aws

Aurora MySQL のリリースノート

Amazon Aurora

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

Amazon Aurora: Aurora MySQL のリリースノート

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

Amazon の商標およびトレードドレスは、お客様に混乱を招く可能性がある態様、または Amazon の信用を傷つけたり、失わせたりする態様において、Amazon のものではない製品またはサービスに関連して使用してはなりません。Amazon が所有しない他の商標はすべてそれぞれの所有者に帰属します。所有者は必ずしも Amazon との提携や関連があるわけではありません。また、Amazon の支援を受けているとはかぎりません。

Table of Contents

Aurora MySQL リリースノート 1
Aurora MySQL バージョン 3 2
Aurora MySQL アップデート:2024-03-07 (バージョン 3.06.0、MySQL 8.0.34 に対応) 3
新機能 4
改良点 5
MySQL Community Edition でのバグ修正の統合 8
Aurora MySQL アップデート:2024-01-31 (バージョン 3.05.2、MySQL 8.0.32 に対応) 8
改良点 9
MySQL Community Edition でのバグ修正の統合 11
Aurora MySQL アップデート:2023-11-21 (バージョン 3.05.1、MySQL 8.0.32 に対応) 11
改良点 11
MySQL Community Edition でのバグ修正の統合 12
Aurora MySQL の更新: 2023-10-30 (バージョン 3.05.0.1、MySQL 8.0.32 互換) ベータ 12
改良点 13
Aurora MySQL の更新: 2023-10-25 (バージョン 3.05.0、MySQL 8.0.32 互換) 13
改良点 14
MySQL Community Edition でのバグ修正の統合 18
Aurora MySQL アップデート:2024-03-15 (バージョン 3.04.2、MySQL 8.0.28 に対応) 18
改良点 19
MySQL Community Edition でのバグ修正の統合 21
Aurora MySQL アップデート:2023-11-13 (バージョン 3.04.1、MySQL 8.0.28 に対応) デフォルト 21
改良点 23
MySQL Community Edition でのバグ修正の統合 24
Aurora MySQL アップデート:2023-07-31 (バージョン 3.04.0、MySQL 8.0.28 に対応) 24
改良点 26
MySQL Community Edition でのバグ修正の統合 31
Aurora MySQL の更新: 2023-12-08 (バージョン 3.03.3、MySQL 8.0.26 互換) 32
改良点 33
MySQL Community Edition でのバグ修正の統合 34
Aurora MySQL の更新: 2023-08-29 (バージョン 3.03.2、MySQL 8.0.26 互換) 34
改良点 35
MySQL Community Edition でのバグ修正の統合 37
Aurora MySQL の更新: 2023-05-11 (バージョン 3.03.1、MySQL 8.0.26 互換) 37

改良点 38
MySQL Community Edition でのバグ修正の統合 40
Aurora MySQL の更新: 2023-03-01 (バージョン 3.03.0、MySQL 8.0.26 互換) このバージョンへのアップグレードはサポートされていません。 40
改良点 41
MySQL Community Edition でのバグ修正の統合 44
Aurora MySQL の更新: 2023-04-17 (バージョン 3.02.3、MySQL 8.0.23 互換) 標準サポートは 2024 年 1 月 15 日に終了。 45
改良点 45
Aurora MySQL の更新: 2022-11-18 (バージョン 3.02.2、MySQL 8.0.23 互換) 標準サポートは 2024 年 1 月 15 日に終了 46
改良点 47
MySQL Community Edition でのバグ修正の統合 44
Aurora MySQL の更新: 2022-09-07 (バージョン 3.02.1、MySQL 8.0.23 互換) 標準サポートは 2024 年 1 月 15 日に終了。このバージョンへのアップグレードはサポートされていません。 50
改良点 51
Aurora MySQL の更新: 2022-04-20 (バージョン 3.02.0、MySQL 8.0.23 互換) 標準サポートは 2024 年 1 月 15 日に終了。このバージョンへのアップグレードはサポートされていません。 52
改良点 53
MySQL Community Edition バグ修正の統合 55
Aurora MySQL の更新: 2022-04-15 (バージョン 3.01.1、MySQL 8.0.23 互換) 標準サポートは 2024 年 1 月 15 日に終了。このバージョンへのアップグレードはサポートされていません。 55
改良点 56
MySQL Community Edition バグ修正の統合 58
Aurora MySQL の更新: 2021-11-18 (バージョン 3.01.0、MySQL 8.0.23 互換) 標準サポートは 2024 年 1 月 15 日に終了。このバージョンへのアップグレードはサポートされていません。 58
改良点 59
Aurora MySQL バージョン 2 60
Aurora MySQL アップデート:2024-03-19 (バージョン 2.12.2、MySQL 5.7.44 に対応) 62
改良点 63
MySQL Community Edition でのバグ修正の統合 64
Aurora MySQL バージョン 2 ではサポートされていない機能 64

MySQL 5.7 の互換性 64
Aurora MySQL の更新: 2023-12-28 (バージョン 2.12.1、MySQL 5.7.40 互換) 65
改良点 66
MySQL Community Edition でのバグ修正の統合 68
Aurora MySQL バージョン 2 ではサポートされていない機能 68
MySQL 5.7 の互換性 69
Aurora MySQL の更新: 2023-10-25 (バージョン 2.12.0.1、MySQL 5.7.40 互換) ベータ 69
改良点 70
Aurora MySQL の更新: 2023-07-25 (バージョン 2.12.0、MySQL 5.7.40 互換) 70
改良点 71
MySQL Community Edition でのバグ修正の統合 74
Aurora MySQL バージョン 2 ではサポートされていない機能 74
MySQL 5.7 の互換性 74
Aurora MySQL アップデート:2024-03-26 (バージョン 2.11.5、MySQL 5.7.12 に対応) 75
改良点 76
Aurora MySQL バージョン 2 ではサポートされていない機能 78
MySQL 5.7 の互換性 78
Aurora MySQL の更新: 2023-10-17 (バージョン 2.11.4、MySQL 5.7.12 互換) 78
改良点 80
MySQL Community Edition でのバグ修正の統合 82
Aurora MySQL バージョン 2 ではサポートされていない機能 82
MySQL 5.7 の互換性 82
Aurora MySQL の更新: 2023-06-09 (バージョン 2.11.3、MySQL 5.7.12 互換) 83
改良点 84
Aurora MySQL バージョン 2 ではサポートされていない機能 85
MySQL 5.7 の互換性 85
Aurora MySQL の更新: 2023-03-24 (バージョン 2.11.2、MySQL 5.7.12 互換) 86
改良点 87
Aurora MySQL バージョン 2 ではサポートされていない機能 87
MySQL 5.7 の互換性 88
Aurora MySQL の更新: 2023-02-14 (バージョン 2.11.1、MySQL 5.7.12 互換) 88
改良点 89
Aurora MySQL バージョン 1 との比較 90
MySQL 5.7 の互換性 91
Aurora MySQL の更新: 2022-10-25 (バージョン 2.11.0、MySQL 5.7.12 互換) このバージョンは新規作成には使用不可。 91

改良点 92
MySQL Community Edition でのバグ修正の統合 96
Aurora MySQL バージョン 1 との比較 98
MySQL 5.7 の互換性 98
Aurora MySQL の更新: 2022-11-01 (バージョン 2.10.3) (廃止) 99
改良点 99
MySQL Community Edition でのバグ修正の統合 100
Aurora MySQL バージョン 1 との比較 101
MySQL 5.7 の互換性 101
Aurora MySQL の更新: 2022-01-26 (バージョン 2.10.2) (廃止) 102
改良点 103
MySQL Community Edition でのバグ修正の統合 44
Aurora MySQL バージョン 1 との比較 106
MySQL 5.7 の互換性 107
Aurora MySQL の更新: 2021-10-21 (バージョン 2.10.1) (廃止) 107
改良点 108
MySQL Community Edition バグ修正の統合 109
Aurora MySQL バージョン 1 との比較 109
MySQL 5.7 の互換性 110
Aurora MySQL の更新: 2021-05-25 (バージョン 2.10.0) (廃止) 110
改良点 111
MySQL Community Edition バグ修正の統合 114
Aurora MySQL バージョン 1 との比較 118
MySQL 5.7 の互換性 118
Aurora MySQL の更新: 2021-11-12 (バージョン 2.09.3) (廃止) 119
改良点 119
MySQL Community Edition バグ修正の統合 122
Aurora MySQL バージョン 1 との比較 122
MySQL 5.7 の互換性 123
Aurora MySQL の更新: 2021-02-26 (バージョン 2.09.2) (廃止) 123
改良点 124
Aurora MySQL バージョン 1 との比較 125
MySQL 5.7 の互換性 126
Aurora MySQL の更新: 2020-12-11 (バージョン 2.09.1) (廃止) 127
改良点 127
MySQL Community Edition バグ修正の統合 129

Aurora MySQL バージョン 1 との比較 129
MySQL 5.7 の互換性 129
Aurora MySQL 更新: 2020-09-17 (バージョン 2.09.0) (廃止) 130
改良点 131
MySQL Community Edition バグ修正の統合 136
Aurora MySQL バージョン 1 との比較 137
MySQL 5.7 の互換性 138
Aurora MySQL の更新: 2022-01-06 (バージョン 2.08.4) (廃止) 138
改良点 139
Aurora MySQL バージョン 1 との比較 139
MySQL 5.7 の互換性 140
Aurora MySQL の更新: 2020-11-12 (バージョン 2.08.3) (廃止) 141
改良点 141
MySQL Community Edition バグ修正の統合 142
Aurora MySQL バージョン 1 との比較 142
MySQL 5.7 の互換性 143
Aurora MySQL の更新: 2020-08-28 (バージョン 2.08.2) (廃止) 144
改良点 144
Aurora MySQL バージョン 1 との比較 145
MySQL 5.7 の互換性 145
Aurora MySQL の更新: 2020-06-18 (バージョン 2.08.1) (廃止) 146
改良点 147
Aurora MySQL バージョン 1 との比較 147
MySQL 5.7 の互換性 148
Aurora MySQL の更新: 2020-06-02 (バージョン 2.08.0) (廃止) 148
改良点 149
MySQL Community Edition バグ修正の統合 152
Aurora MySQL バージョン 1 との比較 152
MySQL 5.7 の互換性 153
Aurora MySQL の更新: 2023-08-15 (バージョン 2.07.10、MySQL 5.7.12 互換) 153
改善点 155
Aurora MySQL バージョン 2 ではサポートされていない機能 155
MySQL 5.7 の互換性 156
Aurora MySQL の更新: 2023-05-04 (バージョン 2.07.9、MySQL 5.7.12 互換) 156
改善点 157
Aurora MySQL バージョン 2 ではサポートされていない機能 158

MySQL 5.7 の互換性 158
Aurora MySQL の更新: 2022-06-16 (バージョン 2.07.8) (廃止) 159
改良点 160
MySQL Community Edition バグ修正の統合 160
Aurora MySQL バージョン 1 との比較 160
MySQL 5.7 の互換性 161
Aurora MySQL の更新: 2021-11-24 (バージョン 2.07.7) (廃止) 162
改良点 162
Aurora MySQL バージョン 1 との比較 163
MySQL 5.7 の互換性 164
Aurora MySQL の更新: 2021-09-02 (バージョン 2.07.6) (廃止) 165
MySQL Community Edition バグ修正の統合 165
Aurora MySQL バージョン 1 との比較 165
MySQL 5.7 の互換性 166
Aurora MySQL の更新: 2021-07-06 (バージョン 2.07.5) (廃止) 166
改良点 167
Aurora MySQL バージョン 1 との比較 167
MySQL 5.7 の互換性 168
Aurora MySQL の更新: 2021-03-04 (バージョン 2.07.4) (廃止) 169
改良点 169
MySQL Community Edition バグ修正の統合 170
Aurora MySQL バージョン 1 との比較 170
MySQL 5.7 の互換性 171
Aurora MySQL の更新: 2020-11-10 (バージョン 2.07.3) (廃止) 172
改良点 172
MySQL Community Edition バグ修正の統合 174
Aurora MySQL バージョン 1 との比較 175
MySQL 5.7 の互換性 175
Aurora MySQL の更新: 2020-04-17 (バージョン 2.07.2) (廃止) 176
改良点 177
MySQL Community Edition バグ修正の統合 178
Aurora MySQL バージョン 1 との比較 178
MySQL 5.7 の互換性 178
Aurora MySQL の更新: 2019-12-23 (バージョン 2.07.1) (廃止) 179
改良点 180
Aurora MySQL バージョン 1 との比較 180

MySQL 5.7 の互換性 181
Aurora MySQL の更新: 2019-11-25 (バージョン 2.07.0) (廃止) 181
改良点 182
MySQL Community Edition バグ修正の統合 184
Aurora MySQL バージョン 1 との比較 184
MySQL 5.7 の互換性 184
Aurora MySQL の更新: 2019-11-22 (バージョン 2.06.0) (廃止) 185
改良点 186
Aurora MySQL バージョン 1 との比較 188
MySQL 5.7 の互換性 189
Aurora MySQL の更新: 2019-11-11 (バージョン 2.05.0) (廃止) 189
改良点 190
MySQL バグ修正の統合 191
Aurora MySQL バージョン 1 との比較 191
MySQL 5.7 の互換性 192
Aurora MySQL の更新: 2020-08-14 (バージョン 2.04.9) (廃止) 193
改良点 193
MySQL バグ修正の統合 196
Aurora MySQL バージョン 1 との比較 197
MySQL 5.7 の互換性 197
Aurora MySQL の更新: 2019-11-20 (バージョン 2.04.8) (廃止) 198
改良点 199
Aurora MySQL バージョン 1 との比較 200
MySQL 5.7 の互換性 200
Aurora MySQL の更新: 2019-11-14 (バージョン 2.04.7) (廃止) 201
改良点 202
Aurora MySQL バージョン 1 との比較 202
MySQL 5.7 の互換性 203
Aurora MySQL の更新: 2019-09-19 (バージョン2.04.6) (廃止) 204
改良点 205
MySQL バグ修正の統合 205
Aurora MySQL バージョン 1 との比較 205
MySQL 5.7 の互換性 206
Aurora MySQL の更新: 2019-07-08 (バージョン 2.04.5) (廃止) 206
改良点 207
Aurora MySQL バージョン 1 との比較 208

MySQL 5.7 の互換性 208
Aurora MySQL の更新: 2019-05-29 (バージョン 2.04.4) (廃止) 209
改良点 210
Aurora MySQL バージョン 1 との比較 210
MySQL 5.7 の互換性 211
Aurora MySQL の更新: 2019-05-09 (バージョン 2.04.3) (廃止) 211
改良点 212
Aurora MySQL バージョン 1 との比較 213
MySQL 5.7 の互換性 213
Aurora MySQL の更新: 2019-05-02 (バージョン 2.04.2) (廃止) 214
改良点 215
MySQL バグ修正の統合 215
Aurora MySQL バージョン 1 との比較 215
MySQL 5.7 の互換性 216
Aurora MySQL 更新: 2019-03-25 (バージョン 2.04.1) (廃止) 216
改良点 217
Aurora MySQL バージョン 1 との比較 218
MySQL 5.7 の互換性 218
Aurora MySQL 更新: 2019-03-25 (バージョン 2.04.0) (廃止) 219
改良点 220
MySQL バグ修正の統合 220
Aurora MySQL バージョン 1 との比較 220
MySQL 5.7 の互換性 221
Aurora MySQL の更新: 2019-02-07 (バージョン 2.03.4) (廃止) 221
改良点 222
Aurora MySQL バージョン 1 との比較 222
MySQL 5.7 の互換性 223
Aurora MySQL の更新: 2019-01-18 (バージョン 2.03.3) (廃止) 224
改良点 224
MySQL バグ修正の統合 225
Aurora MySQL バージョン 1 との比較 226
MySQL 5.7 の互換性 226
Aurora MySQL の更新: 2019-01-09 (バージョン 2.03.2) (廃止) 227
改良点 228
Aurora MySQL バージョン 1 との比較 228
MySQL 5.7 の互換性 229

Aurora MySQL の更新: 2018-10-24 (バージョン 2.03.1) (廃止) 229
改良点 230
Aurora MySQL バージョン 1 との比較 230
MySQL 5.7 の互換性 231
Aurora MySQL の更新: 2018-10-11 (バージョン 2.03) (廃止) 231
改良点 232
MySQL Community Edition バグ修正の統合 233
Aurora MySQL バージョン 1 との比較 233
MySQL 5.7 の互換性 233
Aurora MySQL の更新: 2018-10-08 (バージョン 2.02.5) (廃止) 234
改良点 235
Aurora MySQL バージョン 1 との比較 235
MySQL 5.7 の互換性 236
Aurora MySQL の更新: 2018-09-21 (バージョン 2.02.4) (廃止) 236
改良点 237
MySQL Community Edition バグ修正の統合 237
Aurora MySQL バージョン 1 との比較 238
MySQL 5.7 の互換性 238
Aurora MySQL の更新: 2018-08-23 (バージョン 2.02.3) (廃止) 239
Aurora MySQL バージョン 1 との比較 240
MySQL 5.7 の互換性 241
Aurora MySQL 2.x と Aurora MySQL 1.x の CLI の違い 241
改良点 242
Aurora MySQL の更新: 2018-06-04 (バージョン 2.02.2) (廃止) 242
改良点 243
Aurora MySQL 5.6 との比較 243
MySQL 5.7 の互換性 244
Aurora MySQL 2.x と Aurora MySQL 1.x の CLI の違い 241
改良点 243
Aurora MySQL の更新: 2018-05-03 (バージョン 2.02) (廃止) 245
Aurora MySQL 5.6 との比較 246
MySQL 5.7 の互換性 246
Aurora MySQL 2.x と Aurora MySQL 1.x の CLI の違い 241
改良点 247
MySQL バグ修正の統合 248
Aurora MySQL の更新: 2018-03-13 (バージョン 2.01.1) (廃止) 248

Aurora MySQL 5.6 との比較 249
MySQL 5.7 の互換性 249
Aurora MySQL 2.x と Aurora MySQL 1.x の CLI の違い 250
改良点 250
Aurora MySQL の更新: 2018-02-06 (バージョン 2.01) (廃止) 251
Aurora MySQL 5.6 との比較 251
MySQL 5.7 の互換性 252
Aurora MySQL 2.x と Aurora MySQL 1.x の CLI の違い 241
Aurora MySQL バージョン 1 (廃止) 254
Aurora MySQL の更新: 2021-09-30 (バージョン 1.23.4) (廃止) 256
改良点 256
Aurora MySQL の更新: 2021-06-28 (バージョン 1.23.3) (廃止) 257
改良点 257
Aurora MySQL の更新: 2021-03-18 (バージョン 1.23.2) (廃止) 257
改良点 258
MySQL Community Edition バグ修正の統合 259
Aurora MySQL の更新: 2020-11-24 (バージョン 1.23.1) (廃止) 260
改良点 260
Aurora MySQL の更新: 2020-09-02 (バージョン 1.23.0) (廃止) 261
改良点 262
MySQL Community Edition バグ修正の統合 265
Aurora MySQL の更新: 2021-06-03 (バージョン 1.22.5) (廃止) 267
改良点 267
Aurora MySQL の更新: 2021-03-04 (バージョン 1.22.4) (廃止) 268
改良点 268
Aurora MySQL の更新: 2020-11-09 (バージョン 1.22.3) (廃止) 269
改良点 270
MySQL Community Edition バグ修正の統合 271
Aurora MySQL の更新: 2020-03-05 (バージョン 1.22.2) (廃止) 272
改良点 272
Aurora MySQL の更新: 2019-12-23 (バージョン 1.22.1) (廃止) 273
改良点 274
Aurora MySQL の更新: 2019-11-25 (バージョン 1.22.0) (廃止) 274
改良点 275
MySQL Community Edition バグ修正の統合 279
Aurora MySQL の更新: 2019-11-25 (バージョン 1.21.0) (廃止) 279

改良点 280
MySQL Community Edition バグ修正の統合 281
Aurora MySQL の更新: 2020-03-05 (バージョン 1.20.1) (廃止) 281
改良点 282
Aurora MySQL の更新: 2019-11-11 (バージョン 1.20.0) (廃止) 282
改良点 283
MySQL Community Edition バグ修正の統合 284
Aurora MySQL の更新: 2020-03-05 (バージョン 1.19.6) (廃止) 285
改良点 286
Aurora MySQL の更新: 2019-09-19 (バージョン 1.19.5) (廃止) 286
改良点 287
MySQL Community Edition バグ修正の統合 287
Aurora MySQL の更新: 2019-06-05 (バージョン 1.19.2) (廃止) 288
改良点 289
Aurora MySQL の更新: 2019-05-09 (バージョン 1.19.1) (廃止) 289
改良点 290
Aurora MySQL の更新: 2019-02-07 (バージョン 1.19.0) (廃止) 290
機能 291
改良点 291
MySQL Community Edition バグ修正の統合 292
Aurora MySQL の更新: 2018-09-20 (バージョン 1.18.0) (廃止) 293
機能 293
Aurora MySQL の更新: 2020-03-05 (バージョン 1.17.9) (廃止) 294
改良点 295
Aurora MySQL の更新: 2019-01-17 (バージョン 1.17.8) (廃止) 295
改良点 296
MySQL Community Edition バグ修正の統合 296
Aurora MySQL の更新: 2018-10-08 (バージョン 1.17.7) (廃止) 296
改良点 297
MySQL Community Edition バグ修正の統合 297
Aurora MySQL の更新: 2018-09-06 (バージョン 1.17.6) (廃止) 298
改良点 298
MySQL Community Edition バグ修正の統合 299
Aurora MySQL の更新: 2018-08-14 (バージョン 1.17.5) (廃止) 299
改良点 300
Aurora MySQL の更新: 2018-08-07 (バージョン 1.17.4) (廃止) 300

改良点 300
Aurora MySQL の更新: 2018-06-05 (バージョン 1.17.3) (廃止) 301
改良点 302
Aurora MySQL の更新: 2018-04-27 (バージョン 1.17.2) (廃止) 302
改良点 303
Aurora MySQL の更新: 2018-03-23 (バージョン 1.17.1) (廃止) 303
改良点 304
Aurora MySQL の更新: 2018-03-13 (バージョン 1.17) (廃止) 304
ダウンタイムのないパッチ適用 305
新機能 305
改良点 305
MySQL バグ修正の統合 305
Aurora MySQL の更新: 2017-12-11 (バージョン 1.16) (廃止) 306
ダウンタイムのないパッチ適用 306
新機能 306
改良点 307
MySQL バグ修正の統合 307
Aurora MySQL の更新: 2017-11-20 (バージョン 1.15.1) (廃止) 307
ダウンタイムのないパッチ適用 308
改良点 308
MySQL バグ修正の統合 309
Aurora MySQL の更新: 2017-10-24 (バージョン 1.15) (廃止) 309
ダウンタイムのないパッチ適用 309
新機能 309
改良点 310
MySQL バグ修正の統合 309
Aurora MySQL の更新: 2018-03-13 (バージョン 1.14.4) (廃止) 311
ダウンタイムのないパッチ適用 312
新機能 312
改良点 312
MySQL バグ修正の統合 312
Aurora MySQL の更新: 2017-09-22 (バージョン 1.14.1) (廃止) 313
改良点 313
Aurora MySQL の更新: 2017-08-07 (バージョン 1.14) (廃止) 313
ダウンタイムのないパッチ適用 314
改良点 314

MySQL バグ修正の統合 315
Aurora MySQL の更新: 2017-05-15 (バージョン 1.13) (廃止) 315
ダウンタイムのないパッチ適用 316
新機能 316
改良点 316
MySQL バグ修正の統合 317
Aurora MySQL の更新: 2017-04-05 (バージョン 1.12) (廃止) 318
新機能 318
改良点 318
MySQL バグ修正の統合 319
Aurora MySQL の更新: 2017-02-23 (バージョン 1.11) (廃止) 319
新機能 320
改良点 321
MySQL バグ修正の統合 323
Aurora MySQL の更新: 2017-01-12 (バージョン 1.10.1) (廃止) 323
新機能 323
改良点 323
Aurora MySQL の更新: 2016-12-14 (バージョン 1.10) (廃止) 324
新機能 324
改良点 325
MySQL バグ修正の統合 326
Aurora MySQL の更新: 2016-11-10 (バージョン 1.9.0、1.9.1) (廃止) 326
新機能 326
改良点 327
Aurora MySQL の更新: 2016-10-26 (バージョン 1.8.1) (廃止) 327
改良点 327
MySQL バグ修正の統合 328
Aurora MySQL の更新: 2016-10-18 (バージョン 1.8) (廃止) 328
新機能 328
改良点 328
MySQL バグ修正の統合 329
Aurora MySQL の更新: 2016-09-20 (バージョン 1.7.1) (廃止) 330
改良点 330
Aurora MySQL の更新: 2016-08-30 (バージョン 1.7.0) (廃止) 330
新機能 330
改良点 331

MySQL バグ修正の統合 331
Aurora MySQL の更新: 2016-06-01 (バージョン 1.6.5) (廃止) 332
新機能 332
改良点 332
MySQL バグ修正の統合 332
Aurora MySQL の更新: 2016-04-06 (バージョン 1.6) (廃止) 333
新機能 333
改良点 334
MySQL バグ修正の統合 335
Aurora MySQL の更新: 2016-01-11 (バージョン 1.5) (廃止) 335
改良点 335
MySQL バグ修正の統合 336
Aurora MySQL の更新: 2015-12-03 (バージョン 1.4) (廃止) 336
新機能 336
改良点 337
MySQL バグ修正の統合 337
Aurora MySQL の更新: 2015-10-16 (バージョン 1.2、1.3) (廃止) 338
修正内容 338
改良点 339
MySQL バグ修正の統合 339
Aurora MySQL の更新: 2015-08-24 (バージョン 1.1) (廃止) 341
Aurora MySQL の更新で修正された MySQL のバグ 343
Aurora MySQL 3.x の更新で修正された MySQL のバグ 343
Aurora MySQL 2.x の更新で修正された MySQL のバグ 357
Aurora MySQL 1.x の更新で修正された MySQL のバグ 378
Aurora MySQL で修正されたセキュリティの脆弱性 398
ドキュメント履歴 405
..... cdxx

