原到 Aurora MySQL 2.04.3。我们计划在以后的 Aurora MySQL 2.* 版本中删除这些限制。

您可以将 Aurora MySQL 1.14.*、1.15.*、1.16.*、1.17.*、1.18.*、1.19.*、2.01.*、2.02.*、2.03.* 和 2.04.* 的快照还原到 Aurora MySQL 2.04.3。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

此版本目前不在 AWS GovCloud (美国西部) [-us-gov-west 1] 和中国 (宁夏) [cn-northwest-1] 区域提供。AWS 在提供后，将发布单独的公告。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

- 修复了二进制日志复制中的一个错误，该错误可能导致配置为二进制日志工作线程的 Aurora 实例出现问题。
- 修复了处理大型存储例程时出现 out-of-memory 的情况。
- 修复了处理某类 ALTER TABLE 命令时发生的错误。
- 修复了由于网络协议管理存在错误导致连接中止的问题。

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持，但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 异步键预取 (AKP)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于同步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息，请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)”](#)。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

Aurora MySQL 2.04.3 与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

Aurora MySQL 2.04.3 当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载

- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2019-05-02 (版本 2.04.2) (已弃用)

版本 : 2.04.2

Aurora MySQL 2.04.2 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

在创建新的 Aurora MySQL 数据库集群 (包括还原快照) 时，您可以选择与 MySQL 5.7 或 MySQL 5.6 兼容。我们不允许就地升级 Aurora MySQL 1.* 集群或通过 Amazon S3 备份将 Aurora MySQL 1.* 集群还原到 Aurora MySQL 2.04.2。我们计划在以后的 Aurora MySQL 2.* 版本中删除这些限制。

您可以将 Aurora MySQL

1.14.*、1.15.*、1.16.*、1.17.*、1.18.*、1.19.*、2.01.*、2.02.*、2.03.*、2.04.0 和 2.04.1 的快照还原到 Aurora MySQL 2.04.2。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

此版本目前不在 AWS GovCloud (美国西部) [-us-gov-west 1] 和中国 (宁夏) [cn-northwest-1] 区域提供。AWS 在提供后，将发布单独的公告。

Note

有关如何升级 Aurora MySQL 数据库集群的信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

- 增加了使用自定义证书的 SSL 二进制日志复制支持。有关使用 Aurora MySQL 中的 SSL 二进制日志复制的信息，请参见 [mysql_rds_import_binlog_ssl_material](#)。
- 修复了在优化具有全文搜索索引的表时 Aurora 主实例上出现的死锁问题。
- 修复了使用 SELECT(*) 的某些查询的性能在具有二级索引的表中可能受影响的 Aurora 副本中的问题。
- 修复了导致引发错误 1032 的情况。
- 通过修复多个死锁提高了 Aurora 副本的稳定性。

MySQL 错误修复汇总

- 错误 #24829050 - INDEX_MERGE_INTERSECTION OPTIMIZATION CAUSES WRONG QUERY RESULTS

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持，但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 异步键预取 (AKP)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于同步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息，请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)”](#)。

- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

Aurora MySQL 2.04.2 与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

Aurora MySQL 2.04.2 当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2019-03-25 (版本 2.04.1) (已弃用)

版本：2.04.1

Aurora MySQL 2.04.1 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

在创建新的 Aurora MySQL 数据库集群 (包括还原快照) 时，您可以选择与 MySQL 5.7 或 MySQL 5.6 兼容。我们不允许就地升级 Aurora MySQL 1.* 集群或通过 Amazon S3 备份将 Aurora MySQL 1.* 集群还原到 Aurora MySQL 2.04.1。我们计划在以后的 Aurora MySQL 2.* 版本中删除这些限制。

您可以将 Aurora MySQL

1.14.*、1.15.*、1.16.*、1.17.*、1.18.*、1.19.*、2.01.*、2.02.*、2.03.*、2.04.0 的快照还原到 Aurora MySQL 2.04.1。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

此版本目前不在 AWS GovCloud (美国西部) [us-gov-west-1] 区域提供。在提供后，将发布单独的公告。

Note

升级数据库集群的过程已更改。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

- 修复了 1.16 以下版本的 Aurora MySQL 5.6 快照无法还原到最新的 Aurora MySQL 5.7 集群的问题。

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持，但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 异步键预取 (AKP)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于同步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息，请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16 \) \(已弃用 \)”](#)。

- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2019-03-25 (版本 2.04.0) (已弃用)

版本：2.04

Aurora MySQL 2.04 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

在创建新的 Aurora MySQL 数据库集群（包括还原快照）时，您可以选择与 MySQL 5.7 或 MySQL 5.6 兼容。我们不允许就地升级 Aurora MySQL 1.* 集群或通过 Amazon S3 备份将 Aurora MySQL 1.* 集群还原到 Aurora MySQL 2.04.0。我们计划在以后的 Aurora MySQL 2.* 版本中删除这些限制。

您可以将 Aurora MySQL 1.19.*、2.01.*、2.02.* 和 2.03.* 的快照还原到 Aurora MySQL 2.04.0。您不能将 Aurora MySQL 1.14.* 或更低版本、1.15.*、1.16.*、1.17.*、1.18.* 的快照还原到 Aurora MySQL 2.04.0。Aurora MySQL 2.04.1 已取消此限制。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

此版本目前不在 AWS GovCloud (美国西部) [us-gov-west-1] 区域提供。在提供后，将发布单独的公告。

Note

升级数据库集群的过程已更改。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

- 支持基于 GTID 的复制。有关将基于 GTID 的复制用于 Aurora MySQL 的信息，请参阅《Amazon Aurora 用户指南》中的[将基于 GTID 的复制用于 Aurora MySQL](#)。
- 修复了当删除或更新临时表中的语句包含 InnoDB 子查询时 Aurora 副本错误地引发 Running in read-only mode 错误的问题。

MySQL 错误修复汇总

- 错误 #26225783 : MYSQL CRASH ON CREATE TABLE (REPRODUCEABLE) -> INNODB: ALONG SEMAPHORE WAIT。

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持，但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 异步键预取 (AKP)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于同步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息，请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)”](#)。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

此 Aurora MySQL 版本与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

此 Aurora MySQL 版本当前不支持以下 MySQL 5.7 功能：

- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句

Aurora MySQL 数据库引擎更新 2019-02-07 (版本 2.03.4) (已弃用)

版本：2.03.4

Aurora MySQL 2.03.4 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

在创建新的 Aurora MySQL 数据库集群（包括还原快照）时，您可以选择与 MySQL 5.7 或 MySQL 5.6 的兼容性。

我们不允许将 Aurora MySQL 1.* 集群就地升级到 Aurora MySQL 2.03.4 或通过 Amazon S3 备份还原到 Aurora MySQL 2.03.4。我们计划在以后的 Aurora MySQL 2.* 版本中删除这些限制。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

此版本目前不在 AWS GovCloud（美国西部）[-us-gov-west 1] 和中国（北京）[cn-north-1] 区域推出。在提供后，将发布单独的公告。

Note

升级数据库集群的过程已更改。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

- 支持 UTF8MB4 Unicode 9.0 区分腔调字和区分大小写排序规则 `utf8mb4_0900_as_ci`。

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1（兼容 MySQL 5.6）中受支持，但这些功能目前在 Aurora MySQL 版本 2（兼容 MySQL 5.7）中不受支持。

- 异步键预取 (AKP)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。

- 用于同步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息，请参阅“[Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)](#)”。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

Aurora MySQL 2.03.4 与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

Aurora MySQL 2.03.4 当前不支持以下 MySQL 5.7 功能：

- 全局事务标识符 (GTID)。Aurora MySQL 在版本 2.04 以及更高版本中支持 GTID。
- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句
- X 协议

Aurora MySQL 数据库引擎更新 2019-01-18 (版本 2.03.3) (已弃用)

版本：2.03.3

Aurora MySQL 2.03.3 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

在创建新的 Aurora MySQL 数据库集群（包括还原快照）时，您可以选择与 MySQL 5.7 或 MySQL 5.6 的兼容性。

我们不允许将 Aurora MySQL 1.* 集群就地升级到 Aurora MySQL 2.03.3 或通过 Amazon S3 备份还原到 Aurora MySQL 2.03.3。我们计划在以后的 Aurora MySQL 2.* 版本中删除这些限制。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

此版本目前不在 AWS GovCloud（美国西部）[-us-gov-west 1] 和中国（北京）[cn-north-1] 区域推出。在提供后，将发布单独的公告。

Note

升级数据库集群的过程已更改。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

CVE 修复

- [CVE-2016-5436](#)

关键修复：

- 修复了对索引运行后向扫描时 Aurora 副本可能被死锁的问题。
- 修复了当 Aurora 主实例在分区表中运行就地 DDL 操作时 Aurora 副本可能重启的问题。
- 修复了对 Aurora 主实例执行 DDL 操作后进行查询缓存验证期间 Aurora 副本可能重启的问题。
- 修复了当 Aurora 主实例在表上运行截断时，在该表的 SELECT 查询期间 Aurora 副本可能重启的问题。
- 修复了仅访问索引列的 MyISAM 临时表存在的错误结果问题。

- 修复了约 40,000 次查询后会定期生成错误的大 query_time 和 lock_time 值的慢速日志中的问题。
- 修复了名为“tmp”的架构可能导致 RDS for MySQL 到 Aurora MySQL 的迁移出现停滞的问题。
- 修复了审核日志在日志轮换期间可能存在缺失事件的问题。
- 修复了在启用实验室模式下的 Fast DDL 功能时从 Aurora 5.6 快照还原的 Aurora 主实例可能重启的问题。
- 修复了 CPU 使用完全是由字典 stats 线程引起的问题。
- 修复了运行 CHECK TABLE 语句时 Aurora 副本可能重启的问题。

MySQL 错误修复汇总

- 错误 #25361251 : INCORRECT BEHAVIOR WITH INSERT ON DUPLICATE KEY IN SP
- 错误 #26734162 : INCORRECT BEHAVIOR WITH INSERT OF BLOB + ON DUPLICATE KEY UPDATE
- 错误 #27460607 : INCORRECT BEHAVIOR OF IODKU WHEN INSERT SELECT's SOURCE TABLE IS EMPTY
- 使用 DISTINCT or GROUP BY 子句的查询可能会返回错误的结果。(MySQL 5.7 错误 #79591 , 错误 #22343910)
- 使用 DELETE 子句中的派生表失败并发生错误 1093 (错误 #23074801) 的联接表中的 WHERE。
- GCOLS: INCORRECT BEHAVIOR WITH CHARSET CHANGES (错误 #25287633)。

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持, 但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 异步键预取 (AKP)。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于同步调用 AWS Lambda 函数的本机函数。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息, 请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16 \) \(已弃用 \)”](#)。

- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

Aurora MySQL 2.03.3 与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

Aurora MySQL 2.03.3 当前不支持以下 MySQL 5.7 功能：

- 全局事务标识符 (GTID)。Aurora MySQL 在版本 2.04 以及更高版本中支持 GTID。
- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句
- X 协议

Aurora MySQL 数据库引擎更新 2019-01-09 (版本 2.03.2) (已弃用)

版本：2.03.2

Aurora MySQL 2.03.2 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

在创建新的 Aurora MySQL 数据库集群 (包括还原快照) 时，您可以选择与 MySQL 5.7 或 MySQL 5.6 的兼容性。

我们不允许将 Aurora MySQL 1.* 集群就地升级到 Aurora MySQL 2.03.2 或通过 Amazon S3 备份还原到 Aurora MySQL 2.03.2。我们计划在以后的 Aurora MySQL 2.* 版本中删除这些限制。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

此版本目前不在 AWS GovCloud (美国西部) [-us-gov-west 1] 和中国 (北京) [cn-north-1] 区域推出。在提供后，将发布单独的公告。

Note

升级数据库集群的过程已更改。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

- Aurora 版本选择器 – 从 Aurora MySQL 2.03.2 开始，您可以在 AWS Management Console 上从多个与 MySQL 5.7 兼容的 Aurora 版本中选择。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[通过 AWS 检查或指定 Aurora MySQL 引擎版本](#)。

关键修复：

- [CVE-2016-3495](#)

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持，但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 异步键预取 (AKP)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。

- 用于同步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息，请参阅“[Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)](#)”。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

Aurora MySQL 2.03.2 与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

Aurora MySQL 2.03.2 当前不支持以下 MySQL 5.7 功能：

- 全局事务标识符 (GTID)。Aurora MySQL 在版本 2.04 以及更高版本中支持 GTID。
- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句
- X 协议

Aurora MySQL 数据库引擎更新 2018-10-24 (版本 2.03.1) (已弃用)

版本：2.03.1

Aurora MySQL 2.03.1 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

在创建新的 Aurora MySQL 数据库集群时，您可以选择与 MySQL 5.7 或 MySQL 5.6 的兼容性。在还原与 MySQL 5.6 兼容的快照时，您可以选择与 MySQL 5.7 或 MySQL 5.6 的兼容性。

您可以将 Aurora MySQL 1.14.*、1.15.*、1.16.*、1.17.*、1.18.*、2.01.*、2.02.* 和 2.03 快照还原到 Aurora MySQL 2.03.1 中。

我们不允许将 Aurora MySQL 1.* 集群就地升级到 Aurora MySQL 2.03.1 或通过 Amazon S3 备份还原到 Aurora MySQL 2.03.1。我们计划在以后的 Aurora MySQL 2.* 版本中删除这些限制。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

此版本目前不在 AWS GovCloud (美国西部) [-us-gov-west 1] 和中国 (北京) [cn-north-1] 区域推出。在提供后，将发布单独的公告。

改进

- 修复了一个问题，即，在运行事务死锁检测时，Aurora 写入器可能会重新启动。

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持，但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 异步键预取 (AKP)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于同步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息，请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)”](#)。

- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

Aurora MySQL 2.03.1 与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

Aurora MySQL 2.03.1 当前不支持以下 MySQL 5.7 功能：

- 全局事务标识符 (GTID)。Aurora MySQL 在版本 2.04 以及更高版本中支持 GTID。
- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句
- X 协议

Aurora MySQL 数据库引擎更新 2018-10-11 (版本 2.03) (已弃用)

版本：2.03

Aurora MySQL 2.03 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

在创建新的 Aurora MySQL 数据库集群时，您可以选择与 MySQL 5.7 或 MySQL 5.6 的兼容性。在还原与 MySQL 5.6 兼容的快照时，您可以选择与 MySQL 5.7 或 MySQL 5.6 的兼容性。

您可以将 Aurora MySQL 1.14.*、1.15.*、1.16.*、1.17.*、1.18.*、2.01.* 和 2.02.* 快照还原到 Aurora MySQL 2.03 中。

我们不允许将 Aurora MySQL 1.* 集群就地升级到 Aurora MySQL 2.03 或通过 Amazon S3 备份还原到 Aurora MySQL 2.03。我们计划在以后的 Aurora MySQL 2.* 版本中删除这些限制。

Note

此版本目前不在 AWS GovCloud (美国西部) [-us-gov-west 1] 和中国 (北京) [cn-north-1] 区域推出。在提供后，将发布单独的公告。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

改进

- 提供了性能架构。
- 修复了一个问题，即，处于已终止状态的僵尸会话可能会使用较多 CPU。
- 修复了在只读事务获取 Aurora 写入器上的记录锁定时出现的死锁问题。
- 修复了一个问题，即，没有客户工作负载的 Aurora 副本可能具有较高的 CPU 利用率。
- 对多个问题的修复，这些问题可能导致 Aurora 副本或 Aurora 写入器重新启动。
- 添加了在达到磁盘吞吐量限制时跳过诊断日志记录的功能。
- 修复了在 Aurora 写入器上启用 binlog 时出现的内存泄漏问题。

集成了 MySQL 社区版本错误修复

- 分区表上的反向扫描执行 ICP - ORDER BY DESC (错误 24929748)。
- JSON_OBJECT 创建无效的 JSON 代码 (错误 26867509)。
- 插入大型 JSON 数据花费大量时间 (错误 22843444)。
- 分区表在 5.7 中使用的内存比 5.6 多 (错误 25080442)。

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持，但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 异步键预取 (AKP)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的连接查询](#)。
- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于同步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息，请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)”](#)。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

Aurora MySQL 2.03 与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

Aurora MySQL 2.03 当前不支持以下 MySQL 5.7 功能：

- 全局事务标识符 (GTID)。Aurora MySQL 在版本 2.04 以及更高版本中支持 GTID。
- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句
- X 协议

Aurora MySQL 数据库引擎更新 2018-10-08 (版本 2.02.5) (已弃用)

版本 : 2.02.5

Aurora MySQL 2.02.5 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

在创建新的 Aurora MySQL 数据库集群时，您可以选择与 MySQL 5.7 或 MySQL 5.6 的兼容性。在还原与 MySQL 5.6 兼容的快照时，您可以选择与 MySQL 5.7 或 MySQL 5.6 的兼容性。

您可以将 Aurora MySQL 1.14.*、1.15.*、1.16.*、1.17.*、1.18.*、2.01.* 和 2.02.* 快照还原到 Aurora MySQL 2.02.5 中。您还可以执行从 Aurora MySQL 2.01.* 或 2.02.* 到 Aurora MySQL 2.02.5 的就地升级。

我们不允许将 Aurora MySQL 1.* 集群就地升级到 Aurora MySQL 2.02.5 或通过 Amazon S3 备份还原到 Aurora MySQL 2.02.5。我们计划在以后的 Aurora MySQL 2.* 版本中删除这些限制。

该版本的 Aurora MySQL 5.7 已禁用性能架构。请升级到 Aurora 2.03 以提供性能架构支持。

Note

此版本目前不在 AWS GovCloud (美国西部) [-us-gov-west 1] 和中国 (北京) [cn-north-1] 区域推出。在提供后，将发布单独的公告。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

改进

- 修复了一个问题，即，在一个表上执行反向扫描时，Aurora 副本可能会重新启动。

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持，但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 异步键预取 (AKP)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的连接查询](#)。
- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于同步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息，请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)”](#)。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

Aurora MySQL 2.02.5 与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

Aurora MySQL 2.02.5 当前不支持以下 MySQL 5.7 功能：

- 全局事务标识符 (GTID)。Aurora MySQL 在版本 2.04 以及更高版本中支持 GTID。
- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句
- X 协议

Aurora MySQL 数据库引擎更新 2018-09-21 (版本 2.02.4) (已弃用)

版本 : 2.02.4

Aurora MySQL 2.02.4 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

在创建新的 Aurora MySQL 数据库集群时，您可以选择与 MySQL 5.7 或 MySQL 5.6 的兼容性。在还原与 MySQL 5.6 兼容的快照时，您可以选择与 MySQL 5.7 或 MySQL 5.6 的兼容性。

您可以将 Aurora MySQL 1.14.*、1.15.*、1.16.*、1.17.*、1.18.*、2.01.* 和 2.02.* 快照还原到 Aurora MySQL 2.02.4 中。您还可以执行从 Aurora MySQL 2.01.* 或 2.02.* 到 Aurora MySQL 2.02.4 的就地升级。

我们不允许将 Aurora MySQL 1.* 集群就地升级到 Aurora MySQL 2.02.4 或通过 Amazon S3 备份还原到 Aurora MySQL 2.02.4。我们计划在以后的 Aurora MySQL 2.* 版本中删除这些限制。

该版本的 Aurora MySQL 5.7 已禁用性能架构。请升级到 Aurora 2.03 以提供性能架构支持。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

改进

- 修复了与从 Aurora MySQL 5.6 快照还原的表上的全文搜索索引有关的稳定性问题。

集成了 MySQL 社区版本错误修复

- BUG#13651665 INNODB MAY BE UNABLE TO LOAD TABLE DEFINITION AFTER RENAME
- BUG#21371070 INNODB: CANNOT ALLOCATE 0 BYTES.
- BUG#21378944 FTS ASSERT ENC.SRC_ILIST_PTR != NULL, FTS_OPTIMIZE_WORD(), OPTIMIZE TABLE
- BUG#21508537 ASSERTION FAILURE UT_A(!VICTIM_TRX->READ_ONLY)
- BUG#21983865 UNEXPECTED DEADLOCK WITH INNODB_AUTOINC_LOCK_MODE=0
- BUG#22679185 INVALID INNODB FTS DOC ID DURING INSERT
- BUG#22899305 GCOLS: ASSERTION: !(COL->PRTYPE & 256).

- BUG#22956469 MEMORY LEAK INTRODUCED IN 5.7.8 IN MEMORY/INNODB/OS0FILE
- BUG#22996488 CRASH IN FTS_SYNC_INDEX WHEN DOING DDL IN A LOOP
- BUG#23014521 GCOL:INNODB: ASSERTION: !IS_V
- BUG#23021168 REPLICATION STOPS AFTER TRX IS ROLLED BACK ASYNC
- BUG#23052231 ASSERTION: ADD_AUTOINC < DICT_TABLE_GET_N_USER_COLS
- BUG#23149683 ROTATE INNODB MASTER KEY WITH KEYRING_OKV_CONF_DIR MISSING: SIGSEGV; SIGNAL 11
- BUG#23762382 INSERT VALUES QUERY WITH JOIN IN A SELECT CAUSES INCORRECT BEHAVIOR
- BUG#25209512 CURRENT_TIMESTAMP PRODUCES ZEROS IN TRIGGER
- BUG#26626277 BUG IN "INSERT... ON DUPLICATE KEY UPDATE" QUERY
- BUG#26734162 INCORRECT BEHAVIOR WITH INSERT OF BLOB + ON DUPLICATE KEY UPDATE
- BUG#27460607 INCORRECT WHEN INSERT SELECT'S SOURCE TABLE IS EMPTY

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持, 但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 异步键预取 (AKP)。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于同步调用 AWS Lambda 函数的本机函数。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息, 请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)”](#)。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

MySQL 5.7 兼容性

Aurora MySQL 2.02.4 与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

Aurora MySQL 2.02.4 当前不支持以下 MySQL 5.7 功能：

- 全局事务标识符 (GTID)。Aurora MySQL 在版本 2.04 以及更高版本中支持 GTID。
- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句
- X 协议

Aurora MySQL 数据库引擎更新 2018-08-23 (版本 2.02.3) (已弃用)

版本：2.02.3

Aurora MySQL 2.02.3 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

在创建新的 Aurora MySQL 数据库集群时，您可以选择与 MySQL 5.7 或 MySQL 5.6 的兼容性。在还原与 MySQL 5.6 兼容的快照时，您可以选择与 MySQL 5.7 或 MySQL 5.6 的兼容性。

您可以将 Aurora MySQL 1.14.*、1.15.*、1.16.*、1.17.*、2.01.* 和 2.02.* 快照还原到 Aurora MySQL 2.02.3 中。您还可以执行从 Aurora MySQL 2.01.* 或 2.02.* 到 Aurora MySQL 2.02.3 的就地升级。

我们不允许将 Aurora MySQL 1.* 集群就地升级到 Aurora MySQL 2.02.3 或通过 Amazon S3 备份还原到 Aurora MySQL 2.02.3。我们计划在以后的 Aurora MySQL 2.* 版本中删除这些限制。

该版本的 Aurora MySQL 5.7 已禁用性能架构。请升级到 Aurora 2.03 以提供性能架构支持。

Note

此版本目前不在 AWS GovCloud (美国西部) [-us-gov-west 1] 和中国 (北京) [cn-north-1] 区域推出。在提供后, 将发布单独的公告。

如果您有任何疑问或疑虑, 可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

与 Aurora MySQL 版本 1 进行比较

以下 Amazon Aurora MySQL 功能在 Aurora MySQL 版本 1 (兼容 MySQL 5.6) 中受支持, 但这些功能目前在 Aurora MySQL 版本 2 (兼容 MySQL 5.7) 中不受支持。

- 异步键预取 (AKP)。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于同步调用 AWS Lambda 函数的本机函数。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息, 请参阅“[Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)](#)”。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

目前, Aurora MySQL 2.01 不支持在 Aurora MySQL 1.16 和更高版本中添加的功能。有关 Aurora MySQL 1.16 版的信息, 请参阅[Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)](#)。

MySQL 5.7 兼容性

Aurora MySQL 2.02.3 与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

Aurora MySQL 2.02.3 当前不支持以下 MySQL 5.7 功能：

- 全局事务标识符 (GTID)。Aurora MySQL 在版本 2.04 以及更高版本中支持 GTID。
- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句
- X 协议

Aurora MySQL 2.x 和 Aurora MySQL 1.x 之间的 CLI 差异

- Aurora MySQL 2.x 的引擎名称为 `aurora-mysql`；Aurora MySQL 1.x 的引擎名称仍为 `aurora`。
- Aurora MySQL 2.x 的默认参数组为 `default.aurora-mysql5.7`；Aurora MySQL 1.x 的默认参数组仍为 `default.aurora5.6`。
- Aurora MySQL 2.x 的数据库集群参数组系列名称为 `aurora-mysql5.7`；Aurora MySQL 1.x 的数据库集群参数组系列名称仍为 `aurora5.6`。

有关完整的 CLI 命令集以及 Aurora MySQL 2.x 和 Aurora MySQL 1.x 之间的差异，请参阅 Aurora 文档。

改进

- 修复了一个问题，即，在读取记录时，使用乐观游标还原时 Aurora 副本可能会重新启动。

- 将 `innodb_stats_persistent_sample_pages` 参数的默认值更新为 128 以改进索引统计数据。
- 修复了一个问题，即，在访问在 Aurora 主实例上并发修改的小型表时，Aurora 副本可能会重新启动。
- 修复了 `ANALYZE TABLE` 以停止刷新表定义缓存。
- 修复了一个问题，即，在将地理空间的点查询转换为搜索范围时，Aurora 主实例或 Aurora 副本可能会重新启动。

Aurora MySQL 数据库引擎更新 2018-06-04 (版本 2.02.2) (已弃用)

版本 : 2.02.2

Aurora MySQL 2.02.2 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

在创建新的 Aurora MySQL 数据库集群时，您可以选择与 MySQL 5.7 或 MySQL 5.6 的兼容性。在还原与 MySQL 5.6 兼容的快照时，您可以选择与 MySQL 5.7 或 MySQL 5.6 的兼容性。

您可以将 Aurora MySQL 1.14*、1.15*、1.16*、1.17*、2.01* 和 2.02 快照还原到 Aurora MySQL 2.02.2 中。您还可以执行从 Aurora MySQL 2.01* 或 2.02 到 Aurora MySQL 2.02.2 的就地升级。

我们不允许将 Aurora MySQL 1.* 集群就地升级到 Aurora MySQL 2.02.2 或通过 Amazon S3 备份还原到 Aurora MySQL 2.02.2。我们计划在以后的 Aurora MySQL 2.* 版本中删除这些限制。

该版本的 Aurora MySQL 5.7 已禁用性能架构。请升级到 Aurora 2.03 以提供性能架构支持。

Note

此版本目前不在 AWS GovCloud (美国西部) [-us-gov-west 1] 和中国 (北京) [cn-north-1] 区域推出。在提供后，将发布单独的公告。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

改进

关键修复：

- [CVE-2016-3486](#)

与 Aurora MySQL 5.6 进行比较

在 Aurora MySQL 5.6 中支持以下 Amazon Aurora MySQL 功能，但目前在 Aurora MySQL 5.7 中不支持这些功能。

- 异步键预取 (AKP)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于同步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息，请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)”](#)。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

目前，Aurora MySQL 2.01 不支持在 Aurora MySQL 1.16 和更高版本中添加的功能。有关 Aurora MySQL 1.16 版的信息，请参阅[Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)](#)。

MySQL 5.7 兼容性

Aurora MySQL 2.02.2 与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

Aurora MySQL 2.02.2 当前不支持以下 MySQL 5.7 功能：

- 全局事务标识符 (GTID)。Aurora MySQL 在版本 2.04 以及更高版本中支持 GTID。
- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件

- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句
- X 协议

Aurora MySQL 2.x 和 Aurora MySQL 1.x 之间的 CLI 差异

- Aurora MySQL 2.x 的引擎名称为 `aurora-mysql`；Aurora MySQL 1.x 的引擎名称仍为 `aurora`。
- Aurora MySQL 2.x 的默认参数组为 `default.aurora-mysql5.7`；Aurora MySQL 1.x 的默认参数组仍为 `default.aurora5.6`。
- Aurora MySQL 2.x 的数据库集群参数组系列名称为 `aurora-mysql5.7`；Aurora MySQL 1.x 的数据库集群参数组系列名称仍为 `aurora5.6`。

有关完整的 CLI 命令集以及 Aurora MySQL 2.x 和 Aurora MySQL 1.x 之间的差异，请参阅 Aurora 文档。

改进

- 修复了一个问题，即，在跟踪 Aurora 副本进度时 Aurora 写入器可能会偶尔重新启动。
- 修复了一个问题，即，在 Aurora 写入器上的表中运行索引创建或删除语句后，在访问分区表时，Aurora 副本重新启动或引发错误。
- 修复了一个问题，即，在应用在 Aurora 写入器上运行 ALTER table ADD/DROP column 语句而导致的更改时，无法访问 Aurora 副本中的表。

Aurora MySQL 数据库引擎更新 2018-05-03 (版本 2.02) (已弃用)

版本：2.02

Aurora MySQL 2.02 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

在创建新的 Aurora MySQL 数据库集群时，您可以选择与 MySQL 5.7 或 MySQL 5.6 的兼容性。在还原与 MySQL 5.6 兼容的快照时，您可以选择与 MySQL 5.7 或 MySQL 5.6 的兼容性。

您可以将 Aurora MySQL 1.14*、1.15*、1.16*、1.17* 和 2.01* 快照还原到 Aurora MySQL 2.02 中。您还可以执行从 Aurora MySQL 2.01* 到 Aurora MySQL 2.02 的就地升级。

我们不允许将 Aurora MySQL 1.x 集群就地升级到 Aurora MySQL 2.02 或通过 Amazon S3 备份还原到 Aurora MySQL 2.02。我们计划从以后的 Aurora MySQL 2.x 版本中删除这些限制。

该版本的 Aurora MySQL 5.7 已禁用性能架构。请升级到 Aurora 2.03 以提供性能架构支持。

如果您有任何疑问或疑虑，可以在社区论坛和 AWS 支持部门获得[AWS 支持](#)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

与 Aurora MySQL 5.6 进行比较

在 Aurora MySQL 5.6 中支持以下 Amazon Aurora MySQL 功能，但目前在 Aurora MySQL 5.7 中不支持这些功能。

- 异步键预取 (AKP)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于同步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息，请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)”](#)。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

目前，Aurora MySQL 2.01 不支持在 Aurora MySQL 1.16 和更高版本中添加的功能。有关 Aurora MySQL 1.16 版的信息，请参阅[Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)](#)。

MySQL 5.7 兼容性

Aurora MySQL 2.02 与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

Aurora MySQL 2.02 当前不支持以下 MySQL 5.7 功能：

- 全局事务标识符 (GTID)。Aurora MySQL 在版本 2.04 以及更高版本中支持 GTID。
- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句
- X 协议

Aurora MySQL 2.x 和 Aurora MySQL 1.x 之间的 CLI 差异

- Aurora MySQL 2.x 的引擎名称为 `aurora-mysql`；Aurora MySQL 1.x 的引擎名称仍为 `aurora`。
- Aurora MySQL 2.x 的默认参数组为 `default.aurora-mysql5.7`；Aurora MySQL 1.x 的默认参数组仍为 `default.aurora5.6`。
- Aurora MySQL 2.x 的数据库集群参数组系列名称为 `aurora-mysql5.7`；Aurora MySQL 1.x 的数据库集群参数组系列名称仍为 `aurora5.6`。

有关完整的 CLI 命令集以及 Aurora MySQL 2.x 和 Aurora MySQL 1.x 之间的差异，请参阅 Aurora 文档。

改进

- 修复了一个问题，即，在运行 INSERT 语句和利用快速插入优化时 Aurora 写入器重新启动。
- 修复了一个问题，即，在 Aurora 副本上运行 ALTER DATABASE 语句时，Aurora 副本重新启动。
- 修复了一个问题，即，在对 Aurora 写入器上刚删除的表运行查询时 Aurora 副本重新启动。
- 修复了一个问题，即在 Aurora 副本上将 `innodb_adaptive_hash_index` 设置为 OFF 时，Aurora 副本重新启动。

- 修复了一个问题，即，在 Aurora 写入器上运行 TRUNCATE TABLE 查询时，Aurora 副本重新启动。
- 修复了一个问题，即，在运行 INSERT 语句的某些情况下 Aurora 写入器冻结。在多节点集群上，这会导致故障转移。
- 修复了与设置会话变量关联的内存泄漏。
- 修复了一个问题，即，在特定情况下对具有生成的列的表执行清除撤消时 Aurora 写入器冻结。
- 修复了一个问题，即，在启用二进制日志记录时 Aurora 写入器有时可能会重新启动。

MySQL 错误修复汇总

- 左侧联接在外侧返回错误结果 (错误 22833364)。

Aurora MySQL 数据库引擎更新 2018-03-13 (版本 2.01.1) (已弃用)

版本：2.01.1

Aurora MySQL 2.01.1 已正式发布。Aurora MySQL 2.x 版与 MySQL 5.7 兼容，Aurora MySQL 1.x 版与 MySQL 5.6 兼容。

在创建新的 Aurora MySQL 数据库集群时，您可以选择与 MySQL 5.7 或 MySQL 5.6 的兼容性。在还原与 MySQL 5.6 兼容的快照时，您可以选择与 MySQL 5.7 或 MySQL 5.6 的兼容性。

您可以将 Aurora MySQL 1.14*、1.15*、1.16* 和 1.17* 快照还原到 Aurora MySQL 2.01.1 中。

我们不允许将 Aurora MySQL 1.x 集群就地升级到 Aurora MySQL 2.01.1 或通过 Amazon S3 备份还原到 Aurora MySQL 2.01.1。我们计划从以后的 Aurora MySQL 2.x 版本中删除这些限制。

该版本的 Aurora MySQL 5.7 已禁用性能架构。请升级到 Aurora 2.03 以提供性能架构支持。

与 Aurora MySQL 5.6 进行比较

在 Aurora MySQL 5.6 中支持以下 Amazon Aurora MySQL 功能，但目前在 Aurora MySQL 5.7 中不支持这些功能。

- 异步键预取 (AKP)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。

- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于同步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息，请参阅“[Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)](#)”。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

目前，Aurora MySQL 2.01 不支持在 Aurora MySQL 1.16 和更高版本中添加的功能。有关 Aurora MySQL 1.16 版的信息，请参阅[Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)](#)。

MySQL 5.7 兼容性

Aurora MySQL 2.01.1 与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

Aurora MySQL 2.01.1 当前不支持以下 MySQL 5.7 功能：

- 全局事务标识符 (GTID)。Aurora MySQL 在版本 2.04 以及更高版本中支持 GTID。
- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件
- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句
- X 协议

Aurora MySQL 2.x 和 Aurora MySQL 1.x 之间的 CLI 差异

- Aurora MySQL 2.x 的引擎名称为 `aurora-mysql`；Aurora MySQL 1.x 的引擎名称仍为 `aurora`。
- Aurora MySQL 2.x 的默认参数组为 `default.aurora-mysql5.7`；Aurora MySQL 1.x 的默认参数组仍为 `default.aurora5.6`。
- Aurora MySQL 2.x 的数据库集群参数组系列名称为 `aurora-mysql5.7`；Aurora MySQL 1.x 的数据库集群参数组系列名称仍为 `aurora5.6`。

有关完整的 CLI 命令集以及 Aurora MySQL 2.x 和 Aurora MySQL 1.x 之间的差异，请参阅 Aurora 文档。

改进

- 修复了使用 MySQL 5.7 兼容性还原与 MySQL 5.6 兼容的快照时错误创建 Aurora 特定的数据库权限的快照还原问题。
- 添加了对 1.17 快照还原的支持。

Aurora MySQL 数据库引擎更新 2018-02-06 (版本 2.01) (已弃用)

版本：2.01

Aurora MySQL 2.01 已正式发布。以后，Aurora MySQL 2.x 版将与 MySQL 5.7 兼容，Aurora MySQL 1.x 版将与 MySQL 5.6 兼容。

在创建新的 Aurora MySQL 数据库集群（包括从快照还原的集群）时，您可以选择与 MySQL 5.7 或 MySQL 5.6 的兼容性。

您可以将 Aurora MySQL 1.14*、1.15* 和 1.16* 快照还原到 Aurora MySQL 2.01 中。

我们不允许将 Aurora MySQL 1.x 集群就地升级到 Aurora MySQL 2.01 或通过 Amazon S3 备份还原到 Aurora MySQL 2.01。我们计划从以后的 Aurora MySQL 2.x 版本中删除这些限制。

该版本的 Aurora MySQL 5.7 已禁用性能架构。请升级到 Aurora 2.03 以提供性能架构支持。

与 Aurora MySQL 5.6 进行比较

在 Aurora MySQL 5.6 中支持以下 Amazon Aurora MySQL 功能，但目前在 Aurora MySQL 5.7 中不支持这些功能。

- 异步键预取 (AKP)。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用异步键预取优化 Aurora 索引的联接查询](#)。
- 哈希联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- 用于同步调用 AWS Lambda 函数的本机函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Aurora MySQL 本机函数调用 Lambda 函数](#)。
- 扫描批处理。有关更多信息，请参阅[“Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)”](#)。
- 使用 Amazon S3 存储桶从 MySQL 中迁移数据。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon S3 存储桶从 MySQL 迁移数据](#)。

目前，Aurora MySQL 2.01 不支持在 Aurora MySQL 1.16 和更高版本中添加的功能。有关 Aurora MySQL 1.16 版的信息，请参阅[Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)](#)。

MySQL 5.7 兼容性

Aurora MySQL 2.01 与 MySQL 5.7 数据兼容，包含 JSON 支持、空间索引及生成列等功能。Aurora MySQL 使用 Z 阶曲线原生实现了空间索引功能，使空间数据集的写入性能相比于 MySQL 5.7 提高了 20 倍以上，读取性能提高 10 倍以上。

Aurora MySQL 2.01 当前不支持以下 MySQL 5.7 功能：

- 全局事务标识符 (GTID)。Aurora MySQL 在版本 2.04 以及更高版本中支持 GTID。
- 组复制插件
- 增加的页面大小
- InnoDB 缓冲池启动时加载
- InnoDB 全文分析器插件
- 多源复制
- 在线缓冲池大小调整
- 密码验证插件

- 查询重写插件
- 复制筛选
- CREATE TABLESPACE SQL 语句
- X 协议

Aurora MySQL 2.x 和 Aurora MySQL 1.x 之间的 CLI 差异

- Aurora MySQL 2.x 的引擎名称为 `aurora-mysql`；Aurora MySQL 1.x 的引擎名称仍为 `aurora`。
- Aurora MySQL 2.x 的默认参数组为 `default.aurora-mysql5.7`；Aurora MySQL 1.x 的默认参数组仍为 `default.aurora5.6`。
- Aurora MySQL 2.x 的数据库集群参数组系列名称为 `aurora-mysql5.7`；Aurora MySQL 1.x 的数据库集群参数组系列名称仍为 `aurora5.6`。

有关完整的 CLI 命令集以及 Aurora MySQL 2.x 和 Aurora MySQL 1.x 之间的差异，请参阅 Aurora 文档。

Amazon Aurora MySQL 版本 1 的数据库引擎更新 (已弃用)

以下是 Amazon Aurora 版本 1 的数据库引擎更新：

- [Aurora MySQL 数据库引擎更新 2021-09-30 \(版本 1.23.4 \) \(已弃用 \)](#)
- [Aurora MySQL 数据库引擎更新 2021-06-28 \(版本 1.23.3 \) \(已弃用 \)](#)
- [Aurora MySQL 数据库引擎更新 2021-03-18 \(版本 1.23.2 \) \(已弃用 \)](#)
- [Aurora MySQL 数据库引擎更新 2020-11-24 \(版本 1.23.1 \) \(已弃用 \)](#)
- [Aurora MySQL 数据库引擎更新 2020-09-02 \(版本 1.23.0 \) \(已弃用 \)](#)
- [Aurora MySQL 数据库引擎更新 2021-06-03 \(版本 1.22.5 \) \(已弃用 \)](#)
- [Aurora MySQL 数据库引擎更新 2021-03-04 \(版本 1.22.4 \) \(已弃用 \)](#)
- [Aurora MySQL 数据库引擎更新 2020-11-09 \(版本 1.22.3 \) \(已弃用 \)](#)
- [Aurora MySQL 数据库引擎更新 2020-03-05 \(版本 1.22.2 \) \(已弃用 \)](#)
- [Aurora MySQL 数据库引擎更新 2019-12-23 \(版本 1.22.1 \) \(已弃用 \)](#)
- [Aurora MySQL 数据库引擎更新 2019-11-25 \(版本 1.22.0 \) \(已弃用 \)](#)
- [Aurora MySQL 数据库引擎更新 2019-11-25 \(版本 1.21.0 \) \(已弃用 \)](#)
- [Aurora MySQL 数据库引擎更新 2020-03-05 \(版本 1.20.1 \) \(已弃用 \)](#)
- [Aurora MySQL 数据库引擎更新 2019-11-11 \(版本 1.20.0 \) \(已弃用 \)](#)
- [Aurora MySQL 数据库引擎更新 2020-03-05 \(版本 1.19.6 \) \(已弃用 \)](#)
- [Aurora MySQL 数据库引擎更新 2019-09-19 \(版本 1.19.5 \) \(已弃用 \)](#)
- [Aurora MySQL 数据库引擎更新 2019-06-05 \(版本 1.19.2 \) \(已弃用 \)](#)
- [Aurora MySQL 数据库引擎更新 2019-05-09 \(版本 1.19.1 \) \(已弃用 \)](#)
- [Aurora MySQL 数据库引擎更新 2019-02-07 \(版本 1.19.0 \) \(已弃用 \)](#)
- [Aurora MySQL 数据库引擎更新 2018-09-20 \(版本 1.18.0 \) \(已弃用 \)](#)
- [Aurora MySQL 数据库引擎更新 2020-03-05 \(版本 1.17.9 \) \(已弃用 \)](#)
- [Aurora MySQL 数据库引擎更新 2019-01-17 \(版本 1.17.8 \) \(已弃用 \)](#)
- [Aurora MySQL 数据库引擎更新 2018-10-08 \(版本 1.17.7 \) \(已弃用 \)](#)
- [Aurora MySQL 数据库引擎更新 2018-09-06 \(版本 1.17.6 \) \(已弃用 \)](#)
- [Aurora MySQL 数据库引擎更新 2018-08-14 \(版本 1.17.5 \) \(已弃用 \)](#)
- [Aurora MySQL 数据库引擎更新 2018-08-07 \(版本 1.17.4 \) \(已弃用 \)](#)

- [Aurora MySQL 数据库引擎更新 2018-06-05 \(版本 1.17.3\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2018-04-27 \(版本 1.17.2\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2018-03-23 \(版本 1.17.1\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2018-03-13 \(版本 1.17\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2017-12-11 \(版本 1.16\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2017-11-20 \(版本 1.15.1\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 2017-10-24 \(版本 1.15\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 : 2018-03-13 \(版本 1.14.4\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 : 2017-09-22 \(版本 1.14.1\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 : 2017-08-07 \(版本 1.14\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 : 2017-05-15 \(版本 1.13\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 : 2017-04-05 \(版本 1.12\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 : 2017-02-23 \(版本 1.11\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 : 2017-01-12 \(版本 1.10.1\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 : 2016-12-14 \(版本 1.10\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 : 2016-11-10 \(版本 1.9.0、1.9.1\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 : 2016-10-26 \(版本 1.8.1\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 : 2016-10-18 \(版本 1.8\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 : 2016-09-20 \(版本 1.7.1\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 : 2016-08-30 \(版本 1.7.0\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 : 2016-06-01 \(版本 1.6.5\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 : 2016-04-06 \(版本 1.6\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 : 2016-01-11 \(版本 1.5\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 : 2015-12-03 \(版本 1.4\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 : 2015-10-16 \(版本 1.2、1.3\) \(已弃用\)](#)
- [Aurora MySQL 数据库引擎更新 : 2015-08-24 \(版本 1.1\) \(已弃用\)](#)

Aurora MySQL 数据库引擎更新 2021-09-30 (版本 1.23.4) (已弃用)

版本：1.23.4

Aurora MySQL 1.23.4 已正式发布。Aurora MySQL 2.* 版与 MySQL 5.7 兼容，Aurora MySQL 1.* 版与 MySQL 5.6 兼容。

此引擎版本计划于 2023 年 2 月 28 日弃用。有关更多信息，请参阅[准备终止使用 Amazon Aurora MySQL 兼容版的版本 1](#)。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

要使用较旧版本的 Aurora MySQL 创建集群，请通过 RDS 控制台、AWS CLI 或 Amazon RDS API 指定引擎版本。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

改进

常规改进：

- 修复了由于内部诊断日志文件中的信息性消息记录过多而导致读取器实例 CPU 占用过高的问题。

高优先级修复：

- [CVE-2021-2307](#)
- [CVE-2021-2226](#)
- [CVE-2021-2160](#)
- [CVE-2021-2154](#)
- [CVE-2021-2060](#)
- [CVE-2021-2032](#)
- [CVE-2021-2001](#)

Aurora MySQL 数据库引擎更新 2021-06-28 (版本 1.23.3) (已弃用)

版本：1.23.3

Aurora MySQL 1.23.3 已正式发布。Aurora MySQL 1.* 版与 MySQL 5.6 兼容，Aurora MySQL 2.* 版与 MySQL 5.7 兼容。

此引擎版本计划于 2023 年 2 月 28 日弃用。有关更多信息，请参阅[准备终止使用 Amazon Aurora MySQL 兼容版的版本 1](#)。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

要使用较旧版本的 Aurora MySQL 创建集群，请通过 RDS 控制台、AWS CLI 或 Amazon RDS API 指定引擎版本。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

改进

一般稳定性与可用性增强。

安全修复：

- [CVE-2021-23841](#)
- [CVE-2021-3449](#)
- [CVE-2020-28196](#)

Aurora MySQL 数据库引擎更新 2021-03-18 (版本 1.23.2) (已弃用)

版本：1.23.2

Aurora MySQL 1.23.2 已正式发布。Aurora MySQL 1.* 版与 MySQL 5.6 兼容，Aurora MySQL 2.* 版与 MySQL 5.7 兼容。

此引擎版本计划于 2023 年 2 月 28 日弃用。有关更多信息，请参阅[准备终止使用 Amazon Aurora MySQL 兼容版的版本 1](#)。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

要使用较旧版本的 Aurora MySQL 创建集群，请通过 RDS 控制台、AWS CLI 或 Amazon RDS API 指定引擎版本。

Note

此版本目前在以下区域中不可用：AWS GovCloud (美国东部) [us-gov-east-1]，AWS GovCloud (美国西部) [us-gov-west-1]。在提供后，将发布单独的公告。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

改进

高优先级修复：

- [CVE-2020-14867](#)
- [CVE-2020-14812](#)
- [CVE-2020-14769](#)
- [CVE-2020-14765](#)
- [CVE-2020-14793](#)
- [CVE-2020-14672](#)
- [CVE-2020-1971](#)
- [CVE-2018-3143](#)

可用性改进：

- 修复了动态调整集群存储大小功能中可能导致读取器数据库实例重启的问题。
- 修复了由 RESET QUERY CACHE 语句中的竞争条件导致的故障转移问题。
- 修复了带有查询缓存的嵌套存储流程调用中的崩溃问题。
- 修复了从未完成的分区表或子分区表截断中恢复时防止 mysqld 重复重启的问题。
- 修复了可能导致 MySQL 从本地或 RDS 迁移到 Aurora MySQL 失败的问题。
- 修复了在扩展存储卷期间数据库可能重启的极度竞争条件。

- 修复了锁管理器中的一个问题，其中，竞争条件可能导致两个事务共享一个锁，从而导致数据库重新启动。
- 修复了一个与事务锁内存管理（即，长时间运行的写事务导致数据库重新启动）相关的问题。
- 修复了锁定管理器中导致事务回滚期间数据库重新启动或故障转移的争用条件。
- 修复了从 5.6 升级到 5.7 的过程中，在 5.6 实验室模式下启用了快速在线 DDL 时出现的问题。
- 修复了多个问题，其中，引擎可能会在零停机时间修补期间，在检查数据库活动中的静止点以进行修补时重新启动。
- 修复了由于 DDL 操作中断而导致重复重启的多个相关问题，如 DROP TRIGGER、ALTER TABLE，特别是修改表中分区类型或分区数的 ALTER TABLE 问题。
- 更新了 16XL 和 24XL 实例上 table_open_cache 的默认值，以避免在大型实例类（R4/R5-16XL、R5-12XL、R5-24XL）上发生重复的重启和较高的 CPU 利用率。这影响了 1.21.x 和 1.22.x 版本。
- 修复了导致二进制日志副本停止并出现 HA_ERR_KEY_NOT_FOUND 错误的问题。

集成了 MySQL 社区版本错误修复

- 复制：在执行 SHOW BINLOG EVENTS 语句时，会阻止所有并行事务。此修复确保 SHOW BINLOG EVENTS 进程现在仅在计算文件终止位置期间锁定，因此不会长时间阻止并行交易。（错误 #76618、错误 #20928790）

Aurora MySQL 数据库引擎更新 2020-11-24（版本 1.23.1）（已弃用）

版本：1.23.1

Aurora MySQL 1.23.1 已正式发布。Aurora MySQL 1.* 版与 MySQL 5.6 兼容，Aurora MySQL 2.* 版与 MySQL 5.7 兼容。

此引擎版本计划于 2023 年 2 月 28 日弃用。有关更多信息，请参阅[准备终止使用 Amazon Aurora MySQL 兼容版的版本 1](#)。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

要使用较旧版本的 Aurora MySQL 创建集群，请通过 RDS 控制台、AWS CLI 或 Amazon RDS API 指定引擎版本。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

改进

安全修复：

对托管环境中的处理进行微调的修复和其他增强功能。其他 CVE 修复如下：

- [CVE-2020-14559](#)
- [CVE-2020-14539](#)

不兼容的更改：

此版本新增了影响 `mysqldump` 命令行为的权限更改。用户必须具有访问 `PROCESS` 表的 `INFORMATION_SCHEMA.FILES` 权限。要在不进行任何更改的情况下运行 `mysqldump` 命令，请向 `PROCESS` 命令连接的数据库用户授予 `mysqldump` 权限。您也可以运行带 `mysqldump` 选项的 `--no-tablespaces` 命令。使用该选项，`mysqldump` 输出不包含任何 `CREATE LOGFILE GROUP` 或 `CREATE TABLESPACE` 语句。在这种情况下，`mysqldump` 命令不访问 `INFORMATION_SCHEMA.FILES` 表，您也无需授予 `PROCESS` 权限。

可用性改进：

- 修复了导致运行 1.23.0 的全局数据库辅助集群中的 Aurora 读取器实例重复重新启动的问题。
- 修复了当主区域写入器使用较旧版本时，全局数据库辅助区域的副本可能会在升级到版本 1.23.0 时重新启动的问题。
- 修复了 Aurora MySQL 1.23.0 中引入的动态调整大小功能中的内存泄漏问题。
- 修复了在使用并行查询功能执行查询期间可能导致服务器重启的问题。
- 修复了当数据库引擎在从网络读取或写入网络时遇到错误时可能导致客户端会话挂起的问题。

Aurora MySQL 数据库引擎更新 2020-09-02 (版本 1.23.0) (已弃用)

版本：1.23.0

Aurora MySQL 1.23.0 已正式发布。Aurora MySQL 1.* 版与 MySQL 5.6 兼容，Aurora MySQL 2.* 版与 MySQL 5.7 兼容。

此引擎版本计划于 2023 年 2 月 28 日弃用。有关更多信息，请参阅[准备终止使用 Amazon Aurora MySQL 兼容版的版本 1](#)。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

您可以将 Aurora MySQL 1.* 数据库的快照还原为 Aurora MySQL 1.23.0。

Important

此版本中对 Aurora 存储的改进将可用升级路径限制为从 Aurora MySQL 1.23 到 Aurora MySQL 2.*。将 Aurora MySQL 1.23 集群升级到 2.* 时，您必须升级到 Aurora MySQL 2.09.0 或更新版本。

要使用较旧版本的 Aurora MySQL 创建集群，请通过 RDS 控制台、AWS CLI 或 Amazon RDS API 指定引擎版本。

Note

此版本目前在以下区域中不可用：AWS GovCloud (美国东部) [us-gov-east-1]，AWS GovCloud (美国西部) [us-gov-west-1]。在提供后，将发布单独的公告。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

改进

新功能：

- 现在，您可以通过更改数据库集群参数 `aurora_parallel_query` 的值来为现有集群打开或关闭并行查询。创建集群时，不需要使用 `parallelquery` 参数的 `--engine-mode` 设置。

并行查询现在已扩展到 Aurora MySQL 可用的所有区域。

对于在 Aurora 集群中升级和启用并行查询的过程，还有许多其他功能增强和更改。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon Aurora MySQL 的并行查询](#)。

- 通过此版本，您可以创建具有最多 128 TiB 存储空间的 Amazon Aurora MySQL 数据库实例。新的存储空间限制比之前的 64 TiB 有所增加。128 TiB 存储大小支持更大的数据库。小型实例大小

(db.t2 或 db.t3) 不支持此功能。由于 [InnoDB 具有页面大小为 16 KB 的限制](#)，因此单个表空间无法增长到 64 TiB 以上。

当集群卷大小接近 128 TiB 时，Aurora 发出警报，以便您可以在达到大小限制之前采取措施。警报显示在 mysql 日志和 AWS Management Console 的 RDS 事件中。

- 改进了二进制日志 (binlog) 处理，以便在涉及非常大的事务时缩短崩溃恢复时间和提交时间延迟。
- Aurora 动态调整集群存储空间的大小。通过动态调整大小，当您从 Aurora 数据库集群中删除数据时，该数据库集群的存储空间会自动减少。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[存储扩展](#)。

Note

动态调整大小功能正在分阶段部署到 Aurora 可用的 AWS 区域。根据集群所在的区域，此功能可能尚不可用。有关更多信息，请参阅[新增功能公告](#)。

高优先级修复：

- [CVE-2019-2911](#)
- [CVE-2019-2537](#)
- [CVE-2018-2787](#)
- [CVE-2018-2784](#)
- [CVE-2018-2645](#)
- [CVE-2018-2640](#)

可用性改进：

- 修复了锁管理器中的一个问题，其中，竞争条件可能导致两个事务共享一个锁，从而导致数据库重新启动。
- 修复了一个与事务锁内存管理（即，长时间运行的写事务导致数据库重新启动）相关的问题。
- 修复了锁定管理器中导致事务回滚期间数据库重新启动或故障转移的争用条件。
- 修复了从 5.6 升级到 5.7 的过程中，在启用了快速 DDL 的表上更改 `innodb_file_format` 时出现的一个问题。
- 修复了多个问题，其中，引擎可能会在零停机时间修补期间，在检查数据库活动中的静止点以进行修补时重新启动。

- 修复了一个与 DDL 恢复相关的问题，该问题影响数据库实例在恢复中断的 DROP TRIGGER 操作时重新启动。
- 修复了一个错误，在执行某些分区操作期间发生崩溃时，此错误可能导致数据库不可用。具体而言，一个中断的 ALTER TABLE 操作，此操作修改表中的分区类型或分区数。
- 修复 16XL 和 24XL 实例上 table_open_cache 的默认值，此值可能导致在大型实例类 (R4/R5-16XL、R5-12XL、R5-24XL) 上发生重复的故障转移和较高的 CPU 利用率。这影响了 1.21.x 和 1.22.x。

全局数据库：

- 在 Aurora 全局数据库中主要和辅助AWS区域的 MySQL INFORMATION_SCHEMA.REPLICA_HOST_STATUS 视图中填充缺少的数据。
- 修复了若干个意外查询失败：在主区域和辅助区域之间发生临时网络连接问题之后，由于对主区域中的 UNDO 记录进行垃圾回收，导致在全局数据库辅助区域中可能发生这些意外查询失败。

并行查询：

- 修复了并行查询可能导致长时间运行的查询返回空结果的问题。
- 修复了 Aurora 只读副本上针对小表的查询可能需要一秒以上时间的问题。
- 修复了当并行查询和 DML 语句在繁重的工作负载下同时执行时可能导致重新启动的问题。

常规改进：

- 修复了以下问题：如果在已存在较大空间值的表上创建了空间索引，则使用空间索引的查询可能会返回部分结果。
- 已将审计系统变量 server_audit_incl_users 和 server_audit_excl_users 允许的最大长度从 1024 字节增加到 2000 字节。
- 修复了以下问题：当 Aurora MySQL 二进制日志主服务器从 S3 中以 statement binlog_format 加载数据时，连接到 Aurora MySQL 二进制日志主服务器的二进制日志副本可能显示不完整的数据。
- 遵守社区行为以将 mixed binlog_format 映射到 row (而不是 statement) 以加载数据。
- 修复了以下问题：在用户关闭连接并且会话正在使用临时表时，导致二进制日志复制停止工作。
- 改进了涉及 MyISAM 临时表的查询的响应时间。
- 修复在二进制日志工作线程运行本机 lambda 函数时的权限问题。

- 修复了在尝试查询或轮换慢速日志或常规日志时 Aurora 只读副本上的一个问题。
- 修复了在主实例和副本上将 binlog_checksum 参数设置为不同值时断开逻辑复制的问题。
- 修复了只读副本可能暂时只看到写入器上最近提交的事务的部分结果的问题。
- 解决死锁时，在 show engine innodb status 中包含已回滚事务的事务信息。

集成了 MySQL 社区版本错误修复

- 具有 ALTER TABLE ADD COLUMN ALGORITHM=QUICK 的二进制日志事件将被重写为 ALGORITHM=DEFAULT，以便与社区版本兼容。
- 错误 #22350047 : IF CLIENT KILLED AFTER ROLLBACK TO SAVEPOINT PREVIOUS STMTS COMMITTED
- 错误 #29915479 : RUNNING COM_REGISTER_SLAVE WITHOUT COM_BINLOG_DUMP CAN RESULTS IN SERVER EXIT
- 错误 #30441969 : 错误 #29723340 : MYSQL SERVER CRASH AFTER SQL QUERY WITH DATA ?AST
- 错误 #30628268 : OUT OF MEMORY CRASH
- 错误 #27081349 : UNEXPECTED BEHAVIOUR WHEN DELETE WITH SPATIAL FUNCTION
- 错误 #27230859 : UNEXPECTED BEHAVIOUR WHILE HANDLING INVALID POLYGON"
- 错误 #27081349 : UNEXPECTED BEHAVIOUR WHEN DELETE WITH SPATIAL"
- 错误 #26935001 : ALTER TABLE AUTO_INCREMENT TRIES TO READ INDEX FROM DISCARDED TABLESPACE
- 错误 #29770705 : SERVER CRASHED WHILE EXECUTING SELECT WITH SPECIFIC WHERE CLAUSE
- 错误 #27659490 : SELECT USING DYNAMIC RANGE AND INDEX MERGE USE TOO MUCH MEMORY(OOM)
- 错误 #24786290 : REPLICATION BREAKS AFTER 错误 #74145 HAPPENS IN MASTER
- 错误 #27703912 : EXCESSIVE MEMORY USAGE WITH MANY PREPARE
- 错误 #20527363 : TRUNCATE TEMPORARY TABLE CRASH : !
DICT_TF2_FLAG_IS_SET(TABLE, DICT_TF2_TEMPORARY)
- 错误 #23103937 : PS_TRUNCATE_ALL_TABLES() DOES NOT WORK IN SUPER_READ_ONLY MODE

- 错误 #25053286 : USE VIEW WITH CONDITION IN PROCEDURE CAUSES INCORRECT BEHAVIOR (fixed in 5.6.36)
- 错误 #25586773 : INCORRECT BEHAVIOR FOR CREATE TABLE SELECT IN A LOOP IN SP (fixed in 5.6.39)
- 错误 #27407480 : AUTOMATIC_SP_PRIVILEGES REQUIRES NEED THE INSERT PRIVILEGES FOR MYSQL.USER TABLE
- 错误 #26997096 : relay_log_space 值不以同步的方式更新，因此其值有时大大高于中继日志使用的实际磁盘空间。
- 错误 #15831300 SLAVE_TYPE_CONVERSIONS=ALL_NON_LOSSY 无法按预期工作
- SSL 错误逆向移植错误 #17087862，错误 #20551271
- 错误 #16894092 : PERFORMANCE REGRESSION IN 5.6.6+ FOR INSERT INTO ... SELECT ... FROM (在 5.6.15 中已修复)。
- 移植与 SLAVE_TYPE_CONVERSIONS 相关的错误修复。

Aurora MySQL 数据库引擎更新 2021-06-03 (版本 1.22.5) (已弃用)

版本 : 1.22.5

Aurora MySQL 1.22.5 已正式发布。Aurora MySQL 1.* 版与 MySQL 5.6 兼容，Aurora MySQL 2.* 版与 MySQL 5.7 兼容。

此引擎版本计划于 2023 年 2 月 28 日弃用。有关更多信息，请参阅[准备终止使用 Amazon Aurora MySQL 兼容版的版本 1](#)。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

要使用较旧版本的 Aurora MySQL 创建集群，请通过 RDS 控制台、AWS CLI 或 Amazon RDS API 指定引擎版本。

Note

此版本被指定为长期支持 (LTS) 版本。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[Aurora MySQL 长期支持 \(LTS\) 版本](#)。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

改进

可用性改进：

- 解决了内部清理线程之间的并发冲突可能导致数据库停滞、随后重新启动或故障转移的问题。
- 解决了在 XA 事务处于准备状态时重新启动数据库，然后在提交或回滚这些事务之前再次重新启动可能导致集群不可用的问题。在此修复之前，您可以通过将集群还原到首次重新启动之前的某个时间点来解决此问题。
- 解决了如果在处理 DDL 语句时数据库重新启动，可能导致 InnoDB 清除被阻止的问题。受此影响，InnoDB 历史记录列表长度将增加，集群存储卷将持续增长直到填满为止，从而使数据库不可用。在此修复之前，您可以通过再次重新启动数据库以取消阻止清除来缓解此问题。

Aurora MySQL 数据库引擎更新 2021-03-04 (版本 1.22.4) (已弃用)

版本：1.22.4

Aurora MySQL 1.22.4 已正式发布。Aurora MySQL 1.* 版与 MySQL 5.6 兼容，Aurora MySQL 2.* 版与 MySQL 5.7 兼容。

此引擎版本计划于 2023 年 2 月 28 日弃用。有关更多信息，请参阅 [准备终止使用 Amazon Aurora MySQL 兼容版的版本 1](#)。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

要使用较旧版本的 Aurora MySQL 创建集群，请通过 RDS 控制台、AWS CLI 或 Amazon RDS API 指定引擎版本。

Note

此版本被指定为长期支持 (LTS) 版本。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [Aurora MySQL 长期支持 \(LTS\) 版本](#)。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

改进

安全修复：

对托管环境中的处理进行微调的修复和其他增强功能。其他 CVE 修复如下：

- [CVE-2020-14867](#)
- [CVE-2020-14812](#)
- [CVE-2020-14793](#)
- [CVE-2020-14769](#)
- [CVE-2020-14765](#)
- [CVE-2020-14672](#)
- [CVE-2020-1971](#)

可用性改进：

- 修复了在 `kill session` 命令期间可能会触发数据库重新启动或故障转移的问题。如果遇到此问题，请联系 AWS Support 以在您的实例上启用此修复程序。
- 改进了二进制日志处理，以便在涉及大的事务时缩短崩溃恢复时间和提交延迟。
- 修复了导致二进制日志副本停止并出现 `HA_ERR_KEY_NOT_FOUND` 错误的问题。

Aurora MySQL 数据库引擎更新 2020-11-09 (版本 1.22.3) (已弃用)

版本：1.22.3

Aurora MySQL 1.22.3 已正式发布。Aurora MySQL 1.* 版与 MySQL 5.6 兼容，Aurora MySQL 2.* 版与 MySQL 5.7 兼容。

此引擎版本计划于 2023 年 2 月 28 日弃用。有关更多信息，请参阅 [准备终止使用 Amazon Aurora MySQL 兼容版的版本 1](#)。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

要使用较旧版本的 Aurora MySQL 创建集群，请通过 RDS 控制台、AWS CLI 或 Amazon RDS API 指定引擎版本。

Note

此版本被指定为长期支持 (LTS) 版本。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [Aurora MySQL 长期支持 \(LTS\) 版本](#)。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

改进

安全修复：

对托管环境中的处理进行微调的修复和其他增强功能。其他 CVE 修复如下：

- [CVE-2020-14559](#)
- [CVE-2020-14539](#)
- [CVE-2020-2579](#)
- [CVE-2020-2812](#)
- [CVE-2020-2780](#)
- [CVE-2020-2763](#)

不兼容的更改：

此版本新增了影响 mysqldump 命令行为的权限更改。用户必须具有访问 PROCESS 表的 INFORMATION_SCHEMA.FILES 权限。要在不进行任何更改的情况下运行 mysqldump 命令，请向 PROCESS 命令连接的数据库用户授予 mysqldump 权限。您也可以运行带 mysqldump 选项的 --no-tablespaces 命令。使用该选项，mysqldump 输出不包含任何 CREATE LOGFILE GROUP 或 CREATE TABLESPACE 语句。在这种情况下，mysqldump 命令不访问 INFORMATION_SCHEMA.FILES 表，您也无需授予 PROCESS 权限。

可用性改进：

- 修复了在恢复未提交的 DDL 语句期间可能导致服务器重新启动的问题。

- 修复了锁定管理器中可能导致服务器重启的争用情况。
- 修复了可能导致监控代理在恢复大型事务期间重新启动服务器的问题

常规改进：

- 已将执行 MIXED 时的行为更改为将 binlog_format ROW 映射到 STATEMENT，而不是映射到 LOAD DATA FROM INFILE | S3。
- 修复了以下问题：当主服务器执行了 LOAD DATA FROM S3 且 binlog_format 设置为 STATEMENT 时，连接到 Aurora MySQL 二进制日志主服务器的二进制日志副本可能显示不完整的数据。

集成了 MySQL 社区版本错误修复

- 错误 #26654685：检查外键时遇到的损坏索引 ID 引发断言
- 错误 #15831300：默认情况下，当将整数从主节点上较小的类型提升到从属节点上的较大类型时（例如，从主节点上的 [SMALLINT](#) 列到从属节点上的 [BIGINT](#)）时，提升的值将被视为已经过签名的值。现在，在此情况下，可以使用为 [slave_type_conversions](#) 服务器系统变量指定的值集中的 ALL_SIGNED、ALL_UNSIGNED 中的一个或两个修改或覆盖此行为。有关详细信息，请参阅[基于行的复制：属性升级和降级](#)以及变量的说明。
- 错误 #17449901：使用 foreign_key_checks=0，InnoDB 允许删除外键约束所需的索引，将表置于不一致状态，并导致表加载时发生的外键检查失败。InnoDB 现在可以防止删除外键约束所需的索引，即使在 foreign_key_checks=0 的情况下亦是如此。在删除外键索引之前，必须删除外键约束。
- 错误 #20768847：[ALTER TABLE ... DROP INDEX](#) 对具有外键依赖项的表进行的操作会引发断言。

Aurora MySQL 数据库引擎更新 2020-03-05 (版本 1.22.2) (已弃用)

版本：1.22.2

Aurora MySQL 1.22.2 已正式发布。Aurora MySQL 1.* 版与 MySQL 5.6 兼容，Aurora MySQL 2.* 版与 MySQL 5.7 兼容。

此引擎版本计划于 2023 年 2 月 28 日弃用。有关更多信息，请参阅[准备终止使用 Amazon Aurora MySQL 兼容版的版本 1](#)。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

要使用较旧版本的 Aurora MySQL 创建集群，请通过 RDS 控制台、AWS CLI 或 Amazon RDS API 指定引擎版本。

Note

此版本目前在以下区域中不可用：AWS GovCloud (美国东部) [us-gov-east-1]，AWS GovCloud (美国西部) [us-gov-west-1]。在提供后，将发布单独的公告。

此版本被指定为长期支持 (LTS) 版本。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [Aurora MySQL 长期支持 \(LTS\) 版本](#)。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

改进

高优先级修复：

- 修复了证书轮换后间歇性连接故障的问题。
- 修复了在某些具有高写入负载的数据库集群上，导致克隆花费时间较长的问题。
- 修复了在主实例和副本上将 binlog_checksum 参数设置为不同值时断开逻辑复制的问题。
- 修复了慢日志和常规日志在只读副本上可能未正确轮换的问题。
- 修复了 ANSI Read Committed 隔离级别行为的问题。

Aurora MySQL 数据库引擎更新 2019-12-23 (版本 1.22.1) (已弃用)

版本：1.22.1

Aurora MySQL 1.22.1 已正式发布。Aurora MySQL 1.* 版与 MySQL 5.6 兼容，Aurora MySQL 2.* 版与 MySQL 5.7 兼容。

此引擎版本计划于 2023 年 2 月 28 日弃用。有关更多信息，请参阅 [准备终止使用 Amazon Aurora MySQL 兼容版的版本 1](#)。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

要使用较旧版本的 Aurora MySQL 创建集群，请通过AWS Management Console、AWS CLI 或 RDS API 指定引擎版本。您可以选择将现有 Aurora MySQL 1.* 数据库集群升级到 Aurora MySQL 1.22.1。

Note

此版本目前在以下AWS区域中不可用：AWS GovCloud (美国东部) [us-gov-east-1]、AWS GovCloud (美国西部) [us-gov-west-1]、中国 (宁夏) [cn-northwest-1]、亚太地区 (香港) [ap-east-1] 和中东 (巴林) [me-south-1]。在提供后，将发布单独的公告。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

升级数据库集群的过程已更改。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

关键修复：

- 修复了阻止引擎恢复涉及表锁和临时表的问题。
- 提高了使用临时表时二进制日志的稳定性。

高优先级修复：

- 修复了 Aurora 特定的数据库跟踪和日志记录子系统中的缓慢内存泄漏，此问题会降低可释放内存。

Aurora MySQL 数据库引擎更新 2019-11-25 (版本 1.22.0) (已弃用)

版本：1.22.0

Aurora MySQL 1.22.0 已正式发布。Aurora MySQL 1.* 版与 MySQL 5.6 兼容，Aurora MySQL 2.* 版与 MySQL 5.7 兼容。

此引擎版本计划于 2023 年 2 月 28 日弃用。有关更多信息，请参阅[准备终止使用 Amazon Aurora MySQL 兼容版的版本 1](#)。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

要使用较旧版本的 Aurora MySQL 创建集群，请通过 AWS Management Console、AWS CLI 或 RDS API 指定引擎版本。您可以选择将现有 Aurora MySQL 1.* 数据库集群升级到 Aurora MySQL 1.22.0。

Note

此版本目前在以下 AWS 区域中不可用：AWS GovCloud (美国东部) [us-gov-east-1]、AWS GovCloud (美国西部) [us-gov-west-1]、中国 (宁夏) [cn-northwest-1]、亚太地区 (香港) [ap-east-1]、中东 (巴林) [me-south-1] 和南美洲 (圣保罗) [sa-east-1]。在提供后，将发布单独的公告。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

升级数据库集群的过程已更改。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

新功能：

- Aurora MySQL 集群现在支持实例类型 r5.8xlarge、r5.16xlarge 和 r5.24xlarge。
- Binlog 具有新的增强功能，可在涉及非常大的事务时改进提交时间延迟。
- Aurora MySQL 现在有一种机制，能够使提交时将大型事务的事件写入 binlog 这一过程的时段最小化。这可以有效地防止在该时段内发生数据库崩溃时的漫长离线恢复。此功能还修复了大型事务在 binlog 提交时阻止小型事务的问题。此功能默认情况下关闭，如果您的工作负载需要，可由服务团队启用此功能。启用后，它将在事务大小 > 500 MB 时触发。