Amazon Aurora MySQL 互換エディションのリリースノート

Amazon Aurora MySQL 互換エディションのリリースは定期的に更新されます。更新はシステムメンテナンスの期間中に Aurora MySQL DB クラスターに適用されます。更新が適用されるタイミングは、AWS リージョン やメンテナンスウィンドウの設定、および更新のタイプによって異なります。

Amazon Aurora MySQL のリリースは、数日のうちに、すべての AWS リージョンで使用可能になります。一部のリージョンでは、別のリージョンではまだ使用できないエンジンバージョンが一時的に表示されることがあります。

更新は、DB クラスター内のすべてのインスタンスに同時に適用されます。更新では、DB クラスター内のすべてのインスタンスでデータベースを再起動する必要があります。20 ~ 30 秒間のダウンタイムが発生した後、DB クラスターの使用を再開できます。AWS Management Consoleでメンテナンスウィンドウの設定を表示または変更できます。

トピック

- Amazon Aurora MySQL バージョン 3 のデータベースエンジンの更新
- Amazon Aurora MySQL バージョン 2 のデータベースエンジンの更新
- Amazon Aurora MySQL バージョン 1 のデータベースエンジンの更新 (廃止)
- Aurora MySQL データベースエンジンの更新で修正された MySQL のバグ
- Aurora MySQL で修正されたセキュリティの脆弱性

Amazon Aurora MySQL バージョン 3 のデータベースエンジンの更新

Amazon Aurora MySQL バージョン 3 のデータベースエンジンの更新は次のとおりです。

トピック

- Aurora MySQL データベースエンジンアップデート 2024-03-07 (バージョン 3.06.0、MySQL 8.0.34 に対応)
- Aurora MySQL データベースエンジンアップデート 2024-01-31 (バージョン 3.05.2、MySQL 8.0.32 に対応)
- Aurora MySQL データベースエンジンアップデート 2023-11-21 (バージョン 3.05.1、MySQL 8.0.32 に対応)
- Aurora MySQL データベースエンジンの更新 2023-10-30 (バージョン 3.05.0.1、MySQL 8.0.32 互換) ベータ
- Aurora MySQL データベースエンジンの更新 2023-10-25(バージョン 3.05.0、MySQL 8.0.32 互換)
- Aurora MySQL データベースエンジンアップデート 2024-03-15 (バージョン 3.04.2、MySQL 8.0.28 に対応)
- Aurora MySQL データベースエンジンアップデート 2023-11-13 (バージョン 3.04.1、MySQL 8.0.28 に対応) デフォルト
- Aurora MySQL データベースエンジンアップデート 2023-07-31 (バージョン 3.04.0、MySQL 8.0.28 に対応)
- Aurora MySQL データベースエンジンの更新 2023-12-08 (バージョン 3.03.3、MySQL 8.0.26 互換)
- Aurora MySQL データベースエンジンの更新 2023-08-29 (バージョン 3.03.2、MySQL 8.0.26 互換)
- Aurora MySQL データベースエンジンの更新 2023-05-11 (バージョン 3.03.1、MySQL 8.0.26 互換)
- Aurora MySQL データベースエンジンの更新 2023-03-01 (バージョン 3.03.0、MySQL 8.0.26 互換) このバージョンへのアップグレードはサポートされていません。
- Aurora MySQL データベースエンジンの更新 2023-04-17 (バージョン 3.02.3、MySQL 8.0.23 互換) 標準サポートは 2024 年 1 月 15 日に終了。
- Aurora MySQL データベースエンジンの更新 2022-11-18 (バージョン 3.02.2、MySQL 8.0.23 互換) 標準サポートは 2024 年 1 月 15 日に終了。

- Aurora MySQL データベースエンジンの更新 2022-09-07 (バージョン 3.02.1、MySQL 8.0.23 互換) 標準サポートは 2024 年 1 月 15 日に終了。このバージョンへのアップグレードはサポートされていません。
- Aurora MySQL データベースエンジンの更新 2022-04-20 (バージョン 3.02.0、MySQL 8.0.23 互換) 標準サポートは 2024 年 1 月 15 日に終了。このバージョンへのアップグレードはサポートされていません。
- Aurora MySQL データベースエンジンの更新 2022-04-15 (バージョン 3.01.1、MySQL 8.0.23 互換) 標準サポートは 2024 年 1 月 15 日に終了。このバージョンへのアップグレードはサポートされていません。
- Aurora MySQL データベースエンジンの更新 2021-11-18 (バージョン 3.01.0、MySQL 8.0.23 互換) 標準サポートは 2024 年 1 月 15 日に終了。このバージョンへのアップグレードはサポートされていません。

Aurora MySQL データベースエンジンアップデート 2024-03-07 (バージョン 3.06.0、MySQL 8.0.34 に対応)

バージョン:3.06.0

Aurora MySQL 3.06.0 が一般公開されています。Aurora MySQL 3.06 バージョンは MySQL 8.0.34 と互換性があります。これまでのコミュニティ版の変更点の詳細については、「MySQL 8.0 Release Notes」を参照してください。

Aurora MySQL バージョン 3 の新機能の詳細については、「Aurora MySQL バージョン 3 は MySQL 8.0 との互換性があります」を参照してください。Aurora MySQL バージョン 3 と Aurora MySQL バージョン 2 の違いについての詳細は、「Aurora MySQL バージョン 2 と Aurora MySQL バージョン 3 の比較」を参照してください。Aurora MySQL バージョン 3 と MySQL 8.0 コミュニティエディションの比較については、Amazon Aurora ユーザーガイドの「Aurora MySQL バージョン 3 と MySQL 8.0 コミュニティエディションの比較」を参照してください。

現在サポートされている Aurora MySQL リリースは 2.07.9、2.07.10、2.11.*、2.12.*、3.03.*、3.04.*、3.05.*、3.06.* です。

現在サポートされている Aurora MySQL バージョン 2 クラスターから Aurora MySQL バージョン 3.06.0 クラスターへのインプレースアップグレード、スナップショットの復元、または Amazon RDS ブルー/グリーンデプロイメントによるマネージドブルー/グリーンアップグレードの開始を行うことができます。

Aurora MySQL バージョン 3 へのアップグレードの計画については、「Aurora MySQL バージョン 3 のアップグレード計画」を参照してください。Aurora MySQL のアップグレードに関する一般的な情報については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL DB クラスターのアップグレード」を参照してください。

トラブルシューティング情報については、Amazon Aurora Aurora ユーザーガイドの「Aurora MySQL バージョン 3 のアップグレードに関する問題のトラブルシューティング」を参照してください。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

新機能

- Aurora MySQL バージョン 3.06.0 は Amazon Bedrock との統合をサポートし、新しい予約キーワード`accept`、、、`aws_bedrock_invoke_model`およびを導入しています。`aws_sagemaker_invoke_endpoint` `content_type` `timeout_ms`バージョン 3.06.0 にアップグレードする前に、オブジェクト定義で新しいリザーブドキーワードの使用状況を確認してください。新しい予約キーワードとの競合を軽減するために、オブジェクト定義で使用されている予約済みキーワードを引用符で囲んでください。Amazon Bedrock との統合とリザーブドキーワードの処理に関する詳細については、「Amazon Bedrock とは何ですか?」を参照してください。Amazon Aurora ユーザーガイドに記載されています。追加情報については、MySQL ドキュメントの「キーワードと予約語」、「INFORMATION_SCHEMA キーワードテーブル」、および「スキーマオブジェクト名」を参照してください。
- 複数のセカンダリインデックスを持つ大きなテーブルのトランザクションをレプリケートするときのバイナリログレプリカのパフォーマンスが向上しました。この機能により、バイナリログレプリカにセカンダリインデックスの変更をparallel 適用するスレッドプールが導入されます。この機能は `aurora_binlog_replication_sec_index_parallel_workers` DB クラスターパラメータによって制御されます。このパラメータは、セカンダリインデックスの変更を適用できる parallel スレッドの総数を制御します。詳細については、「Amazon Aurora ユーザーガイド」の「バイナリログのレプリケーションの最適化」を参照してください。
- Aurora MySQL `mysql.rds_set_read_only` `read_only` クラスター内のデータベースインスタンスのグローバルシステム変数の値を変更できる新しいストアドプロシージャを追加しました。詳細については、Amazon Aurora ユーザーガイドの「レプリケーション」を参照してください。

- `mysql.rds_set_binlog_source_ssl`の値を指定してバイナリログレプリカの暗号化を設定できる新しいストアドプロシージャを追加しました。SOURCE_SSL詳細については、Amazon Aurora ユーザーガイドの「レプリケーション」を参照してください。
- Amazon Aurora 機械学習は、Aurora MySQL AWS データベースとMachine Learning (ML) サービスを最適に統合したものです。Amazon Bedrock がサポートされるようになったため、SQL を使用して Aurora MySQL DB クラスターから Amazon Bedrock の機械学習モデルを直接呼び出すことができます。Aurora MySQL DB クラスターで Amazon Bedrock を使用する方法の詳細については、Amazon Aurora ユーザーガイドの「Aurora MySQL での Amazon Aurora 機械学習の使用」を参照してください。
- Aurora MySQL バージョン 3.06 では、UNDO テーブルスペースの自動切り捨てのサポートが追加されています。この最適化により、UNDO ログが消去された後でも UNDO テーブルスペースの未使用領域を再利用できます。

改良点

セキュリティ問題と CVE が修正されました。

このリリースには、次の CVE 修正が含まれています。

- CVE-2020-11104
- CVE-2020-11105
- CVE-2023-38545
- CVE-2023-38546
- CVE-2023-39975

可用性の向上:

- ライター DB インスタンスのワークロードが高い場合、リードレプリカ DB インスタンスを正常に起動できない問題を修正しました。
- Aurora MySQL ライター DB インスタンスが Aurora ストレージとの通信の欠陥によりフェイルオーバーする可能性がある問題を修正しました。この欠陥は、Aurora ストレージインスタンスのソフトウェア更新後に DB インスタンスと基盤となるストレージ間の通信が中断された結果として発生します。
- InnoDB INSERT パーティションテーブルでクエリを処理するときに、インスタンスの空きメモリが徐々に減少する可能性がある問題を修正しました。

- クエリの実行中にハッシュ結合を使用すると、解放可能なメモリが減少するため、Aurora MySQL DB インスタンスが再起動またはフェールオーバーする可能性がある問題を修正しました。
- SHOW STATUS ステートメントと PURGE BINARY LOGS ステートメントを同時に実行するとデータベースインスタンスが再起動する可能性がある問題を修正しました。 PURGE BINARY LOGSは、ユーザーが設定したバイナリログの保持期間に従って実行されるマネージドステートメントです。
- 非仮想列が OR ステートメントで並べ替えられたテーブルでデータ操作言語 (DML) ステートメントを実行すると、サーバーが予期せず終了する問題を修正しました。MODIFY COLUMN CHANGE COLUMN
- データベースインスタンスの再起動中に、さらに再起動が必要になることがある問題を修正しました。
- DELETE仮想列が外部キー制約の列として、UPDATEまたは参照されるテーブルのメンバーとして関与しているテーブルでカスケードまたは外部キー制約が定義されていると、データベースが再起動する可能性がある問題を修正しました。
- Aurora MySQL 2.10 では、読み取りが可能な Aurora DB クラスターを再起動するためのサポートが追加されました。この機能により、ライター DB インスタンスが再起動されている間、リーダー DB インスタンスをオンラインに保つことができます。この機能は Aurora MySQL グローバルデータベースのセカンダリでサポートされるようになったため、 AWS リージョン プライマリクラスターでのライターインスタンスの再起動中でも読み取りリクエストを処理できます。以前は、ライターインスタンスを再起動すると、Aurora MySQL セカンダリクラスターのすべてのリーダーインスタンスも再起動しました。このリリースでは、ライターインスタンスの再起動中もセカンダリクラスターリーダーインスタンスが引き続き読み取りリクエストを処理するようになり、クラスター内の読み取りの可用性が向上しました。詳細については、「読み取り可能な Aurora クラスターの再起動」を参照してください。
- 列を含む負荷の高い挿入操作の実行中に再起動が発生すると、起動時にデータベースの回復が中断されることがある問題を修正しました。AUTO_INCREMENT

全般的な機能強化:

- Aurora クラスターボリュームからデータを読み取る際の一時的なネットワークの問題により、parallel クエリが失敗する可能性がある問題を修正しました。
- ユーザーがクエリを中断したり、クエリのセッションタイムアウトを設定したりできない問題を修正しました。performance_schema

- カスタム SSL 証明書 (mysql.rds_import_binlog_ssl_material) を使用するように構成されたバイナリログ (binlog) のレプリケーションが、レプリケーションインスタンスがホストの交換を行っているときに失敗することがある問題を修正しました。
- メモリが 4 GiB 以下の小規模な DB インスタンスは、DB インスタンスがメモリ不足に陥ったときに、バッファプールを自動的に調整し、メモリ消費量の多い接続を閉じるようになりました。詳細については、Amazon Aurora ユーザーガイドの「Amazon Aurora MySQL out-of-memory の問題」を参照してください。
- 4 GiB を超えるメモリを持つすべての DB インスタンスクラスでのデフォルトレスポンスを空からに変更しました。`aurora_oom_response` `print`詳細については、Amazon Aurora ユーザーガイドの「Amazon Aurora MySQL out-of-memory の問題」を参照してください。
- 監査ログファイル管理に関連する問題を修正しました。これにより、ログファイルにアクセスしてダウンロードやローテーションができなくなり、場合によっては CPU 使用率が高くなる可能性があります。
- `AUTO_INCREMENT`キーリカバリを最適化して、スナップショットの復元、 point-in-time リカバリの実行、データベース内に多数のテーブルがある DB クラスターのクローニングの完了時間を短縮しました。
- wait/io/redo_log_flush イベントがパフォーマンススキーマの待機イベントサマリーテーブルに表示されない問題を修正しました。
- ロックマネージャーのメモリ使用量を追跡するおよびメトリクスを追加しました。`Aurora_lockmgr_memory_used` `Aurora_lockmgr_buffer_pool_memory_used`詳細については、Amazon Aurora ユーザーガイドの「Aurora MySQL グローバルステータス変数」を参照してください。
- 2.11.* より前のバージョンの Aurora MySQL からアップグレードすると、小規模なリードレプリカインスタンスのレプリケーション遅延が増える問題を修正しました。
- スナップショットの復元、バックトラック、またはデータベースのクローニング操作後に、`AUTO_INCREMENT`降順インデックスを使用する列で重複キーエラーが発生する問題を修正しました。
- テーブルに少なくとも 1 つの全文検索 (FTS) インデックスがあり、Aurora ライター DB `TRUNCATE` インスタンスでステートメントが実行されているときに、Aurora `SELECT` リーダーインスタンスのクエリが「テーブルが存在しない」というエラーで失敗する可能性がある問題を修正しました。
- parallel `LEFT JOIN` `RIGHT JOIN` クエリでハッシュ結合アルゴリズムを使用するまたは操作を含むクエリを実行すると、結果セットが不完全になることがある問題を修正しました。

アップグレードと移行:

- FTS_DOC_IDテーブルスキーマにユーザー定義の列があると、メジャーバージョンのアップグレードが失敗する問題を修正しました。
- InnoDB テーブルスペースの処理中の同期の問題により、Aurora MySQL バージョン 2 から Aurora MySQL バージョン 3 へのアップグレードが失敗する可能性がある問題を修正しました。
- Aurora MySQL バージョン 2 の InnoDB システムテーブルにすでに削除されているテーブルスペースの孤立したエントリが存在するため、Aurora MySQL バージョン 3 へのメジャーバージョンアップグレードが失敗する可能性がある問題を修正しました。

MySQL Community Edition でのバグ修正の統合

このリリースには、以下に加えて、8.0.34 以前のすべてのコミュニティバグ修正が含まれています。詳細については、「MySQL 3.x データベースエンジンの更新で修正された MySQL のバグ」を参照してください。

- キャッシュラインの値が正しく計算されず、Graviton ベースのインスタンスでデータベースを再起動する際に障害が発生する問題を修正しました。(コミュニティバグフィックス #35479763)
- ストアドルーチン内のサブクエリの一部のインスタンスが常に正しく処理されない問題を修正しました。(コミュニティバグフィックス #35377192)
- バックグラウンドの TLS 証明書のローテーションが原因で CPU 使用率が高くなることがある問題を修正しました (コミュニティバグフィックス #34284186)。
- Aurora MySQL バージョン 3.05 より前の MySQL INSTANT システムスキーマのテーブルへの列の追加を InnoDB が許可していた問題を修正しました。これにより、Aurora MySQL バージョン 3.05.0 にアップグレードした後に、サーバーが予期せず終了する (データベースインスタンスが再起動する) 可能性がある問題を修正しました。(コミュニティバグフィックス #35625510)。

Aurora MySQL データベースエンジンアップデート 2024-01-31 (バージョン 3.05.2、MySQL 8.0.32 に対応)

バージョン:3.05.2

Aurora MySQL 3.05.2 が一般公開されています。Aurora MySQL 3.05 バージョンは、MySQL 8.0.32 と互換性があります。これまでのコミュニティ版の変更点の詳細については、「MySQL 8.0 Release Notes」を参照してください。

Aurora MySQL バージョン 3 の新機能の詳細については、「Aurora MySQL バージョン 3 は MySQL 8.0 との互換性があります」を参照してください。Aurora MySQL バージョン 3 と Aurora MySQL バージョン 2 の違いについての詳細は、「Aurora MySQL バージョン 2 と Aurora MySQL バージョン 3 の比較」を参照してください。Aurora MySQL バージョン 3 と MySQL 8.0 コミュニティエディションの比較については、Amazon Aurora ユーザーガイドの「Aurora MySQL バージョン 3 と MySQL 8.0 コミュニティエディションの比較」を参照してください。

現在サポートされている Aurora MySQL リリースは、2.07.9、2.07.10、2.11.*、2.12.*、3.03.*、3.04.*、3.05.* です。

Amazon RDS ブルー/グリーンデプロイメントを使用して、現在サポートされている Aurora MySQL バージョン 2 クラスターから Aurora MySQL バージョン 3.05.2 クラスターへの一括アップグレード、スナップショットの復元、またはマネージドブルー/グリーンアップグレードの開始を行うことができます。

Aurora MySQL バージョン 3 へのアップグレードの計画については、「Aurora MySQL バージョン 3 のアップグレード計画」を参照してください。Aurora MySQL のアップグレードに関する一般的な情報については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL DB クラスターのアップグレード」を参照してください。

トラブルシューティング情報については、Amazon Aurora Aurora ユーザーガイドの「Aurora MySQL バージョン 3 のアップグレードに関する問題のトラブルシューティング」を参照してください。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

修正されたセキュリティ問題と CVE:

このリリースには、次の CVE 修正が含まれています。

- CVE-2020-11104
- CVE-2020-11105
- CVE-2023-38545
- CVE-2023-39975

可用性の向上:

- InnoDB INSERT パーティションテーブルでクエリを処理すると、インスタンスの空きメモリが徐々に減少する可能性がある問題を修正しました。
- SHOW STATUS ステートメントと PURGE BINARY LOGS ステートメントを同時に実行すると、データベースインスタンスが再起動する可能性がある問題を修正しました。 PURGE BINARY LOGSは、ユーザーが設定したバイナリログの保持期間に従って実行されるマネージドステートメントです。
- 非仮想列が OR ステートメントで並べ替えられたテーブルでデータ操作言語 (DML) ステートメントを実行すると、サーバーが予期せず終了する問題を修正しました。MODIFY COLUMN CHANGE COLUMN
- データベースインスタンスの再起動中に、さらに再起動が必要になることがある問題を修正しました。

全般的な機能強化:

- ユーザーがクエリを中断したり、クエリのセッションタイムアウトを設定したりできない問題を修正しました。performance_schema
- レプリケーションインスタンスがホストの交換を行っているときに、カスタム SSL 証明書 (mysql.rds_import_binlog_ssl_material) を使用したバイナリログ (binlog) のレプリケーション設定が失敗する可能性がある問題を修正しました。
- 監査ログファイル管理に関連する問題を修正しました。これにより、ログファイルにアクセスしてダウンロードやローテーションができなくなり、場合によっては CPU 使用率が高くなる可能性があります。

アップグレードと移行:

- FTS_DOC_IDテーブルスキーマにユーザー定義列がある場合に Aurora MySQL バージョン 2 から Aurora MySQL バージョン 3 へのアップグレードが失敗する可能性がある問題を修正しました。
- InnoDB テーブルスペースの処理中の同期の問題により、Aurora MySQL バージョン 2 から Aurora MySQL バージョン 3 へのアップグレードが失敗する可能性がある問題を修正しました。
- Aurora MySQL バージョン 2 の InnoDB システムテーブルにすでに削除されているテーブルスペースの孤立したエントリが存在するため、Aurora MySQL バージョン 3 へのメジャーバージョンアップグレードが失敗する可能性がある問題を修正しました。

MySQL Community Edition でのバグ修正の統合

このリリースには、8.0.32 までのすべてのコミュニティバグ修正が含まれています。詳細については、「MySQL 3.x データベースエンジンの更新で修正された MySQL のバグ」を参照してください。

Aurora MySQL データベースエンジンアップデート 2023-11-21 (バージョン 3.05.1、MySQL 8.0.32 に対応)

バージョン:3.05.1

Aurora MySQL 3.05.1 が一般公開されています。Aurora MySQL 3.05 バージョンは、MySQL 8.0.32 と互換性があります。詳細については、MySQL 8.0 リリースノートを参照してください。

現在サポートされている Aurora MySQL リリースは、2.07.*、2.11.*、2.12.*、3.01.*、3.02.*、3.03.*、3.04.*、3.05.* です。

既存の Aurora MySQL 3.* データベースクラスターを Aurora MySQL 3.05.1 にアップグレードできます。また、現在サポートされている Aurora MySQL リリースのスナップショットを Aurora MySQL 3.05.1 に復元することもできます。

Aurora MySQL グローバルデータベースをバージョン 3.05.* にアップグレードする場合、プライマリ DB クラスターとセカンダリ DB クラスターを、パッチレベルを含めてまったく同じバージョンにアップグレードする必要があります。Aurora グローバルデータベースのマイナーバージョンのアップグレードの詳細については、「マイナーバージョンのアップグレード」を参照してください。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

以下のセキュリティの問題と CVE の修正:

このリリースには、MySQL 8.0.32 までのコミュニティ版 CVE の修正がすべて反映されています。

- CVE-2023-38545

MySQL Community Edition でのバグ修正の統合

このリリースには、以下を含め、8.0.32 までのコミュニティ版のバグ修正がすべて反映されています。詳細については、「MySQL 3.x データベースエンジンの更新で修正された MySQL のバグ」を参照してください。

- システムスキーマ内の MySQL テーブルで Aurora MySQL バージョン 3.01 から Aurora MySQL バージョン 3.04 までの間に `INSTANT ADD` 列が追加され、Aurora MySQL がバージョン 3.05.0 にアップグレードされた後に、これらのテーブルの DML によってサーバーが予期せず終了する場合の InnoDB の問題を修正しました。(コミュニティのバグ修正 #35625510)

Aurora MySQL データベースエンジンの更新 2023-10-30 (バージョン 3.05.0.1、MySQL 8.0.32 互換) ベータ

バージョン: 3.05.0.1

Aurora MySQL 3.05.0.1 は、通常、米国東部 (バージニア北部)、米国東部 (オハイオ)、米国西部 (北カリフォルニア)、米国西部 (オレゴン)、(米国東部)、および AWS GovCloud (米国西部) でご利用いただけます。 AWS GovCloud これは初期のセキュリティ修正限定のリリースです。これらの修正は、次回のパッチリリース 3.05.1 で対象範囲を広げ、すべてのリージョンにデプロイされる予定です。Aurora MySQL 3.05 バージョンは、MySQL 8.0.32 と互換性があります。これまでのコミュニティ版の変更点の詳細については、「MySQL 8.0 Release Notes」を参照してください。

現在サポートされている Aurora MySQL リリース
は、2.07.*、2.11.*、2.12.*、3.01.*、3.02.*、3.03.*、3.04.*、3.05.* です。

既存の Aurora MySQL 3.* データベースクラスターを Aurora MySQL 3.05.0.1 にアップグレードできます。現在サポートされている Aurora MySQL リリースから取得したスナップショットを Aurora MySQL 3.05.0.1 で復元することもできます。

Aurora MySQL グローバルデータベースをバージョン 3.05.* にアップグレードする場合、プライマリ DB クラスターとセカンダリ DB クラスターを、パッチレベルを含めてまったく同じバージョンにアップグレードする必要があります。Aurora グローバルデータベースのマイナーバージョンのアップグレードの詳細については、「マイナーバージョンのアップグレード」を参照してください。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

以下のセキュリティの問題と CVE の修正:

このリリースには、MySQL 8.0.32 までのコミュニティ版 CVE の修正がすべて反映されています。

- CVE-2023-38545

Aurora MySQL データベースエンジンの更新 2023-10-25(バージョン 3.05.0、MySQL 8.0.32 互換)

バージョン: 3.05.0

Aurora MySQL 3.05.0 は一般公開されています。Aurora MySQL 3.05 バージョンは、MySQL 8.0.32 と互換性があります。これまでのコミュニティ版の変更点の詳細については、「MySQL 8.0 Release Notes」を参照してください。

Aurora MySQL バージョン 3 の新機能の詳細については、「Aurora MySQL バージョン 3 は MySQL 8.0 との互換性があります」を参照してください。Aurora MySQL バージョン 3 と Aurora MySQL バージョン 2 の違いについての詳細は、「Aurora MySQL バージョン 2 と Aurora MySQL バージョン 3 の比較」を参照してください。Aurora MySQL バージョン 3 と MySQL 8.0 Community Edition の比較については、「Aurora MySQL バージョン 3 と MySQL 8.0 コミュニティエディションの比較」を参照してください。

現在サポートされている Aurora MySQL リリースは、2.07.9、2.07.10、2.11.*、2.12.*、3.03.*、3.04.*、3.05.* です。

現在サポートされている Aurora MySQL バージョン 2 クラスターから Aurora MySQL バージョン 3.05.0 クラスターへのインプレースアップグレード、スナップショットの復元、Amazon RDS ブルー/グリーンデプロイによるマネージドブルー/グリーンアップグレードの開始を行うことができます。

Aurora MySQL バージョン 3 へのアップグレードの計画については、「Amazon Aurora ユーザーガイド」の「Upgrade planning for Aurora MySQL version 3」を参照してください。Aurora MySQL のアップグレードに関する一般的な情報については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL DB クラスターのアップグレード」を参照してください。