- 在只读副本上增加了对 ANSI READ COMMITTED 隔离级别的支持。此隔离级别支持对只读副本执行长时间运行的查询，而不会影响写入器节点上写入的高吞吐量。有关更多信息，请参阅 [Aurora MySQL 隔离级别](#)。
- 全局数据库现在允许为部署在以下AWS区域中的数据库集群添加辅助只读副本区域：美国东部（弗吉尼亚北部）[us-east-1]、美国东部（俄亥俄州）[us-east-2]、美国西部（加利福尼亚北部）[us-west-1]、美国西部（俄勒冈）[us-west-2]、欧洲（爱尔兰）[eu-west-1]、欧洲（伦敦）[eu-west-2]、欧洲（巴黎）[eu-west-3]、亚太地区（东京）[ap-northeast-1]、亚太地区（首尔）[ap-northeast-2]、亚太地区（新加坡）[ap-southeast-1]、亚太地区（悉尼）[ap-southeast-2]、加拿大（中部）[ca-central-1]、欧洲（法兰克福）[eu-central-1] 和亚太地区（孟买）[ap-south-1]。
- 热行争用功能现在正式推出，无需将 Aurora 实验室模式设置为 ON。在很多事务争用同一页面上的行时，该功能大大提高了工作负载的吞吐量。
- 此版本更新了时区文件，以支持针对新集群的最新巴西时区更新。

关键修复：

- [CVE-2019-2922](#)
- [CVE-2019-2923](#)
- [CVE-2019-2924](#)
- [CVE-2019-2910](#)

高优先级修复：

- [CVE-2019-2805](#)
- [CVE-2019-2730](#)
- [CVE-2019-2740](#)
- [CVE-2018-3064](#)
- [CVE-2018-3058](#)
- [CVE-2017-3653](#)
- [CVE-2017-3464](#)
- [CVE-2017-3244](#)
- [CVE-2016-5612](#)
- [CVE-2016-5439](#)
- [CVE-2016-0606](#)

- [CVE-2015-4904](#)
- [CVE-2015-4879](#)
- [CVE-2015-4864](#)
- [CVE-2015-4830](#)
- [CVE-2015-4826](#)
- [CVE-2015-2620](#)
- [CVE-2015-0382](#)
- [CVE-2015-0381](#)
- [CVE-2014-6555](#)
- [CVE-2014-4258](#)
- [CVE-2014-4260](#)
- [CVE-2014-2444](#)
- [CVE-2014-2436](#)
- [CVE-2013-5881](#)
- [CVE-2014-0393](#)
- [CVE-2013-5908](#)
- [CVE-2013-5807](#)
- [CVE-2013-3806](#)
- [CVE-2013-3811](#)
- [CVE-2013-3804](#)
- [CVE-2013-3807](#)
- [CVE-2013-2378](#)
- [CVE-2013-2375](#)
- [CVE-2013-1523](#)
- [CVE-2013-2381](#)
- [CVE-2012-5615](#)
- [CVE-2014-6489](#)
- 修复了 DDL 恢复组件中导致数据库停机时间延长的问题。应更新对包含 TRUNCATE TABLE 列的表执行 AUTO_INCREMENT 查询后变得不可用的集群。
- 修复了 DDL 恢复组件中导致数据库停机时间延长的问题。应更新对多个表并行执行 DROP TABLE 查询后变得不可用的集群。

一般稳定性修复：

- 修复了在长时间运行的事务期间导致只读副本重新启动的问题。遇到副本重新启动同时可用内存加速下降的客户应考虑升级到此版本。
- 修复了针对只读副本上的临时表执行嵌套查询时错误报告 ERROR 1836 的问题。
- 修复了当 Aurora 写入器实例上正在运行繁重写入工作负载时在 Aurora 读取器实例上出现的并行查询中止错误。
- 修复了导致配置为 binlog 主实例的数据库在运行繁重写入工作负载时重新启动的问题。
- 修复了重新启动引擎时长时间不可用的问题。这解决了缓冲池初始化中的一个问题。此问题很少发生，但可能会影响任何受支持的版本。
- 修复了在 `information_schema.replica_host_status` 表中生成不一致数据的问题。
- 修复了并行查询与标准执行路径之间的争用情况，该争用会导致读取器节点间歇性重新启动。
- 提高了在客户端连接数超过 `max_connections` 参数值时的数据库稳定性。
- 通过阻止不支持的 DDL 和 LOAD FROM S3 查询，提高了读取器实例的稳定性。

集成了 MySQL 社区版本错误修复

- 错误 #16346241 - SERVER CRASH IN ITEM_PARAM::QUERY_VAL_STR
- 错误 #17733850 - NAME_CONST() CRASH IN ITEM_NAME_CONST::ITEM_NAME_CONST()
- 错误 #20989615 - INNODB AUTO_INCREMENT PRODUCES SAME VALUE TWICE
- 错误 #20181776 - ACCESS CONTROL DOESN'T MATCH MOST SPECIFIC HOST WHEN IT CONTAINS WILDCARD
- 错误 #27326796 - MYSQL CRASH WITH INNODB ASSERTION FAILURE IN FILE PARS0PARS.CC
- 错误 #20590013 - IF YOU HAVE A FULLTEXT INDEX AND DROP IT YOU CAN NO LONGER PERFORM ONLINE DDL

Aurora MySQL 数据库引擎更新 2019-11-25 (版本 1.21.0) (已弃用)

版本：1.21.0

Aurora MySQL 1.21.0 已正式发布。Aurora MySQL 1.* 版与 MySQL 5.6 兼容，Aurora MySQL 2.* 版与 MySQL 5.7 兼容。

当前支持的 Aurora MySQL 版本有

1.14.*、1.15.*、1.16.*、1.17.*、1.18.*、1.19.*、1.20.*、1.21.*、1.22.*、2.01.*、2.02.*、2.03.*、2.04.*、2.05.*、2.06.* 和 2.07.*。要使用较旧版本的 Aurora MySQL 创建集群，请通过 AWS Management Console、AWS CLI 或 RDS API 指定引擎版本。您可以选择将现有 Aurora MySQL 1.* 数据库集群升级到 Aurora MySQL 1.21.0。

Note

此版本目前在以下 AWS 区域中不可用：AWS GovCloud (美国东部) [us-gov-east-1]、AWS GovCloud (美国西部) [us-gov-west-1]、中国 (宁夏) [cn-northwest-1]、亚太地区 (香港) [ap-east-1]、欧洲 (斯德哥尔摩) [eu-north-1] 和中东 (巴林) [me-south-1]。在提供后，将发布单独的公告。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

Note

升级数据库集群的过程已更改。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

关键修复：

- [CVE-2018-0734](#)
- [CVE-2019-2534](#)
- [CVE-2018-2612](#)
- [CVE-2017-3599](#)
- [CVE-2018-2562](#)
- [CVE-2017-3329](#)
- [CVE-2018-2696](#)

- [CVE-2015-4737](#)

高优先级修复：

- 强烈建议数据库大小接近 64 TiB 的客户升级到此版本，以避免由于稳定性错误影响接近 Aurora 存储限制的卷而导致停机。

一般稳定性修复：

- 修复了当 Aurora 写入器实例上正在运行大量写入工作负载时在 Aurora 读取器实例上出现的并行查询中止错误。
- 修复了 Aurora 读取器实例上的以下问题：当写入器实例有大量事务提交流量时，长时间运行事务期间空闲内存会减少。
- 当数据库重新启动或主机更换之后，现在会保留参数 `aurora_disable_hash_join` 的值。
- 修复了与全文搜索高速缓存相关的问题，该问题会导致 Aurora 实例用尽内存。使用全文搜索的客户应进行升级。
- 当已启用哈希联接功能且实例内存不足时提高数据库的稳定性。使用哈希联接的客户应进行升级。
- 修复了查询高速缓存中的问题，该问题中的“太多连接”错误会导致重启。
- 修正了 T2 实例上的空闲内存计算，以包括交换内存空间，防止不必要的重启。

集成了 MySQL 社区版本错误修复

- 错误 #19929406：HANDLE_FATAL_SIGNAL (SIG=11) IN __MEMMOVE_SSSE3_BACK FROM STRING::COPY
- 错误 #17059925：对于 [UNION](#) 语句，rows-examined 值计算错误。这表现为 Performance Schema 语句表（例如 [events_statements_current](#)）的 ROWS_EXAMINED 列的值太大。
- 错误 #11827369：一些带有 SELECT ... FROM DUAL 嵌套子查询的查询引发断言。
- 错误 #16311231：如果一个查询在 IN 子句中包含了子查询，而 WHERE 子句中包含 [XOR](#) 操作，则会返回错误的结果。

Aurora MySQL 数据库引擎更新 2020-03-05 (版本 1.20.1) (已弃用)

版本 : 1.20.1

Aurora MySQL 1.20.1 已正式发布。Aurora MySQL 1.* 版与 MySQL 5.6 兼容，Aurora MySQL 2.* 版与 MySQL 5.7 兼容。

当前支持的 Aurora MySQL 版本有

1.14.*、1.15.*、1.16.*、1.17.*、1.18.*、1.19.*、1.20.*、1.21.*、1.22.*、2.01.*、2.02.*、2.03.*、2.04.*、2.05.*、2.06.* 和 2.07.*。您可以将 Aurora MySQL 1.* 数据库的快照还原为 Aurora MySQL 1.20.1。

要使用较旧版本的 Aurora MySQL 创建集群，请通过 RDS 控制台、AWS CLI 或 Amazon RDS API 指定引擎版本。

Note

此版本目前在以下区域中不可用：AWS GovCloud (美国东部) [us-gov-east-1]，AWS GovCloud (美国西部) [us-gov-west-1]。在提供后，将发布单独的公告。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

改进

高优先级修复：

- 修复了证书轮换后间歇性连接故障的问题。
- 修复了与并发连接关闭相关的问题，该问题导致在繁重的工作负载下出现故障转移。

一般稳定性修复：

- 修复了在执行复杂查询期间发生的崩溃，该查询涉及在内部使用中间表的多表联接和聚合。

Aurora MySQL 数据库引擎更新 2019-11-11 (版本 1.20.0) (已弃用)

版本 : 1.20.0

Aurora MySQL 1.20.0 已正式发布。Aurora MySQL 1.* 版与 MySQL 5.6 兼容 , Aurora MySQL 2.* 版与 MySQL 5.7 兼容。

当前支持的 Aurora MySQL 版本有

1.14.*、1.15.*、1.16.*、1.17.*、1.18.*、1.19.*、1.20.*、2.01.*、2.02.*、2.03.* 和 2.04.*。要使用较旧版本的 Aurora MySQL 创建集群 , 请通过AWS Management Console、AWS CLI 或 RDS API 指定引擎版本。您可以选择将现有 Aurora MySQL 1.* 数据库集群 (最多 1.19.5) 升级到 Aurora MySQL 1.20.0。

Note

此版本目前在以下AWS区域中不可用 : AWS GovCloud (美国东部) [us-gov-east-1]、AWS GovCloud (美国西部) [us-gov-west-1]、中国 (宁夏) [cn-northwest-1]、亚太地区 (香港) [ap-east-1]、欧洲 (斯德哥尔摩) [eu-north-1] 和中东 (巴林) [me-south-1]。在提供后 , 将发布单独的公告。

如果您有任何问题或疑问 , 可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息 , 请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

升级数据库集群的过程已更改。有关更多信息 , 请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

关键修复 :

- [CVE-2018-0734](#)
- [CVE-2019-2534](#)
- [CVE-2018-2612](#)

- [CVE-2017-3599](#)
- [CVE-2018-2562](#)
- [CVE-2017-3329](#)
- [CVE-2018-2696](#)
- [CVE-2015-4737](#)

高优先级修复：

- 强烈建议数据库大小接近 64 TiB 的客户升级到此版本，以避免由于稳定性错误影响接近 Aurora 存储限制的卷而导致停机。

一般稳定性修复：

- 修复了当 Aurora 写入器实例上正在运行大量写入工作负载时在 Aurora 读取器实例上出现的并行查询中止错误。
- 修复了 Aurora 读取器实例上的以下问题：当写入器实例有大量事务提交流量时，长时间运行事务期间空闲内存会减少。
- 当数据库重新启动或主机更换之后，现在会保留参数 `aurora_disable_hash_join` 的值。
- 修复了与全文搜索高速缓存相关的问题，该问题会导致 Aurora 实例用尽内存。使用全文搜索的客户应进行升级。
- 当已启用哈希联接功能且实例内存不足时提高数据库的稳定性。使用哈希联接的客户应进行升级。
- 修复了查询高速缓存中的问题，该问题中的“太多连接”错误会导致重启。
- 修正了 T2 实例上的空闲内存计算，以包括交换内存空间，防止不必要的重启。

集成了 MySQL 社区版本错误修复

- 错误 #19929406：HANDLE_FATAL_SIGNAL (SIG=11) IN __MEMMOVE_SSSE3_BACK FROM STRING::COPY
- 错误 #17059925：对于 [UNION](#) 语句，rows-examined 值计算错误。这表现为 Performance Schema 语句表（例如 [events_statements_current](#)）的 ROWS_EXAMINED 列的值太大。
- 错误 #11827369：一些带有 SELECT ... FROM DUAL 嵌套子查询的查询引发断言。
- 错误 #16311231：如果一个查询在 IN 子句中包含了子查询，而 WHERE 子句中包含 [XOR](#) 操作，则会返回错误的结果。

Aurora MySQL 数据库引擎更新 2020-03-05 (版本 1.19.6) (已弃用)

版本 : 1.19.6

Aurora MySQL 1.19.6 已正式发布。Aurora MySQL 1.* 版与 MySQL 5.6 兼容，Aurora MySQL 2.* 版与 MySQL 5.7 兼容。

此引擎版本计划于 2023 年 2 月 28 日弃用。有关更多信息，请参阅[准备终止使用 Amazon Aurora MySQL 兼容版的版本 1](#)。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

您可以将 Aurora MySQL 1.* 数据库的快照还原为 Aurora MySQL 1.19.6。

要使用较旧版本的 Aurora MySQL 创建集群，请通过 RDS 控制台、AWS CLI 或 Amazon RDS API 指定引擎版本。

Note

此版本目前在以下区域中不可用：AWS GovCloud (美国东部) [us-gov-east-1]，AWS GovCloud (美国西部) [us-gov-west-1]。在提供后，将发布单独的公告。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

改进

高优先级修复：

- 修复了证书轮换后间歇性连接故障的问题。

Aurora MySQL 数据库引擎更新 2019-09-19 (版本 1.19.5) (已弃用)

版本 : 1.19.5

Aurora MySQL 1.19.5 已正式发布。Aurora MySQL 1.* 版与 MySQL 5.6 兼容，Aurora MySQL 2.* 版与 MySQL 5.7 兼容。

此引擎版本计划于 2023 年 2 月 28 日弃用。有关更多信息，请参阅[准备终止使用 Amazon Aurora MySQL 兼容版的版本 1](#)。

当前支持的 Aurora MySQL 版本有

1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.* 和 3.02.*。

您可以选择将现有数据库集群升级到 Aurora MySQL 1.19.5。您可以将 Aurora MySQL 1.14.*、1.15.*、1.16.*、1.17.*、1.18.*、1.19.1 和 1.19.2 的快照还原到 Aurora MySQL 1.19.5。

要使用 Aurora MySQL 的较旧版本，您可以通过 AWS Management Console、AWS CLI 或 RDS API 指定引擎版本，从而创建新的数据库集群。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

此版本目前在以下 AWS 区域中不可用：欧洲（伦敦）[eu-west-2]、AWS GovCloud（美国东部）[us-gov-east-1]、AWS GovCloud（美国西部）[us-gov-west-1]、中国（宁夏）[cn-northwest-1] 和亚太地区（香港）[ap-east-1]。在提供后，将发布单独的公告。

Note

升级数据库集群的过程已更改。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

- 修复了 Aurora 读取器实例上的以下问题：当写入器实例有大量事务提交流量时，长时间运行事务期间空闲内存会减少。
- 修复了当 Aurora 写入器实例上正在运行大量写入工作负载时在 Aurora 读取器实例上出现的并行查询中止错误。
- 当数据库重新启动或主机更换之后，现在会保留参数 `aurora_disable_hash_join` 的值。

- 修复了与全文搜索高速缓存相关的问题，该问题会导致 Aurora 实例用尽内存。
- 保留 160 GB 空间，以便在没有故障转移的情况下完成恢复工作流程，从而在卷大小接近 64 TiB 卷限制时提高数据库的稳定性。
- 当已启用哈希联接功能且实例内存不足时提高数据库的稳定性。
- 修正了空闲内存计算，以包括 T2 实例上导致它们过早重启的交换内存空间。
- 修复了查询高速缓存中的问题，该问题中的“太多连接”错误会导致重启。

集成了 MySQL 社区版本错误修复

- [CVE-2018-2696](#)
- [CVE-2015-4737](#)
- 错误 #19929406 : HANDLE_FATAL_SIGNAL (SIG=11) IN __MEMMOVE_SSSE3_BACK FROM STRING::COPY
- 错误 #17059925 : 对于 [UNION](#) 语句，rows-examined 值计算错误。这表现为 Performance Schema 语句表 (例如 [events_statements_current](#)) 的 ROWS_EXAMINED 列的值太大。
- 错误 #11827369 : 一些带有 SELECT ... FROM DUAL 嵌套子查询的查询引发断言。
- 错误 #16311231 : 如果一个查询在 IN 子句中包含了子查询，而 WHERE 子句中包含 [XOR](#) 操作，则会返回错误的结果。

Aurora MySQL 数据库引擎更新 2019-06-05 (版本 1.19.2) (已弃用)

版本 : 1.19.2

Aurora MySQL 1.19.2 已正式发布。可以在 1.17.8、1.19.0、1.19.1 或 1.19.2 中创建所有与 MySQL 5.6 兼容的新 Aurora MySQL 数据库集群 (包括从快照还原的集群)。您可以选择 (但不要求这样做) 将现有数据库集群升级到 Aurora MySQL 1.19.2。要使用旧版本，您可以在 Aurora MySQL 1.14.4、Aurora MySQL 1.15.1、Aurora MySQL 1.16、Aurora MySQL 1.17.8 或 Aurora MySQL 1.18 中创建新的数据库集群。您可以使用 AWS CLI 或 Amazon RDS API 并指定引擎版本来执行该操作。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

该版本目前在 AWS GovCloud (美国西部) [us-gov-west-1]、欧洲 (斯德哥尔摩) [eu-north-1]、中国 (宁夏) [cn-northwest-1] 和亚太地区 (香港) [ap-east-1] AWS区域中不可用。在提供后，将发布单独的公告。

Note

升级数据库集群的过程已更改。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

- 修复了在将数据从 Amazon S3 加载到 Aurora 时可能造成故障的问题。
- 修复了在将数据从 Aurora 上传到 Amazon S3 时可能造成故障的问题。
- 修复了会创建处于已终止状态的僵尸会话的问题。
- 修复了在处理网络协议管理错误时导致连接中止的问题。
- 修复了在处理分区表时可能导致崩溃的问题。
- 修复了与触发器创建的二进制日志复制有关的问题。

Aurora MySQL 数据库引擎更新 2019-05-09 (版本 1.19.1) (已弃用)

版本 : 1.19.1

Aurora MySQL 1.19.1 已正式发布。可以通过 1.17.8、1.19.0 或 1.19.1 创建所有与 MySQL 5.6 兼容的新 Aurora MySQL 数据库集群 (包括从快照还原的集群)。您可以选择 (但不要求这样做) 将现有数据库集群升级到 Aurora MySQL 1.19.1。要使用旧版本，您可以在 Aurora MySQL 1.14.4、Aurora MySQL 1.15.1、Aurora MySQL 1.16、Aurora MySQL 1.17.8 或 Aurora MySQL 1.18 中创建新的数据库集群。您可以使用 AWS CLI 或 Amazon RDS API 并指定引擎版本来执行该操作。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

该版本目前在 AWS GovCloud (美国西部) [us-gov-west-1] 和中国 (北京) [cn-north-1] 区域中不可用。在提供后，将发布单独的公告。

Note

升级数据库集群的过程已更改。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

改进

- 修复了二进制日志复制中的一个错误，该错误可能导致配置为二进制日志工作线程的 Aurora 实例出现问题。
- 修复了处理某类 ALTER TABLE 命令时发生的错误。
- 修复了由于网络协议管理存在错误导致连接中止的问题。

Aurora MySQL 数据库引擎更新 2019-02-07 (版本 1.19.0) (已弃用)

版本 : 1.19.0

Aurora MySQL 1.19.0 已正式发布。可以在 1.17.8 或 1.19.0 中创建所有与 MySQL 5.6 兼容的新 Aurora MySQL 数据库集群 (包括从快照还原的集群)。您可以选择 (但不要求这样做) 将现有数据库集群升级到 Aurora MySQL 1.19.0。要使用旧版本，您可以在 Aurora MySQL 1.14.4、Aurora MySQL 1.15.1、Aurora MySQL 1.16、Aurora MySQL 1.17.8 或 Aurora MySQL 1.18.0 中创建新的数据库集群。您可以使用 AWS CLI 或 Amazon RDS API 并指定引擎版本来执行该操作。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

Note

该版本目前在 AWS GovCloud (美国西部) [us-gov-west-1] 和中国 (北京) [cn-north-1] 区域中不可用。在提供后，将发布单独的公告。

Note

升级数据库集群的过程已更改。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[升级 Aurora MySQL 数据库集群的次要版本或补丁程序级别](#)。

特征

- Aurora 版本选择器 - 从 Aurora MySQL 1.19.0 开始，您可以在 Amazon RDS 控制台上从多个与 MySQL 5.6 兼容的 Aurora 版本中选择。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[通过 AWS 检查或指定 Aurora MySQL 引擎版本](#)。

改进

- 修复了与 Aurora 副本上的 CHECK TABLE 查询相关的稳定性问题。
- 引入了新的全局用户变量 aurora_disable_hash_join 来禁用哈希联接。
- 修复了多个表哈希联接期间生成输出行时的稳定性问题。
- 修复了哈希联接适用性检查期间由于计划更改返回错误结果的问题。
- 通过长时间运行事务支持零停机时间修补。此增强功能会在从版本 1.19 升级到更高版本时生效。
- 现在在启用二进制日志时支持零停机时间。此增强功能会在从版本 1.19 升级到更高版本时生效。
- 修复了导致 Aurora 副本上出现与工作负载无关的 CPU 使用率峰值的问题。
- 修复了锁定管理器中导致数据库重启的竞争情况。
- 修复了锁定管理器组件中存在的竞争情况，提高了 Aurora 实例的稳定性。
- 提高了锁定管理器组件内死锁探测器的稳定性。
- INSERT 如果 InnoDB 检测到索引已损坏，将禁止对表执行操作。
- 修复了 Fast DDL 的稳定性问题。
- 通过减少单行子查询的扫描批处理的内存消耗提高了 Aurora 的稳定性。
- 修复了当系统变量 foreign_key_checks 设置为 0 时删除外键后出现的稳定性问题。

- 修复了错误覆盖了对用户输入的 `table_definition_cache` 值进行的更改的 Out Of Memory Avoidance 功能的问题。
- 修复了 Out Of Memory Avoidance 功能的稳定性问题。
- 修复了将 `query_time` 中的 `lock_time` 和 `slow_query_log` 设置为垃圾值的问题。
- 修复了在内部不当处理字符串排序规则触发的并行查询的稳定性问题。
- 修复了二级索引搜索触发的并行查询的稳定性问题。
- 修复了多表更新触发的并行查询的稳定性问题。

集成了 MySQL 社区版本错误修复

- 错误 #32917 : DETECT ORPHAN TEMP-POOL FILES, AND HANDLE GRACEFULLY
- 错误 #63144 CREATE TABLE IF NOT EXISTS METADATA LOCK IS TOO RESTRICTIVE

Aurora MySQL 数据库引擎更新 2018-09-20 (版本 1.18.0) (已弃用)

版本 : 1.18.0

Aurora MySQL 1.18.0 已正式发布。将在 Aurora MySQL 1.18.0 中创建所有与 MySQL 5.6 兼容的新 Aurora MySQL 并行查询集群 (包括从快照还原的集群)。您可以选择 (但不要求这样做) 将现有并行查询集群升级到 Aurora MySQL 1.18.0。您可以在 Aurora MySQL 1.14.4、Aurora MySQL 1.15.1、Aurora MySQL 1.16 或 Aurora MySQL 1.17.6 中创建新数据库集群。您可以使用 AWS CLI 或 Amazon RDS API 并指定引擎版本来执行该操作。

对于 Aurora MySQL 1.18.0 版，我们将使用集群修补模式，在该模式下，将同时修补 Aurora 数据库集群中的所有节点。

Important

Aurora MySQL 1.18.0 仅适用于 Aurora 并行查询集群。如果您升级预配置的 5.6.10a 集群，则生成的版本为 1.17.8。如果您升级并行查询 5.6.10a 集群，则生成的版本为 1.18.0。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

特征

- 并行查询在此版本中可用，适用于新集群和恢复的快照。Aurora MySQL 并行查询是一种优化功能，它并行处理在处理数据密集型查询时涉及的一些 I/O 和计算。并行处理的工作包括从存储中检索行，提取列值以及确定哪些行与 WHERE 子句和联接子句中的条件匹配。该数据密集型工作将委派（在数据库优化术语中为“向下推送”）给 Aurora 分布式存储层中的多个节点。如果不使用并行查询，每个查询将所有扫描的数据传输到 Aurora MySQL 集群中的单个节点（头节点），并在此处执行所有查询处理。
- 如果启用了并行查询功能，Aurora MySQL 引擎将自动确定查询何时可以从中受益，而无需进行 SQL 更改（如提示或表属性）。

有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon Aurora MySQL 的并行查询](#)。

- OOM 避免：该功能监控系统内存，并跟踪各种数据库组件使用的内存。在系统内存不足后，它执行一系列操作以从跟踪的各种组件中释放内存，以尝试防止数据库出现内存不足 (OOM) 问题，从而避免数据库重新启动。将默认为 t2 实例启用该最大努力功能，并且可以通过名为 `aurora_oom_response` 的新实例参数在其他实例类上启用该功能。该实例参数使用一串以逗号分隔的操作，在内存不足时，实例将执行这些操作。有效的操作包括“print”、“tune”、“decline”、“kill_query”或这些操作的任意组合。任何空字符串表示不应执行任何操作，实际上是将该功能禁用。请注意，该功能的默认操作是“print, tune”。用法示例：
 - “print”– 仅输出占用大量内存的查询。
 - “tune”– 调整内部表缓存以将一些内存释放回系统。
 - “decline”– 在实例内存不足后，拒绝新的查询。
 - “kill_query”– 按内存消耗量降序终止查询，直到实例内存高于下限阈值。不会终止数据定义语言 (DDL) 语句。
 - “print, tune”– 执行为“print”和“tune”描述的操作。
 - “tune, decline, kill_query”– 执行为“tune”、“decline”和“kill_query”描述的操作。

有关处理内存不足状况的信息及其他故障排除建议，请参阅《Amazon Aurora 用户指南》中的[Amazon Aurora MySQL 内存不足问题](#)。

Aurora MySQL 数据库引擎更新 2020-03-05 (版本 1.17.9) (已弃用)

版本：1.17.9

Aurora MySQL 1.17.9 已正式发布。Aurora MySQL 1.* 版与 MySQL 5.6 兼容，Aurora MySQL 2.* 版与 MySQL 5.7 兼容。

当前支持的 Aurora MySQL 版本有

1.14.*、1.15.*、1.16.*、1.17.*、1.18.*、1.19.*、1.20.*、1.21.*、1.22.*、2.01.*、2.02.*、2.03.*、2.04.*、2.05.*、2.06.* 和 2.07.*。您可以将 Aurora MySQL 1.* 数据库的快照还原为 Aurora MySQL 1.17.9。

要使用较旧版本的 Aurora MySQL 创建集群，请通过 RDS 控制台、AWS CLI 或 Amazon RDS API 指定引擎版本。

Note

此版本目前在以下区域中不可用：AWS GovCloud (美国东部) [us-gov-east-1]，AWS GovCloud (美国西部) [us-gov-west-1]。在提供后，将发布单独的公告。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

改进

高优先级修复：

- 修复了证书轮换后间歇性连接故障的问题。

Aurora MySQL 数据库引擎更新 2019-01-17 (版本 1.17.8) (已弃用)

版本：1.17.8

Aurora MySQL 1.17.8 已正式发布。将在 Aurora MySQL 1.17.8 中创建所有与 MySQL 5.6 兼容的新 Aurora MySQL 数据库集群 (包括从快照还原的集群)。您可以选择 (但不要求这样做) 将现有数据库集群升级到 Aurora MySQL 1.17.8。要使用旧版本，您可以在 Aurora MySQL 1.14.4、1.15.1、1.16 或 1.17.7 中创建新的数据库集群。您可以使用 AWS CLI 或 Amazon RDS API 并指定引擎版本来执行该操作。

对于 Aurora MySQL 1.17.8 版，我们将使用集群修补模式，在该模式下，将同时修补 Aurora 数据库集群中的所有节点。

Note

该版本目前在 AWS GovCloud (美国西部) [us-gov-west-1] 和中国 (北京) [cn-north-1] 区域中不可用。在提供后，将发布单独的公告。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

改进

- 修复了重启后 Aurora 副本上的 CPU 利用率增加的性能问题。
- 修复了使用哈希连接的 SELECT 查询的稳定性问题。

集成了 MySQL 社区版本错误修复

- 错误 #13418638 : CREATE TABLE IF NOT EXISTS METADATA LOCK IS TOO RESTRICTIVE

Aurora MySQL 数据库引擎更新 2018-10-08 (版本 1.17.7) (已弃用)

版本 : 1.17.7

Aurora MySQL 1.17.7 已正式发布。将在 Aurora MySQL 1.17.7 中创建所有与 MySQL 5.6 兼容的新 Aurora MySQL 数据库集群 (包括从快照还原的集群)。您可以选择 (但不要求这样做) 将现有数据库集群升级到 Aurora MySQL 1.17.7。要使用旧版本，您可以在 Aurora MySQL 1.14.4、1.15.1、1.16 或 1.17.6 中创建新的数据库集群。您可以使用 AWS CLI 或 Amazon RDS API 并指定引擎版本来执行该操作。

对于 Aurora MySQL 1.17.7 版，我们将使用集群修补模式，在该模式下，将同时修补 Aurora 数据库集群中的所有节点。

Note

该版本目前在 AWS GovCloud (美国西部) [us-gov-west-1] 和中国 (北京) [cn-north-1] 区域中不可用。在提供后，将发布单独的公告。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

改进

- 已公开显示 InnoDB 状态变量 `innodb_buffer_pool_size` 以供客户修改。
- 修复了在故障转移期间在 Aurora 集群上出现的稳定性问题。
- 修复了在 TRUNCATE 操作失败后出现的 DDL 恢复问题，从而提高了集群可用性。
- 修复了与 DDL 操作触发的 `mysql.innodb_table_stats` 表更新有关的稳定性问题。
- 修复了在执行 DDL 操作后在查询缓存失效期间触发的 Aurora 副本稳定性问题。
- 修复了在后台的定期字典缓存移出期间由无效的内存访问触发的稳定性问题。

集成了 MySQL 社区版本错误修复

- 错误 16208542：在外键列上删除索引导致表丢失。
- 错误 76349：在 `add_derived_key()` 中发生内存泄漏。
- 错误 16862316：对于分区表，查询可能会返回不同的结果，具体取决于是否使用索引合并。
- 错误 17588348：在针对由 HASH 分区的表运行时，使用 `index_merge` 优化（请参阅 [索引合并优化](#)）的查询可能会返回无效的结果。

Aurora MySQL 数据库引擎更新 2018-09-06 (版本 1.17.6) (已弃用)

版本：1.17.6

Aurora MySQL 1.17.6 已正式发布。将在 Aurora MySQL 1.17.6 中创建所有与 MySQL 5.6 兼容的新 Aurora MySQL 数据库集群（包括从快照还原的集群）。您可以选择（但不要求这样做）将现有数据库集群升级到 Aurora MySQL 1.17.6。要使用旧版本，您可以在 Aurora MySQL 1.14.4、1.15.1、1.16 或 1.17.5 中创建新的数据库集群。您可以使用 AWS CLI 或 Amazon RDS API 并指定引擎版本来执行该操作。

对于 Aurora MySQL 1.17.6 版，我们将使用集群修补模式，在该模式下，将同时修补 Aurora 数据库集群中的所有节点。

Note

该版本目前在 AWS GovCloud (美国西部) [us-gov-west-1] 和中国 (北京) [cn-north-1] 区域中不可用。在提供后，将发布单独的公告。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

改进

- 修复了在 Aurora 写入器在同一个表上执行 DDL 操作时在 SELECT 查询的 Aurora 读取器上出现的稳定性问题。
- 修复了在为使用堆/内存引擎的临时表创建和删除 DDL 日志而导致的稳定性问题。
- 修复了在二进制日志主实例连接不稳定的情况下、复制 DDL 语句时在二进制日志工作线程上出现的稳定性问题。
- 修复了在写入到较慢的查询日志时遇到的稳定性问题。
- 修复了在副本状态表中出现的一个问题，即，显示不正确的 Aurora 读取器滞后信息。

集成了 MySQL 社区版本错误修复

- 对于重命名或更改 [BINARY](#) 列默认值的 [ALTER TABLE](#) 语句，修改是使用表副本完成的，而不是就位完成的。(错误 67141、错误 14735373、错误 69580、错误 17024290)
- 常规表和隐式为组的派生表之间的外部联接可能会导致服务器退出。(错误 16177639)

Aurora MySQL 数据库引擎更新 2018-08-14 (版本 1.17.5) (已弃用)

版本：1.17.5

Aurora MySQL 1.17.5 已正式发布。将在 Aurora MySQL 1.17.5 中创建所有与 MySQL 5.6 兼容的新 Aurora MySQL 数据库集群 (包括从快照还原的集群)。您可以选择 (但不要求这样做) 将现有数据库集群升级到 Aurora MySQL 1.17.5。要使用旧版本，您可以在 Aurora MySQL 1.14.4、1.15.1、1.16 或 1.17.4 中创建新的数据库集群。您可以使用 AWS CLI 或 Amazon RDS API 并指定引擎版本来执行该操作。

对于 Aurora MySQL 1.17.5 版，我们将使用集群修补模式，在该模式下，将同时修补 Aurora 数据库集群中的所有节点。

Note

该版本目前在 AWS GovCloud (美国西部) [us-gov-west-1] 和中国 (北京) [cn-north-1] 区域中不可用。在提供后，将发布单独的公告。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

改进

- 修复了一个问题，即，在使用零停机时间修补功能修补 Aurora 集群后，Aurora 写入器可能会重新启动。

Aurora MySQL 数据库引擎更新 2018-08-07 (版本 1.17.4) (已弃用)

版本：1.17.4

Aurora MySQL 1.17.4 已正式发布。将在 Aurora MySQL 1.17.4 中创建所有与 MySQL 5.6 兼容的新 Aurora MySQL 数据库集群 (包括从快照还原的集群)。您可以选择 (但不要求这样做) 将现有数据库集群升级到 Aurora MySQL 1.17.4。要使用旧版本，您可以在 Aurora MySQL 1.14.4、1.15.1、1.16 或 1.17.3 中创建新的数据库集群。您可以使用 AWS CLI 或 Amazon RDS API 并指定引擎版本来执行该操作。

对于 Aurora MySQL 1.17.4 版，我们将使用集群修补模式，在该模式下，将同时修补 Aurora 数据库集群中的所有节点。

Note

该版本目前在 AWS GovCloud (美国西部) [us-gov-west-1] 和中国 (北京) [cn-north-1] 区域中不可用。在提供后，将发布单独的公告。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

改进

- 复制改进：
 - 由于不将二进制日志记录传输到集群副本，减少了网络流量。默认情况下，启用此改进。
 - 由于压缩复制消息，减少了网络流量。默认情况下，为 8xlarge 和 16xlarge 实例类启用此改进。此类大型实例可以承受大量写入流量，从而产生实际复制消息网络流量。
 - 针对副本缓存查询的修复。
- 修复了一个问题，即，在使用 ORDER BY LOWER(*col_name*) 排序规则时 utf8_bin 可能产生不正确排序。
- 修复了 DDL 语句（尤其是 TRUNCATE TABLE）可能导致 Aurora 副本问题（包括表不稳定或缺失）的问题。
- 修复了一个问题，即，套接字在存储节点重新启动时处于半开状态。
- 提供了以下新的数据库集群参数：
 - aurora_enable_zdr – 在副本重新启动时，允许在 Aurora 副本上打开的连接保持活动状态。
 - aurora_enable_replica_log_compression – 启用复制负载压缩以提高主实例和 Aurora 副本之间的网络带宽利用率。
 - aurora_enable_repl_bin_log_filtering – 在主实例上为 Aurora 副本无法使用的复制记录启用筛选。

Aurora MySQL 数据库引擎更新 2018-06-05（版本 1.17.3）（已弃用）

版本：1.17.3

Aurora MySQL 1.17.3 已正式发布。将在 Aurora MySQL 1.17.3 中创建所有与 MySQL 5.6 兼容的新 Aurora MySQL 数据库集群（包括从快照还原的集群）。您可以选择（但不要求这样做）将现有数据库集群升级到 Aurora MySQL 1.17.3。您可以在 Aurora MySQL 1.14.4、Aurora MySQL 1.15.1 或 Aurora MySQL 1.16 中创建新数据库集群。您可以使用 AWS CLI 或 Amazon RDS API 并指定引擎版本来执行该操作。

对于 Aurora MySQL 1.17.3 版，我们将使用集群修补模式，在该模式下，将同时修补 Aurora 数据库集群中的所有节点。

Note

该版本目前在 AWS GovCloud (美国西部) [us-gov-west-1] 和中国 (北京) [cn-north-1] 区域中不可用。在提供后，将发布单独的公告。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

改进

- 修复了一个问题，即，在读取记录时，使用乐观游标还原时 Aurora 副本可能会重新启动。
- 修复了一个问题，即，在启用性能架构的情况下尝试终止 MySQL 会话 (kill "<session id>") 时，Aurora 写入器重新启动。
- 修复了一个问题，即，在计算垃圾回收阈值时 Aurora 写入器重新启动。
- 修复了一个问题，即，在日志应用程序中跟踪 Aurora 副本进度时，Aurora 写入器偶尔会重新启动。
- 修复了在关闭自动提交时的查询缓存问题，该问题可能会导致过时读取。

Aurora MySQL 数据库引擎更新 2018-04-27 (版本 1.17.2) (已弃用)

版本：1.17.2

Aurora MySQL 1.17.2 已正式发布。将在 Aurora MySQL 1.17.2 中创建所有与 MySQL 5.6 兼容的新 Aurora MySQL 数据库集群 (包括从快照还原的集群)。您可以选择 (但不要求这样做) 将现有数据库集群升级到 Aurora MySQL 1.17.2。您可以在 Aurora MySQL 1.14.4、Aurora MySQL 1.15.1 或 Aurora MySQL 1.16 中创建新数据库集群。您可以使用 AWS CLI 或 Amazon RDS API 并指定引擎版本来执行该操作。

对于 Aurora MySQL 1.17.2 版，我们将使用集群修补模式，在该模式下，将同时修补 Aurora 数据库集群中的所有节点。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

改进

- 修复了在某些 DDL 分区操作期间引起重新启动的问题。
- 修复了导致禁止通过本机 AWS Lambda 函数调用 Aurora MySQL 函数的问题。
- 修复了在 Aurora 副本上引起重启的缓存失效问题。
- 修复了引起重启的锁定管理器问题。

Aurora MySQL 数据库引擎更新 2018-03-23 (版本 1.17.1) (已弃用)

版本 : 1.17.1

Aurora MySQL 1.17.1 已正式发布。将在 Aurora MySQL 1.17.1 中创建所有新的数据库集群 (包括从快照还原的集群)。您可以选择 (但不要求这样做) 将现有数据库集群升级到 Aurora MySQL 1.17.1。您可以在 Aurora MySQL 1.15.1、Aurora MySQL 1.16 或 Aurora MySQL 1.17 中创建新数据库集群。您可以使用 AWS CLI 或 Amazon RDS API 并指定引擎版本来执行该操作。

对于 Aurora MySQL 1.17.1 版, 我们将使用集群修补模式, 在该模式下, 将同时修补 Aurora 数据库集群中的所有节点。此版本修复了一些已知引擎问题以及回归。

如果您有任何问题或疑问, 可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息, 请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

Note

在最新版本的 Aurora MySQL 引擎中存在一个问题。在升级到 1.17.1 后, 引擎版本错误地报告为 1.17。如果已升级到 1.17.1, 您可以在 AWS Management Console 中检查数据库集群的维护列以确认升级。如果显示 none, 则引擎已升级到 1.17.1。

改进

- 修复了二进制日志恢复中导致大型二进制日志索引文件的情况恢复时间更长的问题, 如果经常进行二进制日志轮换, 则可能发生这种情况。
- 修复了查询优化程序中为分区表生成了效率低下的查询计划的问题。
- 修复了查询优化程序中导致范围查询引起数据库引擎重新启动的问题。

Aurora MySQL 数据库引擎更新 2018-03-13 (版本 1.17) (已弃用)

版本 : 1.17

Aurora MySQL 1.17 已正式发布。Aurora MySQL 1.x 版仅与 MySQL 5.6 兼容，而与 MySQL 5.7 不兼容。将在 Aurora 1.17 中创建所有与 5.6 兼容的新数据库集群（包括从快照还原的集群）。您可以选择（但不要求这样做）将现有数据库集群升级到 Aurora 1.17。您可以在 Aurora 1.14.1、Aurora 1.15.1 或 Aurora 1.16 中创建新数据库集群。您可以使用 AWS CLI 或 Amazon RDS API 并指定引擎版本来执行该操作。

对于 Aurora 1.17 版，我们将使用集群修补模式，在该模式下，将同时修补 Aurora 数据库集群中的所有节点。我们支持零停机时间修补，这将通过修补过程尽力保留客户端连接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。

零停机时间修补

零停机时间修补 (ZDP) 功能尝试在引擎修补期间尽力保留客户端连接。有关 ZDP 的更多信息，请参阅《Amazon Aurora 用户指南》中的[使用零停机时间修补](#)。

新特征

- Aurora MySQL 现在支持锁定压缩，后者可优化锁定管理器的内存使用量。从 1.17 版开始，您可以在不启用实验室模式的情况下使用该功能。

改进

- 修复了一个主要在核心较少的实例上出现的问题：即使在数据库处于闲置状态时，单个核心也可能具有 100% 的 CPU 使用率。
- 改善了从 Aurora 集群获取二进制日志的性能。
- 修复了一个问题，即，Aurora 副本尝试将表统计数据写入持久性存储并崩溃。
- 修复了一个问题，即，查询缓存不按 Aurora 副本预期起作用。
- 修复了锁定管理器中导致引擎重启的竞争条件。
- 修复了导致引擎重启的只读、自动提交事务采取的锁定问题。

- 修复了一个问题，即，某些查询不会写入审核日志。
- 修复了故障转移后某些分区维护操作的恢复问题。

MySQL 错误修复汇总

- 使用复制筛选条件时 LAST_INSERT_ID 不正确复制 (错误 69861)
- 根据 INDEX_MERGE 设置的不同，查询返回不同的结果 (错误 16862316)
- 查询过程重新执行存储例程，查询计划效率低下 (错误 16346367)
- INNODB FTS：在 FTS_CACHE_APPEND_DELETED_DOC_IDS 中断言 (错误 18079671)
- 在 ALTER TABLE CHANGE COLUMN 中断言 RBT_EMPTY(INDEX_CACHE->WORDS) (错误 17536995)
- INNODB 全文搜索在涉及保存点时找不到记录 (错误 70333，错误 17458835)

Aurora MySQL 数据库引擎更新 2017-12-11 (版本 1.16) (已弃用)

版本：1.16

Aurora MySQL 1.16 已正式发布。将在 Aurora 1.16 中创建所有新的数据库集群 (包括从快照还原的集群)。您可以选择 (但不要求这样做) 将现有数据库集群升级到 Aurora 1.16。您可以在 Aurora 1.14.1 或 Aurora 1.15.1 中创建新数据库集群。您可以使用 AWS CLI 或 Amazon RDS API 并指定引擎版本来执行该操作。

对于 Aurora 1.16 版，我们将使用集群修补模式，在该模式下，将同时修补 Aurora 数据库集群中的所有节点。我们支持零停机时间修补，这将通过修补过程尽力保留客户端连接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。

零停机时间修补

零停机时间修补 (ZDP) 功能尝试在引擎修补期间尽力保留客户端连接。有关 ZDP 的更多信息，请参阅《Amazon Aurora 用户指南》中的[使用零停机时间修补](#)。

新特征

- Aurora MySQL 现在支持通过本机函数 `lambda_sync()` 同步调用 AWS Lambda。还提供了本机函数 `lambda_async()`，它可以替代异步 Lambda 调用的现有存储过程。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[从 Amazon Aurora MySQL 数据库集群中调用 Lambda 函数](#)。
- Aurora MySQL 现在支持哈希联接以加快 `equijoin` 查询速度。Aurora 基于成本的优化程序可以自动确定何时使用哈希联接，您也可以查询计划中强制使用该联接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用哈希联接优化大型 Aurora MySQL 联接查询](#)。
- Aurora MySQL 现在支持扫描批处理以大大加快内存中面向扫描的查询。该功能通过批处理提高了表完全扫描、索引完全扫描和索引范围扫描的性能。

改进

- 修复了一个问题，即，在对主节点上刚删除的表执行查询时只读副本崩溃。
- 修复了一个问题，即，在具有非常大的 FULLTEXT 索引数量的数据库集群上重新启动写进程时导致时间比预期恢复要长。
- 修复了一个问题，即，刷新二进制日志导致了二进制日志事件中的 LOST_EVENTS 事故。
- 修复了在启用性能架构时计划程序的稳定性问题。
- 修复了一个问题，即，使用临时表的子查询可能会返回部分结果。

MySQL 错误修复汇总

无

Aurora MySQL 数据库引擎更新 2017-11-20 (版本 1.15.1) (已弃用)

版本 : 1.15.1

Aurora MySQL 1.15.1 已正式发布。将在 Aurora 1.15.1 中创建所有新的数据库集群 (包括从快照还原的集群)。您可以选择 (但不要求这样做) 将现有数据库集群升级到 Aurora 1.15.1。您可以在 Aurora 1.14.1 中创建新数据库集群。您可以使用 AWS CLI 或 Amazon RDS API 并指定引擎版本来执行该操作。

对于 Aurora 1.15.1 版，我们将使用集群修补模式，在该模式下，将同时修补 Aurora 数据库集群中的所有节点。我们支持零停机时间修补，这将通过修补过程尽力保留客户端连接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

零停机时间修补

零停机时间修补 (ZDP) 功能尝试在引擎修补期间**尽力**保留客户端连接。有关 ZDP 的更多信息，请参阅《Amazon Aurora 用户指南》中的[使用零停机时间修补](#)。

改进

- 修复了读取请求的自适应区段选择器中的问题，在特定情况下，该问题导致选择两次相同的区段而使读取延迟达到峰值。
- 修复了 Aurora MySQL 中的线程计划程序优化产生的问题。该问题的症状是，在写入较慢的日志时出现虚假的错误，而关联的查询本身可以正常执行。
- 修复了大型卷 (> 5 TB) 上的只读副本的稳定性问题。
- 修复了一个问题，即，由于虚假的未完成连接数而导致工作线程数持续增加。
- 修复了表锁定问题，这会导致在插入工作负载期间出现较长的信号量等待时间。
- 已恢复了 Aurora MySQL 1.15 中包含的以下 MySQL 错误修复：
 - MySQL 实例一直在“执行 SYNC 索引” (错误 73816)
 - 在 ALTER TABLE CHANGE COLUMN 中断言 RBT_EMPTY(INDEX_CACHE->WORDS) (错误 17536995)
 - InnoDB 全文搜索在涉及保存点时找不到记录 (错误 70333)

MySQL 错误修复汇总

无

Aurora MySQL 数据库引擎更新 2017-10-24 (版本 1.15) (已弃用)

版本 : 1.15

Aurora MySQL 1.15 已正式发布。将在 Aurora 1.15 中创建所有新的数据库集群（包括从快照还原的集群）。您可以选择（但不要求这样做）将现有数据库集群升级到 Aurora 1.15。您可以在 Aurora 1.14.1 中创建新数据库集群。您可以使用 AWS CLI 或 Amazon RDS API 并指定引擎版本来执行该操作。

对于 Aurora 1.15 版，我们将使用集群修补模式，在该模式下，将同时修补 Aurora 数据库集群中的所有节点。更新需要重新启动数据库，因此，将出现 20 到 30 秒的停机，之后，您可以继续使用数据库集群。如果您的数据库集群当前运行的是 Aurora 1.14 或 Aurora 1.14.1，则 Aurora MySQL 中的零停机时间修补功能可能允许与您的 Aurora MySQL 主实例的客户端连接持续到升级结束（具体取决于您的工作负载）。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [维护 Amazon Aurora 数据库集群](#)。

零停机时间修补

零停机时间修补 (ZDP) 功能尝试在引擎修补期间尽力保留客户端连接。有关 ZDP 的更多信息，请参阅《Amazon Aurora 用户指南》中的 [使用零停机时间修补](#)。

新特征

- 异步键预取 – 异步键预取 (AKP) 功能在需要键时预先从内存中提取这些键，从而提高了非缓存索引联接的性能。AKP 针对的一个主要使用案例是小型外部表和大型内部表之间的索引联接，其中索引在大型表中具有高选择性。此外，在启用多区间读 (MRR) 接口时，将利用 AKP 进行二级索引到主索引查找。在某些情况下，具有内存限制的小型实例能够利用 AKP，前提是提供正确的键基数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [使用异步键预取优化 Aurora 索引的联接查询](#)。
- 快速 DDL – 我们已将在 [Aurora 1.13](#) 中发布的功能扩展到包含默认值的操作。利用此扩展，快速 DDL 将适用于向表（带或不带默认值）结尾添加可为空的列的操作。该功能仍处于 Aurora 实验室模式。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [使用快速 DDL 在 Amazon Aurora 中修改表](#)。

改进

- 修复了优化 WITHIN/CONTAINS 空间查询（之前生成了一个空的结果集）期间出现的计算错误。
- 修复了 SHOW VARIABLE 命令以显示更新后的 innodb_buffer_pool_size 参数值（只要该值在参数组中发生变化）。
- 提高了在禁用自适应哈希索引且要插入的记录为页面上的第一条记录的情况下，对使用快速 DDL 更改的表进行批量插入期间主实例的稳定性。

- 在以下情况下提高了 Aurora 的稳定性：用户尝试将 `server_audit_events` 数据库集群参数值设置为 **default**。
- 修复了一个问题，即，在 Aurora 副本重新启动之前，未在这些副本上复制在 Aurora 主实例上运行的 ALTER TABLE 语句的数据库字符集更改。
- 通过修复主实例上的争用情况 (之前允许主实例注册 Aurora 副本，即使主实例已关闭其卷)，提高了稳定性。
- 通过更改锁定协议以在索引编制期间支持并发数据操作语言 (DML) 语句，提高了在大型表上创建索引期间的主实例性能。
- 修复了 ALTER TABLE RENAME 查询期间的 InnoDB 元数据不一致情况 (这将提高稳定性)。示例：在同一个 ALTER 语句中将表 `t1(c1, c2)` 的列循环重命名为 `t1(c2,c3)` 时。
- 在以下情况下提高了 Aurora 副本的稳定性：Aurora 副本没有活动工作负载，并且主实例停止响应。
- 在以下情况下提高了 Aurora 副本的可用性：Aurora 副本已明确锁定表，并阻止复制线程应用从主实例接收的任何 DDL 更改。
- 提高了在外键和列将同时从两个单独的会话中添加到表，并且已启用快速 DDL 的情况下，主实例的稳定性。
- 通过在清除撤消记录之前阻止截断该记录，提高了密集型写入工作负载期间主实例上的清除线程的稳定性。
- 通过修复删除表的事务提交过程中的锁释放顺序，提高了稳定性。
- 修复了导致数据库实例无法完成启动的 Aurora 副本缺陷并已端口 3306 已在使用中进行投诉。
- 修复了 SELECT 查询在某些 `information_schema` 表 (`innodb_trx`、`innodb_lock`、`innodb_lock_waits`) 上运行时导致集群稳定性降低的争用情况。

MySQL 错误修复汇总

- CREATE USER 接受插件和密码哈希，但忽略密码哈希 (错误 78033)
- 分区引擎将字段添加到读位集以便能够从分区索引返回排序的条目。这将促使联接缓冲区尝试读取不需要的字段。通过不向 `read_set` 添加所有分区字段，而只是对 `read_set` 中的前缀字段排序来修复该错误。添加了 `DEBUG_ASSERT`，如果执行 `key_cmp`，则至少必须读取第一个字段 (错误 16367691)。
- MySQL 实例一直在“执行 SYNC 索引” (错误 73816)
- 在 ALTER TABLE CHANGE COLUMN 中断言 `RBT_EMPTY(INDEX_CACHE->WORDS)` (错误 17536995)
- InnoDB 全文搜索在涉及保存点时找不到记录 (错误 70333)

Aurora MySQL 数据库引擎更新：2018-03-13（版本 1.14.4）（已弃用）

版本：1.14.4

Aurora MySQL 1.14.4 已正式发布。您可以在 Aurora 1.14.4 中创建新的数据库集群，方法是使用 AWS CLI 或 Amazon RDS API 并指定引擎版本。您可以选择（但不要求这样做）将现有 1.14.x 数据库集群升级到 Aurora 1.14.4。

对于 Aurora 1.14.4 版，我们将使用集群修补模式，在该模式下，将同时修补 Aurora 数据库集群中的所有节点。我们支持零停机时间修补，这将通过修补过程尽力保留客户端连接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

零停机时间修补

零停机时间修补 (ZDP) 功能尝试在引擎修补期间尽力保留客户端连接。有关 ZDP 的更多信息，请参阅《Amazon Aurora 用户指南》中的[使用零停机时间修补](#)。

新特征

- Aurora MySQL 现在支持 db.r4 实例类。

改进

- 修复了一个问题，即，写入大型二进制日志事件时生成 LOST_EVENTS。

MySQL 错误修复汇总

- 可忽略的事件不起作用，并且未经过测试（错误 74683）
- NEW->OLD ASSERT FAILURE 'GTID_MODE > 0' (Bug #20436436)

Aurora MySQL 数据库引擎更新：2017-09-22（版本 1.14.1）（已弃用）

版本：1.14.1

Aurora MySQL 1.14.1 已正式发布。将在 Aurora MySQL 1.14.1 中创建所有新的数据库集群（包括从快照还原的集群）。Aurora MySQL 1.14.1 还将对现有 Aurora MySQL 数据库集群进行强制升级。有关更多信息，请参阅AWS开发人员论坛网站上的[公告：Amazon Aurora 的强制升级计划期限延长](#)。

对于 Aurora MySQL 1.14.1 版，我们将使用集群修补模式，在该模式下，将同时修补 Aurora MySQL 数据库集群中的所有节点。更新需要重新启动数据库，因此，将出现 20 到 30 秒的停机，之后，您可以继续使用数据库集群。如果您的数据库集群当前运行的是 1.13 版或更高版本，则 Aurora MySQL 中的零停机时间修补功能可能允许与您的 Aurora MySQL 主要实例的客户端连接持续到升级结束（具体取决于您的工作负载）。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。

改进

- 修复了与插入和清除相关的争用条件，以提高仍处于 Aurora MySQL 实验室模式的快速 DDL 功能的稳定性。

Aurora MySQL 数据库引擎更新：2017-08-07（版本 1.14）（已弃用）

版本：1.14

Aurora MySQL 1.14 已正式发布。将在 Aurora MySQL 1.14 中创建所有新的数据库集群（包括从快照还原的集群）。Aurora MySQL 1.14 还将对现有 Aurora MySQL 数据库集群进行强制升级。我们将针对早期版本的 Aurora MySQL 的弃用时间表发布单独的公告。

对于 Aurora MySQL 1.14 版，我们将使用集群修补模式，在该模式下，将同时修补 Aurora 数据库集群中的所有节点。更新需要重新启动数据库，因此，将出现 20 到 30 秒的停机，之后，您可以继续使用数据库集群。如果您的数据库集群当前运行的是 1.13 版，Aurora 的零停机时间修补功能可能允许到 Aurora 主实例的客户端连接在升级期间保持活动状态，具体取决于您的工作负载。

如果您有任何问题或疑问，可通过社区论坛和 [AWS Support](#) 联系 AWS Support。

零停机时间修补

零停机时间修补 (ZDP) 功能尝试在引擎修补期间尽力保留客户端连接。有关 ZDP 的更多信息，请参阅《Amazon Aurora 用户指南》中的[使用零停机时间修补](#)。

改进

- 修复了当某个记录在辅助索引中发现但未在主索引中发现时出现的不正确的“未发现记录”错误。
- 修复了可能由在单独的写入跨越 32 页以上时非常强的某个防御性断言 (1.12 中新增) 导致的稳定性问题。此类情况可能在存在诸如较大的 BLOB 值时出现。
- 修复了因表空间缓存与目录缓存不一致而导致的稳定性问题。
- 修复了一个问题，即，Aurora 副本在超出连接主实例的最大尝试次数之后变得无响应。如果不活动的时间超出主实例用于运行状况检查的检测信号时长，Aurora 副本现在会重新启动。
- 修复了一个动态锁，如果一个连接尝试在发出命令 (如 ALTER TABLE) 时获取独占元数据锁 (MDL)，该动态锁可能在非常高的并发度下产生。
- 修复了在执行逻辑/并行预读时 Aurora 只读副本中发生的稳定性问题。
- 对 LOAD FROM S3 进行了以下两个方面的改进：
 1. 通过将软件开发工具包重试与现有重试结合使用来更好地处理 Amazon S3 超时错误。
 2. 通过缓存和重复使用客户端状态来优化加载超大型文件或大量文件时的性能。
- 修复了与针对 ALTER TABLE 运算的快速 DDL 有关的以下稳定性问题：
 1. 当 ALTER TABLE 语句有多个 ADD COLUMN 命令时，列名称不按升序排列。
 2. 要更新的列的名称字符串与该列从内部系统表中提取的对应名称字符串相差一个空终止字符 (/0)。
 3. 正在执行某些 B 树拆分运算。
 4. 当表具有长度可变的主键时的问题。
- 修复了由于实现 Aurora 副本的全文搜索 (FTS) 索引缓存与主实例的全文搜索索引缓存的一致性耗时过长而导致的副本稳定性问题。当主实例上的大部分新创建的 FTS 索引条目尚未刷新到磁盘中时可能出现这种情况。
- 修复了索引创建期间可能出现的稳定性问题。
- 新增了基础设施，用于跟踪每个连接和相关遥测的内存消耗以制定避免内存不足 (OOM) 的策略。
- 修复了一个问题，即，错误地在 Aurora 副本上允许使用 ANALYZE TABLE。现已阻止该操作。
- 修复了因逻辑预读和清除之间的争用情况导致的罕见死锁而造成的稳定性问题。

MySQL 错误修复汇总

- 与派生表 (FROM 子句中的子查询) 相结合的全文搜索导致服务器退出。现在，如果全文操作依赖于派生表，服务器将产生指示无法对具体化表执行全文搜索的错误。(错误 68751、错误 16539903)

Aurora MySQL 数据库引擎更新：2017-05-15 (版本 1.13) (已弃用)

版本：1.13

Note

我们在首次发布后在 Aurora MySQL 1.13 版本中启用了一项新功能 - SELECT INTO OUTFILE S3，并更新了发行说明来反映该更改。

Aurora MySQL 1.13 已正式发布。将在 Aurora MySQL 1.13 中创建所有新的数据库集群 (包括从快照还原的集群)。您可以选择 (但不要求这样做) 将现有数据库集群升级到 Aurora MySQL 1.13。对于 Aurora 1.13 版，我们将使用集群修补模式，在该模式下，将同时修补 Aurora 数据库集群中的所有节点。我们支持零停机时间修补，这将通过修补过程尽力保留客户端连接。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

零停机时间修补

零停机时间修补 (ZDP) 功能尝试在引擎修补期间尽力保留客户端连接。有关 ZDP 的更多信息，请参阅《Amazon Aurora 用户指南》中的[使用零停机时间修补](#)。

新功能：

- SELECT INTO OUTFILE S3 – Aurora MySQL 现在允许您将查询结果上传到 Amazon S3 存储桶中的一个或多个文件。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[将数据从 Amazon Aurora MySQL 数据库集群保存到 Amazon S3 存储桶中的文本文件](#)。

改进：

- 在引擎启动时对 CSV 格式的日志文件进行截断以避免较长的恢复时间。现在，`general_log_backup`、`general_log`、`slow_log_backup` 和 `slow_log` 表在数据库重新启动后不会保留。
- 修复了一个问题，即，无法迁移名为 **test** 的数据库。
- 通过重用正确的锁定段提高了锁定管理器的垃圾收集器的稳定性。
- 通过删除死锁检测算法中的无效断言提高了锁定管理器的稳定性。
- 已重新启用异步复制，并且修复了导致在无负载或只读工作负载下报告错误的副本滞后的相关问题。在 1.10 版中引入了复制管道改进。引入这些改进是为了将日志流更新应用于 Aurora 副本的缓冲区缓存，这有助于提高 Aurora 副本的读取性能和稳定性。
- 修复了一个问题，即，`autocommit=OFF` 导致在服务器重新启动之前，计划事件被阻止以及长时间运行的事务保持打开状态。
- 修复了一个问题，即，一般日志、审核日志和慢速查询日志无法记录由异步提交处理的查询。
- 将逻辑预读 (LRA) 功能的性能提高了最多 2.5 倍。通过允许预提取在 B 树中的中间页面之间继续来做到这一点。
- 增加了面向审核变量的参数验证 (用于去除不必要的空格)。
- 修复了 Aurora MySQL 版本 1.11 中引入的回归，其中，在使用 `SQL_CALC_FOUND_ROWS` 选项并调用 `FOUND_ROWS()` 函数时，查询会返回错误结果。
- 修复了元数据锁列表的格式错误时出现的稳定性问题。
- 提高了在将 `sql_mode` 设置为 `PAD_CHAR_TO_FULL_LENGTH` 并执行命令 `SHOW FUNCTION STATUS WHERE Db='string'` 时的稳定性。
- 修复了因错误的卷一致性检查导致实例在 Aurora 版本升级后无法启动的少数情况。
- 修复了 Aurora MySQL 1.12 版中出现的性能问题，即，在用户具有大量表时，Aurora 写入器的性能将下降。
- 改善了当 Aurora 写入器配置为二进制日志工作线程且连接数接近 16000 时的稳定性情况。
- 修复了一个极少出现的问题，即，在 Aurora 主实例上运行 DDL 期间，在阻塞连接以等待元数据锁定时，Aurora 副本可能会重新启动。

MySQL 错误修复汇总

- 对于空的 InnoDB 表，无法使用 `ALTER TABLE` 语句减小 `auto_increment` 值 (甚至在该表为空的情况下也是如此)。(错误 69882)

- 在具有全文搜索索引的 InnoDB 表上运行时，将长字符串作为 AGAINST() 参数的 MATCH() ...AGAINST 查询可能会导致出现错误。(错误 17640261)
- 将 SQL_CALC_FOUND_ROWS 与 ORDER BY 和 LIMIT 一起处理会产生错误的 FOUND_ROWS() 结果。(错误 68458、错误 16383173)
- 如果外键存在，ALTER TABLE 将不允许更改列的可为 Null 值性。(错误 77591)

Aurora MySQL 数据库引擎更新：2017-04-05 (版本 1.12) (已弃用)

版本：1.12

Aurora MySQL 1.12 现在是创建新数据库集群 (包括从快照还原集群) 的首选版本。

这不会强制升级现有集群。当我们在整个队列范围内完成对 1.11 版的修补后，您可以选择是否将现有集群升级到 1.12 版 (请参阅 Aurora 1.11 [发布说明](#)和相应的[论坛公告](#))。对于 Aurora 1.12 版，我们将使用集群修补模式，在该模式下，将同时修补 Aurora 数据库集群中的所有节点。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

新特征

- 快速 DDL – 现在，几乎可以使用 Aurora MySQL 立即执行 ALTER TABLE tbl_name ADD COLUMN col_name column_definition 操作。完成该操作无需复制表，对其他 DML 语句也没有重大影响。由于该操作不会因复制表而使用临时存储，使得 DDL 语句对于小型实例类中的大型表也很实用。目前，快速 DDL 仅支持将没有默认值且可为空的列添加到表的最后。目前该功能在 Aurora 实验室模式中可用。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[使用快速 DDL 在 Amazon Aurora 中修改表](#)。
- 显示卷状态 – 我们添加了新的监控命令 (SHOW VOLUME STATUS) 以显示卷中的节点和磁盘数量。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[显示 Aurora MySQL 数据库集群的卷状态](#)。

改进

- 对锁定压缩进行更改，进一步降低分配给每个锁定对象的内存。此改进在实验室模式中可用。
- 修复了一个问题，即，即使数据库处于空闲状态，trx_active_transactions 指标也会快速递减。
- 修复了在模拟磁盘和节点的故障时，关于错误注入查询语法的无效错误消息。

- 修复了与锁定管理器中的竞争条件和死锁相关的多个错误。
- 修复了在查询优化程序中导致缓冲区溢出的问题。
- 修复了当底层存储节点的可用内存较低时，Aurora 只读副本的稳定性问题。
- 修复了一个问题，即，在 `wait_timeout` 参数设置结束后，空闲连接持续存在。
- 修复了一个问题，即，在重启实例后，`query_cache_size` 返回意外值。
- 修复了在无法写入存储的事件中，由于诊断线程频繁探查网络而导致的性能问题。

MySQL 错误修复汇总

- 当因空值而导致 `AUTO_INCREMENT` 值被重置时，重新加载已移出的表。(错误 21454472、错误 77743)
- 由于 `purge_node_t structure` 中存在不一致，在回滚中未找到索引记录。产生不一致的来源是警告和错误消息，如“二级索引条目更新出错”、“无法清除记录”以及“尝试清除未标记为删除的二级索引条目”等。(错误 19138298、错误 70214、错误 21126772、错误 21065746)
- `qsort` 操作堆栈大小计算错误导致堆栈溢出。(错误 73979)
- 回滚时索引中未找到记录。(错误 70214、错误 72419)
- `ALTER TABLE` 添加列 `TIMESTAMP` 更新 `CURRENT_TIMESTAMP` 时插入了零数据 (错误 17392)

Aurora MySQL 数据库引擎更新：2017-02-23 (版本 1.11) (已弃用)

版本：1.11

我们将在最新版本发布后的较短时段内使用最新版本修补所有 Aurora MySQL 数据库集群。将使用旧过程修补数据库集群，停机时间为 5-30 秒。

在您为每个数据库实例指定的系统维护时段内执行修补。您可以使用 AWS Management Console 查看或更改此时段。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

或者，您也可以通过以下方法在 AWS Management Console 中立即应用修补程序：选择一个数据库集群，选择集群操作，然后选择立即升级。

对于 Aurora MySQL 1.11 版，我们将使用集群修补模式，在该模式下，将同时修补 Aurora 数据库集群中的所有节点。

新功能

- **LOAD DATA FROM S3 的 MANIFEST 选项** – LOAD DATA FROM S3 是在 1.8 版中发布的。已扩展该命令的选项，您现在可以使用清单文件指定要从 Amazon S3 加载到 Aurora 数据库集群的文件列表。这可让您轻松加载一个或多个位置的特定文件中的数据，与使用 FILE 选项加载单个文件中的数据或使用 PREFIX 选项加载具有相同位置和前缀的多个文件中的数据相反。清单文件格式与 Amazon Redshift 使用的格式相同。有关将 LOAD DATA FROM S3 与 MANIFEST 选项结合使用的更多信息，请参阅《Amazon Aurora 用户指南》中的[使用清单指定要加载的数据文件](#)。
- **默认启用空间索引** – 该功能是在 1.10 版的实验室模式中提供的，现在默认将其启用。空间索引将提高使用空间数据的查询在大型数据集上的查询性能。有关使用空间索引的更多信息，请参阅《Amazon Aurora 用户指南》中的[Amazon Aurora MySQL 和空间数据](#)。
- **高级审核计时更改** – 该功能是在 1.10.1 版中发布的，用于提供高性能设施以审核数据库活动。在此版本中，审核日志时间戳精度已从 1 秒更改为 1 微秒。更准确的时间戳可让您更好地了解审核事件的出现时间。有关审计的更多信息，请参阅《Amazon Aurora 用户指南》中的[在 Amazon Aurora MySQL 数据库集群中使用高级审计](#)。

改进

- 修改了 `thread_handling` 参数以禁止将其设置为 **multiple-connections-per-thread** (Aurora 的线程池支持的唯一模型) 之外的任何其他值。
- 修复了在将 `buffer_pool_size` 或 `query_cache_size` 参数设置为大于数据库集群的总内存的值时导致的问题。在此情况下，Aurora 将修改后的参数设置为默认值，以便数据库集群能够启动且不会发生崩溃。
- 修复了查询缓存中的问题，其中，当表在一个事务中失效的情况下，另一个事务将获得静态读取结果。
- 修复了一个问题，即，标记为删除的二进制日志文件在一小段延迟后被删除而不是立即被删除。
- 修复了一个问题，即，将使用名称 **tmp** 创建的数据视为在临时存储中存储的系统数据库，而不是永久保存到 Aurora 分布式存储的系统数据库。
- 修改了 SHOW TABLES 的行为以排除某些内部系统表。此更改可帮助避免因 mysqldump 锁定 SHOW TABLES 中列出的所有文件而导致的不必要的故障转移，这反而会阻止对内部系统表的写入，从而导致故障转移。
- 修复了一个问题，即，Aurora 副本在从一个查询 (此查询将调用其参数为 InnoDB 表列的函数) 创建临时表时重新启动。

- 修复了与 Aurora 副本节点中的元数据锁冲突有关的问题，该问题导致 Aurora 副本滞后于主数据库集群并最终重新启动。
- 修复了读取器节点中复制管道的死锁，这将导致 Aurora 副本滞后并最终重新启动。
- 修复了一个问题，即，Aurora 副本滞后过多且加密卷大于 1TB。
- 通过使用改进方式读取系统时钟时间改进了 Aurora 副本死锁检测。
- 修复了一个问题，即，Aurora 副本在读取方注销后重新启动两次而不是一次。
- 修复了在临时统计数据导致非唯一索引列上的统计数据不一致时，Aurora 副本上出现的低查询性能问题。
- 修复了一个问题，即，如果在 Aurora 副本处理相关的查询的同时在 Aurora 副本上复制 DDL 语句，Aurora 副本可能会发生崩溃。
- 已将版本 1.10 中引入的复制管道改进从默认启用更改为默认禁用。引入这些改进是为了将日志流更新应用于 Aurora 副本的缓冲区缓存，虽然该功能有助于提高 Aurora 副本的读取性能和稳定性，但它在某些工作负载条件下增加复制滞后时间。
- 修复了一个问题，即，在 DDL 事务的提交阶段，同时对同一个表执行持续性 DDL 语句和挂起的并行预读导致断言失败。
- 增强了一般日志和慢速查询日志以使数据库集群免于重新启动。
- 通过减少 ACL 模块中的内存使用量修复了某些长时间运行的查询的内存不足问题。
- 修复了在以下情况下出现的重新启动问题：表具有非空间索引，查询包含空间谓词，计划程序选择使用非空间索引，计划程序错误地将空间条件下推至索引。
- 修复了一个问题，即，数据库集群在删除、更新或清除外部存储的超大型地理空间对象 (例如 LOB) 时重新启动。
- 修复了一个问题，也即，使用 ALTER SYSTEM SIMULATE ... FOR INTERVAL 进行的故障模拟无法正常工作。
- 修复了因针对锁管理器中错误不变量的断言无效导致的稳定性问题。
- 禁用了对版本 1.10 中引入的 InnoDB 全文搜索功能的以下两项改进，因为这些改进导致一些严苛的工作负载出现稳定性问题：
 - 仅在针对 Aurora 副本的读取请求后更新缓存以便加快全文搜索索引缓存复制速度。
 - 在缓存大小超过总大小的 10% 时将缓存同步任务卸载到单独的线程，以避免 MySQL 查询在 FTS 缓存同步到磁盘期间卡顿太长时间。(错误 22516559，#73816)。