トラブルシューティング情報については、「Aurora MySQL バージョン 3 のアップグレードに関する問題のトラブルシューティング」を参照してください。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

新機能:

- KMS キー (SSE-KMS) で暗号化された Amazon S3 バケット内のテキストファイルに Aurora MySQL データベースクラスターからデータを保存できるようになりました。詳細については、「Amazon Aurora MySQL DB クラスターから Amazon S3 バケット内のテキストファイルへのデータの保存」を参照してください。
- エンジンで使用されているタイムゾーン (TZ) 情報の最新バージョンを示す新しいグローバルステータス変数 `aurora_tmz_version` が追加されました。この値は IANA タイムゾーンデータベースのバージョンに従い、「YYYYsuffix」という形式になっています (例:2022a と 2023c)。詳細については、「Aurora MySQL グローバルステータス変数」を参照してください。

以下のセキュリティの問題と CVE の修正:

マネージド型の環境での処理を微調整するための修正およびその他の機能強化。以下の CVE の追加の修正:

- CVE-2022-37434

可用性の向上:

- パラレルクエリを使用する Aurora MySQL データベースインスタンスで、多数のパラレルクエリを同時に実行すると、データベースが再起動する問題を修正しました。
- 監査ログ記録のスレッドが原因でロック競合が発生し、最終的に CPU 使用率が高くなり、クライアントアプリケーションがタイムアウトする問題を修正しました。
- 拡張バイナリログ (binlog) が有効になっているバイナリログレプリカクラスターで、いずれかのバイナリログソースで `gtid_mode` が `ON` または `ON_PERMISSIVE` に設定されている場合に、実行された GTID セットが誤って復元されることがある問題を修正しました。この問題が原因で、レプリカクラスターのライターインスタンスが復元中にさらに再起動したり、実行された GTID セットを照会したときに誤った結果が返されたりする可能性があります。

- 拡張バイナリログが有効になっている場合に解放可能なメモリが減少し、Aurora MySQL データベースインスタンスが再起動またはフェイルオーバーする原因となる、メモリ管理の問題を修正しました。
- 削除されたテーブルに属するデータベースページを読み取ろうとした場合に、データベースインスタンスが再起動する問題を修正しました。
- ライターインスタンスがデータベースボリュームを拡大させ、160 GB の倍数に達すると、リーダーインスタンスが再起動する問題を修正しました。
- 拡張バイナリログ機能が有効になっている Aurora MySQL データベースインスタンスが、バイナリログの復旧プロセスが実行されている場合に、起動中に停止してしまう問題を修正しました。
- 大規模なロールバックセグメントが初期化されている間、Aurora MySQL データベースインスタンスが起動中に複数回再起動する可能性がある問題を修正しました。
- ダウンタイムのないパッチ適用中にインスタンスが再起動し、データベース接続が予期せず切断される問題を修正しました。
- SHOW STATUS ステートメントと PURGE BINARY LOGS ステートメントを同時に実行すると、デッドラッチが原因でデータベースインスタンスが再起動する可能性がある問題を修正しました。purge binary logs は、ユーザーが設定したバイナリログの保持期間に従って実行されるマネージドステートメントです。
- データベースが内部システムテーブルでトリガーを作成または削除しているときにライターインスタンスが再起動すると、データベースクラスターが使用できない状態になる問題を修正しました。
- Aurora レプリカのあるクラスターで拡張バイナリログ機能を使用する場合、セマフォの待機時間が長くなり、データベースインスタンスが再起動する可能性がある問題を修正しました。
- 集計関数を参照するクエリの実行中に、データベースインスタンスが再起動する可能性がある問題を修正しました。
- Aurora Serverless v2 がスケーリング中に誤ってテーブルキャッシュの更新を試みた場合に、まれにデータベースインスタンスが再起動する可能性がある問題を修正しました。
- 中間一時テーブルを実体化 (マテリアライズ) する際に、共通テーブル式 (CTE) に対してサポートされていないインデックススキャンのアクセス手法が検討され、データベースの再起動や誤ったクエリ結果など、望ましくない挙動が生じる可能性がある問題を修正しました。 TempTable ストレージエンジンを使用するテーブルで、このようなサポートされていないインデックススキャンアクセス方法を使用しないようにすることで、この問題を解決しました。

全般的な機能強化:

- Aurora MySQL 3.04.0 で実行されている Aurora Serverless v2 データベースクラスターで拡張バイナリログが有効になっている場合に、データベースが使用できなくなることがある問題を修正しました。
- 拡張バイナリログ機能が有効になっている場合は、Aurora ストレージへの書き込み前に、未使用のストレージメタデータが削除されます。その結果、ネットワーク上で転送されるバイト数が増えて書き込み遅延が長くなり、データベースの再起動やフェイルオーバーが発生する特定のシナリオを回避できます。
- performance_schema に malloc_stats テーブルと malloc_stats_totals テーブルが追加され、内部メモリアロケータである Jemalloc の動作を制御する 3 つの高度なシステム変数が追加されました。
  - aurora_jemalloc_background_thread.
  - aurora_jemalloc_dirty_decay_ms.
  - aurora_jemalloc_tcache_enabled.
- アップグレード時または移行時に Aurora 固有のパフォーマンススキーマテーブルが作成されない問題を修正しました。
- 新しいシステム変数 aurora_use_vector_instructions が追加されました。このパラメータが有効な場合、Aurora MySQL は、最適化されたベクトル処理命令を使用して I/O 負荷の高いワークロードのパフォーマンスを向上させます。Aurora MySQL 3.05 以降では、この設定がデフォルトで ON になっています。詳細については、「Aurora MySQL 設定パラメータ」を参照してください。
- 拡張バイナリログが有効になっていると、NumBinaryLogFiles CloudWatch メトリクスに誤った結果が表示されることがある問題を修正しました。
- Amazon Sagemaker に対する Aurora MySQL 機械学習オペレーションのリクエストタイムアウトが 3 秒から 30 秒に延長されました。これにより、バッチサイズが大きい場合に、Aurora MySQL 機械学習から Amazon Sagemaker へのリクエストの再試行回数や失敗回数が増えるという問題を解決できます。
- performance_schema データベース内の malloc_stats テーブルと malloc_stats_totals テーブルのサポートが追加されました。
- LOAD DATA FROM S3 コマンドの FROM キーワードが更新され、オプションになりました。詳細については、「Amazon S3 バケットのテキストファイルから Amazon Aurora MySQL DB クラスターへのデータのロード」を参照してください。
- パラメータ innodb_aurora_instant_alter_column_allowed のサポートが追加されました。このパラメータは、INSTANT アルゴリズムを ALTER COLUMN オペレーションに使用できる

かどうかを制御します。詳細については、「クラスターレベルのパラメータ」を参照してください。

- 書き込み転送が有効になっていると、クライアントからデータベースへの新しい接続が確立されないことがある問題を修正しました。
- table_open_cache データベースパラメータの変更が、データベースインスタンスを再起動するまで有効にならないことがある問題を修正しました。
- スナップショットの復元、バックトラック、またはデータベースのクローン作成のオペレーション後に、降順インデックスを使用する AUTO_INCREMENT 列で重複キーエラーが発生する可能性がある問題を修正しました。
- GROUP BY 句を使用し、aurora_parallel_query パラメータを ON に指定した SELECT クエリを実行すると、不正確な結果が返されることがある、インデックススキャンに関する問題を修正しました。
- INFORMATION_SCHEMA INNODB_TABLESPACES テーブルに対してクエリを実行すると、使用可能なメモリが枯渇する可能性がある問題を修正しました。
- リーダーインスタンスがテーブルを開くことができず、ERROR 1146 が発生する問題を修正しました。この問題は、ライターインスタンスで INPLACE アルゴリズムが使用されている間に、特定の種類のオンラインデータ定義言語 (DDL) を実行すると発生します。
- 内部モニタリングプロセスが重複するスケーリングリクエストを誤って送信した場合に、Aurora Serverless v2 のスケーリング中にインスタンスの再起動を回避するため、問題を修正しました。
- 接続先のバイナリログ (binlog) コンシューマーが、重複するバイナリログレプリケーションサーバー ID を使用している場合に、データベースが再起動する問題を修正しました。
- Aurora MySQL が管理するバイナリログレプリカ用のインメモリリレーログキャッシュを導入しました。この改善により、バイナリログのレプリケーションスループットを最大 40% 向上させることができます。この拡張機能は、シングルスレッドのバイナリログレプリケーションを使用する場合や、GTID 自動配置を有効にしたマルチスレッドレプリケーションを使用する場合に自動的に有効になります。

アップグレードと移行:

- MySQL 5.7 から MySQL 8.0 にアップグレードする際に、1 つのデータベースに多数のテーブルがある場合、サーバーがメモリを過剰に消費していました。テーブルをアップグレードできるかどうかを確認する過程で、データディクショナリ Table オブジェクトをすべて事前に取得し、それぞれを処理して名前を取得してから、リストに対してバージョン互換性の確認を実行していたことが判明しました。この場合、すべてのオブジェクトを事前に取得する必要はなく、そのせいでメモリ消費量に大きな影響が出ていました。この問題を解決するために、このような場合には、一度に 1

つずつ Table オブジェクトを取得し、必要なチェックをすべて実行して名前を取得し、オブジェクトを解放してから、次のオブジェクトに進むことにしました。(バグ #34526001)

- データベースインスタンスで利用可能な vCPU をすべて使用してテーブルスペースチェックを並列実行することで、Aurora MySQL バージョン 2 からバージョン 3 へのメジャーバージョンアップグレードのパフォーマンスが向上しました。

MySQL Community Edition でのバグ修正の統合

このリリースには、以下を含め、8.0.32 までのコミュニティ版のバグ修正がすべて反映されています。詳細については、「MySQL 3.x データベースエンジンの更新で修正された MySQL のバグ」を参照してください。

- バックグラウンドの TLS 証明書のローテーションが原因で CPU 使用率が高くなる問題を修正しました。(コミュニティのバグ修正 #34284186)

Aurora MySQL データベースエンジンアップデート 2024-03-15 (バージョン 3.04.2、MySQL 8.0.28 に対応)

バージョン:3.04.2

Aurora MySQL 3.04.2 が一般公開されています。Aurora MySQL 3.04 バージョンは MySQL 8.0.28 と互換性があります。発生したコミュニティの変更の詳細については、MySQL 8.0 リリースノートを参照してください。

Aurora MySQL バージョン 3 の新機能の詳細については、「Aurora MySQL バージョン 3 は MySQL 8.0 との互換性があります」を参照してください。Aurora MySQL バージョン 3 と Aurora MySQL バージョン 2 の違いについての詳細は、「Aurora MySQL バージョン 2 と Aurora MySQL バージョン 3 の比較」を参照してください。Aurora MySQL バージョン 3 と MySQL 8.0 Community Edition の比較については、「Aurora MySQL バージョン 3 と MySQL 8.0 コミュニティエディションの比較」を参照してください。

> **Note**
>
> このバージョンは、長期サポート (LTS) リリースとして指定されています。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL 長期サポート (LTS) リリース」を参照してください。

現在サポートされている Aurora MySQL リリースは
2.07.9、2.7.10、2.11.*、2.12.*、3.03.*、3.04.*、3.05.*、3.06.* です。

現在利用可能な Aurora MySQL バージョン 2 クラスターから Aurora MySQL バージョン 3.04.2 クラスターへのインプレースアップグレード、スナップショットの復元、または Amazon RDS ブルー/グリーンデプロイメントによるマネージドブルー/グリーンアップグレードの開始を行うことができます。

Aurora MySQL バージョン 3 へのアップグレードの計画については、「Amazon Aurora ユーザーガイド」の「Upgrade planning for Aurora MySQL version 3」を参照してください。Aurora MySQL のアップグレードに関する一般的な情報については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL DB クラスターのアップグレード」を参照してください。

トラブルシューティング情報については、「Aurora MySQL バージョン 3 のアップグレードに関する問題のトラブルシューティング」を参照してください。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

修正されたセキュリティ問題と CVE:

このリリースには、次の CVE 修正が含まれています。

- CVE-2020-11104
- CVE-2020-11105
- CVE-2023-38545
- CVE-2023-38546
- CVE-2023-39975

可用性の向上:

- ライター DB インスタンスのワークロードが高い場合、リードレプリカ DB インスタンスを正常に起動できない問題を修正しました。
- Aurora ストレージと通信するコンポーネントの不具合により、Aurora MySQL ライター DB インスタンスがフェイルオーバーする可能性がある問題を修正しました。この不具合は、ソフトウェアの更新後に DB インスタンスと基盤となるストレージ間の通信が切断されたために発生します。

- SHOW STATUS ステートメントと PURGE BINARY LOGS ステートメントを同時に実行すると、データベースインスタンスが再起動する可能性がある問題を修正しました。 PURGE BINARY LOGSは、ユーザーが設定したバイナリログの保持期間に従って実行されるマネージドステートメントです。
- データベースインスタンスの再起動中に、さらに再起動されることがある問題を修正しました。
- 監査ログ記録のスレッドが原因でロック競合が発生し、CPU 使用率が高くなり、クライアントアプリケーションがタイムアウトする問題を修正しました。
- 大規模なロールバックセグメントが初期化されている間、Aurora MySQL データベースインスタンスがインスタンスの起動中に複数回再起動する可能性がある問題を修正しました。
- 集計関数を参照するクエリの実行中に DB インスタンスが再起動する可能性がある問題を修正しました。

全般的な機能強化:

- Aurora DB クラスターボリュームからデータを読み取る際の一時的なネットワークの問題により、parallel クエリが失敗する可能性がある問題を修正しました
- ユーザーがクエリを中断したり、クエリのセッションタイムアウトを設定したりできない問題を修正しました。performance_schema
- カスタム SSL 証明書 (mysql.rds_import_binlog_ssl_material) を使用するように構成されたバイナリログ (binlog) のレプリケーションが、レプリケーションインスタンスがホストの交換を行っているときに失敗することがある問題を修正しました。
- 監査ログファイル管理に関連する問題を修正しました。これにより、ログファイルにアクセスしてダウンロードやローテーションができなくなり、場合によっては CPU 使用率が高くなる可能性があります。
- AUTO_INCREMENTキーリカバリを最適化して、スナップショットの復元、 point-in-time リカバリの実行、データベース内に多数のテーブルがある DB クラスターのクローニングの完了時間を短縮しました。
- Community MySQL から Aurora MySQL バージョン 3.04.0 および 3.04.1 に移行した後に、performance_schema一部のテーブルを参照する SQL ステートメントがこれらのテーブルが見つからないためにエラーを返すことがあるという問題を修正しました。
- 2.11.* より前のバージョンの Aurora MySQL からアップグレードすると、小規模なリードレプリカインスタンスのレプリケーション遅延が増える問題を修正しました。

- スナップショットの復元、バックトラック、またはデータベースのクローニング操作後に、AUTO_INCREMENT降順インデックスを使用する列で重複キーエラーが発生する問題を修正しました。
- DB table_open_cache インスタンスを再起動するまでデータベースパラメータの変更が有効にならないことがある問題を修正しました。
- リーダー DB インスタンスがエラー 1146 でテーブルを開くことができない問題を修正しました。この問題は、ライター DB INPLACE インスタンスでアルゴリズムが使用されているときに、特定の種類のオンラインデータ定義言語 (DDL) ステートメントを実行すると発生します。
- 内部モニタリングプロセスが重複するスケーリングリクエストを誤って送信した場合に、Aurora Serverless v2 のスケーリング中にインスタンスの再起動を回避するため、問題を修正しました。
- 接続しているバイナリログ (binlog) コンシューマーが重複するバイナリログレプリケーションサーバー ID を使用していると、データベースが再起動する可能性がある問題を修正しました。

アップグレードと移行:

- Aurora MySQL バージョン 2 の InnoDB システムテーブルにすでに削除されているテーブルスペースの孤立したエントリが存在するため、Aurora MySQL バージョン 3 へのメジャーバージョンアップグレードが失敗する可能性がある問題を修正しました。

MySQL Community Edition でのバグ修正の統合

このリリースには、以下に加えて、8.0.28 以前のすべてのコミュニティバグ修正が含まれています。詳細については、「MySQL 3.x データベースエンジンの更新で修正された MySQL のバグ」を参照してください。

- キャッシュラインの値が正しく計算されず、Graviton ベースのインスタンスでデータベースを再起動する際に障害が発生する問題を修正しました。(コミュニティバグフィックス #35479763)
- 複数の、、XORまたは条件を含む SELECT OR ステートメントをサブクエリとして持つストアドルーチンを繰り返し実行するとAND、仮想メモリが過剰に消費され、最終的には仮想メモリが枯渇する可能性がありました。(コミュニティバグフィックス #33852530)

Aurora MySQL データベースエンジンアップデート 2023-11-13 (バージョン 3.04.1、MySQL 8.0.28 に対応) デフォルト

バージョン:3.04.1

Aurora MySQL 3.04.1 が一般公開されています。Aurora MySQL 3.04 バージョンは MySQL 8.0.28 と互換性があります。発生したコミュニティの変更の詳細については、MySQL 8.0 リリースノートを参照してください。

> **Note**
>
> このバージョンは、長期サポート (LTS) リリースとして指定されています。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL 長期サポート (LTS) リリース」を参照してください。

Aurora MySQL バージョン 3 の新機能の詳細については、「Aurora MySQL バージョン 3 は MySQL 8.0 との互換性があります」を参照してください。Aurora MySQL バージョン 3 と Aurora MySQL バージョン 2 の違いについての詳細は、「Aurora MySQL バージョン 2 と Aurora MySQL バージョン 3 の比較」を参照してください。Aurora MySQL バージョン 3 と MySQL 8.0 Community Edition の比較については、「Aurora MySQL バージョン 3 と MySQL 8.0 コミュニティエディションの比較」を参照してください。

現在サポートされている Aurora MySQL リリースは 2.07.9、2.7.10、2.11.*、2.12.*、3.01.*、3.02.*、3.03.*、3.04.*、3.05.* です。

現在利用可能な Aurora MySQL バージョン 2 クラスターから Aurora MySQL バージョン 3.04.1 クラスターへのインプレースアップグレード、スナップショットの復元、または Amazon RDS ブルー/グリーンデプロイメントによるマネージドブルー/グリーンアップグレードの開始を行うことができます。

Aurora MySQL バージョン 3 へのアップグレードの計画については、「Amazon Aurora ユーザーガイド」の「Upgrade planning for Aurora MySQL version 3」を参照してください。Aurora MySQL のアップグレードに関する一般的な情報については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL DB クラスターのアップグレード」を参照してください。

トラブルシューティング情報については、「Aurora MySQL バージョン 3 のアップグレードに関する問題のトラブルシューティング」を参照してください。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

可用性の向上:

- パラレルクエリを使用する Aurora MySQL データベースインスタンスで、多数のパラレルクエリを同時に実行すると、データベースが再起動する問題を修正しました。
- 拡張バイナリログが有効になっているバイナリログ (binlog) レプリカクラスターで、いずれかのバイナリログソースがまたはに設定されていると、実行された GTID セットが誤って復元されることがある問題を修正しました。`gtid_mode` `ON` `ON_PERMISSIVE`この問題が原因で、レプリカクラスターのライターインスタンスが復元中にさらに再起動したり、実行された GTID セットを照会したときに誤った結果が返されたりする可能性があります。
- 拡張バイナリログが有効になっている場合に解放可能なメモリが減少し、Aurora MySQL データベースインスタンスが再起動またはフェイルオーバーする原因となる、メモリ管理の問題を修正しました。
- ライターインスタンスがデータベースボリュームを拡大させ、160 GB の倍数に達すると、リーダーインスタンスが再起動する問題を修正しました。
- 拡張バイナリログ機能が有効になっている Aurora MySQL データベースインスタンスが、バイナリログの復旧プロセスが実行されている場合に、起動中に停止してしまう問題を修正しました。
- `SHOW STATUS` ステートメントと `PURGE BINARY LOGS` ステートメントを同時に実行すると、デッドラッチが原因でデータベースインスタンスが再起動する可能性がある問題を修正しました。purge binary logs は、ユーザーが設定したバイナリログの保持期間に従って実行されるマネージドステートメントです。
- データベースが内部システムテーブルでトリガーを作成または削除しているときにライターインスタンスが再起動すると、データベースクラスターが使用できない状態になる問題を修正しました。
- Aurora レプリカのあるクラスターで拡張バイナリログ機能を使用する場合、セマフォの待機時間が長くなり、データベースインスタンスが再起動する可能性がある問題を修正しました。

全般的な機能強化:

- Aurora MySQL 3.04.0 で実行されている Aurora Serverless v2 データベースクラスターで拡張バイナリログが有効になっている場合に、データベースが使用できなくなることがある問題を修正しました。
- 拡張バイナリログ機能が有効になっている場合に Aurora Storage に書き込む前に、未使用のストレージメタデータを削除しました。その結果、ネットワーク上で転送されるバイト数が増えて書き

込み遅延が長くなり、データベースの再起動やフェイルオーバーが発生する特定のシナリオを回避できます。

- アップグレード時または移行時に Aurora 固有のパフォーマンススキーマテーブルが作成されない問題を修正しました。
- 拡張バイナリログが有効になっていると、`NumBinaryLogFiles` CloudWatch 上のメトリックスに誤った結果が表示されることがある問題を修正しました。

アップグレードと移行:

- MySQL 5.7 から MySQL 8.0 にアップグレードする際に、1 つのデータベースに多数のテーブルがある場合、サーバーがメモリを過剰に消費していました。テーブルをアップグレードできるかどうかを確認する過程で、`Table`すべてのデータディクショナリオブジェクトを事前に取得し、それぞれを処理して名前を取得してから、`CHECK TABLE ... FOR UPGRADE`リストに対して実行したことが判明しました。この場合、すべてのオブジェクトを事前に取得する必要はなく、そのせいでメモリ消費量に大きな影響が出ていました。この問題を解決するために、このような場合には、一度に 1 つずつ `Table` オブジェクトを取得し、必要なチェックをすべて実行して名前を取得し、オブジェクトを解放してから、次のオブジェクトに進むことにしました。(バグ #34526001)

MySQL Community Edition でのバグ修正の統合

このリリースには、以下を含め、8.0.28 までのコミュニティ版のバグ修正がすべて反映されています。詳細については、「MySQL 3.x データベースエンジンの更新で修正された MySQL のバグ」を参照してください。

- バックグラウンドの TLS 証明書のローテーションが原因で CPU 使用率が高くなる問題を修正しました (コミュニティのバグ修正 #34284186)。

Aurora MySQL データベースエンジンアップデート 2023-07-31 (バージョン 3.04.0、MySQL 8.0.28 に対応)

バージョン: 3.04.0

Aurora MySQL 3.04.0 は一般公開されています。Aurora MySQL 3.04 バージョンは MySQL 8.0.28 と互換性があり、Aurora MySQL 3.03 バージョンは MySQL 8.0.26 と互換性があり、Aurora MySQL 3.02 バージョンは MySQL 8.0.23 と互換性があります。8.0.23 から 8.0.28 で行われたコミュニティ版の変更点の詳細については、「MySQL 8.0 Release Notes」を参照してください。

> **Note**
>
> このバージョンは、長期サポート (LTS) リリースとして指定されています。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL 長期サポート (LTS) リリース」を参照してください。

Aurora MySQL バージョン 3 の新機能の詳細については、「Aurora MySQL バージョン 3 は MySQL 8.0 との互換性があります」を参照してください。Aurora MySQL バージョン 3 と Aurora MySQL バージョン 2 の違いについての詳細は、「Aurora MySQL バージョン 2 と Aurora MySQL バージョン 3 の比較」を参照してください。Aurora MySQL バージョン 3 と MySQL 8.0 Community Edition の比較については、「Aurora MySQL バージョン 3 と MySQL 8.0 コミュニティエディションの比較」を参照してください。

現在サポートされている Aurora MySQL リリースは、2.07.9、2.11.1、2.11.2、3.01.*、3.02.*、3.03.*、3.04.0 です。

現在サポートされている Aurora MySQL バージョン 2 クラスターから Aurora MySQL バージョン 3.04.0 クラスターへのインプレースアップグレード、スナップショットの復元、Amazon RDS ブルー/グリーンデプロイによるマネージドブルー/グリーンアップグレードの開始を行うことができます。

Aurora MySQL バージョン 3 へのアップグレードの計画については、「Amazon Aurora ユーザーガイド」の「Upgrade planning for Aurora MySQL version 3」を参照してください。Aurora MySQL のアップグレードに関する一般的な情報については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL DB クラスターのアップグレード」を参照してください。

トラブルシューティング情報については、「Aurora MySQL バージョン 3 のアップグレードに関する問題のトラブルシューティング」を参照してください。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> Aurora MySQL の拡張バイナリログ (binlog) は、現在、Aurora MySQL バージョン 3.04.0 の Aurora Serverless v2 データベースインスタンスではサポートされていません。この機能を有効にすると、データベースを使用できなくなる可能性があります。Aurora MySQL バー

ジョン 3.04.0 で拡張バイナリログを使用する必要がある場合は、非サーバーレスのデータベースインスタンスクラスを使用するか、Serverless v2 データベースインスタンスの ACU の最小値と最大値を同じ値に設定することをお勧めします。
Aurora MySQL の拡張バイナリログ記録の詳細については、「Aurora ユーザーガイド」を参照してください。

改良点

新機能:

- InnoDB フルテキストインデックスを使用して自然言語モードでフレーズを検索するクエリのパフォーマンスが向上しました。MySQL の全文検索の詳細については、「Full-Text Search Functions」を参照してください。
- Amazon Aurora MySQL はローカル (クラスター内) 書き込み転送をサポートしています。リーダー DB インスタンスから Aurora MySQL DB クラスター内のライター DB インスタンスに書き込み操作を転送できるようになりました。詳細については、「Amazon Aurora MySQL DB クラスターでのローカル書き込み転送の使用」を参照してください。
- `autocommit` が無効になっているセッションで Amazon Aurora Global Database の書き込み転送機能を使用する場合に、`aurora_replica_read_consistency` パラメータの値を変更する機能が追加されました。詳細については、「書き込み転送の設定パラメータ」を参照してください。
- Aurora MySQL 3.04 以降、グローバルデータベースの書き込み転送機能では、データベースクラスターとデータベースインスタンスのパラメータグループを使用して `aurora_replica_read_consistency` パラメータの値を設定できるようになりました。Aurora MySQL 3.04 より前のバージョンでは、このパラメータの値はセッションレベルでしか設定できませんでした。