MySQL 错误修复汇总

- 将 ALTER table DROP foreign key 与另一个 DROP 操作一起运行会导致表消失。(错误 16095573)

- 一些使用 ORDER BY 的 INFORMATION_SCHEMA 查询不使用文件排序优化，因为它们之前已使用此优化。(错误 16423536)
- FOUND_ROWS () 返回错误的表行数。(错误 68458)
- 服务器失败，而不是在打开的临时表过多时显示错误。(错误 18948649)

Aurora MySQL 数据库引擎更新：2017-01-12 (版本 1.10.1) (已弃用)

版本：1.10.1

Aurora MySQL 版本 1.10.1 是可选择使用的版本，不用于修补数据库实例。它可用于创建新的 Aurora 实例和升级现有实例。您可以在 [Amazon RDS 控制台](#) 中选择一个集群，选择集群操作，然后选择立即升级以应用修补程序。修补要求数据库重新启动，停机时间通常持续 5-30 秒，在此之后可以继续使用数据库集群。此修补程序使用集群修补模式，Aurora 集群中的所有节点同时进行修补。

新功能

- 高级审核 – Aurora MySQL 提供了高性能的高级审核功能，可用于审核数据库活动。有关启用和使用高级审核的更多信息，请参阅《Amazon Aurora 用户指南》中的 [在 Aurora MySQL 数据库集群中使用高级审核](#)。

改进

- 修复在同一个语句中创建列并在其上添加索引时的空间索引问题。
- 修复了一个问题，即，在数据库集群重新启动时，不会永久保存空间统计数据。

Aurora MySQL 数据库引擎更新：2016-12-14 (版本 1.10) (已弃用)

版本：1.10

新功能

- 零停机时间修补 – 通过使用该功能，可以修补数据库实例而不会产生任何停机时间。也就是说，无需断开客户端应用程序或重启数据库即可执行数据库升级。此方法将提高 Aurora 数据库集群在维护

时段内的可用性。请注意，升级过程中将重置临时数据（例如，性能架构中的数据）。该功能适用于维护时段内的服务交付的修补程序以及用户启动的修补程序。

在启动修补程序时，服务可确保不存在未完成的锁定、事务或临时表，然后等待数据库可进行修补和重新启动的合适时段。虽然在修补期间（约 5 秒）吞吐量会下降，但将保留应用程序会话。如果找不到合适的时段，修补将默认为标准修补行为。

零停机时间修补将尽力实施，但受到某些限制，如下所述：

- 该功能当前适用于修补单节点数据库集群或多节点数据库集群中的写入器实例。
- SSL 连接不能与该功能结合使用。如果存在活动 SSL 连接，Amazon Aurora MySQL 将不会执行零停机时间修补，而是定期重试以查看 SSL 连接是否已终止。如果 SSL 连接已终止，零停机时间修补将继续。如果 SSL 连接在几秒后保留，带停机时间的标准修补将继续。
- 该功能在 Aurora 版本 1.10 和更高版本中可用。以后，我们将使用零停机时间修补来标识无法应用的任何版本或修补程序。
- 如果基于二进制日志记录的复制处于活动状态，则该功能将不适用。
- 空间索引 – 空间索引提高了使用空间数据的查询在大型数据集上的查询性能。有关使用空间索引的更多信息，请参阅《Amazon Aurora 用户指南》中的 [Amazon Aurora MySQL 和空间数据](#)。

默认情况下，该功能已禁用，可通过启用 Aurora 实验室模式将其激活。有关信息，请参阅《Amazon Aurora 用户指南》中的 [Amazon Aurora MySQL 实验室模式](#)。

- 复制管道改进 – Aurora MySQL 现在使用改进的机制，以将日志流更新应用于 Aurora 副本的缓冲区缓存。当主实例上的写入负载较重以及 Aurora 副本上的读取负载较重时，该功能可提高该副本的读取性能和稳定性。该功能已默认启用。
- 提高了具有缓存读取的工作负载的吞吐量 – Aurora MySQL 现在使用无锁定并发算法实施读取视图，这会提高缓冲区缓存提供的读取查询的吞吐量。作为此改进和其他改进的结果，对于仅 SysBench SELECT 工作负载，Amazon Aurora MySQL 可实现每秒最高 625K 的读取吞吐量，而 MySQL 5.7 的读取吞吐量只有每秒 164K。
- 提高了具有热行争用的工作负载的吞吐量 – Aurora MySQL 使用新的锁定释放算法以提高性能，特别是在出现热页面争用（即，多个事务争用同一页面上的行）情况时。在 TPC-C 基准测试中，这可能导致将吞吐量（每分钟事务数）提高到 MySQL 5.7 的 16 倍。默认情况下，该功能已禁用，可通过启用 Aurora 实验室模式将其激活。有关信息，请参阅《Amazon Aurora 用户指南》中的 [Amazon Aurora MySQL 实验室模式](#)。

改进

- 通过仅在针对 Aurora 副本的读取请求后更新缓存来加快全文搜索索引缓存复制速度。此方法可避免复制线程从磁盘进行任何读取。
- 修复了一个问题，即，字典缓存失效对数据库名称或表名称中具有特殊字符的表的 Aurora 副本不起作用。
- 修复了在启用存储热管理时分布式存储节点的数据迁移期间的 STUCK IO 问题。
- 修复了锁定管理器中的导致针对事务锁定等待线程的断言检查在准备回滚或提交事务时失败的问题。
- 修复了在通过正确更新对字典表条目的应用计数来打开损坏的字典时出现的问题。
- 修复了数据库集群最小读取点可由较慢的 Aurora 副本保留的错误。
- 修复了查询缓存中的潜在内存泄漏问题。
- 修复了 Aurora 副本在存储过程的 IF 语句中使用查询时对表施加行级别锁定的错误。

MySQL 错误修复汇总

- 对派生的表执行 UNION 将返回错误结果，显示 '1=0/false'-clauses。(错误 69471)
- 在第二次对存储过程执行 ITEM_FUNC_GROUP_CONCAT::FIX_FIELDS 时，服务器发生崩溃。(错误 20755389)
- 通过在缓存大小超过总大小的 10% 时将缓存同步任务卸载到单独的线程，来避免 MySQL 查询在 FTS 缓存同步到磁盘期间卡顿太长时间。(错误 22516559、#73816)

Aurora MySQL 数据库引擎更新：2016-11-10 (版本 1.9.0、1.9.1) (已弃用)

版本：1.9.0、1.9.1

新功能

- 改进了索引构建 – 现在使用自下而上的方式构建索引以实施二级索引，这会消除不必要的页面拆分。可将创建索引或重新构建表所需的时间减少高达 75%，具体取决于 db.r3.8xlarge 数据库实例类。该功能是在 Aurora MySQL 1.7 版的实验室模式中提供的，在 Aurora 1.9 和更高版本中默认将其启用。有关信息，请参阅《Amazon Aurora 用户指南》中的 [Amazon Aurora MySQL 实验室模式](#)。

- 锁定压缩 (实验室模式) – 该实施显著减少锁定管理器占用的内存量 (最多减少 66%)。锁定管理器可以获取更多行锁而不会遇到内存不足异常。默认情况下, 该功能已禁用, 可通过启用 Aurora 实验室模式将其激活。有关信息, 请参阅《Amazon Aurora 用户指南》中的 [Amazon Aurora MySQL 实验室模式](#)。
- 性能架构 – Aurora MySQL 现在包括性能架构支持, 并且对性能产生的影响非常小。在使用 SysBench 的测试中, 启用性能架构会将 MySQL 的性能降低高达 60%。

Aurora 数据库集群的 SysBench 测试显示, 对性能的影响比 MySQL 小 4 倍。即使启用性能架构, 运行 db.r3.8xlarge 数据库实例类每秒产生了 10 万次 SQL 写入和超过 55 万次 SQL 读取。

- 热行争用改进 – 该功能降低了 CPU 使用率, 并在大量连接访问少量热行时增加了吞吐量。该功能还可在出现热行争用时消除 error 188。
- 改进内部不足处理 – 在执行不必要的锁定 SQL 语句并超过保留的内存池限制时, Aurora 强制回滚这些 SQL 语句。该功能可释放内存, 并防止因内存不足异常而导致的引擎崩溃。
- 智能读取选择器 – 该实施在每次读取时从不同区段中选择最佳存储区段以减少读取延迟, 从而提高了读取吞吐量。SysBench 测试显示, 写入工作负载的性能最多提高 27%。

改进

- 修复了一个问题, 即, Aurora 副本在引擎启动期间遇到共享锁定。
- 修复了清除系统中的读取视图指针为 NULL 时, Aurora 副本可能发生崩溃的问题。

Aurora MySQL 数据库引擎更新: 2016-10-26 (版本 1.8.1) (已弃用)

版本: 1.8.1

改进

- 修复了一个问题, 即, 使用调用 AWS Lambda 过程的触发器批量插入失败。
- 修复了一个问题, 即, 全局关闭 autocommit 时目录迁移失败。
- 解决了使用 SSL 时连接 Aurora 失败的问题, 改进了 Diffie-Hellman 组, 使其可处理 LogJam 攻击。

MySQL 错误修复汇总

- 由于 LogJam 问题，OpenSSL 更改了 Diffie-Hellman 密钥长度参数。(错误 18367167)

Aurora MySQL 数据库引擎更新：2016-10-18 (版本 1.8) (已弃用)

版本：1.8

新功能

- AWS Lambda 集成 – 您现在可以使用 AWS Lambda 过程从 Aurora 数据库集群中异步调用 `mysql.lambda_async` 函数。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[从 Amazon Aurora MySQL 数据库集群中调用 Lambda 函数](#)。
- 从 Amazon S3 中加载数据 – 您现在可以使用 `LOAD DATA FROM S3` 或 `LOAD XML FROM S3` 命令，将文本或 XML 文件从 Amazon S3 存储桶加载到 Aurora 数据库集群中。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[将数据从 Amazon S3 存储桶中的文本文件加载到 Amazon Aurora MySQL 数据库集群](#)。
- 目录迁移 – Aurora 现在，在集群卷中永久保存目录元数据以支持版本控制。这样可以在不同版本和还原操作之间实现无缝的目录迁移。
- 集群级别的维护和修补 – Aurora 现在管理整个数据库集群的维护更新。有关更多信息，请参阅《Amazon Aurora 用户指南》中的[维护 Amazon Aurora 数据库集群](#)。

改进

- 修复了一个问题，即，当未授予对正在进行的 DDL 表的元数据锁定时 Aurora 副本崩溃。
- 允许 Aurora 副本修改非 InnoDB 表，从而方便 `log_output=TABLE` 的慢速日志和一般日志 CSV 文件的轮换。
- 修复了将统计数据从主实例更新到 Aurora 副本时的滞后情况。如果没有此修复，Aurora 副本统计数据可能会与主实例统计数据不同步，导致 Aurora 副本上存在不同的 (并可能导致性能欠佳) 查询计划。
- 修复了确保 Aurora 副本不获取锁定的竞用情况。
- 修复了 Aurora 副本无法注册到主实例或者无法从主实例中注销的罕见情况。
- 修复了在打开或关闭卷时可能导致 `db.r3.large` 实例发生死锁的争用情况。

- 修复了由于 Aurora 分布式存储服务中同时存在大量写入工作负载和失败，而导致发生内存不足的问题。
- 修复了由于长时间运行的事务造成的清除线程自旋而导致高 CPU 占用的问题。
- 修复了在高负载下为获取有关锁定的信息而运行信息架构查询时产生的问题。
- 修复了与诊断过程有关的问题，在罕见的情况下，该问题会导致 Aurora 对存储节点的写入操作发生停止并重新启动/故障转移。
- 修复了一个问题：如果在处理 CREATE TABLE [if not exists] 语句期间发生崩溃，则在崩溃恢复期间可能会删除已成功创建的表。
- 修复了一个问题：如果在使用目录迁移时一般日志和慢速日志未存储在磁盘上，则日志轮换过程会中断。
- 修复了当用户在用户定义函数中创建临时表，然后在查询的选择列表中使用该用户定义函数时发生的崩溃情况。
- 修复了在重放 GTID 事件时发生的崩溃。Aurora MySQL 不支持 GTID。

集成 MySQL 错误修复：

- 在具有多个索引的列上删除所有索引时，InnoDB 在外键约束需要索引的情况下无法阻止 DROP INDEX 操作。(错误 16896810)
- 解决了添加外键约束崩溃情况。(错误 16413976)
- 修复了在提取存储过程中的游标并同时分析或刷新表时发生的崩溃。(错误 18158639)
- 修复了当用户更改表来将 AUTO_INCREMENT 值更改为低于最大自动递增列值时出现的自动递增错误。(错误 16310273)

Aurora MySQL 数据库引擎更新：2016-09-20 (版本 1.7.1) (已弃用)

版本：1.7.1

改进

- 修复了一个问题，即，InnoDB 全文搜索缓存已满时 Aurora 副本崩溃。
- 修复了一个问题，即，进程池中的工作进程等待其自己时数据库引擎崩溃。
- 修复了一个问题，即，表的元数据锁导致死锁时 Aurora 副本崩溃。

- 修复了一个问题，即，线程池中的两个工作线程出现争用情况导致数据库引擎崩溃。
- 修复以下问题：监控代理未检测到对分布式存储子系统的写操作提前时，在高负载下进行不必要的故障转移。

Aurora MySQL 数据库引擎更新：2016-08-30（版本 1.7.0）（已弃用）

版本：1.7.0

新功能

- 支持 NUMA 的计划程序 – Aurora MySQL 引擎的任务计划程序现在支持非一致性内存访问 (NUMA)。这将最大程度地减少跨 CPU 套接字争用，从而增加 db.r3.8xlarge 数据库实例类的性能吞吐量。
- 并行预读在后台异步操作 – 已修订并行预读以使用专用线程减少线程争用，从而提高性能。
- 改进了索引构建（实验室模式）– 现在使用自下而上的方式构建索引以实施二级索引，这会消除不必要的页面拆分。这可减少创建索引或重新构建表所需的时间。默认情况下，该功能已禁用，可通过启用 Aurora 实验室模式将其激活。有关信息，请参阅《Amazon Aurora 用户指南》中的 [Amazon Aurora MySQL 实验室模式](#)。

改进

- 修复了一个问题，即，在为实例请求的连接数猛增的情况下建立连接需耗费长时间。
- 修复了一个问题，即，ALTER TABLE 在未使用 InnoDB 的分区表上运行时发生崩溃。
- 修复了一个问题，即，密集型写入工作负载可能导致故障转移。
- 修复了当 RENAME TABLE 在分区表上运行时可能导致故障的错误断言。
- 提高了在插入密集型工作负载期间回滚事务时的稳定性。
- 修复了一个问题，即，全文搜索索引对 Aurora 副本不可用。

MySQL 错误修复汇总

- 通过对 LOCK_grant 锁进行分区，提高了可扩展性。(端口 WL #8355)
- 存储过程中在 SELECT 上打开游标可能导致 segfault。(端口错误 16499751)

- MySQL 提供了有关一些特殊用法的错误结果。(错误 11751794)
- GET_SEL_ARG_FOR_KEYPART 中发生崩溃 – 由错误 11751794 的修补程序导致。(错误 16208709)
- 使用 GROUP BY 的简单查询的错误结果。(错误 17909656)
- 包含范围谓词的半联接查询上的额外行。(错误 16221623)
- 在 IN 子查询后添加 ORDER BY 子句可能导致返回重复行。(错误 16308085)
- 发生崩溃，包含针对带 GROUP BY、MyISAM 的松散扫描的查询的说明。(错误 16222245)
- 包含带引号 int 谓词的松散索引扫描将返回随机数据。(错误 16394084)
- 如果优化程序使用的是松散索引扫描，则服务器可能在尝试创建临时表时退出。(错误 16436567)
- COUNT(DISTINCT) 将不会计算 NULL 值，但会在优化程序使用松散索引扫描时计算这些值。(错误 17222452)
- 如果查询同时包含 MIN()/MAX() 和 aggregate_function(DISTINCT) (例如，SUM(DISTINCT))，并且是使用松散索引扫描执行的，则 MIN()/MAX() 的结果值的设置错误。(错误 17217128)

Aurora MySQL 数据库引擎更新：2016-06-01 (版本 1.6.5) (已弃用)

版本：1.6.5

新功能

- 高效的二进制日志存储 – 现在，默认为所有 Aurora MySQL 数据库集群启用高效的二进制日志存储，并且这是无法配置的。2016 年 4 月版更新中引入了二进制日志的高效存储。有关更多信息，请参阅[Aurora MySQL 数据库引擎更新：2016-04-06 \(版本 1.6\) \(已弃用\)](#)。

改进

- 提高了在主实例遇到密集型工作负载时 Aurora 副本的稳定性。
- 提高了在对已分区表和表名中包含特殊字符的表运行查询时 Aurora 副本的稳定性。
- 修复了使用安全连接时的连接问题。

MySQL 错误修复汇总

- SLAVE CAN'T CONTINUE REPLICATION AFTER MASTER'S CRASH RECOVERY (Port Bug #17632285)

Aurora MySQL 数据库引擎更新：2016-04-06（版本 1.6）（已弃用）

版本：1.6

此更新包括以下改进：

新功能

- 并行预读 – 现在，默认为所有 Aurora MySQL 数据库集群启用并行预读，并且这是无法配置的。2015 年 12 月版更新中引入了并行预读。有关更多信息，请参阅[“Aurora MySQL 数据库引擎更新：2015-12-03（版本 1.4）（已弃用）”](#)。

除了默认启用并行预读之外，此版本还包括对并行预读的改进：

- 改进了逻辑，以使并行预读不太主动，这在数据库集群遇到多个并行工作负载时很有用。
- 改进了小型表上的稳定性。
- 高效的二进制日志存储（实验室模式）– 现在，更高效地在 Aurora MySQL 中存储 MySQL 二进制日志文件。新的存储实现使得删除二进制日志文件变得轻松多了，并且提高了 Aurora MySQL 数据库集群中作为二进制日志复制主实例的实例的系统性能。

要启用二进制日志的高效存储，请在主实例或 Aurora 副本的参数组中将 `aurora_lab_mode` 参数设置为 1。`aurora_lab_mode` 参数是 `default.aurora5.6` 参数组中默认的实例级参数。有关修改数据库参数组的信息，请参阅《Amazon Aurora 用户指南》中的[修改数据库参数组中的参数](#)。有关参数组和 Aurora MySQL 的信息，请参阅《Amazon Aurora 用户指南》中的[Aurora MySQL 配置参数](#)。

仅为 Aurora MySQL 数据库集群中作为 MySQL 二进制日志复制主实例的实例启用二进制日志的高效存储。

- AURORA_VERSION 系统变量 – 您现在可以查询 AURORA_VERSION 系统变量以获取 Aurora MySQL 数据库集群的 Aurora 版本。

要获取 Aurora 版本，请使用下列查询之一：

```
select AURORA_VERSION();
select @@aurora_version;
show variables like '%version';
```

您也可以在修改数据库集群时在AWS Management Console中查看 Aurora 版本，或者调用 [describe-db-engine-versions](#) AWS CLI 命令或 [DescribeDBEngineVersions](#) API 操作以查看该版本。

- 锁定管理器内存使用量指标 – 现在以指标形式提供有关锁定管理器内存使用量的信息。

要获取锁定管理器内存使用量指标，请使用下列查询之一：

```
show global status where variable_name in ('aurora_lockmgr_memory_used');
select * from INFORMATION_SCHEMA.GLOBAL_STATUS where variable_name in
('aurora_lockmgr_memory_used');
```

改进

- 提高了二进制日志和 XA 事务恢复期间的稳定性。
- 修复了因大量连接导致的内存问题。
- 提高了以下指标的准确度：Read Throughput、Read IOPS、Read Latency、Write Throughput、Write IOPS、Write Latency 和 Disk Queue Depth。
- 修复了导致大型实例在崩溃后启动较慢的稳定性问题。
- 提高了有关同步机制和缓存移出的数据字典的并发性。
- Aurora 副本的稳定性和性能改进：
 - 修复了主实例的密集型或突发写入工作负载期间的 Aurora 副本的稳定性问题。
 - 改善了 db.r3.4xlarge 和 db.r3.8xlarge 实例的副本滞后。
 - 通过减少日志记录的应用程序和 Aurora 副本上的并发读取之间的争用来提供性能。
 - 修复了有关刷新 Aurora 副本的统计信息以获取新创建的或更新后的统计信息的问题。
 - 提高了在主实例上有多个事务并且跨相同数据对 Aurora 副本进行并发读取时 Aurora 副本的稳定性。
 - 提高了在运行 UPDATE 和 DELETE 语句与 JOIN 语句时 Aurora 副本的稳定性。
 - 提高了运行 INSERT ... SELECT 语句时 Aurora 副本的稳定性。

MySQL 错误修复汇总

- BACKPORT 错误 18694052 FIX FOR ASSERTION '!M_ORDERED_REC_BUFFER' FAILED TO 5.6 (端口错误 18305270)
- SEGV IN MEMCPY(), HA_PARTITION::POSITION (端口错误 18383840)
- WRONG RESULTS WITH PARTITIONING,INDEX_MERGE AND NO PK (端口错误 18167648)
- FLUSH TABLES FOR EXPORT: ASSERTION IN HA_PARTITION::EXTRA (端口错误 16943907)
- SERVER CRASH IN VIRTUAL HA_ROWS HANDLER::MULTI_RANGE_READ_INFO_CONST (端口错误 16164031)
- 范围优化程序在 SEL_ARG::RB_INSERT() 中崩溃 (端口错误 16241773)

Aurora MySQL 数据库引擎更新：2016-01-11 (版本 1.5) (已弃用)

版本：1.5

此更新包括以下改进：

改进

- 修复了 Aurora 存储部署期间空闲实例的写入操作的 10 秒暂停问题。
- 现在，逻辑预读在 innodb_file_per_table 设置为 No 时运行。有关逻辑预读的更多信息，请参阅 [Aurora MySQL 数据库引擎更新：2015-12-03 \(版本 1.4 \) \(已弃用 \)](#)。
- 修复了与主实例重新连接的 Aurora 副本有关的问题。此改进还修复了在使用错误注入查询测试 Aurora 副本故障时为 quantity 参数指定较大值时的问题。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [测试 Aurora 副本故障](#)。
- 改进了对滞后和重新启动的 Aurora 副本的监控。
- 修复了导致 Aurora 副本滞后、取消注册，然后重新启动的问题。
- 修复了在锁死期间运行 show innodb status 命令时的问题。
- 修复了与高写入吞吐量期间大型实例的故障转移相关的问题。

MySQL 错误修复汇总

- 处理了影响以数字开头的数据库名称所在的表的 MySQL 全文搜索中的未完成修复。(端口错误 17607956)

Aurora MySQL 数据库引擎更新：2015-12-03 (版本 1.4) (已弃用)

版本：1.4

此更新包括以下改进：

新功能

- 快速插入 – 加快按主键排序的并行插入。有关更多信息，请参阅《Amazon Aurora 用户指南》中的 [Amazon Aurora MySQL 性能增强](#)。
- 大型数据集读取性能 – Aurora MySQL 自动检测 IO 密集型工作负载并启动更多线程以提高数据库集群的性能。Aurora 计划程序查看 IO 活动并决定动态调整系统中的最佳线程数，从而以较低开销快速地在 IO 密集型工作负载和 CPU 密集型工作负载之间做出调整。
- 并行预读 – 提高了对主实例或 Aurora 副本上的可用内存来说过大的 B 树扫描的性能 (包括范围查询)。并行预读会自动检测页面读取模式，并在需要页面时提前将页面预提取到缓冲区缓存中。并行预读在同一事务内同时处理多个表。

改进：

- 修复了 Aurora 存储部署期间 Aurora 数据库的短暂可用性问题。
- 正确实施 max_connection 限制。
- 改进了 Aurora 作为 Binlog 主实例且数据库在大量数据加载后重新启动的情况下的二进制日志清除。
- 修复了与表缓存相关的内存管理问题。
- 增加了对共享内存缓冲区缓存中的大页的支持以加快恢复。
- 修复了有关未初始化的线程本地存储的问题。
- 默认情况下允许 16K 连接。
- IO 密集型工作负载的动态线程池。

- 修复了有关适当地使涉及查询缓存中的 UNION 的视图无效的问题。
- 修复了有关字典 stats 线程的稳定性问题。
- 修复了与缓存移出相关的字典子系统内存中的内存泄漏问题。
- 修复了当主实例上的写入负载非常低时，Aurora 副本上的高读取延迟问题。
- 修复了在主实例上的 DDL 分区表上执行操作（如 ALTER TABLE ... REORGANIZE PARTITION）时出现的 Aurora 副本稳定性问题。
- 修复了容量增长期间 Aurora 副本的稳定性问题。
- 修复了有关针对 Aurora 副本中的非集群索引的扫描的性能问题。
- 修复了导致 Aurora 副本滞后并最终取消注册并重启的稳定性问题。

MySQL 错误修复汇总

- FTSPARSE() 中的 SEGV。(错误 16446108)
- InnoDB 数据字典在重名列时不会更新。(错误 19465984)
- FTS 在将表重命名为其他数据库后发生崩溃。(错误 16834860)
- 无法为截断的表准备触发器会生成错误 1054。(错误 18596756)
- 元数据更改可能会导致触发器执行出现问题。(错误 18684393)
- 未为较长的 UTF8 VARCHAR 字段选择具体化。(错误 17566396)
- 将 ORDER BY 与限制 X 结合使用时，执行计划性能下降。(错误 #16697792)
- 将错误 11765744 逆向移植到 5.1、5.5 和 5.6。(错误 17083851)
- 生成 SIG6 的 SQL/SQL_SHOW.CC 中的互斥问题。根源可能是 FILL_VARIABLES。(错误 20788853)
- 将错误 18008907 逆向移植到 5.5+ 版本。(错误 18903155)
- 适应 MySQL 5.7 中的堆栈溢出错误的修复。(错误 19678930)

Aurora MySQL 数据库引擎更新：2015-10-16（版本 1.2、1.3） （已弃用）

版本：1.2、1.3

此更新包括以下改进：

修复

- 解决了长时间运行的事务的新锁定管理器中的内存不足问题
- 解决了使用非 RDS for MySQL 数据库进行复制时的安全漏洞
- 进行了更新，以确保在出现存储故障的情况下 quorum 写入正确重试
- 进行了更新，以更准确地报告副本滞后
- 已通过减少很多并发事务尝试修改同一行时的争用情况来提高性能
- 已解决通过联接 2 个表创建的视图的查询缓存失效
- 禁用了具有 UNCOMMITTED_READ 隔离的事务的查询缓存

改进

- 提高了热缓存上慢速目录查询的性能
- 提高了字典统计数据中的并发度
- 提高了新的查询缓存资源管理器、范围管理、存储在 Amazon Aurora 智能存储中的文件以及日志记录的批量写入的稳定性

MySQL 错误修复汇总

- 终止 innodb 中的查询会导致该查询最终崩溃并显示断言。(错误 1608883)
- 如果无法为事件调度程序、事件执行或新连接创建新线程，则不会将任何消息写入错误日志。(错误 16865959)
- 如果一个连接更改了其默认数据库，同时另一个连接执行了 SHOW PROCESSLIST，则第二个连接可在尝试显示第一个连接的默认数据库内存时访问无效内存。(错误 11765252)
- 虽然 PURGE BINARY LOGS 设计为不删除正在使用或活动的二进制日志文件，但发生该操作时，不会提供任何通知。(错误 13727933)
- 对于某些语句，在优化程序删除不需要的子查询子句时，可能会导致内存泄漏。(错误 15875919)
- 关闭期间，服务器可能会尝试锁定未初始化的互斥。(错误 16016493)
- 使用 GROUP_CONCAT() 的准备好的语句和已指定多个列的 ORDER BY 子句可能会导致服务器退出。(错误 16075310)
- 副本工作线程缺少性能架构检测。(错误 16083949)
- STOP SLAVE 与诸如 SHOW STATUS (检索 Slave_retried_transactions、Slave_heartbeat_period、Slave_received_heartbeats、S

- 和 `Slave_running` 中的一个或多个状态变量的值) 之类的语句同时发出时 , 可能会导致死锁。(错误 16088188)
- 在搜索词为用引号引起来的短语的某些情况下 , 使用布尔值模式的全文查询可能会返回零个结果。(错误 16206253)
 - 在尝试在子查询中的联接的 `ON` 子句中使用子查询执行准备好的语句时 , 优化程序尝试删除引发断言的冗余子查询子句。(错误 16318585)
 - `GROUP_CONCAT` 不稳定 , 在 `ITEM_SUM::CLEAN_UP_AFTER_REMOVAL` 中发生崩溃。(错误 16347450)
 - 尝试通过创建具有与 `INFORMATION_SCHEMA.INNOODB_FT_DEFAULT_STOPWORD` 相同的结构的 InnoDB 表来替换默认 InnoDB 全文搜索 (FTS) 非索引列表将生成错误。(错误 16373868)
 - 在工作线程上的客户端线程执行 `FLUSH TABLES WITH READ LOCK` 并接着在主实例上进行一些更新后 , 工作线程在执行 `SHOW SLAVE STATUS` 时挂起。(错误 16387720)
 - 在全文搜索中分析分隔的搜索字符串 (如“abc-def”) 时 , InnoDB 现在会使用与 MyISAM 相同的单词分隔符。(错误 16419661)
 - `FTS_AST_TERM_SET_WILDCARD` 中发生崩溃。(错误 16429306)
 - 针对 FTS RQG 测试的 `FTS_AST_VISIT()` 中的 `SEGFault`。(错误 16435855)
 - 在优化程序删除指向子查询的 `Item_ref` 后 , 调试版本会导致服务器退出。(错误 16509874)
 - InnoDB 表上的全文搜索无法搜索组合了 + 或 - 运算符的文字短语。(错误 16516193)
 - `START SLAVE` 当一起使用选项 `--master-info-repository=TABLE relay-log-info-repository=TABLE` 和启动服务器且 `autocommit` 设置为 0 时 , 失败。`--skip-slave-start`(错误 16533802)
 - 非常大的 InnoDB 全文搜索 (FTS) 结果可能会占用大量内存。(错误 16625973)
 - 在调试版本中 , 由于二进制数据可能包含 `NULL` 字节和其他无意义的字符 , 因此当直接在搜索字符串中使用二进制数据时 , `OPT_CHECK_ORDER_BY` 中可能发生断言。(错误 16766016)
 - 对于某些语句 , 在优化程序删除不需要的子查询子句时 , 可能会导致内存泄漏。(错误 16807641)
 - 通过在工作线程的新连接中发出 `STOP SLAVE` 来发出 `FLUSH TABLES WITH READ LOCK` , 然后使用原始连接发出 `SHOW SLAVE STATUS` 后 , 可能会导致锁死。(错误 16856735)
 - 具有无效分隔符的 `GROUP_CONCAT()` 可能会导致服务器退出。(错误 16870783)
 - 服务器对任何 `SHOW STATUS LIKE`“模式”语句的 `LOCK_active_mi` 和 `active_mi->rli->data_lock` 互斥对象执行了过度锁定 , 甚至在该模式不匹配使用这些互斥对象的状态变量 (`Slave_heartbeat_period`、`Slave_last_heartbeat`、`Slave_received_heartbeats`、`Slave` 的情况下也是如此。(错误 16904035)
 - 使用 `IN BOOLEAN MODE` 修饰符的全文搜索将导致断言失败。(错误 16927092)

- InnoDB 表上的全文搜索无法执行使用 + 布尔值运算符的搜索。(错误 17280122)
- 4 向锁死：zombie、清除二进制日志、显示 processlist、显示二进制日志。(错误 17283409)
- 当等待提交锁定的 SQL 线程终止并重新启动时，导致在工作线程上跳过事务。(错误 17450876)
- InnoDB 全文搜索失败可能是由于“unended”令牌导致的。应传递字符串和字符串长度以进行字符串比较。(错误 17659310)
- 在 MySQL 5.6 或 5.7 中使用大量已分区的 InnoDB 表可能比在 MySQL Server 的先前版本中使用相同的表所占用的内存更多。(错误 17780517)
- 对于全文查询，未能检查出 num_token 少于 max_proximity_item 可能会导致断言。(错误 18233051)
- 当存在大量空 InnoDB 表时，INFORMATION_SCHEMA TABLES 和 COLUMNS 表的特定查询可能会导致内存使用过度。(错误 18592390)
- 当提交事务时，标记现在用于检查是否创建了线程，而不是使用更多资源来检查线程本身 (尤其是在 master_info_repository=TABLE 的情况下运行服务器时)。(错误 18684222)
- 如果工作线程上的客户端线程在主实例执行 DML 时执行了 FLUSH TABLES WITH READ LOCK，则在同一客户端上执行 SHOW SLAVE STATUS 会受阻，从而导致死锁。(错误 19843808)
- 按 GROUP_CONCAT() 结果排序可能导致服务器退出。(错误 19880368)

Aurora MySQL 数据库引擎更新：2015-08-24 (版本 1.1) (已弃用)

版本：1.1

此更新包括以下改进：

- 提高了在使用 MySQL 数据库进行复制 (二进制日志复制) 时的复制稳定性。有关使用 MySQL 进行 Aurora MySQL 复制的信息，请参阅《Amazon Aurora 用户指南》中的[使用 Amazon Aurora 进行复制](#)。
- 为 Aurora MySQL 数据库集群 (作为复制工作线程) 累积的中继日志的大小限制为 1 GB。这改进了 Aurora 数据库集群的文件管理。
- 提高了预读、递归的外键关系和 Aurora 复制方面的稳定性。
- 集成 MySQL 错误修复。
 - 名称以数字开头的 InnoDB 数据库导致的全文搜索 (FTS) 分析器错误。(错误 17607956)
 - 名称以数字开头的数据库中的 InnoDB 全文搜索失败。(错误 17161372)

- 对于 Windows 上的 InnoDB 数据库，全文搜索 (FTS) 对象 ID 未使用预期的十六进制格式。(错误 16559254)
- MySQL 5.6 中引入的代码回归对 DROP TABLE 和 ALTER TABLE 性能有负面影响。这可能会导致 MySQL Server 5.5.x 和 5.6.x 之间的性能下降。(错误 16864741)
- 简化了日志记录，以减少日志文件的大小和其所需的存储量。

由 Aurora MySQL 数据库引擎更新修复的 MySQL 错误

以下部分标识了已由 Aurora MySQL 数据库引擎更新修复的 MySQL 错误。

主题

- [由 Aurora MySQL 3.x 数据库引擎更新修复的 MySQL 错误](#)
- [由 Aurora MySQL 2.x 数据库引擎更新修复的 MySQL 错误](#)
- [由 Aurora MySQL 1.x 数据库引擎更新修复的 MySQL 错误](#)

由 Aurora MySQL 3.x 数据库引擎更新修复的 MySQL 错误

MySQL 8.0 兼容版本 Aurora 包含其相应 MySQL 兼容版本和之前版本的所有 MySQL 错误修复。下表列出了 Aurora MySQL 数据库引擎更新修复的其他 MySQL 错误以及修复这些错误的更新。