修正されたセキュリティ問題と CVE:

- SSL/TLS プロバイダーが OpenSSL から AWS-LC に変更されました。これに伴い、以下を始めとする多数の変更が加えられました。
  - Aurora MySQL バージョン 3.04.0 から上位バージョンへのアップグレード時に、SSL を使用するデータベース接続を、ダウンタイムのない再起動とダウンタイムのないパッチ適用で復元できるようになりました。

  - TLSv1.3 に対応しました。併せて、TLS_AES_128_GCM_SHA256、TLS_AES_256_GCM_SHA384、TLS_CHACHA20_POLY1305_SH SSL の暗号がサポートされています。
  - 安全性の低い DHE-RSA-* 暗号のサポートは終了しました。

  詳細については、「Aurora MySQL DB クラスターでの TLS の使用」を参照してください。
- 動的権限 `SHOW_ROUTINE` を `rds_superuser_role` に付与しました。この権限があれば、ストアドプロシージャやファンクションなど、すべてのストアドルーチンの定義とプロパティにアクセスできます。詳細については、「SHOW_ROUTINE」を参照してください。
- 監査ログファイルのローテーション中に、監査ログでイベントが見落とされる場合がある問題を修正しました。
- 安全性とパフォーマンスに優れた Transport Layer Security (TLS) 1.3 プロトコルのサポートが有効になりました。TLS 1.2 バージョンとの互換性も確保されています。
- TLS バージョン TLSv1 と TLSv1.1 は、コミュニティ版の MySQL 8.0.26 で非推奨となった経緯から、Aurora MySQL 3.03 でも非推奨となりました。これらのプロトコルはコミュニティ版の MySQL 8.0.28 で削除され、それを受けて Aurora MySQL 3.04 でも削除されました。デフォルトでは、TLS 1.2 以降で通信できないセキュアクライアントは拒否されます。TLS を使用してデータベースインスタンスに接続する方法については、「Amazon Aurora MySQL でのセキュリティ」を参照してください。

このリリースには、次の CVE 修正が含まれています。

- 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

可用性の向上:

- 長時間のトランザクションリカバリ中にデータベースが再起動することがある問題を修正しました。
- データベースアクティビティストリームのイベント暗号化で、データベースが再起動されることがある問題を修正しました。
- 起動中または Aurora Serverless v2 でのスケーリング中に InnoDB バッファプールが初期化される際のメモリ不足エラーに起因する、メモリ管理の問題を修正しました。この問題により、データベースインスタンスが再起動したり、スループットの低下やレイテンシーの増加などのパフォーマンスが低下したりした可能性があります。
- Aurora MySQL parallel クエリ実行プランを使用するクエリの実行中に Aurora MySQL リーダーインスタンスが再起動する可能性がある問題を修正しました。
- 特定の状況で、範囲の推定中に Aurora リーダーインスタンスが再起動する可能性がある問題を修正しました。
- 自動インクリメント列を含む重い挿入操作の実行中に再起動が発生すると、起動時にデータベースの回復が中断されることがある問題を修正しました。
- Aurora の高度な監査で、サーバー変数 `server_audit_events` が `ALL` または `QUERY` に設定されていると、情報メッセージが Aurora MySQL エラーログに過剰に記録される問題を修正しました。この問題により、データベースインスタンスが再起動する可能性があります。
- parallel クエリが有効になっている場合、`INSERT`ステートメントのロールバック中にデータベースが再起動する可能性がある問題を修正しました。
- `EXPLAIN ANALYZEall select tables were optimized awayEXTRA`情報列内の出力を返したクエリに対してプロファイリングツールを実行すると、データベースインスタンスが再起動する問題を修正しました。詳細については、MySQL ドキュメントの「EXPLAIN Output Format」を参照してください。
- 転送された暗黙のコミットステートメントでエラーが発生したときに、グローバル書き込み転送を使用する Aurora グローバルデータベースのセカンダリリージョンリーダーインスタンスが再起動する可能性がある問題を修正しました。
- Aurora `SELECT FOR UPDATE` グローバルデータベースのセカンダリリージョンからのグローバル書き込み転送を使用してクエリを実行すると、Aurora グローバルデータベースのプライマリリージョンのライターインスタンスが再起動する可能性がある問題を修正しました。

全般的な機能強化:

- 新しいストアドプロシージャ mysql.rds_gtid_purged が追加され、ユーザーが GTID_PURGED システム変数を設定できるようになりました。詳細については、「mysql.rds_gtid_purged」を参照してください。
- 2 つの新しいストアドプロシージャ mysql.rds_start_replication_until と mysql.rds_start_replication_until_gtid が追加され、ユーザーがバイナリログのレプリケーションを停止する場所を設定できるようになりました。Aurora MySQL におけるバイナリログレプリケーションの停止場所の設定については、「mysql.rds_start_replication_until」を参照してください。
- Aurora MySQL のレプリケーション制御のストアドプロシージャが、autocommit (自動コミット) モードが無効になっているセッションから呼びされた場合に sql_log_bin 変数を変更できない問題を修正しました。
- データ制御言語 (DCL) ステートメント GRANT/REVOKE および CREATE/DROP/ALTER/RENAME USER の論理レプリケーションのサポートが追加されました。
- InnoDB の統計情報が古くならないように、問題を修正しました。古くなると、最適ではないクエリ実行プランが生成され、クエリの実行時間が長くなる可能性があります。
- 2 つの新しいシステムビュー information_schema.aurora_global_db_instance_status と information_schema.aurora_global_db_status が追加されました。これらのビューを使用して、Aurora MySQL グローバルデータベースクラスター内のプライマリリソースとセカンダリリソースのステータスとトポロジを表示できます。この 2 つのシステムビューの詳細については、「Aurora MySQL — 固有の information_schema テーブル」を参照してください。
- ワイルドカード文字をエスケープした SET ROLE ステートメントを実行した後で、名前にワイルドカード文字が含まれているデータベースにユーザーがアクセスできなくなる問題を修正しました。
- 監査ログのローテーション中に報告されたイベントが監査ログに書き込まれない場合がある問題を修正しました。
- TRIGGER の実行によって内部一時テーブルが作成されると、ライターデータベースインスタンスが再起動する可能性がある問題を修正しました。
- 新しいシステム変数 innodb_aurora_max_partitions_for_range が追加されました。永続的な統計情報が得られない場合は、このパラメータを使用して、パーティション分割テーブルの行数計算の実行時間を短縮できます。詳細については、ドキュメント「Aurora MySQL 設定パラメータ」を参照してください。
- パーティション分割テーブルを作成するときに、ユーザーが ROW_FORMAT を COMPRESSED に設定できてしまう問題を修正しました。テーブルは暗黙的に COMPACT 形式に変換され、Aurora MySQL は圧縮テーブルをサポートしていないことを通知する警告が表示されます。

- replica_parallel_type 変数を LOGICAL_CLOCK に設定し、replica_preserve_commit_order 変数を ON にすると、マルチスレッドのバイナリログレプリケーションが停止する可能性がある問題を修正しました。この問題は、500 MB を超えるトランザクションがソースで実行された場合に発生する可能性があります。
- グローバルデータベースの書き込み転送機能が有効になっている場合に、セカンダリリージョンのリーダーインスタンスの performance_schema 設定を変更すると、プライマリリージョンのライターインスタンスに変更が意図せず転送されることがある問題を修正しました。
- データページが Aurora ストレージファイルシステムから読み取られた後で、サーバーステータス変数 innodb_buffer_pool_reads が更新されないことがある問題を修正しました。
- Aurora I/O-Optimized クラスター設定を選択した場合、Aurora MySQL のパラレルクエリはサポートされません。詳細については、Amazon Aurora MySQL のパラレルクエリの「制限事項」を参照してください。
- パラレルクエリが有効になっている場合に、プライマリインデックスまたはセカンダリインデックスを活用する特定の SELECT クエリについて、クエリプランオプティマイザが非効率的な実行プランを選択する問題を修正しました。
- タイムゾーン定義が IANA 2023c バージョンにアップグレードされました。
- リレーログファイルへの書き込み時の競合を減らすために、バイナリログレプリカに対するファイル管理パフォーマンスの最適化を導入しました。
- ユーザーのワークロードに関係なく、RPO_LAG_IN_MILLISECONDSinformation_schema.aurora_global_db_statusAuroraGlob CloudWatch テーブルとメトリックの列が常にゼロと表示される問題を修正しました。
- 新しいパラメータ aurora_tmptable_enable_per_table_limit が追加されました。このパラメータを有効にすると、tmp_table_size TempTable 変数はストレージエンジンによって作成される個々のインメモリ内部テンポラリテーブルの最大サイズを定義します。詳細については、「内部 (黙示的) 一時テーブルのストレージエンジン」を参照してください。
- グローバルデータベース書き込み転送機能が有効になっている場合に、追加の接続が作成される問題を修正しました。この問題は、リーダーインスタンスの読み取り専用トランザクションが暗黙のコミットを誤ってライターに転送した場合に発生します。
- グローバルデータベース書き込み転送機能を使用した接続で、performance_schema.threads テーブルの PROCESSLIST_USER フィールドと PROCESSLIST_HOST フィールドがプライマリリージョンのライターに入力取り込まれない問題を修正しました。このテーブルとパフォーマンススキーマの詳細については、MySQL リファレンスマニュアルの「The threads Table」および「Amazon Aurora ユーザーガイド」の「Performance Schema の概要」を参照してください。

- グローバルデータベースの書き込み転送機能の使用時に、セカンダリリージョンのリーダーインスタンスで表示される CommitLatency Cloudwatch メトリクスの値が間違っていた問題を修正しました。セカンダリデータベースクラスターで転送された DML ステートメントのレイテンシーをモニタリングするには、ForwardingReplicaDMLLatency メトリクスと ForwardingWriterDMLLatency メトリクスを使用することをお勧めします。コミットレイテンシーは、プライマリリージョンのライターインスタンスの CommitLatency メトリクスを使用して確認することもできます。詳細については、Aurora ユーザーガイドの Amazon CloudWatch の書き込み転送メトリクスをご覧ください。
- replica_parallel_workers 変数に 0 より大きい値を設定してマルチスレッドのバイナリログレプリケーションを構成した場合に、バイナリログレプリケーションの管理と設定に使用される Aurora MySQL のレプリケーション制御のストアドプロシージャが誤ってエラーを報告する問題を修正しました。

アップグレードと移行:

- Aurora グローバルデータベースに対して、Aurora MySQL バージョン 3.01、3.02、または 3.03 から Aurora MySQL バージョン 3.04 以降へのマイナーバージョンアップグレードを実行するには、「エンジンのバージョンを変更して Aurora MySQL をアップグレードする」を参照してください。
- Aurora MySQL 2 から Aurora MySQL 3 へのアップグレード時に、mysql.general_log_backup、mysql.general_log、mysql.slow_log_backup、mysql.sl テーブルに対してスキーマ不整合エラーが報告され、アップグレードの事前チェックが失敗する可能性がある問題を修正しました。アップグレードのトラブルシューティングの詳細については、「Aurora MySQL バージョン 3 のアップグレードに関する問題のトラブルシューティング」を参照してください。
- トリガーの定義に含まれている予約済みのキーワードが引用符で囲まれていない場合に、Aurora MySQL バージョン 3 へのメジャーバージョンアップグレードが失敗する可能性がある問題を修正しました。

MySQL Community Edition でのバグ修正の統合

このリリースには、以下を含め、8.0.28 までのコミュニティ版のバグ修正がすべて反映されています。詳細については、「MySQL 3.x データベースエンジンの更新で修正された MySQL のバグ」を参照してください。

- intrinsic の一時テーブルページを含むバッファブロックがページトラバーサル中に再配置され、アサーションエラーが発生する問題を修正しました (バグ# 33715694)。
- InnoDB: オンライン DDL out-of-bounds 操作がメモリにアクセスしないようにします (バグ番号 34750489、バグ番号 108925)
- 複数の共通テーブル式 (CTE) がネスト構造になっている複雑な SQL ステートメントの処理中に、誤ったクエリ結果が生成されることがある問題を修正しました (バグ #34572040、バグ #34634469、バグ #33856374)。

Aurora MySQL データベースエンジンの更新 2023-12-08 (バージョン 3.03.3、MySQL 8.0.26 互換)

バージョン: 3.03.3

Aurora MySQL 3.03.3 は一般利用可能です。Aurora MySQL 3.03 バージョンは、MySQL 8.0.26 と互換性があります。8.0.23 から 8.0.28 で行われたコミュニティ版の変更点の詳細については、「MySQL 8.0 Release Notes」を参照してください。

Aurora MySQL バージョン 3 の新機能の詳細については、「Aurora MySQL バージョン 3 は MySQL 8.0 との互換性があります」を参照してください。Aurora MySQL バージョン 3 と Aurora MySQL バージョン 2 の違いについての詳細は、「Aurora MySQL バージョン 2 と Aurora MySQL バージョン 3 の比較」を参照してください。Aurora MySQL バージョン 3 と MySQL 8.0 Community Edition の比較については、「Aurora MySQL バージョン 3 と MySQL 8.0 コミュニティエディションの比較」を参照してください。

現在利用可能な Aurora MySQL リリースは、2.07.9、2.07.10、2.11.*、2.12.*、3.01.*、3.02.*、3.03.*、3.04.*、3.05.* です。

現在利用可能な Aurora MySQL バージョン 2 クラスターから Aurora MySQL バージョン 3.03.3 クラスターへのインプレースアップグレード、スナップショットの復元、Amazon RDS ブルー/グリーンデプロイによるマネージドブルー/グリーンアップグレードの開始を行うことができます。

Aurora MySQL バージョン 3 へのアップグレードの計画については、「Amazon Aurora ユーザーガイド」の「Upgrade planning for Aurora MySQL version 3」を参照してください。Aurora MySQL のアップグレードに関する一般的な情報については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL DB クラスターのアップグレード」を参照してください。

トラブルシューティング情報については、「Aurora MySQL バージョン 3 のアップグレードに関する問題のトラブルシューティング」を参照してください。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

以下のセキュリティの問題と CVE の修正:

マネージド型の環境での処理を微調整するための修正およびその他の機能強化。以下の CVE の追加の修正:

- CVE-2023-38545

可用性の向上:

- パラレルクエリを使用する Aurora MySQL データベースインスタンスで、多数のパラレルクエリを同時に実行すると、データベースが再起動する問題を修正しました。
- 拡張バイナリログ (binlog) が有効になっているバイナリログレプリカクラスターで、いずれかのバイナリログソースで `gtid_mode` が `ON` または `ON_PERMISSIVE` に設定されている場合に、実行された GTID セットが誤って復元されることがある問題を修正しました。この問題が原因で、レプリカクラスターのライターインスタンスが復元中にさらに再起動したり、実行された GTID セットを照会したときに誤った結果が返されたりする可能性があります。
- 拡張バイナリログが有効になっている場合に解放可能なメモリが減少し、Aurora MySQL データベースインスタンスが再起動またはフェイルオーバーする原因となる、メモリ管理の問題を修正しました。
- ライターインスタンスがデータベースボリュームを拡大させ、160 GB の倍数に達すると、リーダーインスタンスが再起動する問題を修正しました。
- 拡張バイナリログ機能が有効になっている Aurora MySQL データベースインスタンスが、バイナリログの復旧プロセスが実行されている場合に、起動中に停止してしまう問題を修正しました。
- ダウンタイムのないパッチ適用中にインスタンスが再起動し、データベース接続が予期せず切断される問題を修正しました。
- `SHOW STATUS` ステートメントと `PURGE BINARY LOGS` ステートメントを同時に実行すると、デッドラッチが原因でデータベースインスタンスが再起動する可能性がある問題を修正しました。purge binary logs は、ユーザーが設定したバイナリログの保持期間に従って実行されるマネージドステートメントです。

- Aurora レプリカのあるクラスターで拡張バイナリログ機能を使用する場合、セマフォの待機時間が長くなり、データベースインスタンスが再起動する可能性がある問題を修正しました。

全般的な機能強化:

- 拡張バイナリログ機能が有効になっている場合は、Aurora ストレージへの書き込み前に、未使用のストレージメタデータが削除されます。その結果、ネットワーク上で転送されるバイト数が増えて書き込み遅延が長くなり、データベースの再起動やフェイルオーバーが発生する特定のシナリオを回避できます。
- 拡張バイナリログが有効になっていると、`NumBinaryLogFiles` CloudWatch 指標に誤った結果が表示されることがある問題を修正しました。
- `table_open_cache` データベースパラメータの変更が、データベースインスタンスを再起動するまで有効にならないことがある問題を修正しました。
- 接続先のバイナリログ (binlog) コンシューマーが、重複するバイナリログレプリケーションサーバー ID を使用している場合に、データベースが再起動する問題を修正しました。

MySQL Community Edition でのバグ修正の統合

このリリースには、以下を含め、8.0.26 までのコミュニティ版のバグ修正がすべて反映されています。詳細については、「MySQL 3.x データベースエンジンの更新で修正された MySQL のバグ」を参照してください。

- バックグラウンドの TLS 証明書のローテーションが原因で CPU 使用率が高くなる問題を修正しました (コミュニティのバグ修正 #34284186)。

Aurora MySQL データベースエンジンの更新 2023-08-29 (バージョン 3.03.2、MySQL 8.0.26 互換)

バージョン: 3.03.2

Aurora MySQL 3.03.2 は一般公開されています。Aurora MySQL 3.04 バージョンは MySQL 8.0.28 と互換性があり、Aurora MySQL 3.03 バージョンは MySQL 8.0.26 と互換性があり、Aurora MySQL 3.02 バージョンは MySQL 8.0.23 と互換性があります。8.0.23 から 8.0.28 で行われたコミュニティ版の変更点の詳細については、「MySQL 8.0 Release Notes」を参照してください。

Aurora MySQL バージョン 3 の新機能の詳細については、「Aurora MySQL バージョン 3 は MySQL 8.0 との互換性があります」を参照してください。Aurora MySQL バージョン 3 と Aurora MySQL バージョン 2 の違いについての詳細は、「Aurora MySQL バージョン 2 と Aurora MySQL バージョン 3 の比較」を参照してください。Aurora MySQL バージョン 3 と MySQL 8.0 Community Edition の比較については、「Aurora MySQL バージョン 3 と MySQL 8.0 コミュニティエディションの比較」を参照してください。

現在入手可能な Aurora MySQL リリースは、2.07.9、2.07.10、2.11.*、3.01.*、3.02.*、3.03.*、3.04.* です。

現在入手可能な Aurora MySQL バージョン 2 クラスターから Aurora MySQL バージョン 3.03.2 クラスターへのインプレースアップグレード、スナップショットの復元、Amazon RDS ブルー/グリーンデプロイによるマネージドブルー/グリーンアップグレードの開始を行うことができます。

Aurora MySQL バージョン 3 へのアップグレードの計画については、「Amazon Aurora ユーザーガイド」の「Upgrade planning for Aurora MySQL version 3」を参照してください。Aurora MySQL のアップグレードに関する一般的な情報については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL DB クラスターのアップグレード」を参照してください。

トラブルシューティング情報については、「Aurora MySQL バージョン 3 のアップグレードに関する問題のトラブルシューティング」を参照してください。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

修正されたセキュリティ問題と CVE:

- 監査ログファイルのローテーション中に監査ログにイベントが見落とされることがある問題を修正しました。

このリリースには、次の CVE 修正が含まれています。

- CVE-2023-21963
- CVE-2023-21912
- CVE-2023-0215

- CVE-2022-43551
- CVE-2022-37434

可用性の向上:

- 長時間のトランザクションリカバリ中にデータベースが再起動することがある問題を修正しました。
- データベースが内部システムテーブルでトリガーを作成または削除しているときにライターインスタンスを再起動すると、データベースクラスターが使用できなくなる問題を修正しました。
- 集計関数を参照するクエリの実行中にデータベースインスタンスが再起動することがある問題を修正しました。
- parallel クエリが有効になっている場合、INSERTステートメントのロールバック中にデータベースが再起動する可能性がある問題を修正しました。
- 高速挿入は、Aurora MySQL バージョン 3.03.2 以降の通常の InnoDB テーブルでのみ有効です。この最適化は、InnoDB 一時テーブルでは機能しません。高速挿入最適化の詳細については、「Amazon Aurora MySQL のパフォーマンス強化」を参照してください。

全般的な機能強化:

- リーダーインスタンスがテーブルを開くことができず、ERROR 1146 が発生する問題を修正しました。この問題は、ライターインスタンスで INPLACE アルゴリズムが使用されている間に、特定の種類のオンラインデータ定義言語 (DDL) を実行すると発生します。
- リレーログファイルへの書き込み時の競合を減らすために、バイナリログレプリカに対するファイル管理パフォーマンスの最適化を導入しました。
- パラレルクエリが有効になっている場合に、プライマリインデックスまたはセカンダリインデックスを活用する特定の SELECT クエリについて、クエリプランオプティマイザが非効率的な実行プランを選択する問題を修正しました。
- データ制御言語 (DCL) ステートメント GRANT/REVOKE および CREATE/DROP/ALTER/RENAME USER の論理レプリケーションのサポートが追加されました。
- Aurora I/O-Optimized クラスター設定を選択した場合、Amazon Aurora MySQL のパラレルクエリはサポートされません。詳細については、Aurora MySQL のパラレルクエリの「制限事項」を参照してください。

アップグレードと移行:

- Aurora グローバルデータベースに対して、Aurora MySQL バージョン 3.01 または 3.02 から Aurora MySQL バージョン 3.03 以降へのマイナーバージョンアップグレードを実行するには、「エンジンのバージョンを変更して Aurora MySQL をアップグレードする」を参照してください。
- トリガーの定義に含まれている予約済みのキーワードが引用符で囲まれていない場合に、Aurora MySQL バージョン 3 へのメジャーバージョンアップグレードが失敗する可能性がある問題を修正しました。

MySQL Community Edition でのバグ修正の統合

このリリースには、以下を含め、8.0.26 までのコミュニティ版のバグ修正がすべて反映されています。詳細については、「MySQL 3.x データベースエンジンの更新で修正された MySQL のバグ」を参照してください。

- 複数の共通テーブル式 (CTE) がネスト構造になっている複雑な SQL ステートメントの処理中に、誤ったクエリ結果が生成されることがある問題を修正しました。(バグ #34572040、バグ #34634469、バグ #33856374)
- InnoDB: 同じテーブルの統計を初期化解除および初期化しようとするスレッド間で競合状態になり、アサーション障害が発生していました。(バグ #33135425)
- InnoDB: オンライン DDL 操作がメモリにアクセスできないようにします out-of-bounds 。(バグ #34750489、バグ #108925)

Aurora MySQL データベースエンジンの更新 2023-05-11 (バージョン 3.03.1、MySQL 8.0.26 互換)

バージョン: 3.03.1

Aurora MySQL 3.03.1 は一般公開されています。Aurora MySQL 3.03 バージョンは MySQL 8.0.26 と互換性があり、Aurora MySQL 3.02 バージョンは MySQL 8.0.23 と互換性があります。8.0.23 から 8.0.26 で行われたコミュニティ版の変更点の詳細については、「MySQL 8.0 Release Notes」を参照してください。

Aurora MySQL バージョン 3 の新機能の詳細については、「Aurora MySQL バージョン 3 は MySQL 8.0 との互換性があります」を参照してください。Aurora MySQL バージョン 3 と Aurora MySQL バージョン 2 の違いについての詳細は、「Aurora MySQL バージョン 2 と Aurora MySQL バージョン 3 の比較」を参照してください。Aurora MySQL バージョン 3 と MySQL 8.0 Community Edition

の比較については、「Aurora MySQL バージョン 3 と MySQL 8.0 コミュニティエディションの比較」を参照してください。

現在サポートされている Aurora MySQL リリース
は、2.07.9、2.11.1、2.11.2、3.01.*、3.02.*、3.03.* です。

現在サポートされている Aurora MySQL バージョン 2 クラスターから Aurora MySQL 3.03.1 へのインプレースアップグレードまたはスナップショットの復元を実行できます。

Aurora MySQL バージョン 3 へのアップグレードの計画については、「Amazon Aurora ユーザーガイド」の「Upgrade planning for Aurora MySQL version 3」を参照してください。Aurora MySQL のアップグレードに関する一般的な情報については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL DB クラスターのアップグレード」を参照してください。

トラブルシューティング情報については、「Aurora MySQL バージョン 3 のアップグレードに関する問題のトラブルシューティング」を参照してください。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

新機能:

- Aurora I/O Optimized ストレージ設定は、バージョン 3.03.1 以降で使用できます。詳細については、「Amazon Aurora DB クラスターのストレージ設定」を参照してください。
- 新しいシステム変数 `innodb_aurora_max_partitions_for_range` が追加されました。永続的な統計情報が得られない場合は、このパラメータを使用して、パーティション分割テーブルの行数計算の実行時間を短縮できます。詳細については、ドキュメント「Aurora MySQL 設定パラメータ」を参照してください。

可用性の向上:

- トランザクションのコミット直後に接続を切断した場合に、無効なメモリに誤ってアクセスした結果、データベースインスタンスが再起動する可能性がある問題を修正しました。
- Aurora の高度な監査で、サーバー変数 `server_audit_events` が ALL または QUERY に設定されていると、情報メッセージが Aurora MySQL エラーログに過剰に記録される問題を修正しました。この問題が原因で、データベースインスタンスが再起動する可能性があります。

- 特定の状況下で、アクセスできなくなったページの読み取りを範囲の推定中に試みた場合に、Aurora リーダーインスタンスが再起動する可能性がある問題を修正しました。
- Aurora パラレルクエリの実行プランを利用するクエリの実行中に、Aurora MySQL リーダーインスタンスが再起動する問題を修正しました。
- バイナリログレプリケーションを使用するデータベースインスタンスで、バイナリログレプリケーションのコンシューマーが複数アタッチされていると CPU 使用率が増加し、接続障害が発生する可能性がある問題を修正しました。
- 中間一時テーブルを実体化 (マテリアライズ) する際に、共通テーブル式 (CTE) に対してサポートされていないインデックススキャンのアクセス手法が検討され、データベースの再起動や誤ったクエリ結果など、望ましくない挙動が生じる可能性がある問題を修正しました。 TempTable ストレージエンジンを使用するテーブルで、このようなサポートされていないインデックススキャンアクセス方法を使用しないようにすることで、この問題を解決しています。
- この Aurora MySQL バージョンでは高速挿入が有効になっていません。、、などのクエリの実行時に不整合が生じる可能性があるためです。`INSERT INTO SELECT FROM`高速挿入最適化の詳細については、「Amazon Aurora MySQL のパフォーマンス強化」を参照してください。

全般的な機能強化:

- SHOW BINARY LOGS ステートメントの実行時間が予想よりも長くなることがある問題を修正しました。これに伴い、データベースのコミットスループットが低下する可能性があります。
- Instant ADD COLUMN 機能を使用して列を追加したユーザーテーブルの並列エクスポートが失敗する可能性がある問題を修正しました。
- 監査ログのローテーション中に報告されたイベントが監査ログに書き込まれない場合がある問題を修正しました。
- INFORMATION_SCHEMA INNODB_TABLESPACES テーブルに対してクエリを実行すると、使用可能なメモリが枯渇する可能性がある問題を修正しました。
- パーティション分割テーブルを作成するときに、ユーザーが ROW_FORMAT を COMPRESSED に設定できてしまう問題を修正しました。テーブルは暗黙的に COMPACT 形式に変換され、Aurora MySQL は圧縮テーブルをサポートしていないことを通知する警告が表示されます。

アップグレードと移行:

- Aurora グローバルデータベースに対して、Aurora MySQL バージョン 3.01 または 3.02 から Aurora MySQL バージョン 3.03 以降へのマイナーバージョンアップグレードを実行するには、

「エンジンのバージョンを変更して Aurora MySQL をアップグレードする」を参照してください。

- Aurora MySQL 2 から Aurora MySQL 3 へのアップグレード時に、mysql.general_log_backup、mysql.general_log、mysql.slow_log_backup、mysql.sl テーブルに対してスキーマ不整合エラーが報告され、アップグレードの事前チェックが失敗する可能性がある問題を修正しました。アップグレードのトラブルシューティングの詳細については、「Aurora MySQL バージョン 3 のアップグレードに関する問題のトラブルシューティング」を参照してください。

MySQL Community Edition でのバグ修正の統合

このリリースには、以下を含め、8.0.26 までのコミュニティ版のバグ修正がすべて反映されています。詳細については、「MySQL 3.x データベースエンジンの更新で修正された MySQL のバグ」を参照してください。

- intrinsic の一時テーブルページを含むバッファブロックがページトラバーサル中に再配置され、アサーションエラーが発生する問題を修正しました。(バグ #33715694)

Aurora MySQL データベースエンジンの更新 2023-03-01 (バージョン 3.03.0、MySQL 8.0.26 互換) このバージョンへのアップグレードはサポートされていません。

バージョン: 3.03.0

Aurora MySQL 3.03.0 は一般公開されています。Aurora MySQL 3.03 バージョンは MySQL 8.0.26 と互換性があり、Aurora MySQL 3.02 バージョンは MySQL 8.0.23 と互換性があります。8.0.23 から 8.0.26 で行われたコミュニティ版の変更点の詳細については、「MySQL 8.0 Release Notes」を参照してください。

Aurora MySQL バージョン 3 の新機能の詳細については、「Aurora MySQL バージョン 3 は MySQL 8.0 との互換性があります」を参照してください。Aurora MySQL バージョン 3 と Aurora MySQL バージョン 2 の違いについての詳細は、「Aurora MySQL バージョン 2 と Aurora MySQL バージョン 3 の比較」を参照してください。Aurora MySQL バージョン 3 と MySQL 8.0 Community Edition の比較については、「Aurora MySQL バージョン 3 と MySQL 8.0 コミュニティエディションの比較」を参照してください。

現在サポートされている Aurora MySQL リリースは、2.07.*、2.11.*、3.01.*、3.02.*、3.03.* です。

現在サポートされている Aurora MySQL バージョン 2 クラスターから Aurora MySQL 3.03.0 へのインプレースアップグレードまたはスナップショットの復元を実行できます。

Aurora MySQL バージョン 3 へのアップグレードの計画については、「Amazon Aurora ユーザーガイド」の「Upgrade planning for Aurora MySQL version 3」を参照してください。Aurora MySQL のアップグレードに関する一般的な情報については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL DB クラスターのアップグレード」を参照してください。

トラブルシューティング情報については、「Aurora MySQL バージョン 3 のアップグレードに関する問題のトラブルシューティング」を参照してください。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

以下のセキュリティの問題と CVE の修正:

マネージド型の環境での処理を微調整するための修正およびその他の機能強化。以下の CVE の追加の修正:

- CVE-2022-32221
- CVE-2022-21451
- CVE-2022-21444

可用性の向上:

- バッファプールの初期化に予想以上に時間がかかり、大きな DB インスタンスクラスの再起動時に問題が発生する可能性がありました。この問題を修正しました。
- バイナリログ記録が有効になっていると、データベースの復旧中に DB インスタンスが再起動する可能性がある問題を修正しました。
- GRANT や REVOKE などのデータ制御言語 (DCL) ステートメントの実行中や、ライターインスタンスで新しい接続を確立している間に、リーダーインスタンスで接続障害が発生する可能性がある問題を修正しました。
- DELETE、UPDATE ステートメントなどのデータ操作言語 (DML) オペレーションにパラレルクエリが誤って使用され (現在サポートされていません)、DB インスタンスが再起動する問題を修正しま

した。パラレルクエリでサポートされているオペレーションの詳細については、Aurora MySQL パラレルクエリの「制限事項」を参照してください。

- 大規模な更新オペレーションまたはデータ定義言語 (DDL) ワークロードをライターインスタンスで、同じテーブルセットに対する読み取りオペレーションを Aurora レプリカで同時に実行すると、まれに Aurora レプリカが再起動することがある問題を修正しました。
- Aurora Serverless v2 リーダーインスタンスのスケールダウン操作時に、リーダーインスタンスが再起動する場合や、まれにデータの不整合が生じる場合がある問題を修正しました。
- DB インスタンスへの接続が切断されたときに、無効なメモリロケーションに誤ってアクセスした結果、DB インスタンスが再起動する可能性がある問題を修正しました。
- `GROUP BY` 句を使用し、DECIMAL 列の小数点以下を切り捨てるクエリを処理する際に、まれに DB インスタンスが再起動する可能性がある問題を修正しました。
- 空間インデックスを使用して範囲クエリを実行する際に、レコードに誤ってアクセスした結果、DB インスタンスが再起動する可能性がある問題を修正しました。
- デフォルトまたは顧客設定のメモリまたは mmap の値を内部一時テーブルが超えた場合に、Aurora MySQL レプリカインスタンスで DB インスタンスが再起動する可能性がある問題を修正しました。
- 高度な監査ログのローテーションが原因でメモリ管理の問題が発生する場合がありました。この問題を修正しました。
- この Aurora MySQL バージョンでは高速挿入が有効になっていません。、、などのクエリの実行時に不整合が生じる可能性があるためです。`INSERT INTO SELECT FROM`高速挿入最適化の詳細については、「Amazon Aurora MySQL のパフォーマンス強化」を参照してください。

全般的な機能強化:

- 読み込み整合性が GLOBAL に設定されたグローバルデータベース書き込み転送セッションの読み取りクエリのレイテンシーが改善されました。
- クライアントセッションで `reset_connection` または `change_user` コマンドが実行された後で、`wait_timeout` パラメータの値が適用されない問題を修正しました。
- DB インスタンスへの接続中に、そのインスタンスへの着信接続が急増している場合、アプリケーションでレイテンシーが増大する可能性がある問題を修正しました。Aurora MySQL DB インスタンスの接続確立遅延のトラブルシューティングに役立つ`AuroraSlowConnectionHandleCount`、`AuroraSlowHandshakeCount`と 2 CloudWatch つの新しいメトリックスが導入されました。これらのメトリクスの詳細については、Aurora メトリ

クス定義ドキュメント「Amazon Aurora CloudWatch の Amazon CloudWatch メトリックス」を参照してください。

- `temptable_use_mmap` パラメータは非推奨であり、今後の MySQL リリースではサポート対象外になる予定です。詳細については、「内部 (黙示的) 一時テーブルのストレージエンジン」を参照してください。
- `SHOW BINARY LOGS` ステートメントの実行時間が予想よりも長くなることがある問題を修正しました。これに伴い、データベースのコミットスループットが低下する可能性があります。

アップグレードと移行:

- Aurora グローバルデータベースに対して、Aurora MySQL バージョン 3.01 または 3.02 から Aurora MySQL バージョン 3.03 以降へのマイナーバージョンアップグレードを実行するには、「エンジンのバージョンを変更して Aurora MySQL をアップグレードする」を参照してください。
- クラスターに多数のテーブル (750K 超) がある場合に、Aurora MySQL バージョン 2 から Aurora MySQL バージョン 3 へのメジャーバージョンアップグレードが失敗する可能性がある問題を修正しました。
- `mysql.innodb_table_stats` テーブルと `mysql.innodb_index_stats` テーブルの移行に予想以上に時間がかかり、Aurora MySQL バージョン 2 から Aurora MySQL バージョン 3 へのメジャーバージョンアップグレードが失敗する可能性がある問題を修正しました。この問題の影響があったのは主に、数百万のテーブルがある DB クラスターです。
- スキーマの不整合エラーが原因で、Aurora MySQL バージョン 2 から Aurora MySQL バージョン 3 へのアップグレードが失敗する可能性がある問題を修正しました。これらのエラーは、`mysql.general_log_template` および `mysql.slow_log_template` テーブルに対するアップグレードの事前チェックで報告されます。アップグレードのトラブルシューティングの詳細については、「Aurora MySQL バージョン 3 のアップグレードに関する問題のトラブルシューティング」を参照してください。
- `schemaInconsistencyCheck` エラーが原因で Aurora MySQL バージョン 2 から Aurora MySQL バージョン 3 へのアップグレードが失敗する可能性がある問題を修正しました。このエラーは、`upgrade-prechecks.log` で報告されているように、`mysql.table_migration_index_info` テーブル内のスキーマの不整合が原因です。Aurora MySQL バージョン 3 へのアップグレードのトラブルシューティングの詳細については、「Aurora MySQL バージョン 3 のアップグレードに関する問題のトラブルシューティング」を参照してください。

MySQL Community Edition でのバグ修正の統合

このリリースには、以下を含め、8.0.26 までのコミュニティ版のバグ修正がすべて反映されています。詳細については、「MySQL 3.x データベースエンジンの更新で修正された MySQL のバグ」を参照してください。

- JSON や TEXT など、一部の列タイプのソート時に、ソートバッファのサイズがソートの最大行の 15 倍未満の場合、バッファが枯渇してしまう場合がある問題を修正しました。現在は、ソートバッファのサイズは最大のソートキーの 15 倍あれば十分です。(バグ #103325、バグ #105532、バグ #32738705、バグ #33501541)
- InnoDB がテーブルパーティションの一部の有効な名前を正しく処理しない場合がある問題を修正しました。(バグ #32208630)
- 特定の条件下で、OR 条件を使用したクエリを実行した場合に nullability プロパティの計算が不正確になり、誤った結果が返される可能性がある問題を修正しました。(バグ #34060289)
- 次の 2 つの条件が当てはまる場合に、特定の条件下で誤った結果が返される可能性がある問題を修正しました。
  - 派生テーブルが外側のクエリブロックにマージされている。
  - クエリに左結合と IN サブクエリが含まれている。

  (バグ #34060289)
- 整数列の最大値を超えた場合に、不正な AUTO_INCREMENT 値が生成されていた問題を修正しました。このエラーは、列の最大値が考慮されていないことが原因でした。この場合、以前の有効な AUTO_INCREMENT 値が返され、重複キーエラーが発生すべきでした。(バグ #87926、バグ #26906787)
- パフォーマンススキーマの DROP 権限を取り消すことができなかった問題を修正しました。(バグ #33578113)
- IF ステートメントで EXISTS を使用したストアドプロシージャが、処理対象の 1 つ以上のテーブルが実行の合間に削除され、再作成された場合に、最初の呼び出し以降の呼び出しで正しく実行されない問題を修正しました。(バグ #32855634)
- サブクエリのビューと外部クエリブロックを参照するクエリが、予期しない再起動を引き起こす可能性がある問題を修正しました。(バグ #32324234)

Aurora MySQL データベースエンジンの更新 2023-04-17 (バージョン 3.02.3、MySQL 8.0.23 互換) 標準サポートは 2024 年 1 月 15 日に終了。

バージョン: 3.02.3

Aurora MySQL 3.02.3 は一般公開されています。Aurora MySQL 3.02 バージョンは MySQL 8.0.23 と互換性があり、Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があります。

Aurora MySQL バージョン 3 の新機能や、Aurora MySQL バージョン 3 と Aurora MySQL バージョン 2 またはコミュニティ版 MySQL 8.0 との違いについては、「Amazon Aurora ユーザーガイド」の「Aurora MySQL バージョン 2 と Aurora MySQL バージョン 3 の比較」を参照してください。

現在サポートされている Aurora MySQL リリース
は、2.07.*、2.11.1、2.11.2、3.01.*、3.02.*、3.03.* です。

現在サポートされている Aurora MySQL バージョン 2 クラスターから Aurora MySQL 3.02.3 へのインプレースアップグレードまたはスナップショットの復元を実行できます。

Aurora MySQL バージョン 3 へのアップグレードの計画については、「Amazon Aurora ユーザーガイド」の「Upgrade planning for Aurora MySQL version 3」を参照してください。アップグレード手順自体については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL バージョン 3 へのアップグレード」を参照してください。Aurora MySQL のアップグレードに関する一般的な情報については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL DB クラスターのアップグレード」を参照してください。

トラブルシューティング情報については、「Aurora MySQL バージョン 3 のアップグレードに関する問題のトラブルシューティング」を参照してください。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

可用性の向上:

- トランザクションのコミット直後に接続を切断した場合に、無効なメモリに誤ってアクセスした結果、データベースインスタンスが再起動する可能性がある問題を修正しました。

- この Aurora MySQL バージョンでは高速挿入が有効になっていません。、、などのクエリの実行時に不整合が生じる可能性があるためです。INSERT INTO SELECT FROM高速挿入最適化の詳細については、「Amazon Aurora MySQL のパフォーマンス強化」を参照してください。

全般的な機能強化:

- 中間一時テーブルを実体化 (マテリアライズ) する際に、共通テーブル式 (CTE) に対してサポートされていないインデックススキャンのアクセス手法が検討され、データベースの再起動や誤ったクエリ結果など、望ましくない挙動が生じる可能性がある問題を修正しました。この問題は、ストレージエンジンを使用するテーブルで、このようなサポートされていないインデックススキャンアクセス方法を使用しないようにすることで修正されました。 TempTable
- Aurora MySQL ライターインスタンスで大規模な更新オペレーションまたはデータ定義言語 (DDL) オペレーションを同時に実行しているテーブルにアクセスした場合に、まれに Aurora MySQL リーダーインスタンスが再起動することがある問題を修正しました。
- 特定の状況下で、アクセスできなくなったページの読み取りを範囲の推定中に試みた場合に、Aurora MySQL リーダーインスタンスが再起動する可能性がある問題を修正しました。
- バイナリログレプリケーションを使用するデータベースインスタンスで、バイナリログレプリケーションのコンシューマーが複数アタッチされていると CPU 使用率が増加し、接続障害が発生する可能性がある問題を修正しました。
- Aurora パラレルクエリの実行プランを利用するクエリの実行中に、Aurora MySQL リーダーインスタンスが再起動する問題を修正しました。

Aurora MySQL データベースエンジンの更新 2022-11-18 (バージョン 3.02.2、MySQL 8.0.23 互換) 標準サポートは 2024 年 1 月 15 日に終了。

バージョン: 3.02.2

Aurora MySQL 3.02.2 は一般公開されています。Aurora MySQL 3.02 バージョンは MySQL 8.0.23 と互換性があり、Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

Aurora MySQL バージョン 3 の新機能や、Aurora MySQL バージョン 3 と Aurora MySQL バージョン 2 またはコミュニティ版 MySQL 8.0 との違いについては、「Amazon Aurora ユーザーガイド」の「Aurora MySQL バージョン 2 と Aurora MySQL バージョン 3 の比較」を参照してください。

現在サポートされている Aurora MySQL リリース
は、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

現在サポートされている Aurora MySQL バージョン 2 クラスターから取得したスナップショットを Aurora MySQL 3.02.2 で復元できます。

Aurora MySQL バージョン 3 へのアップグレードの計画については、「Amazon Aurora ユーザーガイド」の「Upgrade planning for Aurora MySQL version 3」を参照してください。アップグレード手順自体については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL バージョン 3 へのアップグレード」を参照してください。Aurora MySQL のアップグレードに関する一般的な情報については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL DB クラスターのアップグレード」を参照してください。

トラブルシューティング情報については、「Aurora MySQL バージョン 3 のアップグレードに関する問題のトラブルシューティング」を参照してください。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

Aurora MySQL バージョン 3.02.2 は一般公開されています。コミュニティ版 MySQL 8.0.23 と互換性があります。

以下のセキュリティの問題と CVE の修正:

マネージド型の環境での処理を微調整するための修正およびその他の機能強化。以下の CVE の追加の修正:

- CVE-2022-21451
- CVE-2021-36222
- CVE-2021-22926

- CVE-2022-21444

可用性の向上:

- データベースインスタンスへの接続が明示的または暗黙的に切断されたときに、無効なメモリに誤ってアクセスした結果、データベースインスタンスが再起動する可能性がある問題を修正しました。
- バッファプールの初期化に予想以上に時間がかかり、比較的大きなインスタンスクラスでデータベースの起動が繰り返し中断される可能性がある問題を修正しました。
- Aurora Serverless v2 がスケーリング中に誤ってテーブルキャッシュの更新を試みた場合に、まれにデータベースインスタンスが再起動する可能性がある問題を修正しました。
- GROUP BY 句を使用し、DECIMAL 列の小数点以下を切り捨てるクエリを処理する際に、まれにデータベースが再起動する可能性がある問題を修正しました。
- この Aurora MySQL バージョンでは高速挿入が有効になっていません。、、などのクエリの実行時に不整合が生じる可能性があるためです。`INSERT INTO SELECT FROM`高速挿入最適化の詳細については、「Amazon Aurora MySQL のパフォーマンス強化」を参照してください。

全般的な機能強化:

- mysql.host テーブルのメタデータの不整合が原因で、Aurora MySQL バージョン 2 (MySQL 5.7 互換) から Aurora MySQL バージョン 3 (MySQL 8.0 互換) にアップグレードできない場合がある問題を修正しました。
- パフォーマンスが向上し、Aurora MySQL バージョン 2 (MySQL 5.7 互換) から Aurora MySQL バージョン 3 (MySQL 8.0 互換) へのアップグレードの所要時間が短くなりました。一部のアップグレード手順を並行処理することで、db.r6g.16xlarge や db.r5.24xlarge などの大きいインスタンスクラスを使用している場合はさらに時間短縮になります。
- Aurora MySQL バージョン 2 (MySQL 5.7 互換) から Aurora MySQL バージョン 3 (MySQL 8.0 互換) にアップグレードする際に、すべてのエラーが表示されるようになりました。以前のバージョンでは、50 件のエラーしか表示されませんでした。
- Aurora MySQL バージョン 2 (MySQL 5.7 互換) から Aurora MySQL バージョン 3 (MySQL 8.0 互換) へのメジャーバージョンアップグレード後に、まれに自動増分カウンターが不正確になる問題を修正しました。
- `mysql.innodb_table_stats` テーブルと `mysql.innodb_index_stats` テーブルの移行に予想以上に時間がかかり、Aurora MySQL バージョン 2 から Aurora MySQL バージョン 3 へのメジャーバージョンアップグレードが失敗する可能性がある問題を修正しました。この問題の影響があったのは主に、テーブルの数が多い (150 万を超える) データベースクラスターです。
- AMS 8.0 エンジンのアップグレードワークフローの欠陥が原因で、ログレコードが Aurora ストレージクラスターボリュームに蓄積されて通常の書き込み操作が停止し、Aurora MySQL バージョン 2 から Aurora MySQL バージョン 3 へのメジャーバージョンアップグレードが失敗する可能性

がある問題を修正しました。この問題の影響があったのは主に、テーブルの数が多い (約 750k を超える) データベースクラスターです。

- MySQL のパージスレッドが誤ってアクティブに保たれていたせいで、Aurora MySQL Serverless v2 のアイドルインスタンスを 0.5 ACU にスケールダウンできない問題を修正しました。
- データベースインスタンスへの接続中に、そのインスタンスへの着信接続が急増している場合、アプリケーションでレイテンシーが増大する可能性がある問題を修正しました。
- Aurora MySQL データベースインスタンスの接続確立遅延のトラブルシューティングに役立つ 2 つの新しい Amazon CloudWatch メトリクスを導入しました。 AuroraSlowHandshakeCount AuroraSlowConnectionHandleCount およびメトリックの詳細については、Aurora CloudWatch メトリクスの定義を参照してください。

MySQL Community Edition でのバグ修正の統合

このリリースには、以下を含め、8.0.23 までのコミュニティ版のバグ修正がすべて反映されています。詳細については、「MySQL 3.x データベースエンジンの更新で修正された MySQL のバグ」を参照してください。

- 特定の条件下で、OR 条件を使用したクエリを実行した場合に nullability プロパティの計算が不正確になり、誤った結果が返される可能性がある問題を修正しました。(バグ #34060289)
- 次の 2 つの条件が当てはまる場合に、特定の条件下で誤った結果が返される可能性がある問題を修正しました。
  - 派生テーブルが外側のクエリブロックにマージされている。
  - クエリに左結合と IN サブクエリが含まれている。(バグ #34060289)
- パフォーマンススキーマの DROP 権限を取り消すことができなかった問題を修正しました。(バグ #33578113)
- IF ステートメントで EXISTS を使用したストアドプロシージャが、処理対象の 1 つ以上のテーブルが実行の合間に削除され、再作成された場合に、最初の呼び出し以降の呼び出しで正しく実行されない問題を修正しました。(MySQL バグ #32855634)
- 整数列の最大値を超えた場合に、不正な AUTO_INCREMENT 値が生成されていました。このエラーは、列の最大値が考慮されていないことが原因でした。この場合、以前の有効な AUTO_INCREMENT 値が返され、重複キーエラーが発生すべきでした。(バグ #87926、バグ #26906787)
- 特定のテーブル ID を持つユーザー作成のテーブルを含む Aurora MySQL バージョン 1 (MySQL 5.6 互換) データベースクラスターをアップグレードする際に、エラーが発生する可能性がある問

題を修正しました。Aurora MySQL バージョン 2 (MySQL 5.7 互換) から Aurora MySQL バージョン 3 (MySQL 8.0 互換) にアップグレードする際に、これらのテーブル ID を割り当てると、データディクショナリテーブル ID が競合する可能性があります (バグ #33919635)。

Aurora MySQL データベースエンジンの更新 2022-09-07 (バージョン 3.02.1、MySQL 8.0.23 互換) 標準サポートは 2024 年 1 月 15 日に終了。このバージョンへのアップグレードはサポートされていません。

バージョン: 3.02.1

Aurora MySQL 3.02.1 は一般公開されています。Aurora MySQL 3.02 バージョンは MySQL 8.0.23 と互換性があり、Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

Aurora MySQL バージョン 3 の新機能や、Aurora MySQL バージョン 3 と Aurora MySQL バージョン 2 またはコミュニティ版 MySQL 8.0 との違いについては、「Amazon Aurora ユーザーガイド」の「Aurora MySQL バージョン 2 と Aurora MySQL バージョン 3 の比較」を参照してください。

現在サポートされている Aurora MySQL リリース
は、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

Aurora MySQL バージョン 3 へのアップグレードの計画については、「Amazon Aurora ユーザーガイド」の「Upgrade planning for Aurora MySQL version 3」を参照してください。アップグレード手順自体については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL バージョン 3 へのアップグレード」を参照してください。Aurora MySQL のアップグレードに関する一般的な情報については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL DB クラスターのアップグレード」を参照してください。

トラブルシューティング情報については、「Aurora MySQL バージョン 3 のアップグレードに関する問題のトラブルシューティング」を参照してください。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

Aurora MySQL バージョン 3.02.1 は一般公開されています。MySQL 8.0.23 コミュニティ版と互換性があります。

以下のセキュリティの問題と CVE の修正:

マネージド型の環境での処理を微調整するための修正およびその他の機能強化。以下の CVE の追加の修正:

- CVE-2022-0778

可用性の向上:

- 複数の MySQL バイナリログ (binlog) レプリカが Aurora ライターノードにアタッチされている場合や、実行時間が長いクエリが多数同時に実行されている場合に、新しい接続リクエストが急増すると、接続障害やレイテンシーの増大につながる可能性がある問題を修正しました。
- CONNECT イベントに対して高度な監査が有効になっている場合に、データベースが再起動する可能性がある問題を修正しました。
- 顧客設定値またはデフォルト値として設定された、メモリと mmap ファイルの割り当てサイズを内部一時テーブルが使い切った場合に、Aurora MySQL リードレプリカインスタンスでデータベースが再起動する可能性がある問題を修正しました。
- ストアドプロシージャで複数の DDL オペレーションが同時に実行されている間に、リードレプリカが繰り返し再起動する問題を修正しました。
- この Aurora MySQL バージョンでは高速挿入が有効になっていません。、、などのクエリの実行時に不整合が生じる可能性があるためです。INSERT INTO SELECT FROM高速挿入最適化の詳細については、「Amazon Aurora MySQL のパフォーマンス強化」を参照してください。

全般的な機能強化:

- R6i インスタンスのサポートが追加されました。

追加情報:

- Aurora MySQL バージョン 3.02.1 では、Aurora MySQL バージョン 2 (MySQL 5.7 互換) からのメジャーバージョンアップグレードはサポートされていません。このバージョンへのメジャーバージョンアップグレードを実行するには、まず Aurora MySQL バージョン 3.02.0 へのメジャーバー

ジョンアップグレードを実行してから、Aurora MySQL バージョン 3.02.1 へのインプレースマイナーバージョンアップグレードを実行します。

Aurora MySQL データベースエンジンの更新 2022-04-20 (バージョン 3.02.0、MySQL 8.0.23 互換) 標準サポートは 2024 年 1 月 15 日に終了。このバージョンへのアップグレードはサポートされていません。