数据库引擎更新	MySQL 兼容版本	版本	已修复的 MySQL 错误
Aurora MySQL 数据库引擎更新 2024-03-07 (版本 3.06.0, 兼容 MySQL 8.0.34)	8.0.34	3.06.0	<ul style="list-style-type: none"> • 修复了缓存行值计算不正确的问 题，该问题会导致在 Graviton 实例上重启数据库时失败。（社区错误修复 #35479763） • 修复了存储例程中的某些子查询实例并非总是能正确处理的问题。（社区错误修复 #35377192） • 修复了后台 TLS 证书轮换可能导致更高 CPU 使用率的问题。（社区错误修复 #34284186） • 修复了在 Aurora MySQL 版本低于 3.05 的版本中，InnoDB 允许向 MySQL 系统架构中的表添加 INSTANT 列的问题，这可能会导致服务器在升级到 Aurora MySQL 版本 3.05.0 后意外关闭

数据库引擎更新	MySQL 兼容版本	版本	已修复的 MySQL 错误
			(数据库实例重新启动)。(社区错误修复 35625510)
Aurora MySQL 数据库引擎更新 2024-01-31 (版本 3.05.2 , 兼容 MySQL 8.0.32)	8.0.32	3.05.2	<ul style="list-style-type: none"> 重复运行存储的例程，将包含多个、或XOR条件的 SELECT 语句作为子查询 ANDOR，会导致过度消耗虚拟内存，并可能最终耗尽虚拟内存。(社区错误修复 #33852530)
Aurora MySQL 数据库引擎更新 2023-11-21 (版本 3.05.1 , 兼容 MySQL 8.0.32)	8.0.32	3.05.1	<ul style="list-style-type: none"> 修复了 InnoDB 中的一个问题，即在 Aurora MySQL 版本 3.01 至 Aurora MySQL 版本 3.04 之间，如果系统架构中的 MySQL 表添加了 INSTANT ADD 列，在 Aurora MySQL 升级到 3.05.0 版本之后，这些表上的 DML 将导致服务器意外关闭。(社区错误修复 35625510)

数据库引擎更新	MySQL 兼容版本	版本	已修复的 MySQL 错误
Aurora MySQL 数据库引擎更新 2023-10-25 (版本 3.05.0, 与 MySQL 8.0.32 兼容)	8.0.32	3.05.0	<ul style="list-style-type: none"> 修复了因后台 TLS 证书轮换而可能导致更高 CPU 使用率的问题 (社区错误修复 #34284186)
Aurora MySQL 数据库引擎更新 2024-03-15 (版本 3.04.2, 兼容 MySQL 8.0.28)	8.0.28	3.04.2	<ul style="list-style-type: none"> 修复了在基于 Graviton 的实例上可能无法正确计算缓存行值, 从而导致数据库重启失败的问题。(社区错误修复 #35479763) 重复运行存储的例程, 将包含多个、或XOR条件的 SELECT 语句作为子查询 ANDOR, 会导致过度消耗虚拟内存, 并可能最终耗尽虚拟内存。(社区错误修复 #33852530)

数据库引擎更新	MySQL 兼容版本	版本	已修复的 MySQL 错误
Aurora MySQL 数据库引擎更新 2023-11-13 (版本 3.04.1, 兼容 MySQL 8.0.28) 默认	8.0.28	3.04.1	<ul style="list-style-type: none"> 修复了因后台 TLS 证书轮换而可能导致更高 CPU 使用率的问题 (社区错误修复 #34284186)
Aurora MySQL 数据库引擎更新 2023-07-31 (版本 3.04.0, 兼容 MySQL 8.0.28)	8.0.28	3.04.0	<ul style="list-style-type: none"> 修复了在页面遍历期间重新定位包含内部临时表页面的缓冲块, 从而导致断言失败的问题 (错误 #33715694) InnoDB: 阻止在线 DDL 操作访问 out-of-bounds 内存 (错误# 34750489, 错误# 108925) 修复了在处理由多个嵌套的公用表表达式 (CTE) 组成的复杂 SQL 语句时, 有时会生成错误查询结果的问题 (错误 #34572040、错误 #34634469、错误 #33856374)

数据库引擎更新	MySQL 兼容版本	版本	已修复的 MySQL 错误
Aurora MySQL 数据库引擎更新 2023-12-08 (版本 3.03.3, 与 MySQL 8.0.26 兼容)	8.0.26	3.03.3	<ul style="list-style-type: none"> 修复了因后台 TLS 证书轮换而可能导致更高 CPU 使用率的问题 (社区错误修复 #34284186)
Aurora MySQL 数据库引擎更新 2023-08-29 (版本 3.03.2, 与 MySQL 8.0.26 兼容)	8.0.26	3.03.2	<ul style="list-style-type: none"> 修复了在处理由多个嵌套的公用表表达式 (CTE, Common Table Expressions) 组成的复杂 SQL 语句时, 有时会生成错误查询结果的问题 (错误 34572040、错误 34634469、错误 33856374) InnoDB: 尝试取消初始化和初始化同一个表的统计信息的线程之间的争用条件会导致断言失败 (错误 33135425) InnoDB: 阻止在线 DDL 操作访问 out-of-bounds 内存 (错误 #34750489, 错误 #108925)

数据库引擎更新	MySQL 兼容版本	版本	已修复的 MySQL 错误
Aurora MySQL 数据库引擎更新 2023-05-11 (版本 3.03.1, 与 MySQL 8.0.26 兼容)	8.0.26	3.03.1	<ul style="list-style-type: none">修复了在页面遍历期间重新定位包含内部临时表页面的缓冲块，从而导致断言失败的问题 (错误 33715694)

数据库引擎更新	MySQL 兼容版本	版本	已修复的 MySQL 错误
Aurora MySQL 数据库引擎更新 2023-03-01 (版本 3.03.0, 与 MySQL 8.0.26 兼容) 不支持升级到此版本。	8.0.26	3.03.0	<ul style="list-style-type: none"> • 修复了以下问题：排序缓冲区的大小必须最少是排序中最大行的 15 倍，否则某些列类型（包括 JSON 和 TEXT）的排序有时会耗尽排序缓冲区。现在，排序缓冲区的大小只需是最大排序键的 15 倍。（错误 #103325、错误 #105532、错误 #32738705、错误 #33501541） • 修复了 InnoDB 并不总能正确处理表分区的某些合法名称的问题。（错误 #32208630） • 修复了在执行具有 OR 条件的查询时，由于空性属性的计算不准确，在某些条件下可能会返回错误结果的问题。（错误 #34060289） • 修复了在某些情况下，满足以下两个条件时可能返回错误结果的问题： <ul style="list-style-type: none"> • 将一个派生表合并到了外部查询块中。 • 查询包括一个左连接和一个 IN 子查询。 （错误 #34060289） • 修复了在超过最大整数列值时生成不正确的 AUTO_INCREMENT 值的问题。该错误由于未考虑最大列值所致。在这种情况下，应返回上一个有效的 AUTO_INCREMENT 值，而没有这样做就

数据库引擎更新	MySQL 兼容版本	版本	已修复的 MySQL 错误
			<p>导致了键重复错误。(错误 #87926、错误 #26906787)</p> <ul style="list-style-type: none">• 修复了无法撤销性能架构上的 DROP 权限的问题。(错误 #33578113)• 修复了包含使用 EXISTS 的 IF 语句的存储过程无法正确执行的问题，该语句对在两次执行之间删除并重新创建的一个或多个表执行操作，但在第一次执行之后的后续调用中无法正确执行。(错误 #32855634)。• 修复了引用子查询中的视图和外部查询块的查询可能导致意外重启的问题 (错误 #32324234)

数据库引擎更新	MySQL 兼容版本	版本	已修复的 MySQL 错误
Aurora MySQL 数据库引擎更新 2022-11-18 (版本 3.02.2, 与 MySQL 8.0.23 兼容) 标准支持将于 2024 年 1 月 15 日结束。	8.0.23	3.02.2	<ul style="list-style-type: none"> • 修复了在执行具有 OR 条件的查询时，由于空性属性的计算不准确，在某些条件下可能会返回错误结果的问题。(错误 #34060289) • 修复了在某些情况下，满足以下两个条件时可能返回错误结果的问题： <ul style="list-style-type: none"> • 将一个派生表合并到了外部查询块中。 • 查询包括一个左联接和一个 IN 子查询。(错误 #34060289) • 修复了无法撤销性能架构上的 DROP 权限的问题。(错误 #33578113) • 修复了包含使用 EXISTS 的 IF 语句的存储过程无法正确执行的问题，该语句对在两次执行之间删除并重新创建的一个或多个表执行操作，但在第一次执行之后的后续调用中无法正确执行。(MySQL 错误 #32855634)。 • 当超过最大整数列值时，生成的 AUTO_INCREMENT 值不正确。该错误由于未考虑最大列值所致。在这种情况下，应返回上一个有效的 AUTO_INCREMENT 值，而没有这样做就导致了键重复错误。(错误 #87926、错误 #26906787) • 修复了在升级 Aurora MySQL 版本 1 (与 MySQL 5.6 兼容)

数据库引擎更新	MySQL 兼容版本	版本	已修复的 MySQL 错误
			<p>数据库集群时，如果该集群包含用户创建的带有特定表 ID 的表，则可能导致失败的问题。从 Aurora MySQL 版本 2 (与 MySQL 5.7 兼容) 升级到 Aurora MySQL 版本 3 (与 MySQL 8.0 兼容) 时，分配这些表 ID 可能会导致数据字典表 ID 冲突。(错误 33919635)</p>
<p>Aurora MySQL 数据库引擎更新 2022-04-20 (版本 3.02.0, 与 MySQL 8.0.23 兼容) 标准支持将于 2024 年 1 月 15 日结束。不支持升级到此版本。</p>	<p>8.0.23</p>	<p>3.02.0</p>	<p>修复了存储过程中用于游标的临时表处理不当的问题，这可能会导致服务器出现意外行为。(错误 #32416811)</p>

数据库引擎更新	MySQL 兼容版本	版本	已修复的 MySQL 错误
Aurora MySQL 数据库引擎更新 2022-04-15 (版本 3.01.1, 与 MySQL 8.0.23 兼容) 标准支持将于 2024 年 1 月 15 日结束。不支持升级到此版本。	8.0.23	3.01.1	修复了存储过程中用于游标的临时表处理不当的问题，这可能会导致服务器出现意外行为。 (错误 #32416811)

由 Aurora MySQL 2.x 数据库引擎更新修复的 MySQL 错误

MySQL 5.7 兼容版本 Aurora 包含 MySQL 5.7.40 和之前版本的所有 MySQL 错误修复。下表列出了 Aurora MySQL 数据库引擎更新修复的其他 MySQL 错误以及修复这些错误的更新。

数据库引擎更新	Version	已修复的 MySQL 错误
Aurora MySQL 数据库引擎更新 2023-12-28 (版本 2.12.1, 与 MySQL 5.7.40 兼容)	2.12.1	<ul style="list-style-type: none"> 修复了在有 SHOW PROCESSLIST 语句同时运行时，可能导致现有和新的远程连接停顿的问题 (社区错误 34857411) 复制：某些二进制日志事件的处理并不总是正确的 (错误 34617506) 修复了全文搜索 (FTS, Full-Text Search) 解析器插件对单字符令牌的处理 (错误 35432973)

数据库引擎更新	Version	已修复的 MySQL 错误
Aurora MySQL 数据库引擎更新 2023-07-25 (版本 2.12.0, 与 MySQL 5.7.40 兼容)	2.12.0	<ul style="list-style-type: none">修复了由于后台 TLS 证书轮换可能导致更高 CPU 使用率的问题。(社区错误修复 #34284186)
Aurora MySQL 数据库引擎更新 2023-10-17 (版本 2.11.4, 与 MySQL 5.7.12 兼容)	2.11.4	<ul style="list-style-type: none">复制: 某些二进制日志事件的处理并不总是正确的。(错误 #34617506)修复了由于后台 TLS 证书轮换可能导致更高 CPU 使用率的问题。(社区错误修复 #34284186)在预编译语句中, 某些类型的子查询可能会导致服务器退出。(错误 #33100586)

数据库引擎更新	Version	已修复的 MySQL 错误
Aurora MySQL 数据库引擎更新 2022-10-25 (版本 2.11.0, 与 MySQL 5.7.12 兼容) 此版本不适用于新创建。	2.11.0	<ul style="list-style-type: none"> • 修复了从性能架构语句事件表 (例如 , events_statements_current) 中读取字符集信息的代码无法阻止同时写入该字符集信息的问题。因此 , SQL 查询文本字符集可能无效 , 从而可能导致服务器退出。通过此修复 , 无效的字符集会 导致 SQL_TEXT 列被截断并阻止服务器退出。 (错误 #23540008) • InnoDB : 社区错误 #25189192、错误 #84038 修复的向后移植。修复了在将表移至其他架构的 RENAME TABLE 操作后 , InnoDB 无法更新 INNODB_SYS_DATAFILES 数据字典表的问题。这会导致重启时出错 , 表明它找不到表空间数据文件。 • InnoDB : 修复了以下问题 : 在添加新索引时 , 服务器会丢弃内部定义的外键索引 , 并试图使用在虚拟生成的列上定义的二级索引作为外键索引 , 从而导致服务器退出。InnoDB 现在允许外键约束引用在虚拟生成的列上定义的二级索引。 (错误 23533396) • 修复了两个会话同时执行 INSERT ... ON DUPLICATE KEY UPDATE 操作时 , 会生成死锁的问题。在元组的部分回滚期间 , 另一个会话可能会对其进行更新。此错误的修复撤销了对错误 #11758237、错误 #17604730 和错误 #20040791 的修复。 (错误 #25966845) • 修复了即使启用了 automatic_sp_privileges , 也无法正确向例程创建者授予 EXECUTE 和 ALTER ROUTINE 权限的问题。 (错误 #27407480) • 社区错误 #24671968 修复的向后移植 : 修复了以下问题 : 如果 WHERE 子句包含依赖子查询 , 表在选择列表中的列上有二级索引 (后跟子查询中的列) , 且 GROUP BY 或 DISTINCT 允许查询使用松散索引扫描 , 则查询可能会产生错误的结果。 • 修复了如果对多个带外键的表发出多表删除语句 , 则复制会中断的问题。 (错误 #80821) • 修复了在特殊情况下 , 即使启用了 slave_skip_errors , 某些从属服务器错误也不会被忽略的问题。如果打开和锁定表失败 , 或者在运行基于行的复制的服务器上进行字段转换失败 , 则该错误被视为严重错误 , slave_skip_errors 状态将被忽略。

数据库引擎更新	Version	已修复的 MySQL 错误
		<p>该修复可确保在启用 slave_skip_errors 的情况下，应用事务期间报告的所有错误都得到正确处理。（错误 #70640、错误 #17653275）</p> <ul style="list-style-type: none"> 修复了在 SET PASSWORD 语句从 MySQL 5.6 主服务器复制到 MySQL 5.7 从属服务器，或者从 log_builtin_as_identified_by_password 系统变量设置为 ON 的 MySQL 5.7 主服务器复制到 MySQL 5.7 从属服务器时，密码哈希本身在存储在从属服务器上之前也经过哈希处理的问题。该问题现已得到修复，复制的密码哈希按最初传递给从属服务器的原样存储。（错误 #24687073） 修复了由包裹在多级 JSON 数组和/或对象中的大型子文档组成的 JSON 值的序列化有时需要过长时间才能完成的问题。（错误 #23031146） 无法解析的语句（例如，由于语法错误）将不再写入慢速查询日志。（错误 #33732907）
Aurora MySQL 数据库引擎更新 2022-11-01（版本 2.10.3）（已弃用）	2.10.3	<ul style="list-style-type: none"> 修复了从性能架构语句事件表（例如，events_statements_current）中读取字符集信息的代码无法阻止同时写入该字符集信息的问题。因此，SQL 查询文本字符集可能无效，从而可能导致服务器退出。通过此修复，无效的字符集会导致 SQL_TEXT 列被截断并阻止服务器退出。（错误 #23540008） 修复了当 UPDATE 需要一个主键大于 1024 个字节的临时表，而该表是使用 InnoDB 创建的时，服务器可能会退出的问题。（错误 #25153670）

数据库引擎更新	Version	已修复的 MySQL 错误
Aurora MySQL 数据库引擎更新 2022-01-26 (版本 2.10.2) (已弃用)	2.10.2	<ul style="list-style-type: none"> • 修复了 InnoDB 中一个的问题：与表统计数据相关的代码错误导致 dict0stats.cc (http://dict0stats.cc/) 源文件出现断言。(错误 #24585978) • 在线构建索引时，虚拟列上的二级索引已损坏。对于 UPDATE (https://dev.mysql.com/doc/refman/5.7/en/update.html) 语句，我们按以下方式修复此问题：如果索引记录的虚拟列值设置为 NULL，就从集群索引记录中生成此值。(错误 #30556595) • ASSERTION "!OTHER_LOCK" IN LOCK_REC_ADD_TO_QUEUE (错误 #29195848) • HANDLE_FATAL_SIGNAL (SIG=11) IN __STRCHR_SSE2 (错误 #28653104) • 修复了锁定等待期间查询中断可能导致 InnoDB 错误的问题。(错误 #28068293) • 当事务隔离级别设置为 REPEATABLE READ 时，交错事务有时可能会使副本应用程序死锁。(错误 #25040331) • 修复了锁定等待超时导致二进制日志副本停顿的问题。(错误 #27189701)
Aurora MySQL 数据库引擎更新 2021-10-21 (版本 2.10.1) (已弃用)	2.10.1	CURRENT_TIMESTAMP PRODUCES ZEROS IN TRIGGER。(错误 25209512)

数据库引擎更新	Version	已修复的 MySQL 错误
Aurora MySQL 数据库引擎更新 2021-05-25 (版本 2.10.0) (已弃用)	2.10.0	<ul style="list-style-type: none"> • 当事务隔离级别设置为 REPEATABLE READ 时，交错事务有时可能会使副本应用程序死锁。(错误 #25040331) • 当存储过程包含引用一个视图的语句，而该语句又引用了另一个视图时，该过程不能多次成功调用。(错误 #87858、错误 #26864199) • 对于具有多个 OR 条件的查询，优化程序现在的内存效率更高，并且超过 range_optimizer_max_mem_size 系统变量规定内存限制的可能性更小。此外，该变量的默认值已从 1,536,000 提高到 8,388,608。(错误 #79450、错误 #22283790) • 复制：在由副本的 SQL 线程调用以从中继日志中读取下一个事件的 <code>next_event()</code> 函数中，SQL 线程在遇到错误(例如，由于中继日志关闭)时没有释放获取的 <code>relaylog_log_lock</code>，导致等待从中继日志中获取锁定的所有其他线程挂起。采用此修复方式后，锁定在这种情况下将于 SQL 线程离开函数之前释放。(错误 #21697821) • 使用虚拟列修复 ALTER TABLE 内存损坏。(错误 #24961167、错误 #24960450) • 复制：如果多线程副本需要处理大于该大小的事务，则无法使用 slave_pending_jobs_size_max 来配置较小的队列大小。任何大于 slave_pending_jobs_size_max 的数据包都会被拒绝并显示错误 ER_MTS_EVENT_BIGGER_PENDING_JOBS_SIZE_MAX，即使该数据包小于 slave_max_allowed_packet 设置的限制。采用此修复方式后，slave_pending_jobs_size_max 将成为软限制而不是硬限制。如果数据包的大小超过了 slave_pending_jobs_size_max 但小于 slave_max_allowed_packet，则该事务将保留直到所有副本工作线程都有空队列为止，然后进行处理。所有后续事务将一直保留到大型事务完成。因此，副本工作线程的队列大小可以受到限制，同时仍允许偶尔执行较大事务。(错误 #21280753、错误 #77406) • 复制：使用多线程副本时，应用程序错误显示的工作线程 ID 数据与性能架构副本表中外部化的数据不一致。(错误 #25231367)

数据库引擎更新	Version	已修复的 MySQL 错误
		<ul style="list-style-type: none"> • 复制：在基于 GTID-mode=on、-log-bin=off 和使用 - 的复制副本上，当遇到应忽略的错误时 <code>slave-skip-errors</code>，未正确更新，导致与的同步失败。 <code>Exec_Master_Log_Pos</code> <code>Exec_Master_Log_Pos</code> <code>Read_master_log_pos</code> 如果未指定 <code>GTID_NEXT</code>，则从单语句事务回滚时，副本永远不会更新其 GTID 状态。 <code>Exec_Master_Log_Pos</code> 不会更新，因为即使事务已完成，其 GTID 状态也会显示为其他状态。此修复消除了只在指定 <code>GTID_NEXT</code> 的情况下回滚事务时才更新 GTID 状态的限制。（错误 #22268777） • 复制：禁用二进制日志记录时，部分失败的语句未正确使用自动生成或指定的 GTID。此修复可确保在禁用二进制日志记录时，部分失败的 DROP TABLE、部分失败的 DROP USER 或部分失败的 DROP VIEW 会分别使用相关的 GTID 并将其保存到 <code>@@GLOBAL.GTID_EXECUTED</code> 和 <code>mysql.gtid_executed</code> 表中。（错误 #21686749） • 复制：由于检索不属于 MySQL 5.5 的 server_uuid 时出错，运行 MySQL 5.7 的副本无法连接到 MySQL 5.5 源。这是由于检索 <code>server_uuid</code> 的方法更改造成的。（错误 #22748612） • 二进制日志复制：在此修复之前，GTID 事务跳过机制无法正常作用于 XA 事务。服务器有一个机制，如果它在过去已经执行了一个 GTID 事务，就会（静默地）跳过此类事务。（错误 #25041920） • 因给出了错误的事务 ID 而失败的 XA ROLLBACK 语句可以用正确的事务 ID 记录在二进制日志中，因此可以由复制副本执行操作。现在，在进行二进制日志记录之前会检查错误情况，不会记录失败的 XA ROLLBACK 语句。（错误 #26618925） • 复制：如果使用未指定源日志文件名和源日志位置的 CHANGE MASTER TO 语句设置副本，则在发出 START SLAVE 之前将其关闭，然后使用选项 <code>-set</code> 重新启动，则复制不会开始。发生这种情况的原因是，在尝试恢复中继日志之前没有启动接收方线程，因此中继日志中没有可用的日志轮换事件来提供源日志文件名和源日志位置。在这种情况下，现在副本跳过中继日志恢复并记录警告，然后继续开始复制。（错误 #28996606、错误 #93397）

数据库引擎更新	Version	已修复的 MySQL 错误
		<ul style="list-style-type: none"> • 复制：在基于行的复制中，从含有 utf8mb3 列的表复制到使用 utf8mb4 字符集定义该列的具有相同定义的表时，会返回错误显示字段长度的消息。（错误 #25135304、错误 #83918） • 复制：当在使用 GTID 的复制副本上发出 RESET SLAVE 语句时，现有的中继日志文件将被清除，但是在清除该通道收到的 GTID 集之前，生成了替换的新中继日志文件。因此，以前的 GTID 集作为 PREVIOUS_GTIDS 事件写入新的中继日志文件，导致复制过程中出现致命错误，指出副本的 GTID 比源多，即使两个服务器的 gtid_executed 集都是空的。现在，当 RESET SLAVE 发出时，收到的 GTID 集将在生成新的中继日志文件之前清除，就不会发生这种情况了。（错误 #27411175） • 复制：当 GTID 用于复制时，那些包括导致解析错误 (ER_PARSE_ERROR) 语句的事务，无法通过注入具有相同 GTID 的空事务或替换事务的推荐方法手动跳过。此操作应导致副本将 GTID 识别为已使用，从而跳过共享其 GTID 的不需要的事务。但是，尽管无论如何都要跳过事务，在解析错误的情况下，由于语句在检查 GTID 以查看是否需要跳过之前解析了语句，因此复制应用程序线程由于解析错误而停止。采用此修复方式后，如果因为已经使用了 GTID 而需要跳过相关事务，则复制应用程序线程现在会忽略解析错误。请注意，此行为更改不适用于由 mysqlbinlog 生成的二进制日志输出组成的工作负载。在这种情况下，紧跟着跳过的事务之后出现解析错误的事务也会以无提示的方式跳过，因为该事务应引发错误。（错误 #27638268） • 复制：启用 SQL 线程以通过 GTID 跳过部分事务。（错误 #25800025） • 复制：向 <code>WAIT_UNTIL_SQL_THREAD_AFTER_GTIDS()</code> 提供负数或小数超时参数时，服务器出现意外行为。采用此修复方式： <ul style="list-style-type: none"> • 小数超时值按原样读取，没有舍入。 • 如果服务器处于严格 SQL 模式，则拒绝负超时值并显示错误；如果服务器不处于严格 SQL 模式，则该值会使函数立即

数据库引擎更新	Version	已修复的 MySQL 错误
		<p>返回 NULL 而无需等待，然后发出警告。(错误 #24976304、错误 #83537)</p> <ul style="list-style-type: none"> 复制：如果 <code>WAIT_FOR_EXECUTED_GTID_SET()</code> 函数与包括小数部分 (例如 1.5) 的超时值一起使用，则转换逻辑中的错误意味着超时向下舍入到最接近的整秒，对于小于 1 秒的值 (例如，0.1)，则为零。转换逻辑现已被纠正，以便按照最初指定的方式应用超时值，而不舍入。感谢 Dirkjan Bussink 的贡献。(错误 #29324564、错误 #94247) 启用 GTID 后，多语句事务中断开的 XA 事务上的 XA COMMIT 会引发断言。(错误 #22173903) 复制：如果手动设置 <code>gtid_next</code> 值时为未知的事务标识符发出 XA ROLLBACK 语句，则在调试版本中引发了断言。如果 XA ROLLBACK 语句失败并出现错误，现在服务器不会尝试更新 GTID 状态。(错误 #27928837、错误 #90640) 修复了 ORDER BY 子句中使用多个 CASE 函数时排序顺序错误的问题。(错误 #22810883) 某些使用排序的查询可能会在优化期间访问未初始化的列并导致服务器退出。(错误 #27389294)
Aurora MySQL 数据库引擎更新 2021-11-12 (版本 2.09.3) (已弃用)	2.09.3	<ul style="list-style-type: none"> ASSERTION !M_PREBUILT->TRX->CHECK_FOREIGNS。(错误 23533396) 复制：* <code>WAIT_FOR_EXECUTED_GTID_SET()</code> 函数中的锁定问题可能会导致服务器在某些情况下挂起。这个问题现在已得到纠正。(错误 29550513)

数据库引擎更新	Version	已修复的 MySQL 错误
Aurora MySQL 数据库引擎更新 2020-12-11 (版本 2.09.1) (已弃用)	2.09.1	<ul style="list-style-type: none"> 复制：当事务隔离级别设置为 REPEATABLE READ 时，交错事务有时可能会使从属应用程序死锁。（错误 #25040331） 对于 TIMESTAMP 或 DATETIME 列默认值为 CURRENT_TIMESTAMP 的表，列可以在表具有 BEFORE INSERT 触发条件时初始化为 0000-00-00 00:00:00。（错误 #25209512、错误 #84077） 对于 VALUES 列表使用包含联接的子查询为第二行或后面的行生成值的 INSERT 语句，服务器可能会在解析所需的特权失败后退出。（错误 #23762382）
Aurora MySQL 数据库引擎更新 2020-11-12 (版本 2.08.3) (已弃用)	2.08.3	<ul style="list-style-type: none"> 错误 #23762382 - INSERT VALUES QUERY WITH JOIN IN A SELECT CAUSES INCORRECT BEHAVIOR。 错误 #25209512 - CURRENT_TIMESTAMP PRODUCES ZEROS IN TRIGGER。
Aurora MySQL 数据库引擎更新 2020-06-02 (版本 2.08.0) (已弃用)	2.08.0	<ul style="list-style-type: none"> 错误 #25289359：如果全文缓存大小超过全文缓存大小限制，则未释放数据同步时执行的全文缓存锁定。 错误 #29138644：在 MySQL 服务器运行时手动更改系统时间引发了页面清理器线程延迟。 错误 #25222337：在填充受外键约束影响的虚拟列时进行字段名称比较期间，虚拟索引中的 NULL 虚拟列字段名称引发了服务器退出。 错误 #25053286：执行包含访问视图的查询的存储过程可能会分配直到会话结束才释放的内存。 错误 #25586773：执行包含从某些 SELECT 语句的内容创建表的语句的存储过程可能会导致内存泄漏。 错误 #28834208：在日志应用程序期间，在 OPTIMIZE TABLE 操作之后，InnoDB 在检查虚拟列索引更新之前没有填充虚拟列。 错误 #26666274：由于 32 位无符号整数溢出，性能架构缓冲区容器中出現无限循环。

数据库引擎更新	Version	已修复的 MySQL 错误
Aurora MySQL 数据库引擎更新 2022-06-16 (版本 2.07.8) (已弃用)	2.07.8	当 UPDATE 需要一个主键大于 1024 个字节的临时表，而该表是使用 InnoDB 创建的时，服务器可能会退出。(错误 #25153670)
Aurora MySQL 数据库引擎更新 2021-09-02 (版本 2.07.6) (已弃用)	2.07.6	<ul style="list-style-type: none">• 插入 64K 大小的记录需要太多的时间。(错误 23031146)
Aurora MySQL 数据库引擎更新 2021-03-04 (版本 2.07.4) (已弃用)	2.07.4	<ul style="list-style-type: none">• 修复了在处理包含“ ”(空格)、“%”或“,”的令牌时，全文 ngram 解析器中的问题。如果使用 ngram 解析器，客户应重建其 FTS 索引。(错误 #25873310)• 修复了在使用嵌套 SQL 视图执行查询期间可能导致引擎重启的问题。(错误 #27214153、错误 #26864199)

数据库引擎更新	Version	已修复的 MySQL 错误
Aurora MySQL 数据库引擎更新 2020-11-10 (版本 2.07.3) (已弃用)	2.07.3	<ul style="list-style-type: none"> • InnoDB：在主节点上成功运行到 XA 准备阶段的并发 XA 事务在从属节点上重播时会发生冲突，导致应用程序线程中的锁定等待超时。冲突是由于在从属节点上连续重播事务时，GAP 锁定范围会有所不同。为了防止这种类型的冲突，当 XA 事务进入准备阶段时，XA 事务在 READ COMMITTED 隔离级别中采取的 GAP 锁定将被释放 (而且不再继承)。(错误 #27189701、错误 #25866046) • InnoDB：使用 READ COMMITTED 隔离级别时，在验证外键期间多余地使用了 gap 锁定。(错误 #25082593) • 复制：使用 XA 事务时，如果复制从属节点上的应用程序 (SQL) 线程出现锁定等待超时或死锁，则自动重试不起作用。原因是，尽管 SQL 线程会进行回滚，但它不会回滚 XA 事务。这意味着，当重试交易时，第一个事件是 XA START，由于 XA 事务已在进行中，导致 XAER_RMFAIL 错误，该事件无效。(错误 #24764800) • 复制：当事务隔离级别设置为 REPEATABLE READ 时，交错事务有时可能会使从属应用程序死锁。(错误 #25040331) • 复制：所有现有中继日志文件 (Relay_Log_Space) 的总组合大小的 SHOW SLAVE STATUS 语句返回的值可能会比中继日志文件实际使用的磁盘空间大得多。I/O 线程在更新值时没有锁定变量，因此 SQL 线程可以在 I/O 线程完成值的更新之前自动删除中继日志文件并写入减少的值。然后，输入/输出线程写入其原始大小计算，从而忽略了 SQL 线程的更新，将删除文件的的空间添加回来。Relay_Log_Space 值现已在更新期间被锁定，以防止并发更新，确保计算准确。(错误 #26997096，错误 #87832) • 对于 VALUES 列表使用包含联接的子查询为第二行或后面的行生成值的 INSERT 语句，服务器可能会在解析所需的特权失败后退出。(错误 #23762382) • 对于 TIMESTAMP 或 DATETIME 列默认值为 CURRENT_TIMESTAMP 的表，列可以在表具有 BEFORE INSERT 触发条件时初始化为 0000-00-00 00:00:00。(错误 #25209512、错误 #84077)

数据库引擎更新	Version	已修复的 MySQL 错误
Aurora MySQL 数据库引擎更新 2020-04-17 (版本 2.07.2) (已弃用)	2.07.2	<ul style="list-style-type: none"> • 服务器退出可能是由于多个线程同时尝试注册和取消注册元数据性能架构对象造成的。(错误 #26502135) • 执行包含从某些 SELECT 语句的内容创建表的语句的存储过程可能会导致内存泄漏。(错误 #25586773) • 执行包含访问视图的查询的存储过程可能会分配直到会话结束才释放的内存。(错误 #25053286) • 某些情况下的子查询具体化可能会导致服务器退出。这些查询现在会产生建议禁止具体化的错误。(错误 #26402045) • 在使用联接缓冲 (例如, 使用块嵌套循环算法) 的情况下, 带有许多左联接的查询会很慢。(错误 #18898433、错误 #72854) • 在对第二列执行带有 LIKE 子句的内部联接时, 优化程序跳过了复合索引中的第二列。(错误 #28086754) <ul style="list-style-type: none"> • 错误 #23104498 : 修复了性能架构中报告使用的总内存的问题。(https://github.com/mysql/mysql-server/commit/20b6840df5452f47313c6f9a6ca075bfbc00a96b) • 错误 #22551677 : 修复了性能架构中可能导致数据库引擎在尝试脱机时崩溃的问题。(https://github.com/mysql/mysql-server/commit/05e2386eccd32b6b444b900c9f8a87a1d8d531e9) • 错误 #23550835、错误 #23298025、错误 #81464 : 修复了性能架构中导致数据库引擎崩溃的问题 (因为超出内部缓冲区的容量)。(https://github.com/mysql/mysql-server/commit/b4287f93857bf2f99b18fd06f555bbe5b12debfc、https://github.com/mysql/mysql-server/commit/b4287f93857bf2f99b18fd06f555bbe5b12debfc)

数据库引擎更新	Version	已修复的 MySQL 错误
Aurora MySQL 数据库引擎更新 2019-11-25 (版本 2.07.0) (已弃用)	2.07.0	<ul style="list-style-type: none">• 错误 #26251621 : INCORRECT BEHAVIOR WITH TRIGGER AND GCOL• 错误 #22574695 : ASSERTION `!TABLE (!TABLE->READ_SET BITMAP_IS_SET(TABLE->READ_SET, FIEL• 错误 #25966845 : INSERT ON DUPLICATE KEY GENERATE A DEADLOCK• 错误 #23070734 : CONCURRENT TRUNCATE TABLES CAUSE STALL• 错误 #26191879 : FOREIGN KEY CASCADES USE EXCESSIVE MEMORY• 错误 #20989615 : INNODB AUTO_INCREMENT PRODUCES SAME VALUE TWICE
Aurora MySQL 数据库引擎更新 2019-11-11 (版本 2.05.0) (已弃用)	2.05.0	<ul style="list-style-type: none">• 错误 #23054591: PURGE BINARY LOGS TO 正在读取整个 binlog 文件并导致停滞 MySql