バージョン: 3.02.0

Aurora MySQL 3.02.0 は一般公開されています。Aurora MySQL 3.02 バージョンは MySQL 8.0.23 と互換性があり、Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

Aurora MySQL バージョン 3 の新機能や、Aurora MySQL バージョン 3 と Aurora MySQL バージョン 2 またはコミュニティ版 MySQL 8.0 との違いについては、「Amazon Aurora ユーザーガイド」の「Aurora MySQL バージョン 2 と Aurora MySQL バージョン 3 の比較」を参照してください。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

現在サポートされている Aurora MySQL バージョン 2 クラスターから取得したスナップショットを Aurora MySQL 3.02.0 で復元できます。

Aurora MySQL バージョン 3 へのアップグレードの計画については、「Amazon Aurora ユーザーガイド」の「Upgrade planning for Aurora MySQL version 3」を参照してください。アップグレード手順自体については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL バージョン 3 へのアップグレード」を参照してください。Aurora MySQL のアップグレードに関する一般的な情報については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL DB クラスターのアップグレード」を参照してください。

トラブルシューティング情報については、「Aurora MySQL バージョン 3 のアップグレードに関する問題のトラブルシューティング」を参照してください。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

Aurora MySQL バージョン 3.02.0 は一般公開されています。MySQL 8.0.23 コミュニティ版と互換性があります。

以下のセキュリティの問題と CVE の修正:

マネージド型の環境での処理を微調整するための修正およびその他の機能強化。以下の CVE の追加の修正:

- CVE-2021-22946

新機能:

- Amazon Aurora Serverless v2 は一般公開されています。詳細については、Amazon Aurora Serverless の概要、ブログ、「Aurora Serverless v2 を使用する」ドキュメントを参照してください。 AWS Management Consoleでいくつかの手順を踏むだけで Aurora Serverless v2 データベースをすぐに作成できます。今すぐ始めましょう。

可用性の向上:

- レコードを削除したり、2 つ以上の可変長列 (VARCHAR、VARBINARY、BLOB、TEXT タイプ) を含むテーブルを削除したりするときに、サーバーが再起動ループに陥り、使用できなくなる可能性がある問題を修正しました。列タイプの詳細については、を参照してくださいinnodb-row-format。
- バイナリログが有効で、少なくとも 1 つのバイナリログコンシューマーが接続されているクラスターで、既存の接続がタイムアウトし、新しい接続を確立できず、アプリケーションとコンシューマー間でリソースの競合が発生する問題を修正しました。
- `FreeableMemory` CloudWatch 空きメモリは指標で示されます。詳細については、「Amazon Aurora CloudWatch のアマゾンメトリックス」を参照してください。
  - バイナリログのレプリケーションが有効になっている場合に解放可能なメモリが減少し、DB インスタンスが再起動またはフェイルオーバーする可能性がある問題を修正しました。
  - セッション変数の設定時に解放可能なメモリが減少し、DB インスタンスが再起動またはフェイルオーバーする可能性がある問題を修正しました。
  - データベースプロセスが既存のファイルを開くときに解放可能なメモリが減少し、DB インスタンスが再起動またはフェイルオーバーする可能性がある問題を修正しました。

- スナップショットから復元されたクラスターで AUTO_INCREMENT 列を含むテーブルに新しい行を挿入すると、まれに重複エントリエラーが発生する問題を修正しました。
- この Aurora MySQL バージョンでは高速挿入が有効になっていません。、、などのクエリの実行時に不整合が生じる可能性があるためです。INSERT INTO SELECT FROM高速挿入最適化の詳細については、「Amazon Aurora MySQL のパフォーマンス強化」を参照してください。

全般的な機能強化:

- SHOW VOLUME STATUS コマンドの使用時にボリュームステータスが表示されない問題を修正しました。詳細については、「SQL.管理」を参照してください。AuroraMy VolumeStatus。
- mysql_rds_import_binlog_ssl_material への呼び出しが失敗し、MySQL サーバーエラー 3512 が発生する問題を修正しました。
- 削除された Aurora リーダーインスタンスについて、Aurora レプリカの遅延が誤って報告される問題を修正しました。

アップグレード/移行:

- ibdata ファイルとテーブルスペースを Aurora ストレージにコピーする際に問題が発生し、MySQL 8.0.x データベースの Aurora MySQL バージョン 3 への移行が失敗する可能性がある問題を修正しました。
- データベーステーブルに大量のデータが含まれている場合に、Aurora MySQL バージョン 2 から Aurora MySQL バージョン 3 へのクラスターのアップグレードが失敗する可能性がある問題を修正しました。
- テーブルのシリアライズデータディクショナリ情報 (SDI) を作成できないことが原因で、Aurora MySQL バージョン 2 から Aurora MySQL バージョン 3 へのクラスターの復元時に障害が起きる可能性がある問題を修正しました。
- RDS システムテーブルのアップグレードの事前チェックによって報告されたスキーマ不整合エラーが原因で、Aurora MySQL バージョン 2 から Aurora MySQL バージョン 3 へのアップグレードが失敗する可能性がある問題を修正しました。
- RDS マネージドストアドプロシージャの構文が無効なため、MySQL 8.0 または Aurora MySQL バージョン 2 用の RDS から Aurora MySQL バージョン 3 のデータベースへの移行または復元時に障害が起きる可能性がある問題を修正しました。
- 一般ログテーブルとスローログのアップグレードの事前チェックによって報告されたスキーマ不整合エラーが原因で、Aurora MySQL 2 から Aurora MySQL 3 へのアップグレードが失敗する可能性がある問題を修正しました。

MySQL Community Edition バグ修正の統合

このリリースには、以下を含め、8.0.23 までのコミュニティ版のバグ修正がすべて反映されています。詳細については、「MySQL 3.x データベースエンジンの更新で修正された MySQL のバグ」を参照してください。

- 予期しないサーバーの動作を引き起こす可能性のある、ストアドプロシージャ内のカーソルに使用される一時テーブルの不適切な処理を修正しました。mysqld-8-0-24-bug。(バグ #32416811)

Aurora MySQL データベースエンジンの更新 2022-04-15 (バージョン 3.01.1、MySQL 8.0.23 互換) 標準サポートは 2024 年 1 月 15 日に終了。このバージョンへのアップグレードはサポートされていません。

バージョン: 3.01.1

Aurora MySQL 3.01.1 は一般公開されています。Aurora MySQL 3.01 バージョンは MySQL 8.0.23 と互換性があり、Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

Aurora MySQL バージョン 3 の新機能や、Aurora MySQL バージョン 3 と Aurora MySQL バージョン 2 またはコミュニティ版 MySQL 8.0 との違いについては、「Amazon Aurora ユーザーガイド」の「Aurora MySQL バージョン 2 と Aurora MySQL バージョン 3 の比較」を参照してください。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

現在サポートされている Aurora MySQL バージョン 2 クラスターから取得したスナップショットを Aurora MySQL 3.01.1 で復元できます。

Aurora MySQL バージョン 3 へのアップグレードの計画については、「Amazon Aurora ユーザーガイド」の「Upgrade planning for Aurora MySQL version 3」を参照してください。アップグレード手順自体については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL バージョン 3 へのアップグレード」を参照してください。Aurora MySQL のアップグレードに関する一般的な情報については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL DB クラスターのアップグレード」を参照してください。

トラブルシューティング情報については、「Aurora MySQL バージョン 3 のアップグレードに関する問題のトラブルシューティング」を参照してください。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

Aurora MySQL バージョン 3.01.1 は一般公開されています。MySQL 8.0.23 コミュニティ版と互換性があります。

MySQL 8.0 互換の Aurora データベースへのアップグレードと移行には、Aurora MySQL バージョン 3.01.1 が推奨されています。

以下のセキュリティの問題と CVE の修正:

マネージド型の環境での処理を微調整するための修正およびその他の機能強化。以下の CVE の追加の修正:

- CVE-2021-36222
- CVE-2021-22946
- CVE-2021-22926

可用性の向上:

- `FreeableMemory` CloudWatch 空きメモリは指標で示されます。詳細については、「Amazon Aurora CloudWatch のアマゾンメトリックス」を参照してください。
  - バイナリログのレプリケーションが有効になっている場合に解放可能なメモリが減少し、DB インスタンスが再起動またはフェイルオーバーする可能性がある問題を修正しました。
  - セッション変数の設定時に解放可能なメモリが減少し、DB インスタンスが再起動またはフェイルオーバーする可能性がある問題を修正しました。
  - データベースプロセスが既存のファイルを開くときに解放可能なメモリが減少し、DB インスタンスが再起動またはフェイルオーバーする可能性がある問題を修正しました。
- スナップショットから復元されたクラスターで `AUTO_INCREMENT` 列を含むテーブルに新しい行を挿入すると、まれに重複エントリエラーが発生する問題を修正しました。

- この Aurora MySQL バージョンでは高速挿入が有効になっていません。、、などのクエリの実行時に不整合が生じる可能性があるためです。INSERT INTO SELECT FROM高速挿入最適化の詳細については、「Amazon Aurora MySQL のパフォーマンス強化」を参照してください。

全般的な機能強化:

- SHOW VOLUME STATUS コマンドの使用時にボリュームステータスが表示されない問題を修正しました。詳細については、「SQL.管理」を参照してください。AuroraMy VolumeStatus。
- mysql_rds_import_binlog_ssl_material への呼び出しが失敗し、MySQL サーバーエラー 3512 が発生する問題を修正しました。
- 削除された Aurora リーダーインスタンスについて、Aurora レプリカの遅延が誤って報告される問題を修正しました。

アップグレード/移行:

- ibdata ファイルとテーブルスペースを Aurora ストレージにコピーする際に問題が発生し、MySQL 8.0.x データベースの Aurora MySQL バージョン 3 への移行が失敗する可能性がある問題を修正しました。
- データベーステーブルに大量のデータが含まれている場合に、Aurora MySQL バージョン 2 から Aurora MySQL バージョン 3 へのクラスターのアップグレードが失敗する可能性がある問題を修正しました。
- テーブルのシリアライズデータディクショナリ情報 (SDI) を作成できないことが原因で、Aurora MySQL バージョン 2 から Aurora MySQL バージョン 3 へのクラスターの復元時に障害が起きる可能性がある問題を修正しました。
- RDS システムテーブルのアップグレードの事前チェックによって報告されたスキーマ不整合エラーが原因で、Aurora MySQL バージョン 2 から Aurora MySQL バージョン 3 へのアップグレードが失敗する可能性がある問題を修正しました。
- RDS マネージドストアドプロシージャの構文が無効なため、MySQL 8.0 または Aurora MySQL バージョン 2 用の RDS から Aurora MySQL バージョン 3 のデータベースへの移行または復元時に障害が起きる可能性がある問題を修正しました。
- 一般ログテーブルとスローログのアップグレードの事前チェックによって報告されたスキーマ不整合エラーが原因で、Aurora MySQL 2 から Aurora MySQL 3 へのアップグレードが失敗する可能性がある問題を修正しました。

MySQL Community Edition バグ修正の統合

このリリースには、以下を含め、8.0.23 までのコミュニティ版のバグ修正がすべて反映されています。詳細については、「MySQL 3.x データベースエンジンの更新で修正された MySQL のバグ」を参照してください。

- 予期しないサーバーの動作を引き起こす可能性のある、ストアドプロシージャ内のカーソルに使用される一時テーブルの不適切な処理を修正しました。mysqld-8-0-24-bug。(バグ #32416811)

Aurora MySQL データベースエンジンの更新 2021-11-18 (バージョン 3.01.0、MySQL 8.0.23 互換) 標準サポートは 2024 年 1 月 15 日に終了。このバージョンへのアップグレードはサポートされていません。

バージョン 3.01.0

Aurora MySQL 3.01.0 は一般公開されています。Aurora MySQL 3.01 バージョンは MySQL 8.0.23 と互換性があり、Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

Aurora MySQL バージョン 3 の新機能や、Aurora MySQL バージョン 3 と Aurora MySQL バージョン 2 またはコミュニティ版 MySQL 8.0 との違いについては、「Amazon Aurora ユーザーガイド」の「Aurora MySQL バージョン 2 と Aurora MySQL バージョン 3 の比較」を参照してください。

現在サポートされている Aurora MySQL リリース
は、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

現在サポートされている Aurora MySQL バージョン 2 クラスターから取得したスナップショットを Aurora MySQL 3.01.0 で復元できます。

Aurora MySQL バージョン 3 へのアップグレードの計画については、「Amazon Aurora ユーザーガイド」の「Upgrade planning for Aurora MySQL version 3」を参照してください。アップグレード手順自体については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL バージョン 3 へのアップグレード」を参照してください。Aurora MySQL のアップグレードに関する一般的な情報については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL DB クラスターのアップグレード」を参照してください。

トラブルシューティング情報については、「Aurora MySQL バージョン 3 のアップグレードに関する問題のトラブルシューティング」を参照してください。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

Aurora MySQL バージョン 3.01.0 は通常、コミュニティ MySQL 8.0.23 と互換性があります。このバージョンには、コミュニティ MySQL 8.0.23 における一般的な脆弱性および露出 (CVE) の問題に対するセキュリティ修正が含まれています。

Aurora MySQL バージョン 3.01.0 には、Aurora MySQL バージョン 2.10.0 による Aurora 固有のバグ修正がすべて含まれています。

Aurora MySQL バージョン 3 の新機能の詳細については、「Amazon Aurora ユーザーガイド」の「MySQL 8.0 コミュニティエディションからの機能」と「新しいパラレルクエリの最適化」を参照してください。

可用性の向上:

- この Aurora MySQL バージョンでは高速挿入が有効になっていません。、、などのクエリの実行時に不整合が生じる可能性があるためです。`INSERT INTO SELECT FROM`高速挿入最適化の詳細については、「Amazon Aurora MySQL のパフォーマンス強化」を参照してください。

Amazon Aurora MySQL バージョン 2 のデータベースエンジンの更新

Amazon Aurora MySQL バージョン 2 のデータベースエンジンの更新は次のとおりです。

- Aurora MySQL データベースエンジンアップデート 2024-03-19 (バージョン 2.12.2、MySQL 5.7.44 と互換)
- Aurora MySQL データベースエンジンの更新 2023-12-28 (バージョン 2.12.1、MySQL 5.7.40 互換)
- Aurora MySQL データベースエンジンの更新 2023-10-25 (バージョン 2.12.0.1、MySQL 5.7.40 互換) ベータ
- Aurora MySQL データベースエンジンの更新 2023-07-25 (バージョン 2.12.0、MySQL 5.7.40 互換)
- Aurora MySQL データベースエンジンアップデート 2024-03-26 (バージョン 2.11.5、MySQL 5.7.12 に対応)
- Aurora MySQL データベースエンジンの更新 2023-10-17 (バージョン 2.11.4、MySQL 5.7.12 互換)
- Aurora MySQL データベースエンジンの更新 2023-06-09 (バージョン 2.11.3、MySQL 5.7.12 互換)
- Aurora MySQL データベースエンジンの更新 2023-03-24 (バージョン 2.11.2、MySQL 5.7.12 互換)
- Aurora MySQL データベースエンジンの更新 2023-02-14 (バージョン 2.11.1、MySQL 5.7.12 互換)
- Aurora MySQL データベースエンジンの更新 2022-10-25 (バージョン 2.11.0、MySQL 5.7.12 互換) このバージョンは新規作成には使用不可。
- Aurora MySQL データベースエンジンの更新 2022-11-01 (バージョン 2.10.3) (廃止)
- Aurora MySQL データベースエンジンの更新 2022-01-26 (バージョン 2.10.2) (廃止)
- Aurora MySQL データベースエンジンの更新 2021-10-21 (バージョン 2.10.1) (廃止)
- Aurora MySQL データベースエンジンの更新 2021-05-25 (バージョン 2.10.0) (廃止)
- Aurora MySQL データベースエンジンの更新 2021-11-12 (バージョン 2.09.3) (廃止)
- Aurora MySQL データベースエンジンの更新 2021-02-26 (バージョン 2.09.2) (廃止)
- Aurora MySQL データベースエンジンの更新 2020-12-11 (バージョン 2.09.1) (廃止)
- Aurora MySQL データベースエンジンの更新 2020-09-17 (バージョン 2.09.0) (廃止)
- Aurora MySQL データベースエンジンの更新 2022-01-06 (バージョン 2.08.4) (廃止)
- Aurora MySQL データベースエンジンの更新 2020-11-12 (バージョン 2.08.3) (廃止)

- Aurora MySQL データベースエンジンの更新 2020-08-28 (バージョン 2.08.2) (廃止)
- Aurora MySQL データベースエンジンの更新 2020-06-18 (バージョン 2.08.1) (廃止)
- Aurora MySQL データベースエンジンの更新 2020-06-02 (バージョン 2.08.0) (廃止)
- Aurora MySQL データベースエンジンの更新 2023-08-15 (バージョン 2.07.10、MySQL 5.7.12 互換)
- Aurora MySQL データベースエンジンの更新 2023-05-04 (バージョン 2.07.9、MySQL 5.7.12 互換)
- Aurora MySQL データベースエンジンの更新 2022-06-16 (バージョン 2.07.8) (廃止)
- Aurora MySQL データベースエンジンの更新 2021-11-24 (バージョン 2.07.7) (廃止)
- Aurora MySQL データベースエンジンの更新 2021-09-02 (バージョン 2.07.6) (廃止)
- Aurora MySQL データベースエンジンの更新 2021-07-06 (バージョン 2.07.5) (廃止)
- Aurora MySQL データベースエンジンの更新 2021-03-04 (バージョン 2.07.4) (廃止)
- Aurora MySQL データベースエンジンの更新 2020-11-10 (バージョン 2.07.3) (廃止)
- Aurora MySQL データベースエンジンの更新 2020-04-17 (バージョン 2.07.2) (廃止)
- Aurora MySQL データベースエンジンの更新 2019-12-23 (バージョン 2.07.1) (廃止)
- Aurora MySQL データベースエンジンの更新 2019-11-25 (バージョン 2.07.0) (廃止)
- Aurora MySQL データベースエンジンの更新 2019-11-22 (バージョン 2.06.0) (廃止)
- Aurora MySQL データベースエンジンの更新 2019-11-11 (バージョン 2.05.0) (廃止)
- Aurora MySQL データベースエンジンの更新 2020-08-14 (バージョン 2.04.9) (廃止)
- Aurora MySQL データベースエンジンの更新 2019-11-20 (バージョン 2.04.8) (廃止)
- Aurora MySQL データベースエンジンの更新 2019-11-14 (バージョン 2.04.7) (廃止)
- Aurora MySQL データベースエンジンの更新 2019-09-19 (バージョン 2.04.6) (廃止)
- Aurora MySQL データベースエンジンの更新 2019-07-08 (バージョン 2.04.5) (廃止)
- Aurora MySQL データベースエンジンの更新 2019-05-29 (バージョン 2.04.4) (廃止)
- Aurora MySQL データベースエンジンの更新 2019-05-09 (バージョン 2.04.3) (廃止)
- Aurora MySQL データベースエンジンの更新 2019-05-02 (バージョン 2.04.2) (廃止)
- Aurora MySQL データベースエンジンの更新 2019-03-25 (バージョン 2.04.1) (廃止)
- Aurora MySQL データベースエンジンの更新 2019-03-25 (バージョン 2.04.0) (廃止)
- Aurora MySQL データベースエンジンの更新 2019-02-07 (バージョン 2.03.4) (廃止)
- Aurora MySQL データベースエンジンの更新 2019-01-18 (バージョン 2.03.3) (廃止)
- Aurora MySQL データベースエンジンの更新 2019-01-09 (バージョン 2.03.2) (廃止)

- Aurora MySQL データベースエンジンの更新 2018-10-24 (バージョン 2.03.1) (廃止)
- Aurora MySQL データベースエンジンの更新 2018-10-11 (バージョン 2.03) (廃止)
- Aurora MySQL データベースエンジンの更新 2018-10-08 (バージョン 2.02.5) (廃止)
- Aurora MySQL データベースエンジンの更新 2018-09-21 (バージョン 2.02.4) (廃止)
- Aurora MySQL データベースエンジンの更新 2018-08-23 (バージョン 2.02.3) (廃止)
- Aurora MySQL データベースエンジンの更新 2018-06-04 (バージョン 2.02.2) (廃止)
- Aurora MySQL データベースエンジンの更新 2018-05-03 (バージョン 2.02) (廃止)
- Aurora MySQL データベースエンジンの更新 2018-03-13 (バージョン 2.01.1) (廃止)
- Aurora MySQL データベースエンジンの更新 2018-02-06 (バージョン 2.01) (廃止)

Aurora MySQL データベースエンジンアップデート 2024-03-19 (バージョン 2.12.2、MySQL 5.7.44 と互換)

バージョン:2.12.2

Aurora MySQL 2.12.2 が一般公開されています。Aurora MySQL 2.12 バージョンは MySQL 5.7.44 まで互換性があります。コミュニティの変更点の詳細については、「MySQL 5.7.44 (2022-10-11、一般公開) での変更点」を参照してください。

現在サポートされている Aurora MySQL リリースは
2.07.9、2.07.10、2.11.*、2.12.*、3.03.*、3.04.*、3.05.*、3.06.* です。

既存の Aurora MySQL 2.* データベースクラスターを Aurora MySQL 2.12.2 にアップグレードできます。また、現在サポートされている Aurora MySQL リリースのスナップショットを Aurora MySQL 2.12.2 に復元することもできます。

Aurora MySQL グローバルデータベースをバージョン 2.12.* にアップグレードし、書き込み転送を有効にした場合、引き続き書き込み転送を使用するには、プライマリ DB クラスターとセカンダリ DB クラスターをパッチレベルを含めてまったく同じバージョンにアップグレードする必要があります。Aurora グローバルデータベースのマイナーバージョンのアップグレードの詳細については、「マイナーバージョンのアップグレード」を参照してください。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

以下のセキュリティの問題と CVE の修正:

このリリースには、MySQL 5.7.44 までのコミュニティ CVE 修正がすべて含まれています。次の CVE 修正が含まれています。

- CVE-2024-20963
- CVE-2023-39975
- CVE-2023-38545

セキュリティ問題:

- ソースが暗号化接続をサポートしている場合、設定に関係なく、バイナリログレプリカがデフォルトで SSL/TLS を使用するようにする修正が追加されました。`MASTER_SSL`

可用性の向上:

- ライターインスタンスに高い負荷がかかると、リードレプリカインスタンスが正常に起動しなくなることがある問題を修正しました。
- Aurora ストレージと通信するコンポーネントの不具合により、Aurora MySQL データベースライターインスタンスがフェールオーバーする可能性がある問題を修正しました。この欠陥は、Aurora ストレージインスタンスのソフトウェア更新後に、データベースインスタンスと基盤となるストレージ間の通信が中断された結果として発生します。
- まれに、リーダーインスタンスが再起動することがある問題を修正しました。
- 特権ユーザーがユーザー rdsadmin に関連付けられているリソース制限を変更できる問題を修正しました。これらのリソース制限を誤って設定すると、RDS 監視エージェントがデータベースインスタンスの状態を監視できなくなり、データベースが使用できなくなる可能性があります。

アップグレードと移行:

- Amazon RDS MySQL 5.7 から移行した、サポートされていないストアドプロシージャを含む Aurora MySQL クラスターのバイナリログレプリケーションを開始しようとしたときに発生する問題を修正しました。
- Aurora MySQL バージョン 3 へのメジャーバージョンアップグレード中に、データベースイベントスケジューラを無効にしました。この更新により、メジャーバージョンアップグレードの進行中にイベントが実行されてデータベースが変更されるのを防ぐことができます。

MySQL Community Edition でのバグ修正の統合

このリリースには、5.7.44 以前のすべてのコミュニティバグ修正が含まれています。詳細については、「Aurora MySQL 2.x データベースエンジンの更新で修正された MySQL のバグ」を参照してください。

Aurora MySQL バージョン 2 ではサポートされていない機能

以下の機能は、現時点では Aurora MySQL バージョン 2 (MySQL 5.7 互換) ではサポートされていません。

- スキャンバッチ処理

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- `CREATE TABLESPACE` SQL ステートメント
- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション

- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- X プロトコル

Aurora MySQL データベースエンジンの更新 2023-12-28 (バージョン 2.12.1、MySQL 5.7.40 互換)

バージョン: 2.12.1

Aurora MySQL 2.12.1 は一般利用可能です。Aurora MySQL 2.12 バージョンは MySQL 5.7.40 までとの互換性があります。コミュニティ版の変更点の詳細については、「Changes in MySQL 5.7.40 (2022-10-11, General Availability)」を参照してください。

現在サポートされている Aurora MySQL リリースは、2.07.*、2.11.*、2.12.*、3.01.*、3.02.*、3.03.*、3.04.*、3.05.* です。

既存の Aurora MySQL 2.* データベースクラスターを Aurora MySQL 2.12.1 にアップグレードできます。現在サポートされている Aurora MySQL リリースから取得したスナップショットを Aurora MySQL 2.12.1 に復元することもできます。

Aurora MySQL グローバルデータベースをバージョン 2.12.* にアップグレードし、書き込み転送を有効にした場合、引き続き書き込み転送を使用するには、プライマリ DB クラスターとセカンダリ DB クラスターをパッチレベルを含めてまったく同じバージョンにアップグレードする必要があります。Aurora グローバルデータベースのマイナーバージョンのアップグレードの詳細については、「マイナーバージョンのアップグレード」を参照してください。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS Support AWS から サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

以下のセキュリティの問題と CVE の修正:

このリリースには、MySQL 5.7.44CVEs 修正が含まれています。

マネージド型の環境での処理を微調整するための修正およびその他の機能強化。以下の CVE の追加の修正:

- 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
- 全文検索 (FTS) パーサープラグインによる単一文字トークンの処理を修正しました (バグ #35432973)
- 監査ログのローテーションの処理中に報告されたイベントが監査ログに書き込まれない可能性がある問題を修正しました。

新機能:

- マルチスレッドのバイナリログ (binlog) レプリケーションのサポートが追加されました。これにより、binlog レプリカの SQL スレッドは、可能な場合はバイナリログイベントを並列で適用します。マルチスレッドレプリケーションの微調整に役立つ設定オプションの詳細については、「Aurora ユーザーガイド」を参照してください。

可用性の向上:

- パラレルクエリを使用する Aurora MySQL データベースインスタンスで、多数のパラレルクエリを同時に実行すると、データベースが再起動する問題を修正しました。
- 監査ログ記録のスレッドが原因でロック競合が発生し、CPU 使用率が高くなり、クライアントアプリケーションがタイムアウトする問題を修正しました。

- 削除されたテーブルに属するデータベースページを読み取ろうとした場合に、データベースインスタンスが再起動する問題を修正しました。
- ライターインスタンスがデータベースボリュームを拡大させ、160 GB の倍数に達すると、リーダーインスタンスが再起動する問題を修正しました。
- 分離レベルを READ_COMMITED または READ_UNCOMMITED に設定し、XA トランザクションが使用されているか、バイナリログ (binlog) が有効になっている状態で 2 段階コミットを処理すると、再起動またはフェイルオーバーが発生するロックマネージャーの問題を修正しました。
- データベースが内部システムテーブルでトリガーを作成または削除しているときにライターインスタンスが再起動すると、データベースクラスターが使用できない状態になる問題を修正しました。
- データベース接続の数が max_connections パラメータで設定された値に近づくと、データベースインスタンスが再起動する場合がある問題を修正しました。
- 全文インデックスを含むテーブルに対してデータ操作言語 (DML) クエリを実行すると、Aurora リーダーインスタンスが再起動する場合がある問題を修正しました。
- この Aurora MySQL バージョンではSELECT、高速挿入は有効になっていません。これは、INSERT INTO、、 などのクエリを実行するときに不整合が発生する可能性があるためですFROM。高速挿入の最適化の詳細については、「Amazon Aurora MySQL のパフォーマンス強化」を参照してください。

全般的な機能強化:

- Aurora クラスターボリュームからデータを読み取る際の一時的なネットワークの問題により、並列クエリが失敗する場合がある問題を修正しました。
- 監査ログファイル管理に関連する問題を修正しました。この問題では、ダウンロードやローテーションのためにログファイルにアクセスできなくなり、場合によっては CPU 使用率が高くなることがあります。
- 2.11 より前のバージョンからアップグレードした後で、小さいリードレプリカインスタンスでレプリケーションの遅延が長くなる問題を修正しました。*
- procs_priv grant テーブルを参照して、ストアドルーチンに関するリクエストを検証すると、ログメッセージが過剰に生成されることがある問題を修正しました。
- ハッシュ結合の最適化を使用してクエリを実行する際に、データベースインスタンスがメモリを過剰に使用する原因となる、メモリ管理の問題を修正しました。
- 書き込み転送を使用すると、information_schema および performance_schema グローバルステータステーブルの変数 Threads_running の値が不正確に場合がある問題を修正しました。

- SELECT ステートメントを (古い ha_partition パーティションハンドラーをサポートするバージョンの MySQL で作成した) パーティションテーブルで実行し、クエリプランナーでパラレルクエリが選択されると、データベースが再起動する問題を修正しました。
- 書き込み転送が有効になっていると、クライアントからデータベースへの新しい接続が確立されないことがある問題を修正しました。
- デフォルトのデータベースを USE コマンドで定義していない状態で、ソースのバイナリログファイルに書き込まれる QUERY イベントを Aurora MySQL のバイナリログ (binlog) レプリカが実行した場合に、バイナリログのレプリケーションの遅延が軽減されます。
- innodb_flush_log_at_trx_commit パラメータが 1 に設定されていない場合にCommitLatency CloudWatch メトリクスが誤って報告される問題を修正しました。
- データベース接続が確立される前に切断してしまう問題を修正しました。接続を頻繁に確立したり切断したりするデータベースインスタンスが、この問題の影響を受けやすい傾向にあります。
- 接続先のバイナリログ (binlog) コンシューマーが、重複するバイナリログレプリケーションサーバー ID を使用している場合に、データベースが再起動する問題を修正しました。

MySQL Community Edition でのバグ修正の統合

このリリースには、以下を含め、5.7.40 までのコミュニティ版のバグ修正がすべて反映されています。詳細については、「Aurora MySQL 2.x データベースエンジンの更新で修正された MySQL のバグ」を参照してください。

- SHOW PROCESSLIST ステートメントと同時に実行すると、既存および新規のリモート接続が停止する問題を修正しました (コミュニティバグ #34857411)
- レプリケーション: 一部のバイナリログイベントが必ずしも正しく処理されていませんでした (バグ #34617506)

Aurora MySQL バージョン 2 ではサポートされていない機能

以下の機能は、現時点では Aurora MySQL バージョン 2 (MySQL 5.7 互換) ではサポートされていません。

- スキャンバッチ処理

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- `CREATE TABLESPACE` SQL ステートメント
- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- X プロトコル

Aurora MySQL データベースエンジンの更新 2023-10-25 (バージョン 2.12.0.1、MySQL 5.7.40 互換) ベータ

バージョン: 2.12.0.1

Aurora MySQL 2.12.0.1 は、通常、米国東部 (バージニア北部）、米国東部 (オハイオ）、米国西部 (北カリフォルニア）、米国西部 (オレゴン） AWS GovCloud 、 (米国東部）、および AWS GovCloud （米国西部) の各リージョンで利用できます。これは初期のセキュリティ修正限定のリリースです。これらの修正は、次回のパッチリリース 2.12.1 で対象範囲を広げ、すべてのリージョンにデプロイされる予定です。Aurora MySQL 2.12 バージョンは、MySQL 5.7.40 と互換性があります。

現在サポートされている Aurora MySQL リリースは、2.07.*、2.11.*、2.12.*、3.01.*、3.02.*、3.03.*、3.04.* です。

既存の Aurora MySQL 2.* データベースクラスターを Aurora MySQL 2.12.0.1 にアップグレードできます。現在サポートされている Aurora MySQL リリースから取得したスナップショットを Aurora MySQL 2.12.0.1 で復元することもできます。

Aurora MySQL グローバルデータベースをバージョン 2.12.* にアップグレードする場合、プライマリ DB クラスターとセカンダリ DB クラスターを、パッチレベルを含めてまったく同じバージョンにアップグレードする必要があります。Aurora グローバルデータベースのマイナーバージョンのアップグレードの詳細については、「マイナーバージョンのアップグレード」を参照してください。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS Support AWS から サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

以下のセキュリティの問題と CVE の修正:

このリリースには、MySQL 5.7.40 までのコミュニティ版 CVE の修正がすべて反映されています。

- CVE-2023-38545

可用性の向上:

- この Aurora MySQL バージョンではSELECT、高速挿入は有効になっていません。これは、INSERT INTO、、 などのクエリを実行するときに不整合が発生する可能性があるためですFROM。高速挿入の最適化の詳細については、「Amazon Aurora MySQL のパフォーマンス強化」を参照してください。

Aurora MySQL データベースエンジンの更新 2023-07-25 (バージョン 2.12.0、MySQL 5.7.40 互換)

バージョン: 2.12.0

Aurora MySQL 2.12.0 は一般公開されています。Aurora MySQL 2.12 バージョンは MySQL 5.7.40 までとの互換性があります。コミュニティ版の変更点の詳細については、「Changes in MySQL 5.7.40 (2022-10-11, General Availability)」を参照してください。

現在サポートされている Aurora MySQL リリースは、2.07.*、2.11.*、2.12.*、3.01.*、3.02.*、3.03.* です。

既存の Aurora MySQL 2.* データベースクラスターを Aurora MySQL 2.12.0 にアップグレードできます。現在サポートされている Aurora MySQL リリースから取得したスナップショットを Aurora MySQL 2.12.0 で復元することもできます。

Aurora MySQL グローバルデータベースをバージョン 2.12.* にアップグレードし、書き込み転送を有効にした場合、引き続き書き込み転送を使用するには、プライマリ DB クラスターとセカンダリ DB クラスターをパッチレベルを含めてまったく同じバージョンにアップグレードする必要があります。Aurora グローバルデータベースのマイナーバージョンのアップグレードの詳細については、「マイナーバージョンのアップグレード」を参照してください。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS Support AWS から サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

以下のセキュリティの問題と CVE の修正:

このリリースには、MySQL 5.7.40 までのコミュニティ版 CVE の修正がすべて反映されています。

- Aurora MySQL が使用するデフォルトの SSL 暗号が更新され、安全性の低い DES-CBC3-SHA 値が SSL_CIPHER データベースパラメータから除外されました。DES-CBC3-SHA 暗号の削除により SSL 接続に問題が生じた場合は、適切かつ安全な暗号を使用してください。詳細については、「Aurora MySQL DB クラスターへの接続用暗号スイートを設定する」を参照してください。MySQL クライアント接続の暗号設定については、MySQL ドキュメントの「Connection Cipher Configuration」を参照してください。
- 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

可用性の向上:

- データベースアクティビティストリームイベントの暗号化が原因で、データベースが再起動する問題を修正しました。
- データ定義言語 (DDL) クエリの実行中にデータベースの再起動が失敗する原因となる 2 つの問題を修正しました。
- 接続の急増によりクエリのレイテンシーが長くなったり、データベースインスタンスが再起動したりする問題を修正しました。
- 大規模な更新オペレーションまたはデータ定義言語 (DDL) ワークロードをライターインスタンスで、同じテーブルセットに対する読み取りオペレーションを Aurora レプリカで同時に実行すると、まれに Aurora レプリカが再起動することがある問題を修正しました。
- 接続が急増すると、接続確立プロセスが完了するまで時間がかかったり、タイムアウトエラーが発生して失敗したりする問題を修正しました。
- 解放可能なメモリが高度な監査ログのローテーションによって減少し、データベースインスタンスの再起動につながる可能性がある問題を修正しました。
- Aurora パラレルクエリの実行プランを利用するクエリの実行中に、Aurora MySQL リーダーインスタンスが再起動する問題を修正しました。
- 全文検索 (FTS) インデックスのあるテーブルで `OPTIMIZE TABLE` クエリを実行しているときに、ライターインスタンスが再起動する問題を修正しました。
- Aurora グローバルデータベースのセカンダリ AWS リージョンからグローバル書き込み転送を使用して`SELECT FOR UPDATE`クエリが実行されると、Aurora グローバルデータベースのプライマリリージョンのライターインスタンスが再起動することがある問題を修正しました。
- 転送された暗黙的なコミットステートメントでエラーが発生すると、グローバル書き込み転送を使用する Aurora グローバルデータベースのセカンダリ AWS リージョンリーダーインスタンスが再起動する問題を修正しました。
- この Aurora MySQL バージョンでは`SELECT`、高速挿入は有効になっていません。これは、`INSERT INTO`、、 などのクエリを実行するときに不整合が発生する可能性があるためで

すFROM。高速挿入の最適化の詳細については、「Amazon Aurora MySQL のパフォーマンス強化」を参照してください。

全般的な機能強化:

- リレーログファイルへの書き込み時の競合を減らすために、バイナリログレプリカに対するファイル管理パフォーマンスの最適化を導入しました。
- information_schema メトリクスで buffer_pool_read_requests カウンターが誤って報告されることがある問題を修正しました。
- LOAD FROM S3 または SELECT INTO S3 オペレーションを実行するとローカルストレージがいっぱいになることがある問題を修正しました。この問題は、CPU 使用率の上昇、メモリ不足によるデータベースの再起動、これらのクエリのレイテンシーの増大につながる可能性もあります。
- バイナリログレプリケーションを使用する DB インスタンスで、バイナリログレプリケーションのコンシューマーが複数アタッチされていると CPU 使用率が増加し、接続障害が発生する可能性がある問題を修正しました。
- SSL サーバーのステータス変数の値が指定されない問題を修正しました。
- データ操作言語 (DML) ステートメントが重複して書き込みを行うと、エラーのログ記録が過剰になり、クエリのレイテンシーが長くなる可能性がある問題を修正しました。
- タイムゾーン定義が IANA 2023c バージョンにアップグレードされました。
- セッションレベルのバイナリログ記録を有効化または無効化できるようになりました。「Amazon Aurora ユーザーガイド」の「ストアドプロシージャ - レプリケーション」を参照してください。
- セッションレベルでバイナリログの形式を設定できるようになりました。Amazon Aurora ユーザーガイドの「ストアドプロシージャ - レプリケーション」を参照してください。
- aurora_disable_hash_join パラメータを 1 または ON に設定しても、オプティマイザーによるハッシュ結合の使用を阻止できない場合がある問題を修正しました。
- GROUP BY 句を使用し、aurora_parallel_query パラメータを ON に指定した SELECT クエリを実行すると、不正確な結果が返されることがある、インデックススキャンに関する問題を修正しました。
- ライターインスタンスで大規模な更新オペレーションまたはデータ定義言語 (DDL) オペレーションを同時に実行しているテーブルにアクセスした場合に、まれに Amazon Aurora リーダーインスタンスが再起動することがある問題を修正しました。
- information_schema メトリクスで buffer_pool_read_requests カウンターが誤って報告されることがある問題を修正しました。

- ソースのシステム変数 server uuid が指定されていないか、その値が無効な場合に、バイナリログレプリカが再起動する問題を修正しました。
- InnoDB の統計情報が古くならないように、問題を修正しました。古くなると、最適ではないクエリ実行プランが生成され、クエリの実行時間が長くなる可能性があります。
- ユーザーワークロードに関係なく`AuroraGlobalDBRPOLag` CloudWatch メトリクスが常にゼロで表示される問題を修正しました。

アップグレードと移行:

- Aurora グローバルデータベースに対して、Aurora MySQL バージョン 2.07 または 2.11 から Aurora MySQL バージョン 2.12 以降へのマイナーバージョンアップグレードを実行するには、「エンジンのバージョンを変更して Aurora MySQL をアップグレードする」を参照してください。

MySQL Community Edition でのバグ修正の統合

このリリースには、以下を含め、5.7.40 までのコミュニティ版のバグ修正がすべて反映されています。詳細については、「Aurora MySQL 2.x データベースエンジンの更新で修正された MySQL のバグ」を参照してください。

- バックグラウンドの TLS 証明書のローテーションが原因で CPU 使用率が高くなる問題を修正しました (コミュニティのバグ修正 #34284186)。

Aurora MySQL バージョン 2 ではサポートされていない機能

以下の機能は、現時点では Aurora MySQL バージョン 2 (MySQL 5.7 互換) ではサポートされていません。

- スキャンバッチ処理。

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- CREATE TABLESPACE SQL ステートメント
- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- X プロトコル

Aurora MySQL データベースエンジンアップデート 2024-03-26 (バージョン 2.11.5、MySQL 5.7.12 に対応)

バージョン:2.11.5

Aurora MySQL 2.11.5 が一般公開されています。Aurora MySQL 2.11 バージョンは、MySQL 5.7.12 と互換性があります。コミュニティ版の変更点の詳細については、「Changes in MySQL 5.7.12 (2016-04-11, General Availability)」を参照してください。

現在サポートされている Aurora MySQL リリースは 2.07.9、2.07.10、2.11.*、2.12.*、3.01.*、3.02.*、3.03.*、3.04.*、3.04.*、3.05.*、3.06.* です。

既存の Aurora MySQL 2.* データベースクラスターを Aurora MySQL 2.11.5 にアップグレードできます。また、現在サポートされている Aurora MySQL バージョン 2 の下位リリースからスナップショットを Aurora MySQL 2.11.5 に復元することもできます。

Aurora MySQL グローバルデータベースをバージョン 2.11.* にアップグレードする場合、プライマリ DB クラスターとセカンダリ DB クラスターを、パッチレベルを含めてまったく同じバージョンにアップグレードする必要があります。Aurora グローバルデータベースのマイナーバージョンのアップグレードの詳細については、「マイナーバージョンのアップグレード」を参照してください。

Aurora MySQL 2.11.* へのエンジンバージョンのインプレースアップグレードが実行された直後に、db.r4、db.r5、db.t2、db.t3 の DB インスタンスクラスで影響を受けるすべてのインスタンスにオペレーティングシステムのアップグレードが自動的に適用されます (インスタンスが古いオペレーティングシステムバージョンを実行している場合)。マルチ AZ DB クラスターでは、まず、すべてのリーダーインスタンスがオペレーティングシステムのアップグレードを適用します。最初のリーダーインスタンスでオペレーティングシステムのアップグレードが完了すると、フェイルオーバーが発生し、以前のライターインスタンスがアップグレードされます。

> **Note**
>
> メジャーバージョンのアップグレード中は、Aurora グローバルデータベースにはオペレーティングシステムのアップグレードは自動適用されません。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

修正されたセキュリティ問題と CVE:

このリリースには、次の CVE 修正が含まれています。

- 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

可用性の向上:

- Aurora ストレージと通信するコンポーネントの不具合により、Aurora MySQL ライター DB インスタンスがフェイルオーバーする可能性がある問題を修正しました。この不具合は、ソフトウェアの更新後に DB インスタンスと基盤となるストレージ間の通信が切断されたために発生します。
- まれに、リーダー DB インスタンスが再起動することがある問題を修正しました。
- 監査ログ記録のスレッドが原因でロック競合が発生し、CPU 使用率が高くなり、クライアントアプリケーションがタイムアウトする問題を修正しました。

全般的な機能強化:

- Aurora DB クラスターボリュームからデータを読み取る際の一時的なネットワークの問題により、parallel クエリが失敗する可能性がある問題を修正しました。
- 監査ログファイル管理に関連する問題を修正しました。これにより、ログファイルにアクセスしてダウンロードやローテーションができなくなり、場合によっては CPU 使用率が高くなる可能性があります。
- 書き込み転送を使用すると、`Threads_running``information_schema`および`performance_schema`グローバルステータステーブルの変数の値が不正確になることがある問題を修正しました。

アップグレードと移行:

- RDS for MySQL 5.7 に移行した Aurora MySQL DB クラスターでバイナリログのレプリケーションを開始できない問題を修正しました。
- Aurora MySQL バージョン 3 へのメジャーバージョンアップグレード中に、データベースイベントスケジューラを無効にしました。これにより、メジャーバージョンアップグレードの進行中にイベント実行によってデータベースが変更されるのを防ぐことができます。

Aurora MySQL バージョン 2 ではサポートされていない機能

以下の機能は、現時点では Aurora MySQL バージョン 2 (MySQL 5.7 互換) ではサポートされていません。

- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2023-10-17 (バージョン 2.11.4、MySQL 5.7.12 互換)

バージョン: 2.11.4

Aurora MySQL 2.11.4 は一般公開されています。Aurora MySQL 2.11 バージョンは、MySQL 5.7.12 と互換性があります。コミュニティ版の変更点の詳細については、「Changes in MySQL 5.7.12 (2016-04-11, General Availability)」を参照してください。

現在サポートされている Aurora MySQL リリース
は、2.07.9、2.07.10、2.11.*、2.12.*、3.01.*、3.02.*、3.03.*、3.04.* です。

既存の Aurora MySQL 2.* データベースクラスターを Aurora MySQL 2.11.4 にアップグレードできます。現在サポートされている Aurora MySQL リリースから取得したスナップショットを Aurora MySQL 2.11.4 で復元することもできます。

Aurora MySQL グローバルデータベースをバージョン 2.11.* にアップグレードする場合、プライマリ DB クラスターとセカンダリ DB クラスターを、パッチレベルを含めてまったく同じバージョンにアップグレードする必要があります。Aurora グローバルデータベースのマイナーバージョンのアップグレードの詳細については、「マイナーバージョンのアップグレード」を参照してください。

Aurora MySQL 2.11.* へのエンジンバージョンのインプレースアップグレードが実行された直後に、db.r4、db.r5、db.t2、db.t3 の DB インスタンスクラスで影響を受けるすべてのインスタンスにオペレーティングシステムのアップグレードが自動的に適用されます (インスタンスが古いオペレーティングシステムバージョンを実行している場合)。マルチ AZ DB クラスターでは、まず、すべてのリーダーインスタンスがオペレーティングシステムのアップグレードを適用します。最初のリーダーインスタンスでオペレーティングシステムのアップグレードが完了すると、フェイルオーバーが発生し、以前のライターインスタンスがアップグレードされます。

> **Note**
>
> メジャーバージョンのアップグレード中は、Aurora グローバルデータベースにはオペレーティングシステムのアップグレードは自動適用されません。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

以下のセキュリティの問題と CVE の修正:

- 監査ログのローテーション中に報告されたイベントが監査ログに書き込まれない場合がある問題を修正しました。
- CVE-2022-24407

可用性の向上:

- パラレルクエリを使用する Aurora MySQL データベースインスタンスで、多数のパラレルクエリを同時に実行すると、データベースが再起動する問題を修正しました。
- I/O 負荷の高い読み取りワークロードの実行中にデータベースインスタンスが再起動する問題を修正しました。
- 削除されたテーブルに属するデータベースページを読み取ろうとした場合に、データベースインスタンスが再起動する問題を修正しました。
- ライターインスタンスがデータベースボリュームを拡大させ、160 GB の倍数に達すると、リーダーインスタンスが再起動する問題を修正しました。
- データベースが内部システムテーブルでトリガーを作成または削除しているときにライターインスタンスが再起動すると、データベースクラスターが使用できない状態になる問題を修正しました。
- 全文インデックスを含むテーブルに対してデータ操作言語 (DML) クエリを実行すると、リーダーインスタンスが再起動する問題を修正しました。
- Aurora パラレルクエリの実行プランを利用するクエリの実行中に、リーダーインスタンスが再起動する問題を修正しました。
- 全文検索 (FTS) インデックスのあるテーブルで `OPTIMIZE TABLE` クエリを実行しているときに、ライターインスタンスが再起動する問題を修正しました。
- この Aurora MySQL バージョンでは高速挿入が有効になっていません。、、などのクエリの実行時に不整合が生じる可能性があるためです。`INSERT INTO SELECT FROM`高速挿入最適化の詳細については、「Amazon Aurora MySQL のパフォーマンス強化」を参照してください。

全般的な機能強化:

- 2.11.* より前のバージョンからアップグレードした後で、小さいリードレプリカインスタンスでレプリケーションの遅延が長くなる問題を修正しました。
- procs_priv grant テーブルを参照して、ストアドルーチンに関するリクエストを検証すると、ログメッセージが過剰に生成されることがある問題を修正しました。
- ハッシュ結合の最適化を使用してクエリを実行する際に、データベースインスタンスがメモリを過剰に使用する原因となる、メモリ管理の問題を修正しました。
- SELECT ステートメントを (古い ha_partition パーティションハンドラーをサポートするバージョンの MySQL で作成した) パーティションテーブルで実行し、クエリプランナーでパラレルクエリが選択されると、データベースが再起動する問題を修正しました。
- 書き込み転送が有効になっていると、クライアントからデータベースへの新しい接続が確立されないことがある問題を修正しました。
- デフォルトのデータベースを USE コマンドで定義していない状態で、ソースのバイナリログファイルに書き込まれる QUERY イベントを Aurora MySQL のバイナリログ (binlog) レプリカが実行した場合に、バイナリログのレプリケーションの遅延が軽減されます。
- GROUP BY 句を使用し、aurora_parallel_query パラメータを ON に指定した SELECT クエリを実行すると、不正確な結果が返されることがある、インデックススキャンに関する問題を修正しました。
- セッションレベルのバイナリログ記録を有効化または無効化できるようになりました。「Amazon Aurora ユーザーガイド」の「ストアドプロシージャ - レプリケーション」を参照してください。
- ソースのシステム変数 server_uuid が指定されていないか、その値が無効な場合に、バイナリログレプリカが再起動する問題を修正しました。
- セッションレベルでバイナリログの形式を設定できるようになりました。「Amazon Aurora ユーザーガイド」の「ストアドプロシージャ - レプリケーション」を参照してください。
- innodb_flush_log_at_trx_commitパラメータが 1 CommitLatency CloudWatch に設定されていない場合にメトリックスが誤って報告されることがある問題を修正しました。
- InnoDB の統計情報が古くならないように、問題を修正しました。古くなると、最適ではないクエリ実行プランが生成され、クエリの実行時間が長くなる可能性があります。
- 接続先のバイナリログ (binlog) コンシューマーが、重複するバイナリログレプリケーションサーバー ID を使用している場合に、データベースが再起動する問題を修正しました。

MySQL Community Edition でのバグ修正の統合

このリリースには、以下を含め、5.7.12 までのコミュニティ版のバグ修正がすべて反映されています。詳細については、「Aurora MySQL 2.x データベースエンジンの更新で修正された MySQL のバグ」を参照してください。

- レプリケーション: 一部のバイナリログイベントが必ずしも正しく処理されていませんでした。(バグ #34617506)
- バックグラウンドの TLS 証明書のローテーションが原因で CPU 使用率が高くなる問題を修正しました (コミュニティのバグ修正 #34284186)。
- プリペアドステートメントで、一部のタイプのサブクエリでサーバーが終了する場合がありました。(バグ #33100586)

Aurora MySQL バージョン 2 ではサポートされていない機能

以下の機能は、現時点では Aurora MySQL バージョン 2 (MySQL 5.7 互換) ではサポートされていません。

- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更

- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- CREATE TABLESPACE SQL ステートメント

Aurora MySQL データベースエンジンの更新 2023-06-09 (バージョン 2.11.3、MySQL 5.7.12 互換)

バージョン: 2.11.3

Aurora MySQL 2.11.3 は一般公開されています。Aurora MySQL 2.11 バージョンは、MySQL 5.7.12 と互換性があります。コミュニティ版の変更点の詳細については、「Changes in MySQL 5.7.12 (2016-04-11, General Availability)」を参照してください。

現在サポートされている Aurora MySQL リリースは、2.07.*、2.11.*、3.01.*、3.02.*、3.03.* です。

既存の Aurora MySQL 2.* データベースクラスターを Aurora MySQL 2.11.3 にアップグレードできます。現在サポートされている Aurora MySQL リリースから取得したスナップショットを Aurora MySQL 2.11.3 で復元することもできます。

Aurora MySQL グローバルデータベースをバージョン 2.11.* にアップグレードする場合、プライマリ DB クラスターとセカンダリ DB クラスターを、パッチレベルを含めてまったく同じバージョンにアップグレードする必要があります。Aurora グローバルデータベースのマイナーバージョンのアップグレードの詳細については、「マイナーバージョンのアップグレード」を参照してください。

Aurora MySQL 2.11.* へのエンジンバージョンのインプレースアップグレードが実行された直後に、db.r4、db.r5、db.t2、db.t3 の DB インスタンスクラスで影響を受けるすべてのインスタンスにオペレーティングシステムのアップグレードが自動的に適用されます (インスタンスが古いオペレーティングシステムバージョンを実行している場合)。マルチ AZ DB クラスターでは、まず、すべてのリーダーインスタンスがオペレーティングシステムのアップグレードを適用します。最初のリーダーインスタンスでオペレーティングシステムのアップグレードが完了すると、フェイルオーバーが発生し、以前のライターインスタンスがアップグレードされます。