数据库引擎更新	Version	已修复的 MySQL 错误
Aurora MySQL 数据库引擎更新 2020-08-14 (版本 2.04.9) (已弃用)	2.04.9	<ul style="list-style-type: none"> • 错误 #23070734、错误 #80060：并发 TRUNCATE TABLE 导致停机 • 错误 #23103937：PS_TRUNCATE_ALL_TABLES() 在 SUPER_READ_ONLY 模式下无法运行 • 错误 #22551677：使服务器脱机时，性能架构中的争用条件可能导致服务器退出。 • 错误 #27082268：无效的 FTS 同步。 • 错误 #12589870：修复了在启用查询缓存的情况下导致使用多查询语句重新启动的问题。 • 错误 #26402045：某些情况下的子查询具体化可能会导致服务器退出。这些查询现在会产生建议禁止具体化的错误。 • 错误 #18898433：在使用联接缓冲（例如，使用块嵌套循环算法）的情况下，带有许多左联接的查询会很慢。 • 错误 #25222337：在填充受外键约束影响的虚拟列的同时比较字段名称的过程中，虚拟索引中的 NULL 虚拟列字段名称引发了服务器退出。https://github.com/mysql/mysql-server/commit/273d5c9d7072c63b6c47dbef6963d7dc491d5131) • 错误 #25053286：执行包含访问视图的查询的存储过程可能会分配直到会话结束才释放的内存。https://github.com/mysql/mysql-server/commit/d7b37d4d141a95f577916448650c429f0d6e193d) • 错误 #25586773：执行包含从某些 SELECT (https://dev.mysql.com/doc/refman/5.7/en/select.html) 语句的内容创建表的语句的存储过程可能会导致内存泄漏。https://github.com/mysql/mysql-server/commit/88301e5adab65f6750f66af284be410c4369d0c1) • 错误 #26666274：性能架构缓冲区容器中的无限循环。 • 错误 #23550835、错误 #23298025、错误 #81464：当内部缓冲区已满时，对性能架构表执行 SELECT 可能会导致服务器退出。

数据库引擎更新	Version	已修复的 MySQL 错误
Aurora MySQL 数据库引擎更新 2019-09-19 (版本 2.04.6) (已弃用)	2.04.6	<ul style="list-style-type: none">错误 #23054591: PURGE BINARY LOGS TO 正在读取整个 binlog 文件并导致停滞 MySql
Aurora MySQL 数据库引擎更新 2019-05-02 (版本 2.04.2) (已弃用)	2.04.2	错误 #24829050 - INDEX_MERGE_INTERSECTION OPTIMIZATION CAUSES WRONG QUERY RESULTS
Aurora MySQL 数据库引擎更新 2018-10-11 (版本 2.03) (已弃用)	2.03	<ul style="list-style-type: none">分区表上的反向扫描执行 ICP - ORDER BY DESC (错误 24929748)。JSON_OBJECT 创建无效的 JSON 代码 (错误 26867509)。插入大型 JSON 数据花费大量时间 (错误 22843444)。分区表在 5.7 中使用的内存比 5.6 多 (错误 25080442)。

数据库引擎更新	Version	已修复的 MySQL 错误
Aurora MySQL 数据库引擎更新 2018-09-21 (版本 2.02.4) (已弃用)	2.02.4	<ul style="list-style-type: none"> • BUG#13651665 INNODB MAY BE UNABLE TO LOAD TABLE DEFINITION AFTER RENAME • BUG#21371070 INNODB: CANNOT ALLOCATE 0 BYTES. • BUG#21378944 FTS ASSERT ENC.SRC_ILIST_PTR != NULL, FTS_OPTIMIZE_WORD(), OPTIMIZE TABLE • BUG#21508537 ASSERTION FAILURE UT_A(!VICTIM_TRX->READ_ONLY) • BUG#21983865 UNEXPECTED DEADLOCK WITH INNODB_AUTOINC_LOCK_MODE=0 • BUG#22679185 INVALID INNODB FTS DOC ID DURING INSERT • BUG#22899305 GCOLS: ASSERTION: !(COL->PR TYPE & 256). • BUG#22956469 MEMORY LEAK INTRODUCED IN 5.7.8 IN MEMORY/INNODB/OS0FILE • BUG#22996488 CRASH IN FTS_SYNC_INDEX WHEN DOING DDL IN A LOOP • BUG#23014521 GCOL:INNODB: ASSERTION: !IS_V • BUG#23021168 REPLICATION STOPS AFTER TRX IS ROLLED BACK ASYNC • BUG#23052231 ASSERTION: ADD_AUTOINC < DICT_TABLE_GET_N_USER_COLS • BUG#23149683 ROTATE INNODB MASTER KEY WITH KEYRING_OKV_CONF_DIR MISSING: SIGSEGV; SIGNAL 11 • BUG#23762382 INSERT VALUES QUERY WITH JOIN IN A SELECT CAUSES INCORRECT BEHAVIOR • BUG#25209512 CURRENT_TIMESTAMP PRODUCES ZEROS IN TRIGGER • BUG#26626277 BUG IN "INSERT... ON DUPLICATE KEY UPDATE" QUERY

数据库引擎更新	Version	已修复的 MySQL 错误
		<ul style="list-style-type: none"> • BUG#26734162 INCORRECT BEHAVIOR WITH INSERT OF BLOB + ON DUPLICATE KEY UPDATE • BUG#27460607 INCORRECT WHEN INSERT SELECT'S SOURCE TABLE IS EMPTY
Aurora MySQL 数据库引擎更新 2018-05-03 (版本 2.02) (已弃用)	2.02.0	左侧联接在外侧返回错误结果 (错误 #22833364)

由 Aurora MySQL 1.x 数据库引擎更新修复的 MySQL 错误

MySQL 5.6 兼容版本 Aurora 包含 MySQL 5.6.10 和之前的所有 MySQL 错误修复。下表列出了 Aurora MySQL 数据库引擎更新修复的其他 MySQL 错误以及修复这些错误的更新。

数据库引擎更新	Version	已修复的 MySQL 错误
Aurora MySQL 数据库引擎更新 2021-03-18 (版本 1.23.2) (已弃用)	1.23.2	<ul style="list-style-type: none"> • 复制：在执行 SHOW BINLOG EVENTS 语句时，会阻止所有并行事务。此修复确保 SHOW BINLOG EVENTS 进程现在仅在计算文件终止位置期间锁定，因此不会长时间阻止并行交易。(错误 #76618、错误 #20928790)
Aurora MySQL 数据库引擎更新 2020-09-02 (版本 1.23.0) (已弃用)	1.23.0	<ul style="list-style-type: none"> • 具有 ALTER TABLE ADD COLUMN ALGORITHM=QUICK 的二进制日志事件将被重写为 ALGORITHM=DEFAULT，以便与社区版本兼容。 • 错误 #22350047：IF CLIENT KILLED AFTER ROLLBACK TO SAVEPOINT PREVIOUS STMTS COMMITTED • 错误 #29915479：RUNNING COM_REGISTER_SLAVE WITHOUT COM_BINLOG_DUMP CAN RESULTS IN SERVER EXIT • 错误 #30441969：错误 #29723340：MYSQL SERVER CRASH AFTER SQL QUERY WITH DATA ?AST • 错误 #30628268：OUT OF MEMORY CRASH

数据库引擎更新	Version	已修复的 MySQL 错误
		<ul style="list-style-type: none"> • 错误 #27081349 : UNEXPECTED BEHAVIOUR WHEN DELETE WITH SPATIAL FUNCTION • 错误 #27230859 : UNEXPECTED BEHAVIOUR WHILE HANDLING INVALID POLYGON" • 错误 #27081349 : UNEXPECTED BEHAVIOUR WHEN DELETE WITH SPATIAL" • 错误 #26935001 : ALTER TABLE AUTO_INCREMENT TRIES TO READ INDEX FROM DISCARDED TABLESPACE • 错误 #29770705 : SERVER CRASHED WHILE EXECUTING SELECT WITH SPECIFIC WHERE CLAUSE • 错误 #27659490 : SELECT USING DYNAMIC RANGE AND INDEX MERGE USE TOO MUCH MEMORY(OOM) • 错误 #24786290 : REPLICATION BREAKS AFTER 错误 #74145 HAPPENS IN MASTER • 错误 #27703912 : EXCESSIVE MEMORY USAGE WITH MANY PREPARE • 错误 #20527363 : TRUNCATE TEMPORARY TABLE CRASH : !DICT_TF2_FLAG_IS_SET(TABLE, DICT_TF2_TEMPORARY) • 错误 #23103937 : PS_TRUNCATE_ALL_TABLES() DOES NOT WORK IN SUPER_READ_ONLY MODE • 错误 #25053286 : USE VIEW WITH CONDITION IN PROCEDURE CAUSES INCORRECT BEHAVIOR (fixed in 5.6.36) • 错误 #25586773 : INCORRECT BEHAVIOR FOR CREATE TABLE SELECT IN A LOOP IN SP (fixed in 5.6.39) • 错误 #27407480 : AUTOMATIC_SP_PRIVILEGES REQUIRES NEED THE INSERT PRIVILEGES FOR MYSQL.USER TABLE • 错误 #26997096 : relay_log_space 值不以同步的方式更新，因此其值有时大大高于中继日志使用的实际磁盘空间。

数据库引擎更新	Version	已修复的 MySQL 错误
		<ul style="list-style-type: none"> • 错误 #15831300 SLAVE_TYPE_CONVERSIONS=ALL_NON_LOSSY 无法按预期工作 • SSL 错误逆向移植错误 #17087862，错误 #20551271 • 错误 #16894092：PERFORMANCE REGRESSION IN 5.6.6+ FOR INSERT INTO ... SELECT ... FROM (在 5.6.15 中已修复)。 • 移植与 SLAVE_TYPE_CONVERSIONS 相关的错误修复。
Aurora MySQL 数据库引擎更新 2020-11-09 (版本 1.22.3) (已弃用)	1.22.3	<ul style="list-style-type: none"> • 错误 #26654685：检查外键时遇到的损坏索引 ID 引发断言 • 错误 #15831300：默认情况下，当将整数从主节点上较小的类型提升到从属节点上的较大类型时（例如，从主节点上的 SMALLINT 列到从属节点上的 BIGINT）时，提升的值将被视为已经过签名的值。现在，在此情况下，可以使用为 slave_type_conversions 服务器系统变量指定的值集中的 ALL_SIGNED、ALL_UNSIGNED 中的一个或两个修改或覆盖此行为。有关详细信息，请参阅基于行的复制：属性升级和降级以及变量的说明。 • 错误 #17449901：使用 foreign_key_checks=0，InnoDB 允许删除外键约束所需的索引，将表置于不一致状态，并导致表加载时发生的外键检查失败。InnoDB 现在可以防止删除外键约束所需的索引，即使在 foreign_key_checks=0 的情况下亦是如此。在删除外键索引之前，必须删除外键约束。 • 错误 #20768847：ALTER TABLE ... DROP INDEX 对具有外键依赖项的表进行的操作会引发断言。

数据库引擎更新	Version	已修复的 MySQL 错误
Aurora MySQL 数据库引擎更新 2019-11-25 (版本 1.22.0) (已弃用)	1.22.0	<ul style="list-style-type: none"> 错误 #16346241 - SERVER CRASH IN ITEM_PARAM::QUERY_VAL_STR 错误 #17733850 - NAME_CONST() CRASH IN ITEM_NAME_CONST::ITEM_NAME_CONST() 错误 #20989615 - INNODB AUTO_INCREMENT PRODUCES SAME VALUE TWICE 错误 #20181776 - ACCESS CONTROL DOESN'T MATCH MOST SPECIFIC HOST WHEN IT CONTAINS WILDCARD 错误 #27326796 - MYSQL CRASH WITH INNODB ASSERTION FAILURE IN FILE PARS0PARS.CC 错误 #20590013 - IF YOU HAVE A FULLTEXT INDEX AND DROP IT YOU CAN NO LONGER PERFORM ONLINE DDL
Aurora MySQL 数据库引擎更新 2019-11-25 (版本 1.21.0) (已弃用)	1.21.0	<ul style="list-style-type: none"> 错误 #19929406 : HANDLE_FATAL_SIGNAL (SIG=11) IN __MEMMOVE_SSSE3_BACK FROM STRING::COPY 错误 #17059925 : 对于 UNION 语句，rows-examined 值计算错误。这表现为 Performance Schema 语句表 (例如 events_statements_current) 的 ROWS_EXAMINED 列的值太大。 错误 #11827369 : 一些带有 SELECT ... FROM DUAL 嵌套子查询的查询引发断言。 错误 #16311231 : 如果一个查询在 IN 子句中包含了子查询，而 WHERE 子句中包含 XOR 操作，则会返回错误的结果。

数据库引擎更新	Version	已修复的 MySQL 错误
Aurora MySQL 数据库引擎更新 2019-11-11 (版本 1.20.0) (已弃用)	1.20.0	<ul style="list-style-type: none"> 错误 #19929406 : HANDLE_FATAL_SIGNAL (SIG=11) IN __MEMMOVE_SSSE3_BACK FROM STRING::COPY 错误 #17059925 : 对于 UNION 语句, rows-examined 值计算错误。这表现为 Performance Schema 语句表 (例如 events_statements_current) 的 ROWS_EXAMINED 列的值太大。 错误 #11827369 : 一些带有 SELECT ... FROM DUAL 嵌套子查询的查询引发断言。 错误 #16311231 : 如果一个查询在 IN 子句中包含了子查询, 而 WHERE 子句中包含 XOR 操作, 则会返回错误的结果。
Aurora MySQL 数据库引擎更新 2019-09-19 (版本 1.19.5) (已弃用)	1.19.5	<ul style="list-style-type: none"> CVE-2018-2696 CVE-2015-4737 错误 #19929406 : HANDLE_FATAL_SIGNAL (SIG=11) IN __MEMMOVE_SSSE3_BACK FROM STRING::COPY 错误 #17059925 : 对于 UNION 语句, rows-examined 值计算错误。这表现为 Performance Schema 语句表 (例如 events_statements_current) 的 ROWS_EXAMINED 列的值太大。 错误 #11827369 : 一些带有 SELECT ... FROM DUAL 嵌套子查询的查询引发断言。 错误 #16311231 : 如果一个查询在 IN 子句中包含了子查询, 而 WHERE 子句中包含 XOR 操作, 则会返回错误的结果。
Aurora MySQL 数据库引擎更新 2019-02-07 (版本 1.19.0) (已弃用)	1.19.0	<ul style="list-style-type: none"> 错误 #32917 : DETECT ORPHAN TEMP-POOL FILES, AND HANDLE GRACEFULLY 错误 #63144 CREATE TABLE IF NOT EXISTS METADATA LOCK IS TOO RESTRICTIVE

数据库引擎更新	Version	已修复的 MySQL 错误
Aurora MySQL 数据库引擎更新 2019-01-17 (版本 1.17.8) (已弃用)	1.17.8	<ul style="list-style-type: none"> 错误 #13418638 : CREATE TABLE IF NOT EXISTS METADATA LOCK IS TOO RESTRICTIVE
Aurora MySQL 数据库引擎更新 2018-10-08 (版本 1.17.7) (已弃用)	1.17.7	<ul style="list-style-type: none"> 在外键列上删除索引导致表丢失。(错误 16208542) 在 add_derived_key() 中发生内存泄漏。(错误 76349) 对于分区表，查询可能会返回不同的结果，具体取决于是否使用索引合并。(错误 16862316) 在针对由 HASH 分区的表运行时，使用 index_merge 优化的查询 (请参阅索引合并优化) 可能会返回无效的结果。(错误 17588348)
Aurora MySQL 数据库引擎更新 2018-09-06 (版本 1.17.6) (已弃用)	1.17.6	<ul style="list-style-type: none"> 对于重命名或更改 BINARY 列默认值的 ALTER TABLE 语句，修改是使用表副本完成的，而不是就位完成的。(错误 67141、错误 14735373、错误 69580、错误 17024290) 常规表和隐式为组的派生表之间的外部联接可能会导致服务器退出。(错误 16177639)
Aurora MySQL 数据库引擎更新 2018-03-13 (版本 1.17) (已弃用)	1.17.0	<ul style="list-style-type: none"> 使用复制筛选条件时 LAST_INSERT_ID 不正确复制 (错误 69861) 根据 INDEX_MERGE 设置的不同，查询返回不同的结果 (错误 16862316) 查询过程重新执行存储例程，查询计划效率低下 (错误 16346367) InnoDB FTS : 在 FTS_CACHE_APPEND_DELETED_DOC_IDS 中断言 (错误18079671) 在 ALTER TABLE CHANGE COLUMN 中断言 RBT_EMPTY (INDEX_CACHE->WORDS) (错误 17536995) INNODB 全文搜索在涉及保存点时找不到记录 (错误 70333、错误 17458835)

数据库引擎更新	Version	已修复的 MySQL 错误
Aurora MySQL 数据库引擎更新 2017-11-20 (版本 1.15.1) (已弃用)	1.15.1	<ul style="list-style-type: none"> 已恢复 — MySQL 实例一直在“执行 SYNC 索引” (错误 73816) 已恢复 — 在 ALTER TABLE CHANGE COLUMN 中断言 RBT_EMPTY(INDEX_CACHE->WORDS) (错误 17536995) 已恢复 — InnoDB 全文搜索在涉及保存点时找不到记录 (错误 70333)
Aurora MySQL 数据库引擎更新 2017-10-24 (版本 1.15) (已弃用)	1.15.0	<ul style="list-style-type: none"> CREATE USER 接受插件和密码哈希，但忽略密码哈希 (错误 78033) 分区引擎将字段添加到读位集以便能够从分区索引返回排序的条目。这将促使联接缓冲区尝试读取不需要的字段。通过不向 read_set 添加所有分区字段，而只是对 read_set 中的前缀字段排序来修复该错误。添加了 DEBUG_ASSERT，如果执行 key_cmp，则至少必须读取第一个字段 (错误 16367691)。 MySQL 实例一直在“执行 SYNC 索引” (错误 73816) 在 ALTER TABLE CHANGE COLUMN 中断言 RBT_EMPTY (INDEX_CACHE->WORDS) (错误 17536995) InnoDB 全文搜索在涉及保存点时找不到记录 (错误 70333)
Aurora MySQL 数据库引擎更新：2018-03-13 (版本 1.14.4) (已弃用)	1.14.4	<ul style="list-style-type: none"> 可忽略的事件不起作用，并且未经过测试 (错误 74683) NEW->OLD ASSERT FAILURE 'GTID_MODE > 0' (Bug #20436436)
Aurora MySQL 数据库引擎更新：2017-08-07 (版本 1.14) (已弃用)	1.14.0	与派生表 (FROM 子句中的子查询) 相结合的全文搜索导致服务器退出。现在，如果全文操作依赖于派生表，服务器将产生指示无法对具体化表执行全文搜索的错误。(错误 68751、错误 16539903)

数据库引擎更新	Version	已修复的 MySQL 错误
Aurora MySQL 数据库引擎更新：2017-05-15 (版本 1.13) (已弃用)	1.13.0	<ul style="list-style-type: none"> 当因空值而导致 AUTO_INCREMENT 值被重置时，重新加载已移出的表。(错误 21454472、错误 77743) 由于 purge_node_t structure 中存在不一致，在回滚中未找到索引记录。产生不一致的来源是警告和错误消息，如“二级索引条目更新出错”、“无法清除记录”以及“尝试清除未标记为删除的二级索引条目”等。(错误 19138298、错误 70214、错误 21126772、错误 21065746) qsort 操作堆栈大小计算错误导致堆栈溢出。(错误 73979) 回滚时索引中未找到记录。(错误 70214、错误 72419) ALTER TABLE 添加列 TIMESTAMP 更新 CURRENT_TIMESTAMP 时插入了零数据 (错误 17392)
Aurora MySQL 数据库引擎更新：2017-04-05 (版本 1.12) (已弃用)	1.12.0	<ul style="list-style-type: none"> 当因空值而导致 AUTO_INCREMENT 值被重置时，重新加载已移出的表。(错误 21454472、错误 77743) 由于 purge_node_t structure 中存在不一致，在回滚中未找到索引记录。产生不一致的来源是警告和错误消息，如“二级索引条目更新出错”、“无法清除记录”以及“尝试清除未标记为删除的二级索引条目”等。(错误 19138298、错误 70214、错误 21126772、错误 21065746) qsort 操作堆栈大小计算错误导致堆栈溢出。(错误 73979) 回滚时索引中未找到记录。(错误 70214、错误 72419) ALTER TABLE 添加列 TIMESTAMP 更新 CURRENT_TIMESTAMP 时插入了零数据 (错误 17392)
Aurora MySQL 数据库引擎更新：2017-02-23 (版本 1.11) (已弃用)	1.11.0	<ul style="list-style-type: none"> 将 ALTER table DROP foreign key 与另一个 DROP 操作一起运行会导致表消失。(错误 16095573) 一些使用 ORDER BY 的 INFORMATION_SCHEMA 查询不使用文件排序优化，因为它们之前已使用此优化。(错误 16423536) FOUND_ROWS () 返回错误的表行数。(错误 68458) 服务器失败，而不是在打开的临时表过多时显示错误。(错误 18948649)

数据库引擎更新	Version	已修复的 MySQL 错误
Aurora MySQL 数据库引擎更新：2016-12-14 (版本 1.10) (已弃用)	1.10.0	<ul style="list-style-type: none"> 对派生的表执行 UNION 将返回错误结果，显示 '1=0/false'-clauses。(错误 69471) 在第二次对存储过程执行 ITEM_FUNC_GROUP_CONCAT::FIX_FIELDS 时，服务器发生崩溃。(错误 20755389) 通过在缓存大小超过总大小的 10% 时将缓存同步任务卸载到单独的线程，来避免 MySQL 查询在 FTS 缓存同步到磁盘期间卡顿太长时间。(错误 22516559，#73816)
Aurora MySQL 数据库引擎更新：2016-10-26 (版本 1.8.1) (已弃用)	1.8.1	<ul style="list-style-type: none"> 由于这个问题，OpenSSL 更改了 Diffie-Hellman 密钥长度参数。LogJam (错误 18367167)
Aurora MySQL 数据库引擎更新：2016-10-18 (版本 1.8) (已弃用)	1.8.0	<ul style="list-style-type: none"> 在具有多个索引的列上删除所有索引时，InnoDB 在外键约束需要索引的情况下无法阻止 DROP INDEX 操作。(错误 16896810) 解决了添加外键约束崩溃情况。(错误 16413976) 修复了在提取存储过程中的游标并同时分析或刷新表时发生的崩溃。(错误 18158639) 修复了当用户更改表来将 AUTO_INCREMENT 值更改为低于最大自动递增列值时出现的自动递增错误。(错误 16310273)

数据库引擎更新	Version	已修复的 MySQL 错误
Aurora MySQL 数据库引擎更新：2016-08-30 (版本 1.7.0) (已弃用)	1.7.0	<ul style="list-style-type: none"> • 通过对 LOCK_grant 锁进行分区，提高了可扩展性。(端口 WL #8355) • 存储过程中在 SELECT 上打开游标可能导致 segfault。(端口错误 16499751) • MySQL 提供了有关一些特殊用法的错误结果。(错误 11751794) • GET_SEL_ARG_FOR_KEYPART 中发生崩溃 – 由错误 11751794 的修补程序导致。(错误 16208709) • 使用 GROUP BY 的简单查询的错误结果。(错误 17909656) • 包含范围谓词的半联接查询上的额外行。(错误 16221623) • 在 IN 子查询后添加 ORDER BY 子句可能导致返回重复行。(错误 16308085) • 发生崩溃，包含针对带 GROUP BY、MyISAM 的松散扫描的查询的说明。(错误 16222245) • 包含带引号 int 谓词的松散索引扫描将返回随机数据。(错误 16394084) • 如果优化程序使用的是松散索引扫描，则服务器可能在尝试创建临时表时退出。(错误 16436567) • COUNT(DISTINCT) 将不会计算 NULL 值，但会在优化程序使用松散索引扫描时计算这些值。(错误 17222452) • 如果查询同时包含 MIN()/MAX() 和 aggregate_function (DISTINCT) (例如，SUM(DISTINCT))，并且是使用松散索引扫描执行的，则 MIN()/MAX() 的结果值的设置错误。(错误 17217128)
Aurora MySQL 数据库引擎更新：2016-06-01 (版本 1.6.5) (已弃用)	1.6.5	<ul style="list-style-type: none"> • SLAVE CAN'T CONTINUE REPLICATION AFTER MASTER'S CRASH RECOVERY (Port Bug #17632285)

数据库引擎更新	Version	已修复的 MySQL 错误
Aurora MySQL 数据库引擎更新 : 2016-04-06 (版本 1.6) (已弃用)	1.6.0	<ul style="list-style-type: none"> • BACKPORT 错误 18694052 FIX FOR ASSERTION `! M_ORDERED_REC_BUFFER' FAILED TO 5.6 (端口错误 18305270) • SEGV IN MEMCPY(), HA_PARTITION::POSITION (端口错误 18383840) • WRONG RESULTS WITH PARTITIONING, INDEX_MERGE AND NO PK (端口错误 18167648) • FLUSH TABLES FOR EXPORT: ASSERTION IN HA_PARTITION::EXTRA (端口错误 16943907) • SERVER CRASH IN VIRTUAL HA_ROWS_HANDLER::MULTI_RANGE_READ_INFO_CONST (端口错误 16164031) • 范围优化程序在 SEL_ARG::RB_INSERT() 中崩溃 (端口错误 16241773)
Aurora MySQL 数据库引擎更新 : 2016-01-11 (版本 1.5) (已弃用)	1.5.0	<ul style="list-style-type: none"> • 处理了影响以数字开头的数据库名称所在的表的 MySQL 全文搜索中的未完成修复。(端口错误 17607956)

数据库引擎更新	Version	已修复的 MySQL 错误
Aurora MySQL 数据库引擎更新：2015-12-03 (版本 1.4) (已弃用)	1.4	<ul style="list-style-type: none"> • FTSPARSE() 中的 SEGV。(错误 16446108) • InnoDB 数据字典在重命名列时不会更新。(错误 19465984) • FTS 在将表重命名为其他数据库后发生崩溃。(错误 16834860) • 无法为截断的表准备触发器会生成错误 1054。(错误 18596756) • 元数据更改可能会导致触发器执行出现问题。(错误 18684393) • 未为较长的 UTF8 VARCHAR 字段选择具体化。(错误 17566396) • 将 ORDER BY 与限制 X 结合使用时，执行计划性能下降。(错误 #16697792) • 将错误 11765744 逆向移植到 5.1、5.5 和 5.6。(错误 17083851) • 生成 SIG6 的 SQL/SQL_SHOW.CC 中的互斥问题。根源可能是 FILL_VARIABLES。(错误 20788853) • 将错误 18008907 逆向移植到 5.5+ 版本。(错误 18903155) • 适应 MySQL 5.7 中的堆栈溢出错误的修复。(错误 19678930)

数据库引擎更新	Version	已修复的 MySQL 错误
Aurora MySQL 数据库引擎更新：2015-10-16 (版本 1.2、1.3) (已弃用)	1.2、1.3	<ul style="list-style-type: none"> • 终止 innodb 中的查询会导致该查询最终崩溃并显示断言。(错误 1608883) • 如果无法为事件调度程序、事件执行或新连接创建新线程，则不会将任何消息写入错误日志。(错误 16865959) • 如果一个连接更改了其默认数据库，同时另一个连接执行了 SHOW PROCESSLIST，则第二个连接可在尝试显示第一个连接的默认数据库内存时访问无效内存。(错误 11765252) • 虽然 PURGE BINARY LOGS 设计为不删除正在使用或活动的二进制日志文件，但发生该操作时，不会提供任何通知。(错误 13727933) • 对于某些语句，在优化程序删除不需要的子查询子句时，可能会导致内存泄漏。(错误 15875919) • 关闭期间，服务器可能会尝试锁定未初始化的互斥。(错误 16016493) • 使用 GROUP_CONCAT() 的准备好的语句和已指定多个列的 ORDER BY 子句可能会导致服务器退出。(错误 16075310) • 副本工作线程缺少性能架构检测。(错误 16083949) • STOP SLAVE 与诸如 SHOW STATUS (检索 Slave_retrieved_transactions、Slave_heartbeat_period、Slave_received_heartbeats、Slave_last_heartbeat 和 Slave_running 中的一个或多个状态变量的值) 之类的语句同时发出时，可能会导致死锁。(错误 16088188) • 在搜索词为用引号引起来的短语的某些情况下，使用布尔值模式的全文查询可能会返回零个结果。(错误 16206253) • 在尝试在子查询中的联接的 ON 子句中使用子查询执行准备好的语句时，优化程序尝试删除引发断言的冗余子查询子句。(错误 16318585) • GROUP_CONCAT 不稳定，在 ITEM_SUM::CLEAN_UP_AFTER_REMOVAL 中发生崩溃。(错误 16347450) • 尝试通过创建具有与 INFORMATION_SCHEMA.INNODB_FT_DEFAULT_STOPWORD 相同的结构的 InnoDB 表来替换默

数据库引擎更新	Version	已修复的 MySQL 错误
		<p>认 InnoDB 全文搜索 (FTS) 非索引字列表将生成错误。(错误 16373868)</p> <ul style="list-style-type: none"> • 在工作线程上的客户端线程执行 FLUSH TABLES WITH READ LOCK 并接着在主实例上进行一些更新后，工作线程在执行 SHOW SLAVE STATUS 时挂起。(错误 16387720) • 在全文搜索中分析分隔的搜索字符串 (如“abc-def”) 时，InnoDB 现在会使用与 MyISAM 相同的单词分隔符。(错误 16419661) • FTS_AST_TERM_SET_WILDCARD 中发生崩溃。(错误 16429306) • 针对 FTS RQG 测试的 FTS_AST_VISIT() 中的 SEGFAULT。(错误 16435855) • 在优化程序删除指向子查询的 Item_ref 后，调试版本会导致服务器退出。(错误 16509874) • InnoDB 表上的全文搜索无法搜索组合了 + 或 - 运算符的文字短语。(错误 16516193) • START SLAVE 当服务器启动时，选项 --master-info-repository=TABLE relay-log-info-repository=TABLE 且自动提交设置为 0 时失败。--skip-slave-start (错误 16533802) • 非常大的 InnoDB 全文搜索 (FTS) 结果可能会占用大量内存。(错误 16625973) • 在调试版本中，由于二进制数据可能包含 NULL 字节和其他无意义的字符，因此当直接在搜索字符串中使用二进制数据时，OPT_CHECK_ORDER_BY 中可能发生断言。(错误 16766016) • 对于某些语句，在优化程序删除不需要的子查询子句时，可能会导致内存泄漏。(错误 16807641) • 通过在工作线程的新连接中发出 STOP SLAVE 来发出 FLUSH TABLES WITH READ LOCK，然后使用原始连接发出 SHOW SLAVE STATUS 后，可能会导致锁死。(错误 16856735) • 具有无效分隔符的 GROUP_CONCAT() 可能会导致服务器退出。(错误 16870783)

数据库引擎更新	Version	已修复的 MySQL 错误
		<ul style="list-style-type: none"> • 服务器对任何 SHOW STATUS LIKE“模式”语句的 LOCK_active_mi 和 active_mi->rli->data_lock 互斥对象执行了过度锁定，甚至在该模式不匹配使用这些互斥对象的状态变量 (Slave_heartbeat_period 、 Slave_last_heartbeat 、 Slave_received_heartbeats 、 Slave_retried_transactions 、 Slave_running) 的情况下也是如此。(错误 16904035) • 使用 IN BOOLEAN MODE 修饰符的全文搜索将导致断言失败。(错误 16927092) • InnoDB 表上的全文搜索无法执行使用 + 布尔值运算符的搜索。(错误 17280122) • 4 向锁死：zombie、清除二进制日志、显示 processlist、显示二进制日志。(错误 17283409) • 当等待提交锁定的 SQL 线程终止并重新启动时，导致在工作线程上跳过事务。(错误 17450876) • InnoDB 全文搜索失败可能是由于“unended”令牌导致的。应传递字符串和字符串长度以进行字符串比较。(错误 17659310) • 在 MySQL 5.6 或 5.7 中使用大量已分区的 InnoDB 表可能比在 MySQL Server 的先前版本中使用相同的表所占用的内存更多。(错误 17780517) • 对于全文查询，未能检查出 num_token 少于 max_proximity_item 可能会导致断言。(错误 18233051) • 当存在大量空 InnoDB 表时，INFORMATION_SCHEMA TABLES 和 COLUMNS 表的特定查询可能会导致内存使用过度。(错误 18592390) • 当提交事务时，标记现在用于检查是否创建了线程，而不是使用更多资源来检查线程本身 (尤其是在 master_info_repository=TABLE 的情况下运行服务器时)。(错误 18684222) • 如果工作线程上的客户端线程在主实例执行 DML 时执行了 FLUSH TABLES WITH READ LOCK，则在同一客户端上执行 SHOW SLAVE STATUS 会受阻，从而导致死锁。(错误 19843808)

数据库引擎更新	Version	已修复的 MySQL 错误
		<ul style="list-style-type: none">按 GROUP_CONCAT() 结果排序可能导致服务器退出。(错误 19880368)
Aurora MySQL 数据库引擎更新 : 2015-08-24 (版本 1.1) (已弃用)	1.1	<ul style="list-style-type: none">名称以数字开头的 InnoDB 数据库导致的全文搜索 (FTS) 分析器错误。(错误 17607956)名称以数字开头的数据库中的 InnoDB 全文搜索失败。(错误 17161372)对于 Windows 上的 InnoDB 数据库，全文搜索 (FTS) 对象 ID 未使用预期的十六进制格式。(错误 16559254)MySQL 5.6 中引入的代码回归对 DROP TABLE 和 ALTER TABLE 性能有负面影响。这可能会导致 MySQL Server 5.5.x 和 5.6.x 之间的性能下降。(错误 16864741)

Aurora MySQL 中修复的安全漏洞

常见漏洞和风险 (CVE) 是公开已知网络安全漏洞的条目列表。每个条目均包含标识号、描述和至少一个公共参考。

您可以在此页面上找到 Aurora MySQL 中修复的安全漏洞列表。有关 Aurora 安全性的一般信息，请参阅《Amazon Aurora 用户指南》中的 [Amazon Aurora 中的安全性](#)。有关 Aurora MySQL 的其他安全性信息，请参阅《Amazon Aurora 用户指南》中的 [使用 Amazon Aurora MySQL 实现高安全性](#)。

我们建议您始终升级到最新 Aurora 版本，以防止已知漏洞。您可以使用此页面验证 Aurora MySQL 的特定版本是否修复了特定安全漏洞。如果您的集群没有安全修复，您可以看到应该升级到哪个 Aurora MySQL 版本以获取该修复。

在 Aurora MySQL 版本 1、2 和 3 中修复的 CVE 也列在该版本的发布说明中：

- [Amazon Aurora MySQL 版本 3 的数据库引擎更新](#)
- [Amazon Aurora MySQL 2 的数据库引擎更新](#)
- [Amazon Aurora MySQL 版本 1 的数据库引擎更新 \(已弃用\)](#)

Note

Aurora MySQL 版本 3 的初始版本包括社群 MySQL 8.0.23 之前修复的所有 CVE。对于未来修复的 CVE，请查找此处和 Aurora MySQL 版本 3 发行说明中列出的它们。

CVE 和最低固定 Aurora MySQL 版本

- [CVE-2024-20963](#) : [2.12.2](#)、[2.11.5](#)
- [CVE-2023-39975](#) : [3.06.0](#)、[3.05.2](#)、[3.04.2](#)、[2.12.2](#)、[2.11.5](#)
- [CVE-2023-38546](#) : [3.06.0](#)、[3.04.2](#)、[2.11.5](#)
- [CVE-2023-38545](#) : [3.06.0](#)、[3.05.2](#)、[3.0](#)
[5.1](#)、[3.05.1](#)、[3.05.1](#)、[3.04.2](#)、[3.03.3](#)、[2.12.2](#)、[2.12.1](#)、[2.12.1](#)、[2.12.0.1](#)、[2.12.0.1](#)、[2.11.5](#)
- [CVE-2023-22084](#) : [2.11.5](#)
- [CVE-2023-22053](#) : [2.12.1](#)
- [CVE-2023-22028](#) : [2.12.1](#)、[2.11.5](#)

- [CVE-2023-22026](#) : [2.12.1](#)、[2.11.5](#)
- [CVE-2023-22015](#) : [2.12.1](#)、[2.11.5](#)
- [CVE-2023-21963](#) : [3.04.0](#)、[3.03.2](#)、[2.12.0](#)、[2.11.3](#)
- [CVE-2023-21912](#) : [3.04.0](#)、[3.03.2](#)、[2.12.0](#)、[2.11.3](#)
- [CVE-2023-21840](#) : [2.12.0](#)
- [CVE-2023-0215](#) : [3.04.0](#)、[3.03.2](#)、[2.12.0](#)、[2.11.3](#)
- [CVE-2022-43551](#) : [3.04.0](#)、[3.03.2](#)、[2.12.0](#)、[2.11.3](#)
- [CVE-2022-37434](#) : [3.05.0](#)、[3.04.0](#)、[3.03.2](#)、[2.12.0](#)、[2.11.3](#)
- [CVE-2022-32221](#) : [3.03.0](#)、[2.12.0](#)、[2.11.1](#)、[2.07.9](#)
- [CVE-2022-24407](#) : [2.12.1](#)、[2.11.4](#)
- [CVE-2022-21635](#) : [3.04.0](#)
- [CVE-2022-21556](#) : [3.04.0](#)
- [CVE-2022-21460](#) : [2.11.0](#)
- [CVE-2022-21451](#) : [3.03.0](#)、[3.02.2](#)、[2.11.0](#)
- [CVE-2022-21444](#) : [3.03.0](#)、[3.02.2](#)、[2.11.0](#)、[2.10.3](#)
- [CVE-2022-21417](#) : [2.11.0](#)
- [CVE-2022-21352](#) : [3.04.0](#)
- [CVE-2022-21344](#) : [2.10.3](#)
- [CVE-2022-21304](#) : [2.11.0](#)、[2.10.3](#)
- [CVE-2022-21303](#) : [2.11.0](#)
- [CVE-2022-21245](#) : [2.11.0](#)、[2.10.3](#)、[2.07.8](#)
- [CVE-2022-0778](#) : [3.02.1](#)、[2.11.0](#)
- [CVE-2021-36222](#) : [3.02.2](#)、[3.01.1](#)、[2.12.0](#)、[2.11.1](#)、[2.11.0](#)、[2.10.3](#)、[2.10.2](#)、[2.07.8](#)
- [CVE-2021-35630](#) : [3.04.0](#)
- [CVE-2021-35624](#) : [3.04.0](#)、[2.10.2](#)
- [CVE-2021-35604](#) : [2.10.2](#)
- [CVE-2021-28196](#) : [2.11.0](#)
- [CVE-2021-23841](#) : [2.11.0](#)、[2.10.0](#)、[2.09.3](#)、[1.23.3](#)
- [CVE-2021-22946](#) : [3.02.0](#)、[3.01.1](#)、[2.12.0](#)
- [CVE-2021-22926](#) : [3.02.2](#)、[3.01.1](#)、[2.11.1](#)、[2.11.0](#)、[2.10.3](#)、[2.10.22.07.8](#)

- [CVE-2021-3712](#) : [2.09.3](#)
- [CVE-2021-3449](#) : [2.11.0](#)、[2.10.0](#)、[2.09.3](#)、[1.23.3](#)
- [CVE-2021-2390](#) : [2.10.2](#)
- [CVE-2021-2389](#) : [2.10.2](#)
- [CVE-2021-2385](#) : [2.10.2](#)
- [CVE-2021-2356](#) : [2.10.2](#)
- [CVE-2021-2307](#) : [2.11.0](#)、[2.10.1](#)、[2.09.3](#)、[1.23.4](#)
- [CVE-2021-2226](#) : [2.11.0](#)、[2.10.1](#)、[2.09.3](#)、[1.23.4](#)
- [CVE-2021-2202](#) : [2.11.0](#)
- [CVE-2021-2194](#) : [2.11.0](#)、[2.10.1](#)
- [CVE-2021-2179](#) : [2.11.0](#)
- [CVE-2021-2178](#) : [2.11.0](#)
- [CVE-2021-2174](#) : [2.11.0](#)、[2.10.1](#)、[2.09.3](#)
- [CVE-2021-2171](#) : [2.11.0](#)、[2.10.1](#)、[2.09.3](#)
- [CVE-2021-2169](#) : [2.12.0](#)、[2.11.1](#)、[2.11.0](#)、[2.10.1](#)、[2.09.3](#)
- [CVE-2021-2166](#) : [2.11.0](#)、[2.10.1](#)、[2.09.3](#)
- [CVE-2021-2160](#) : [2.11.0](#)、[2.10.1](#)、[1.23.4](#)
- [CVE-2021-2154](#) : [2.11.0](#)、[2.10.1](#)、[2.09.3](#)、[1.23.4](#)
- [CVE-2021-2144](#) : [2.07.3](#)
- [CVE-2021-2060](#) : [2.10.1](#)、[2.09.3](#)、[1.23.4](#)
- [CVE-2021-2032](#) : [2.10.1](#)、[2.09.3](#)、[1.23.4](#)
- [CVE-2021-2001](#) : [2.10.1](#)、[2.09.3](#)、[1.23.4](#)
- [CVE-2020-28196](#) : [2.10.0](#)、[2.09.3](#)、[1.23.3](#)
- [CVE-2020-14867](#) : [1.23.2](#)、[1.22.4](#)
- [CVE-2020-14812](#) : [2.09.2](#)、[2.07.4](#)、[1.23.2](#)、[1.22.4](#)
- [CVE-2020-14793](#) : [2.09.2](#)、[2.07.4](#)、[1.23.2](#)、[1.22.4](#)
- [CVE-2020-14790](#) : [2.10.0](#)、[2.09.2](#)、[2.07.4](#)
- [CVE-2020-14776](#) : [2.10.0](#)
- [CVE-2020-14775](#) : [2.09.2](#)、[2.07.4](#)
- [CVE-2020-14769](#) : [2.09.3](#)、[2.09.2](#)、[2.07.4](#)、[1.23.2](#)、[1.22.4](#)

- [CVE-2020-14765](#) : [2.09.2](#)、[2.07.4](#)、[1.23.2](#)、[1.22.4](#)
- [CVE-2020-14760](#) : [2.09.2](#)、[2.07.4](#)
- [CVE-2020-14672](#) : [2.09.2](#)、[2.07.4](#)、[1.23.2](#)、[1.22.4](#)
- [CVE-2020-14567](#) : [2.10.0](#)、[2.09.1](#)、[2.08.3](#)、[2.07.3](#)
- [CVE-2020-14559](#) : [2.10.0](#)、[2.09.1](#)、[2.08.3](#)、[2.07.3](#)、[1.23.1](#)、[1.22.3](#)
- [CVE-2020-14553](#) : [2.10.0](#)、[2.09.1](#)、[2.08.3](#)、[2.07.3](#)
- [CVE-2020-14547](#) : [2.10.0](#)、[2.09.1](#)、[2.08.3](#)、[2.07.3](#)
- [CVE-2020-14540](#) : [2.10.0](#)、[2.09.1](#)、[2.08.3](#)、[2.07.3](#)
- [CVE-2020-14539](#) : [2.10.0](#)、[1.23.1](#)、[1.22.3](#)
- [CVE-2020-11105](#) : [3.06.0](#)、[3.05.2](#)、[3.04.2](#)、[2.12.1](#)、[2.11.5](#)
- [CVE-2020-11104](#) : [3.06.0](#)、[3.05.2](#)、[3.04.2](#)、[2.12.1](#)、[2.11.5](#)
- [CVE-2020-2812](#) : [2.09.1](#)、[2.08.3](#)、[2.07.3](#)、[1.22.3](#)
- [CVE-2020-2806](#) : [2.09.1](#)、[2.08.3](#)、[2.07.3](#)
- [CVE-2020-2780](#) : [2.09.1](#)、[2.08.3](#)、[2.07.3](#)、[1.22.3](#)
- [CVE-2020-2765](#) : [2.09.1](#)、[2.08.3](#)、[2.07.3](#)
- [CVE-2020-2763](#) : [2.09.1](#)、[2.08.3](#)、[2.07.3](#)、[1.22.3](#)
- [CVE-2020-2760](#) : [2.09.1](#)、[2.08.3](#)、[2.07.3](#)、[2.04.9](#)
- [CVE-2020-2579](#) : [2.09.1](#)、[2.08.3](#)、[2.07.3](#)、[1.22.3](#)
- [CVE-2020-1971](#) : [2.09.2](#)、[2.07.4](#)、[1.23.2](#)、[1.22.4](#)
- [CVE-2019-17543](#) : [2.10.2](#)、[2.09.3](#)、[2.07.7](#)
- [CVE-2019-5443](#) : [2.08.0](#)、[2.04.9](#)
- [CVE-2019-3822](#) : [2.08.0](#)、[2.04.9](#)
- [CVE-2019-2960](#) : [2.10.2](#)、[2.09.3](#)、[2.07.7](#)
- [CVE-2019-2948](#) : [2.09.0](#)
- [CVE-2019-2924](#) : [2.07.0](#)、[2.04.9](#)、[1.22.0](#)
- [CVE-2019-2923](#) : [2.07.0](#)、[2.04.9](#)、[1.22.0](#)
- [CVE-2019-2922](#) : [2.07.0](#)、[2.04.9](#)、[1.22.0](#)
- [CVE-2019-2911](#) : [2.09.0](#)、[2.04.9](#)、[1.23.0](#)
- [CVE-2019-2910](#) : [2.07.0](#)、[2.04.9](#)、[1.22.0](#)
- [CVE-2019-2805](#) : [2.06.0](#)、[2.04.9](#)、[1.22.0](#)

- [CVE-2019-2791](#) : [2.06.0](#)、[2.04.9](#)
- [CVE-2019-2778](#) : [2.06.0](#)、[2.04.9](#)
- [CVE-2019-2758](#) : [2.06.0](#)、[2.04.9](#)
- [CVE-2019-2740](#) : [2.07.3](#)、[2.04.9](#)、[1.22.0](#)
- [CVE-2019-2739](#) : [2.06.0](#)、[2.04.9](#)
- [CVE-2019-2731](#) : [2.09.0](#)
- [CVE-2019-2730](#) : [2.06.0](#)、[2.04.9](#)、[1.22.0](#)
- [CVE-2019-2628](#) : [2.04.9](#)
- [CVE-2019-2581](#) : [2.09.0](#)
- [CVE-2019-2537](#) : [2.09.0](#)、[1.23.0](#)
- [CVE-2019-2534](#) : [2.05.0](#)、[2.04.3](#)、[1.21.0](#)、[1.20.0](#)、[1.19.1](#)
- [CVE-2019-2482](#) : [2.09.0](#)
- [CVE-2019-2434](#) : [2.09.0](#)
- [CVE-2019-2420](#) : [2.09.0](#)
- [CVE-2018-3284](#) : [2.09.0](#)
- [CVE-2018-3251](#) : [2.10.0](#)
- [CVE-2018-3156](#) : [2.10.0](#)
- [CVE-2018-3155](#) : [2.05.0](#)、[2.04.3](#)
- [CVE-2018-3143](#) : [2.10.0](#)、[1.23.2](#)
- [CVE-2018-3065](#) : [2.09.0](#)
- [CVE-2018-3064](#) : [2.06.0](#)、[2.04.9](#)、[1.22.0](#)
- [CVE-2018-3058](#) : [2.06.0](#)、[2.04.9](#)、[1.22.0](#)
- [CVE-2018-3056](#) : [2.05.0](#)、[2.04.4](#)
- [CVE-2018-2813](#) : [2.04.9](#)
- [CVE-2018-2787](#) : [2.09.0](#)、[1.23.0](#)
- [CVE-2018-2786](#) : [2.06.0](#)、[2.04.9](#)
- [CVE-2018-2784](#) : [2.09.0](#)、[1.23.0](#)
- [CVE-2018-2696](#) : [2.05.0](#)、[2.04.5](#)、[1.21.0](#)、[1.20.0](#)、[1.19.5](#)
- [CVE-2018-2645](#) : [2.09.0](#)、[1.23.0](#)
- [CVE-2018-2640](#) : [2.09.0](#)、[1.23.0](#)

- [CVE-2018-2612](#) : [2.05.0](#)、[2.04.3](#)、[1.21.0](#)、[1.20.0](#)、[1.19.1](#)
- [CVE-2018-2562](#) : [2.05.0](#)、[2.04.4](#)、[1.21.0](#)、[1.20.0](#)、[1.19.2](#)
- [CVE-2018-0734](#) : [2.05.0](#)、[2.04.3](#)、[1.21.0](#)、[1.20.0](#)、[1.19.1](#)
- [CVE-2017-3653](#) : [2.06.0](#)、[2.04.9](#)、[1.22.0](#)
- [CVE-2017-3599](#) : [2.05.0](#)、[2.04.3](#)、[1.21.0](#)、[1.20.0](#)、[1.19.1](#)
- [CVE-2017-3465](#) : [2.06.0](#)、[2.04.9](#)
- [CVE-2017-3464](#) : [1.22.0](#)、[2.04.9](#)
- [CVE-2017-3455](#) : [2.06.0](#)、[2.04.9](#)
- [CVE-2017-3329](#) : [2.05.0](#)、[2.04.4](#)、[1.21.0](#)、[1.20.0](#)、[1.19.2](#)
- [CVE-2017-3244](#) : [2.06.0](#)、[2.04.9](#)、[1.22.0](#)
- [CVE-2016-8287](#) : [2.07.2](#)
- [CVE-2016-5634](#) : [2.07.2](#)
- [CVE-2016-5612](#) : [2.06.0](#)、[2.04.9](#)、[1.22.0](#)
- [CVE-2016-5440](#) : [2.10.0](#)
- [CVE-2016-5439](#) : [1.22.0](#)、[2.03.3](#)
- [CVE-2016-5436](#) : [2.04.9](#)、[2.03.3](#)
- [CVE-2016-3518](#) : [2.04.5](#)
- [CVE-2016-3495](#) : [2.03.2](#)
- [CVE-2016-3486](#) : [2.02.2](#)
- [CVE-2016-0606](#) : [1.22.0](#)
- [CVE-2015-4904](#) : [1.22.0](#)
- [CVE-2015-4879](#) : [1.22.0](#)
- [CVE-2015-4864](#) : [1.22.0](#)
- [CVE-2015-4830](#) : [1.22.0](#)
- [CVE-2015-4826](#) : [1.22.0](#)
- [CVE-2015-4737](#) : [1.21.0](#)、[1.20.0](#)、[1.19.5](#)
- [CVE-2015-2620](#) : [1.22.0](#)
- [CVE-2015-0382](#) : [1.22.0](#)
- [CVE-2015-0381](#) : [1.22.0](#)
- [CVE-2014-6555](#) : [1.22.0](#)

- [CVE-2014-6489](#) : [1.22.0](#)
- [CVE-2014-4260](#) : [1.22.0](#)
- [CVE-2014-4258](#) : [1.22.0](#)
- [CVE-2014-2444](#) : [1.22.0](#)
- [CVE-2014-2436](#) : [1.22.0](#)
- [CVE-2014-0393](#) : [1.22.0](#)
- [CVE-2013-5908](#) : [1.22.0](#)
- [CVE-2013-5881](#) : [1.22.0](#)
- [CVE-2013-5807](#) : [1.22.0](#)
- [CVE-2013-3811](#) : [1.22.0](#)
- [CVE-2013-3807](#) : [1.22.0](#)
- [CVE-2013-3806](#) : [1.22.0](#)
- [CVE-2013-3804](#) : [1.22.0](#)
- [CVE-2013-2381](#) : [1.22.0](#)
- [CVE-2013-2378](#) : [1.22.0](#)
- [CVE-2013-2375](#) : [1.22.0](#)
- [CVE-2013-1523](#) : [1.22.0](#)
- [CVE-2012-5615](#) : [1.22.0](#)

Aurora MySQL 发布说明的文档历史记录

下表介绍了 Aurora MySQL 发布说明的文档版本。

变更	说明	日期
Aurora MySQL 版本 2.11.5 , 兼容 MySQL 5.7.12	Aurora MySQL 2.11.5 版本已上市。此版本与 MySQL 5.7.12 兼容。有关完整的详细信息，请参阅 与 MySQL 5.7 兼容的 Aurora MySQL 版本 2 。	2024年3月26日
Aurora MySQL 版本 2.12.2 , 兼容 MySQL 5.7.44	Aurora MySQL 2.12.2 版本已上市。此版本与 MySQL 5.7.44 兼容。有关完整的详细信息，请参阅 与 MySQL 5.7 兼容的 Aurora MySQL 版本 2 。	2024年3月19日
Aurora MySQL 版本 3.04.2 , 兼容 MySQL 8.0.28	Aurora MySQL 3.04.2 版本已上市。此版本与 MySQL 8.0.28 兼容。有关完整的详细信息，请参阅 与 MySQL 8.0 兼容的 Aurora MySQL 版本 3 。	2024年3月15日
Aurora MySQL 版本 3.06.0 , 兼容 MySQL 8.0.34	Aurora MySQL 3.06.0 版本已上市。此版本与 MySQL 8.0.34 兼容。有关完整的详细信息，请参阅 与 MySQL 8.0 兼容的 Aurora MySQL 版本 3 。	2024年3月7日
Aurora MySQL 版本 3.05.2 , 兼容 MySQL 8.0.32	Aurora MySQL 3.05.2 版本已上市。此版本与 MySQL 8.0.32 兼容。有关完整的详细信息，请参阅 与 MySQL 8.0 兼容的 Aurora MySQL 版本 3 。	2024年1月31日
Aurora MySQL 版本 2.12.1 , 与 MySQL 5.7.40 兼容	Aurora MySQL 版本 2.12.1 已发布。此版本与 MySQL 5.7.40	2023年12月28日

	兼容。有关完整的详细信息，请参阅 与 MySQL 5.7 兼容的 Aurora MySQL 版本 2 。	
Aurora MySQL 版本 3.03.3，与 MySQL 8.0.26 兼容	Aurora MySQL 版本 3.03.3 已发布。此版本与 MySQL 8.0.26 兼容。有关完整的详细信息，请参阅 与 MySQL 8.0 兼容的 Aurora MySQL 版本 3 。	2023 年 12 月 8 日
Aurora MySQL 版本 3.05.1，与 MySQL 8.0.32 兼容	Aurora MySQL 版本 3.05.1 已发布。此版本与 MySQL 8.0.32 兼容。有关完整的详细信息，请参阅 与 MySQL 8.0 兼容的 Aurora MySQL 版本 3 。	2023 年 11 月 21 日
Aurora MySQL 版本 3.04.1，与 MySQL 8.0.28 兼容	Aurora MySQL 版本 3.04.1 已发布。此版本与 MySQL 8.0.28 兼容。有关完整的详细信息，请参阅 与 MySQL 8.0 兼容的 Aurora MySQL 版本 3 。	2023 年 11 月 13 日
Aurora MySQL 版本 3.05.0.1，与 MySQL 8.0.32 兼容，测试版	Aurora MySQL 版本 3.05.0.1 已发布。此版本与 MySQL 8.0.32 兼容。有关完整的详细信息，请参阅 与 MySQL 8.0 兼容的 Aurora MySQL 版本 3 。	2023 年 10 月 30 日
Aurora MySQL 版本 3.05.0，与 MySQL 8.0.32 兼容	Aurora MySQL 版本 3.05.0 已发布。此版本与 MySQL 8.0.32 兼容。有关完整的详细信息，请参阅 与 MySQL 8.0 兼容的 Aurora MySQL 版本 3 。	2023 年 10 月 25 日

Aurora MySQL 版本 2.12.0.1，与 MySQL 5.7.40 兼容，测试版	Aurora MySQL 版本 2.12.0.1 测试版已发布。此版本与 MySQL 5.7.40 兼容。有关完整的详细信息，请参阅 与 MySQL 5.7 兼容的 Aurora MySQL 版本 2 。	2023 年 10 月 25 日
Aurora MySQL 版本 2.11.4，与 MySQL 5.7.12 兼容	Aurora MySQL 版本 2.11.4 已发布。此版本与 MySQL 5.7.12 兼容。有关完整的详细信息，请参阅 与 MySQL 5.7 兼容的 Aurora MySQL 版本 2 。	2023 年 10 月 17 日
Aurora MySQL 版本 3.03.2，与 MySQL 8.0.26 兼容	Aurora MySQL 版本 3.03.2 已发布。此版本与 MySQL 8.0.26 兼容。有关完整的详细信息，请参阅 与 MySQL 8.0 兼容的 Aurora MySQL 版本 3 。	2023 年 8 月 29 日
Aurora MySQL 版本 2.07.10，与 MySQL 5.7.12 兼容	Aurora MySQL 版本 2.07.10 已发布。此版本与 MySQL 5.7.12 兼容。有关完整的详细信息，请参阅 与 MySQL 5.7 兼容的 Aurora MySQL 版本 2 。	2023 年 8 月 15 日
Aurora MySQL 版本 3.04.0，与 MySQL 8.0.28 兼容	Aurora MySQL 版本 3.04.0 已发布。此版本与 MySQL 8.0.28 兼容。有关完整的详细信息，请参阅 与 MySQL 8.0 兼容的 Aurora MySQL 版本 3 。	2023 年 7 月 31 日
Aurora MySQL 版本 2.12.0，与 MySQL 5.7.40 兼容	Aurora MySQL 版本 2.12.0 已发布。此版本与 MySQL 5.7.40 兼容。有关完整的详细信息，请参阅 与 MySQL 5.7 兼容的 Aurora MySQL 版本 2 。	2023 年 7 月 25 日

Aurora MySQL 版本 2.11.3 , 与 MySQL 5.7.12 兼容	Aurora MySQL 版本 2.11.3 已发布。此版本与 MySQL 5.7.12 兼容。有关完整的详细信息，请参阅 与 MySQL 5.7 兼容的 Aurora MySQL 版本 2 。	2023 年 6 月 9 日
Aurora MySQL 版本 3.03.1 , 与 MySQL 8.0.26 兼容	Aurora MySQL 版本 3.03.1 已发布。此版本与 MySQL 8.0.26 兼容。有关完整的详细信息，请参阅 与 MySQL 8.0 兼容的 Aurora MySQL 版本 3 。	2023 年 5 月 11 日
Aurora MySQL 版本 2.07.9 , 与 MySQL 5.7.12 兼容。	Aurora MySQL 版本 2.07.9 已发布。此版本与 MySQL 5.7.12 兼容。有关完整的详细信息，请参阅 与 MySQL 5.7 兼容的 Aurora MySQL 版本 2 。	2023 年 5 月 4 日
Aurora MySQL 版本 3.02.3 , 与 MySQL 8.0.23 兼容	Aurora MySQL 版本 3.02.3 已发布。此版本与 MySQL 8.0.23 兼容。有关完整的详细信息，请参阅 与 MySQL 8.0 兼容的 Aurora MySQL 版本 3 。	2023 年 4 月 17 日
Aurora MySQL 版本 2.11.2 , 与 MySQL 5.7.12 兼容	Aurora MySQL 版本 2.11.2 已发布。此版本与 MySQL 5.7.12 兼容。有关完整的详细信息，请参阅 与 MySQL 5.7 兼容的 Aurora MySQL 版本 2 。	2023 年 3 月 24 日
Aurora MySQL 版本 3.03.0 , 与 MySQL 8.0.26 兼容	Aurora MySQL 版本 3.03.0 已发布。此版本与 MySQL 8.0.26 兼容。有关完整的详细信息，请参阅 与 MySQL 8.0 兼容的 Aurora MySQL 版本 3 。	2023 年 3 月 1 日

[Aurora MySQL 版本 2.11.1 ,
与 MySQL 5.7.12 兼容。](#)

Aurora MySQL 版本 2.11.1 已发布。此版本与 MySQL 5.7.12 兼容。有关完整的详细信息，请参阅[与 MySQL 5.7 兼容的 Aurora MySQL 版本 2。](#)

2023 年 2 月 14 日

[Aurora MySQL 版本 3.02.2 ,
与 MySQL 8.0.23 兼容](#)

Aurora MySQL 版本 3.02.2 已发布。此版本与 MySQL 8.0.23 兼容。有关完整的详细信息，请参阅[与 MySQL 8.0 兼容的 Aurora MySQL 版本 3。](#)

2022 年 11 月 18 日

[Aurora MySQL 版本 2.10.3 ,
与 MySQL 5.7 兼容](#)

Aurora MySQL 版本 2.10.3 已发布。此版本与 MySQL 5.7 兼容。有关完整的详细信息，请参阅[与 MySQL 5.7 兼容的 Aurora MySQL 版本 2。](#)

2022 年 11 月 1 日

[Aurora MySQL 版本 2.11.0 ,
与 MySQL 5.7.12 兼容](#)

Aurora MySQL 版本 2.11.0 已发布。此版本与 MySQL 5.7.12 兼容。有关完整的详细信息，请参阅[与 MySQL 5.7 兼容的 Aurora MySQL 版本 2。](#)

2022 年 10 月 25 日

[Aurora MySQL 版本 3.02.1 ,
与 MySQL 8.0.23 兼容](#)

Aurora MySQL 版本 3.02.1 已发布。此版本与 MySQL 8.0.23 兼容。有关完整的详细信息，请参阅[与 MySQL 8.0 兼容的 Aurora MySQL 版本 3。](#)

2022 年 9 月 7 日

[与 MySQL 5.6 兼容的 Aurora Serverless v1 的就地升级](#)

您可以对与 MySQL 5.6 兼容的 Aurora Serverless v1 集群执行就地升级，以将现有集群更改为与 MySQL 5.7 兼容的 Aurora Serverless v1 集群。有关更多信息，请参阅 [Aurora MySQL Serverless 5.7 引擎更新 2020-06-18 \(版本 2.07.1\)](#)。

2022 年 6 月 16 日

[与 MySQL 5.6 兼容的 Aurora Serverless v1 的就地升级](#)

您可以对与 MySQL 5.6 兼容的 Aurora Serverless v1 集群执行就地升级，以将现有集群更改为与 MySQL 5.7 兼容的 Aurora Serverless v1 集群。有关更多信息，请参阅 [Aurora MySQL Serverless 5.7 引擎更新 2020-06-18 \(版本 2.07.1\)](#)。

2022 年 6 月 16 日

[Aurora MySQL 数据库引擎更新 2022-06-16 \(版本 2.07.8\) 已发布。](#)

Aurora MySQL 版本 2.07.8 已发布。

2022 年 6 月 16 日

[Aurora MySQL 版本 3.02.0 , 与 MySQL 8.0.23 兼容](#)

Aurora MySQL 版本 3.02.0 已发布。此版本与 MySQL 8.0.23 兼容。有关完整的详细信息，请参阅 [与 MySQL 8.0 兼容的 Aurora MySQL 版本 3。](#)

2022 年 4 月 20 日

[Aurora MySQL 版本 3.01.1 , 与 MySQL 8.0.23 兼容](#)

Aurora MySQL 版本 3.01.1 已发布。此版本与 MySQL 8.0.23 兼容。有关完整的详细信息，请参阅 [与 MySQL 8.0 兼容的 Aurora MySQL 版本 3。](#)

2022 年 4 月 15 日

初始版本	Aurora MySQL 发布说明的初始版本。	2022 年 3 月 22 日
Aurora MySQL 版本 2.10.2	Aurora MySQL 版本 2.10.2 已发布。	2022 年 1 月 26 日
Aurora MySQL 版本 2.08.4	Aurora MySQL 版本 2.08.4 已发布。	2022 年 1 月 6 日
Aurora MySQL 版本 2.07.7	Aurora MySQL 版本 2.07.7 已发布。	2021 年 11 月 24 日
Aurora MySQL 版本 3.01.0 , 与 MySQL 8.0.23 兼容	Aurora MySQL 版本 3.01.0 已发布。此版本与 MySQL 8.0.23 兼容。有关完整的详细信息，请参阅 与 MySQL 8.0 兼容的 Aurora MySQL 版本 3 。	2021 年 11 月 18 日
Aurora MySQL 版本 2.09.3	Aurora MySQL 版本 2.09.3 已发布。	2021 年 11 月 12 日
Aurora MySQL 版本 2.10.1	Aurora MySQL 版本 2.10.1 已发布。	2021 年 10 月 21 日
Aurora MySQL 版本 1.23.4	Aurora MySQL 版本 1.23.4 已发布。	2021 年 9 月 30 日
Aurora MySQL 版本 2.07.6	Aurora MySQL 版本 2.07.6 已发布。	2021 年 9 月 2 日
Aurora MySQL 版本 2.07.5	Aurora MySQL 版本 2.07.5 已发布。	2021 年 7 月 6 日
Aurora MySQL 版本 1.23.3	Aurora MySQL 版本 1.23.3 已发布。	2021 年 6 月 28 日
Aurora MySQL 版本 1.22.5	Aurora MySQL 版本 1.22.5 已发布。	2021 年 6 月 3 日

Aurora MySQL 版本 2.10.0	Aurora MySQL 版本 2.10.0 已发布。其中一些要点包括 提高了写入器重启期间读取器实例的可用性 、 改进了零停机时间修补 (ZDP) 、 改进了零停机时间重启 (ZDR) 以及 优化了二进制日志 I/O 缓存 。	2021 年 5 月 25 日
Aurora MySQL 版本 1.23.2	Aurora MySQL 版本 1.23.2 已发布。	2021 年 3 月 18 日
Aurora MySQL 版本 2.07.4	Aurora MySQL 版本 2.07.4 已发布。	2021 年 3 月 4 日
Aurora MySQL 版本 1.22.4	Aurora MySQL 版本 1.22.4 已发布。	2021 年 3 月 4 日
Aurora MySQL 版本 2.09.2	Aurora MySQL 版本 2.09.2 已发布。	2021 年 2 月 26 日
Aurora MySQL 版本 2.09.1	Aurora MySQL 版本 2.09.1 已发布。	2020 年 12 月 11 日
Aurora MySQL 版本 1.23.1	Aurora MySQL 版本 1.23.1 已发布。	2020 年 11 月 24 日
Aurora MySQL 版本 2.08.3	Aurora MySQL 版本 2.08.3 已发布。	2020 年 11 月 12 日
Aurora MySQL 版本 2.07.3	Aurora MySQL 版本 2.07.3 已发布。	2020 年 11 月 10 日
Aurora MySQL 版本 1.22.3	Aurora MySQL 版本 1.22.3 已发布。	2020 年 11 月 9 日
Aurora MySQL 版本 2.09.0	Aurora MySQL 版本 2.09.0 已发布。	2020 年 9 月 17 日

Aurora MySQL 版本 1.23.0	Aurora MySQL 版本 1.23.0 已发布。	2020 年 9 月 2 日
Aurora MySQL 版本 2.08.2	Aurora MySQL 版本 2.08.2 已发布。	2020 年 8 月 28 日
Aurora MySQL 版本 2.04.9	Aurora MySQL 版本 2.04.9 已发布。	2020 年 8 月 14 日
Aurora MySQL 版本 2.08.1	Aurora MySQL 版本 2.08.1 已发布。	2020 年 6 月 18 日
适用于并行查询集群的 Aurora MySQL 版本 1.22.2	当您创建并行查询集群时，Aurora MySQL 版本 1.22.2 可用。	2020 年 6 月 18 日
适用于并行查询集群的 Aurora MySQL 版本 1.20.1	当您创建并行查询集群时，Aurora MySQL 版本 1.20.1 可用。	2020 年 6 月 11 日
Aurora MySQL 版本 2.08.0	Aurora MySQL 版本 2.08.0 已发布。	2020 年 6 月 2 日
适用于并行查询集群的 Aurora MySQL 版本 1.19.6	当您创建并行查询集群时，Aurora MySQL 版本 1.19.6 可用。	2020 年 6 月 2 日
Aurora MySQL 版本 2.07.2	Aurora MySQL 版本 2.07.2 已发布。	2020 年 4 月 17 日
Aurora MySQL 版本 1.22.2	Aurora MySQL 版本 1.22.2 已发布。	2020 年 3 月 5 日
Aurora MySQL 版本 1.20.1	Aurora MySQL 版本 1.20.1 已发布。	2020 年 3 月 5 日
Aurora MySQL 版本 1.19.6	Aurora MySQL 版本 1.19.6 已发布。	2020 年 3 月 5 日

Aurora MySQL 版本 1.17.9	Aurora MySQL 版本 1.17.9 已发布。	2020 年 3 月 5 日
Aurora MySQL 版本 2.07.1	Aurora MySQL 版本 2.07.1 已发布。	2019 年 12 月 23 日
Aurora MySQL 版本 1.22.1	Aurora MySQL 版本 1.22.1 已发布。	2019 年 12 月 23 日
Aurora MySQL 版本 2.07.0	Aurora MySQL 版本 2.07.0 已发布。	2019 年 11 月 25 日
Aurora MySQL 版本 1.22.0	Aurora MySQL 版本 1.22.0 已发布。	2019 年 11 月 25 日
Aurora MySQL 版本 1.21.0	Aurora MySQL 版本 1.21.0 已发布。	2019 年 11 月 25 日
Aurora MySQL 版本 2.06.0	Aurora MySQL 版本 2.06.0 已发布。	2019 年 11 月 22 日
Aurora MySQL 版本 2.04.8	Aurora MySQL 版本 2.04.8 已发布。	2019 年 11 月 20 日
Aurora MySQL 版本 2.04.7	Aurora MySQL 版本 2.04.7 已发布。	2019 年 11 月 14 日
Aurora MySQL 版本 2.05.0	Aurora MySQL 版本 2.05.0 已发布。	2019 年 11 月 11 日
Aurora MySQL 版本 1.20.0	Aurora MySQL 版本 1.20.0 已发布。	2019 年 11 月 11 日
Aurora MySQL 版本 2.04.6	Aurora MySQL 版本 2.04.6 已发布。	2019 年 9 月 19 日
Aurora MySQL 版本 1.19.5	Aurora MySQL 版本 1.19.5 已发布。	2019 年 9 月 19 日

Aurora MySQL 版本 2.04.5	Aurora MySQL 版本 2.04.5 已发布。	2019 年 7 月 8 日
Aurora MySQL 版本 1.19.2	Aurora MySQL 版本 1.19.2 已发布。	2019 年 6 月 5 日
Aurora MySQL 版本 2.04.4	Aurora MySQL 版本 2.04.4 已发布。	2019 年 5 月 29 日
Aurora MySQL 版本 2.04.3	Aurora MySQL 版本 2.04.3 已发布。	2019 年 5 月 9 日
Aurora MySQL 版本 1.19.1	Aurora MySQL 版本 1.19.1 已发布。	2019 年 5 月 9 日
Aurora MySQL 版本 2.04.2	Aurora MySQL 版本 2.04.2 已发布。	2019 年 5 月 2 日
Aurora MySQL 版本 2.04.1	Aurora MySQL 版本 2.04.1 已发布。	2019 年 3 月 25 日
Aurora MySQL 版本 2.04	Aurora MySQL 版本 2.04 已发布。	2019 年 3 月 25 日
Aurora MySQL 版本 2.03.4	Aurora MySQL 版本 2.03.4 已发布。	2019 年 2 月 7 日
Aurora MySQL 版本 1.19.0	Aurora MySQL 版本 1.19.0 已发布。	2019 年 2 月 7 日
Aurora MySQL 版本 2.03.3	Aurora MySQL 版本 2.03.3 已发布。	2019 年 1 月 18 日
Aurora MySQL 版本 1.17.8	Aurora MySQL 版本 1.17.8 已发布。	2019 年 1 月 17 日
Aurora MySQL 版本 2.03.2	Aurora MySQL 版本 2.03.2 已发布。	2019 年 1 月 9 日

Aurora MySQL 版本 2.03.1	Aurora MySQL 版本 2.03.1 已发布。	2018 年 10 月 24 日
Aurora MySQL 版本 2.03	Aurora MySQL 版本 2.03 已发布。	2018 年 10 月 11 日
Aurora MySQL 版本 2.02.5	Aurora MySQL 版本 2.02.5 已发布。	2018 年 10 月 8 日
Aurora MySQL 版本 1.17.7	Aurora MySQL 版本 1.17.7 已发布。	2018 年 10 月 8 日
Aurora MySQL 版本 2.02.4	Aurora MySQL 版本 2.02.4 已发布。	2018 年 9 月 21 日
Aurora MySQL 版本 1.18.0	Aurora MySQL 版本 1.18.0 已发布。	2018 年 9 月 20 日
Aurora MySQL 版本 1.17.6	Aurora MySQL 版本 1.17.6 已发布。	2018 年 9 月 6 日

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。