> **Note**
>
> メジャーバージョンのアップグレード中は、Aurora グローバルデータベースにはオペレーティングシステムのアップグレードは自動適用されません。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS Support AWS から サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

以下のセキュリティの問題と CVE の修正:

- Aurora MySQL が使用するデフォルトの SSL 暗号を更新し、安全性の低い DES-CBC3-SHA 値を SSL_CIPHER データベースパラメータから除外しました。暗号の削除により SSL DES-CBC3-SHA 接続の問題が発生した場合は、このリストから該当する安全な暗号を使用してくださいConfiguringCipherSuites。MySQL クライアント接続の暗号設定については、MySQL ドキュメントの「Connection Cipher Configuration」を参照してください。
- CVE-2023-21963
- CVE-2023-21912
- CVE-2023-0215
- CVE-2022-43551
- CVE-2022-37434

可用性の向上:

- データベースアクティビティストリーム (DAS) イベントの暗号化が原因で、データベースが再起動する問題を修正しました。
- データ定義言語 (DDL) クエリの実行中にデータベースの再起動が失敗する原因となる 2 つの問題を修正しました。
- この Aurora MySQL バージョンではSELECT、高速挿入は有効になっていません。これは、INSERT INTO、、 などのクエリを実行するときに不整合が発生する可能性があるためですFROM。高速挿入の最適化の詳細については、「Amazon Aurora MySQL のパフォーマンス強化」を参照してください。

全般的な機能強化:

- リレーログファイルへの書き込み時の競合を減らすために、バイナリログレプリカに対するファイル管理パフォーマンスの最適化を導入しました。
- `aurora_disable_hash_join` パラメータを 1 または ON に設定しても、オプティマイザーによるハッシュ結合の使用を阻止できない場合がある問題を修正しました。
- `information_schema` メトリクスで `buffer_pool_read_requests` カウンターが誤って報告されることがある問題を修正しました。
- `LOAD FROM S3` または `SELECT INTO S3` オペレーションを実行するとローカルストレージがいっぱいになることがある問題を修正しました。この問題は、CPU 使用率の上昇、メモリ不足によるデータベースの再起動、これらのクエリのレイテンシーの増大につながる可能性もあります。

Aurora MySQL バージョン 2 ではサポートされていない機能

以下の機能は、現時点では Aurora MySQL バージョン 2 (MySQL 5.7 互換) ではサポートされていません。

- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更

- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- CREATE TABLESPACE SQL ステートメント

Aurora MySQL データベースエンジンの更新 2023-03-24 (バージョン 2.11.2、MySQL 5.7.12 互換)

バージョン: 2.11.2

Aurora MySQL 2.11.2 は一般公開されています。Aurora MySQL 2.11 バージョンは、MySQL 5.7.12 と互換性があります。コミュニティ版の変更点の詳細については、「Changes in MySQL 5.7.12 (2016-04-11, General Availability)」を参照してください。

現在サポートされている Aurora MySQL リリースは、2.07.*、2.11.*、3.01.*、3.02.*、3.03.* です。

既存の Aurora MySQL 2.* データベースクラスターを Aurora MySQL 2.11.2 にアップグレードできます。現在サポートされている Aurora MySQL リリースから取得したスナップショットを Aurora MySQL 2.11.2 で復元することもできます。

Aurora MySQL グローバルデータベースをバージョン 2.11.* にアップグレードする場合、プライマリ DB クラスターとセカンダリ DB クラスターを、パッチレベルを含めてまったく同じバージョンにアップグレードする必要があります。Aurora グローバルデータベースのマイナーバージョンのアップグレードの詳細については、「マイナーバージョンのアップグレード」を参照してください。

Aurora MySQL 2.11.* へのエンジンバージョンのインプレースアップグレードが実行された直後に、db.r4、db.r5、db.t2、db.t3 の DB インスタンスクラスで影響を受けるすべてのインスタンスにオペレーティングシステムのアップグレードが自動的に適用されます (インスタンスが古いオペレーティングシステムバージョンを実行している場合)。マルチ AZ DB クラスターでは、まず、すべてのリーダーインスタンスがオペレーティングシステムのアップグレードを適用します。最初のリーダーインスタンスでオペレーティングシステムのアップグレードが完了すると、フェイルオーバーが発生し、以前のライターインスタンスがアップグレードされます。

> **Note**
>
> メジャーバージョンのアップグレード中は、Aurora グローバルデータベースにはオペレーティングシステムのアップグレードは自動適用されません。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS Support AWS から サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

全般的な機能強化:

- バイナリログレプリケーションを使用する DB インスタンスで、バイナリログレプリケーションのコンシューマーが複数アタッチされていると CPU 使用率が増加し、接続障害が発生する可能性がある問題を修正しました。
- プライマリデータベースライターが Aurora MySQL バージョン 2.10 を使用している場合、グローバルデータベースのセカンダリリージョンのリーダーインスタンスが Aurora MySQL バージョン 2.11 へのアップグレード後に同期しなくなる問題を修正しました。

可用性の向上:

- この Aurora MySQL バージョンではSELECT、高速挿入は有効になっていません。これは、INSERT INTO、、 などのクエリを実行するときに不整合が発生する問題があるためですFROM。高速挿入の最適化の詳細については、「Amazon Aurora MySQL のパフォーマンス強化」を参照してください。

Aurora MySQL バージョン 2 ではサポートされていない機能

以下の機能は、現時点では Aurora MySQL バージョン 2 (MySQL 5.7 互換) ではサポートされていません。

- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2023-02-14 (バージョン 2.11.1、MySQL 5.7.12 互換)

バージョン: 2.11.1

Aurora MySQL 2.11.1 は一般公開されています。Aurora MySQL 2.11 バージョンは、MySQL 5.7.12 と互換性があります。コミュニティ版の変更点の詳細については、「Changes in MySQL 5.7.12 (2016-04-11, General Availability)」を参照してください。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.07.*、2.09.*、2.10.*、2.11.*、3.01.*、3.02.*. です。

既存の Aurora MySQL 2.* データベースクラスターを Aurora MySQL 2.11.1 にアップグレードできます。Aurora MySQL バージョン 1 を実行しているクラスターの場合、既存の Aurora MySQL 1.23 以降のクラスターを 2.11.1 に直接アップグレードできます。現在サポートされている Aurora MySQL リリースから取得したスナップショットを Aurora MySQL 2.11.1 で復元することもできます。

Aurora MySQL グローバルデータベースをバージョン 2.11.* にアップグレードし、書き込み転送を有効にした場合、引き続き書き込み転送を使用するには、プライマリ DB クラスターとセカンダリ DB クラスターをパッチレベルを含めてまったく同じバージョンにアップグレードする必要があります。Aurora グローバルデータベースのマイナーバージョンのアップグレードの詳細については、「マイナーバージョンのアップグレード」を参照してください。

Aurora MySQL 2.11.* へのエンジンバージョンのインプレースアップグレードが実行された直後に、db.r4、db.r5、db.t2、db.t3 の DB インスタンスクラスで影響を受けるすべてのインスタンスにオペレーティングシステムのアップグレードが自動的に適用されます (インスタンスが古いオペレーティングシステムバージョンを実行している場合)。マルチ AZ DB クラスターでは、まず、すべてのリーダーインスタンスがオペレーティングシステムのアップグレードを適用します。最初のリーダーインスタンスでオペレーティングシステムのアップグレードが完了すると、フェイルオーバーが発生し、以前のライターインスタンスがアップグレードされます。

> **Note**
>
> メジャーバージョンのアップグレード中は、Aurora グローバルデータベースにはオペレーティングシステムのアップグレードは自動適用されません。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS Support AWS から サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

以下のセキュリティの問題と CVE の修正:

マネージド型の環境での処理を微調整するための修正およびその他の機能強化。以下の CVE の追加の修正:

- CVE-2022-32221

- CVE-2021-36222
- CVE-2021-22926
- CVE-2021-2169

可用性の向上:

- 接続の急増によりクエリのレイテンシーが長くなったり、データベースインスタンスが再起動したりする問題を修正しました。
- 大規模な更新オペレーションまたはデータ定義言語 (DDL) ワークロードをライターインスタンスで、同じテーブルセットに対する読み取りオペレーションを Aurora レプリカで同時に実行すると、まれに Aurora レプリカが再起動することがある問題を修正しました。
- 接続が急増すると、接続確立プロセスが完了するまで時間がかかったり、タイムアウトエラーが発生して失敗したりする問題を修正しました。
- 解放可能なメモリが高度な監査ログのローテーションによって減少し、データベースインスタンスの再起動につながる可能性がある問題を修正しました。
- この Aurora MySQL バージョンでは`SELECT`、高速挿入は有効になっていません。これは、`INSERT INTO`、、 などのクエリを実行するときに不整合が発生する問題があるためです`FROM`。高速挿入の最適化の詳細については、「Amazon Aurora MySQL のパフォーマンス強化」を参照してください。

全般的な機能強化:

- SSL サーバーのステータス変数の値が指定されない問題を修正しました。
- データ操作言語 (DML) ステートメントが重複して書き込みを行うと、エラーのログ記録が過剰になり、クエリのレイテンシーが長くなる可能性がある問題を修正しました。

Aurora MySQL バージョン 1 との比較

以下の Amazon Aurora MySQL 機能は Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。

- 関数を同期的に呼び出すためのネイティブ AWS Lambda 関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2022-10-25 (バージョン 2.11.0、MySQL 5.7.12 互換) このバージョンは新規作成には使用不可。

バージョン: 2.11.0

Aurora MySQL 2.11.0 は一般公開されています。Aurora MySQL 2.x バージョンは、MySQL 5.7.12 と互換性があります。コミュニティ版の変更点の詳細については、「Changes in MySQL 5.7.12 (2016-04-11, General Availability)」を参照してください。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、2.11.*、3.01.*、3.02.* です。

既存の Aurora MySQL 2.* データベースクラスターを Aurora MySQL 2.11.0 にアップグレードできます。Aurora MySQL バージョン 1 を実行しているクラスターの場合、既存の Aurora MySQL 1.23 以降のクラスターを 2.11.0 に直接アップグレードできます。現在サポートされている Aurora MySQL リリースから取得したスナップショットを Aurora MySQL 2.11.0 で復元することもできます。

Aurora MySQL グローバルデータベースをバージョン 2.11.* にアップグレードし、書き込み転送を有効にした場合、引き続き書き込み転送を使用するには、プライマリ DB クラスターとセカンダリ DB クラスターをパッチレベルを含めてまったく同じバージョンにアップグレードする必要があります。Aurora グローバルデータベースのマイナーバージョンのアップグレードの詳細については、「マイナーバージョンのアップグレード」を参照してください。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS Support AWS から サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

以下のセキュリティの問題と CVE の修正:

マネージド型の環境での処理を微調整するための修正およびその他の機能強化。以下の CVE の追加の修正:

- 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

新機能:

- Aurora MySQL バージョン 2.11 のリリースに伴い、新しいオペレーティングシステムのアップグレードが提供されています。バージョン 2.11 にアップグレードした後、この保留中の OS アップデートをすべての Aurora MySQL データベースインスタンスに適用することをお勧めします。詳細については、「オペレーティングシステムアップデートの操作」を参照してください。
- 新しい動的な構成オプション innodb_deadlock_detect を使用して、デッドロック検出を無効にすることができます。高並行性システムでは、多数のスレッドが同じロックを待機すると、デッド

ロック検出によって速度が低下する可能性があります。デッドロック検出を無効にして、デッドロックの発生時は innodb_lock_wait_timeout 設定でトランザクションをロールバックした方が場合によっては効率的です。(バグ #23477773) Innodb デッドロック検出の詳細については、MySQL のドキュメントを参照してください。

- MySQL 8.0 の UUID_TO_BIN、BIN_TO_UUID、IS_UUID 関数が追加されました。これらの関数の使用方法の詳細については、「MySQL Miscellaneous function」を参照してください。
- オプティマイザヒントのサポートが追加され、ユーザーが Aurora MySQL パラレルクエリをテーブル単位またはクエリ単位で有効または無効にできるようになりました。
  - Amazon Aurora MySQL のパラレルクエリの使用
  - Aurora MySQL のヒント
- R3 インスタンスタイプのサポートが削除されました。
- R6i インスタンスのサポートが追加されました。

可用性の向上:

- エラーログに書き込まれたバイナリログファイルの名前と位置が正確でないことが原因で、データベースクラスターでクロスリージョンの論理レプリケーションが妨げられる問題を修正しました。この問題は、DDL ステートメントの実行後にエンジンが再起動すると発生する場合があります。
- GRANT や FLUSH などのアクセスコントロールリスト (ACL) ステートメントをライターインスタンスで実行すると、まれに Aurora リーダーインスタンスが再起動することがある問題を修正しました。ユーザーと ACL オペレーション (アクセス許可の変更など) の数が多いリーダーインスタンスが、この問題の影響を受けやすい傾向にあります。
- 別のトランザクションによって削除中の行にトランザクションがアクセスすると、まれにライターインスタンスが再起動またはフェイルオーバーすることがある問題を修正しました。
- 全文フレーズ検索のパフォーマンスが向上し、全文インデックスがあるテーブル内のフレーズ検索にかかる時間が大幅に短縮されました。
- ライターインスタンスの再起動後に、復旧に時間がかかり、その後再び再起動する問題を修正しました。この問題は、初回の再起動時にデータベースにコミットされていない行が大量にある場合に発生します。
- デッドロック検出スレッドが停止した際にセマフォの待機時間が長くなり、まれにデータベースサーバーが再起動することがある問題を修正しました。
- I/O スレッドがデッドロック状態になった場合にセマフォの待機時間が長くなり、まれにデータベースサーバーが再起動することがある問題を修正しました。

- この Aurora MySQL バージョンではSELECT、高速挿入は有効になっていません。これは、INSERT INTO、、 などのクエリを実行するときに不整合が発生する問題があるためですFROM。高速挿入の最適化の詳細については、「Amazon Aurora MySQL のパフォーマンス強化」を参照してください。

全般的な機能強化:

- 以下の条件がすべて当てはまる場合に、データベースサーバーが再起動する問題を修正しました。
  - ALLOW_INVALID_DATES が SQL MODE で無効になっている。
  - データベースサーバーで、DATETIME 型の値が無効な (例えば、月が 1 ~ 12 ではない) INSERT、UPDATE、DELETE、または SELECT ステートメントを処理している。
- log-bin が OFF に設定されている場合にバイナリログの保持期間が守られず、ストレージ使用率が予想よりも高くなる問題を修正しました。この修正の適用後は、バイナリログが保持期間に基づいて消去されます。バイナリログの保持期間を設定する方法については、「Aurora MySQL ユーザーガイド」を参照してください。
- GRANT、FLUSH PRIVILEGES など、特定のデータ制御言語 (DCL) の SQL ステートメントをデータベースインスタンスで実行すると、そのインスタンスで解放可能なメモリが減少する場合がある問題を修正しました。このようなステートメントを頻繁に使用すると、解放可能なメモリが減少し続け、 out-of-memory 問題によりデータベースインスタンスが再起動する可能性があります。このようなステートメントをライターインスタンスで使用すると、リーダーインスタンスでも解放可能なメモリが減る可能性があります。
- 読み取り I/O オペレーションの数を最小限に抑えるため、リレーログから実行される読み取り用の読み取りバッファサイズを大きくしました。その結果、I/O スレッドと SQL スレッド間の競合が減少します。
- mysql.rds_rotate_slow_log ストアドプロシージャが失敗し、「Table 'mysql.slow_log_backup' doesn't exist」というエラーメッセージが表示される問題を修正しました。
- クエリキャッシュを無効化し過ぎた結果、リードレプリカがクエリキャッシュからではなくディスクからデータを読み取る必要が生じ、リードレプリカの CPU 使用率とレイテンシーが想定以上に増大する問題を修正しました。
- ユーザーがリーダーインスタンスで INSTALL PLUGIN コマンドと UNINSTALL PLUGIN コマンドを実行できるため、LOCK_plugin、LOCK_system_variables_hash、LOCK_global_system_variables でデッドロックが発生する可能性がある問題を修正しました。これらのステートメントは、データベースクラスター内のライターインスタンスでのみ実行できるようになりました。

- バイナリログ記録が有効な場合に、クラスターのコミットレイテンシーが想定以上に長くなる問題を修正しました。これは、大きな (サイズが 500 MB を超える) バイナリログイベントを生成するすべてのトランザクションに影響します。
- INFORMATION_SCHEMA.INNODB_METRICS テーブルの trx_active_transactions メトリクスの値が正しくなくなる問題を修正しました。
- 大規模なトランザクションでセーブポイントへのロールバックの実行中にバイナリログファイルの整合性がなくなり、論理レプリケーションが停止する問題を修正しました。
- 整合性のあるマスクシークレットを使用して slow-query-log、一般ログおよび監査ログの認証情報ハッシュをマスクします。これは aurora_mask_password_hashes_type パラメータで設定できます。
- ユーザーが観察したイベントで、ダウンタイムのない再起動 (ZDR) の期間が誤って報告される問題を修正しました。
- mysql_rds_import_binlog_ssl_material への呼び出しが失敗し、MySQL サーバーエラー 1457 が発生する問題を修正しました。
- ダンプスレッドの初期化が、バイナリログを消去するスレッドでデッドロック状態になる可能性がある問題を修正しました。これにより、アクティブなバイナリログファイルのローテーションが停止し、代わりに肥大化したり、新しいバイナリログレプリカの接続で問題が発生したりする可能性があります。
- クエリキャッシュが Aurora リードレプリカで古い結果を返すことがある問題を修正しました。

MySQL Community Edition でのバグ修正の統合

このリリースには、以下を含め、5.7 までのコミュニティ版のバグ修正がすべて反映されています。詳細については、「Aurora MySQL 2.x データベースエンジンの更新で修正された MySQL のバグ」を参照してください。

- パフォーマンススキーマのステートメントイベントテーブル (events_statements_current など) から文字セット情報を読み取るコードが、その文字セット情報への同時書き込みを阻止できなかった問題を修正しました。その結果、SQL クエリテキストの文字セットが無効になり、サーバーが終了する可能性がありました。今回の修正により、無効な文字セットがあると SQL_TEXT 列が切り捨てられ、サーバーの終了が阻止されるようになりました。(バグ #23540008)
- InnoDB: コミュニティバグ #25189192、バグ #84038 の修正のバックポート。テーブルを別のスキーマに移動する RENAME TABLE オペレーションの後、InnoDB が INNODB_SYS_DATAFILES データディクショナリテーブルを更新できなかった問題を修正しました。その結果、テーブルスペースデータファイルが見つからないというエラーが再起動時に発生していました。

- InnoDB: 新しいインデックスの追加時に、サーバーが内部定義の外部キーインデックスを削除し、仮想生成列に定義されたセカンダリインデックスを外部キーインデックスとして使用しようとすると、サーバーが終了する問題を修正しました。InnoDB は、外部キー制約が仮想生成列で定義されたセカンダリインデックスを参照することを許可するようになりました。(バグ #23533396)
- 2 つのセッションが INSERT ... ON DUPLICATE KEY UPDATE オペレーションを同時に実行している場合にデッドロック状態になる問題を修正しました。タプルの部分的ロールバック中に、別のセッションがタプルを更新する可能性があります。このバグの修正に伴い、バグ #11758237、バグ #17604730、バグ #20040791 の修正が取り消されます。(バグ #25966845)
- コミュニティバグ #27407480 の修正のバックポート: automatic_sp_privileges が有効になっていても、EXECUTE 権限と ALTER ROUTINE 権限がルーチン作成者に正しく付与されない問題を修正しました。
- コミュニティバグ #24671968 の修正のバックポート: WHERE 句に相関サブクエリが含まれていて、テーブルのセカンダリインデックスが選択リストの列とサブクエリの列に定義されており、`GROUP BY` または `DISTINCT` でクエリにルースインデックススキャンを適用できる場合に、クエリが誤った結果を生成する可能性がある問題を修正しました。
- 外部キーを持つ複数のテーブルに対して複数テーブルの DELETE ステートメントを発行すると、レプリケーションが中断される問題を修正しました。(バグ #80821)
- slave_skip_errors が有効になっていても、特殊なケースで特定のスレーブエラーが無視されない問題を修正しました。行ベースのレプリケーションを実行しているサーバーでテーブルを開いたりロックしたりできない場合や、フィールド変換に失敗した場合、エラーは重大と見なされ、slave_skip_errors の状態は無視されます。今回の修正により、slave_skip_errors が有効な場合は、トランザクションの適用中に報告されたすべてのエラーが正しく処理されるようになりました。(バグ #70640、バグ #17653275)
- `SET PASSWORD` ステートメントが MySQL 5.6 マスターから MySQL 5.7 スレーブにレプリケートされる場合や、log_builtin_as_identified_by_password システム変数が ON に設定された MySQL 5.7 マスターから MySQL 5.7 スレーブにレプリケートされる場合に、パスワードハッシュ自体もスレーブに格納される前にハッシュされる問題を修正しました。この問題は修正され、レプリケートされたパスワードハッシュは、スレーブに当初渡されたままの形で保存されます。(バグ #24687073)
- 多数のレベルの JSON 配列、オブジェクト、またはその両方でラップされた大きなサブドキュメントで構成される JSON 値のシリアル化が、完了するまで長時間かかることがあった問題を修正しました。(バグ #23031146)
- 構文エラーなどの理由で解析できないステートメントは、スロークエリログに書き込まれなくなりました。(バグ #33732907)

Aurora MySQL バージョン 1 との比較

以下の Amazon Aurora MySQL 機能は Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- 関数を同期的に呼び出すためのネイティブ AWS Lambda 関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2022-11-01 (バージョン 2.10.3) (廃止)

バージョン: 2.10.3

Aurora MySQL 2.10.3 は一般公開されています。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、2.11.*、3.01.*、3.02.* です。

既存の Aurora MySQL 2.* データベースクラスターを Aurora MySQL 2.10.3 にアップグレードできます。Aurora MySQL バージョン 1 を実行しているクラスターの場合、既存の Aurora MySQL 1.23 以降のクラスターを 2.10.3 に直接アップグレードできます。現在サポートされている Aurora MySQL リリースから取得したスナップショットを Aurora MySQL 2.10.3 で復元することもできます。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

以下のセキュリティの問題と CVE の修正:

マネージド型の環境での処理を微調整するための修正およびその他の機能強化。以下の CVE の追加の修正:

- CVE-2022-21444
- CVE-2022-21344
- CVE-2022-21304
- CVE-2022-21245

- CVE-2021-36222
- CVE-2021-22926

全般的な機能強化:

- デッドロック検出スレッドが停止した際にセマフォの待機時間が長くなり、まれにデータベースサーバーが再起動することがある問題を修正しました。
- GRANT、FLUSH PRIVILEGES など、特定のデータ制御言語 (DCL) の SQL ステートメントをデータベースインスタンスで実行すると、そのインスタンスで解放可能なメモリが減少する場合がある問題を修正しました。このようなステートメントを頻繁に使用すると、解放可能なメモリが減り続けてメモリ不足になり、データベースインスタンスが再起動する可能性があります。このようなステートメントをライターインスタンスで使用すると、リーダーインスタンスでも解放可能なメモリが減る可能性があります。
- "wait/io/aurora_respond_to_client" performance_schema 待機イベントが有効な場合に、データベースインスタンスの負荷が高くなると、"performance_schema.events_waits_summary_global_by_event_name" テーブルに対するクエリに時間がかかる問題を修正しました。
- セカンダリインデックスの制約違反が原因で、トランザクションが部分的にロールバックした場合に、まれにデータベースサーバーが停止し、再起動することがある問題を修正しました。
- 別のトランザクションによって削除中の行にトランザクションがアクセスすると、まれにライターインスタンスが再起動またはフェイルオーバーすることがある問題を修正しました。
- I/O スレッドがデッドロック状態になった場合にセマフォの待機時間が長くなり、まれにデータベースサーバーが再起動することがある問題を修正しました。
- Unix ソケットロックファイルを使用している場合に、まれに、フェイルオーバー中にリードレプリカが再起動することがある問題を修正しました。
- クエリキャッシュを無効化し過ぎた結果、リードレプリカがクエリキャッシュからではなくディスクからデータを読み取る必要が生じ、リードレプリカの CPU 使用率とレイテンシーが想定以上に増大する問題を修正しました。

MySQL Community Edition でのバグ修正の統合

このリリースには、以下を含め、5.7 までのコミュニティ版のバグ修正がすべて反映されています。詳細については、「Aurora MySQL 2.x データベースエンジンの更新で修正された MySQL のバグ」を参照してください。

- パフォーマンススキーマのステートメントイベントテーブル (events_statements_current など) から文字セット情報を読み取るコードが、その文字セット情報への同時書き込みを阻止できなかった問題を修正しました。その結果、SQL クエリテキストの文字セットが無効になり、サーバーが終了する可能性がありました。今回の修正により、無効な文字セットがあると SQL_TEXT カラムが切り捨てられ、サーバーの終了が阻止されるようになりました。(バグ #23540008)
- プライマリキーが 1024 バイト超の一時テーブルを UPDATE が必要とし、そのテーブルが InnoDB を使用して作成された場合に、サーバーが終了することがある問題を修正しました。(バグ #25153670)
- 2 つのセッションが INSERT ... ON DUPLICATE KEY UPDATE オペレーションを同時に実行している場合にデッドロック状態になる問題を修正しました。タプルの部分的ロールバック中に、別のセッションがタプルを更新する可能性があります。このバグの修正に伴い、バグ #11758237、バグ #17604730、バグ #20040791 の修正が取り消されます。(バグ #25966845)

Aurora MySQL バージョン 1 との比較

以下の Amazon Aurora MySQL 機能は Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2022-01-26 (バージョン 2.10.2) (廃止)

バージョン: 2.10.2

Aurora MySQL 2.10.2 は一般公開されています。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

既存の Aurora MySQL 2.* データベースクラスタを Aurora MySQL 2.10.0 にアップグレードできます。Aurora MySQL バージョン 1 を実行しているクラスターの場合、既存の Aurora MySQL 1.23 以降のクラスターを 2.10.0 に直接アップグレードできます。現在サポートされている Aurora MySQL リリースから取得したスナップショットを Aurora MySQL 2.10.0 で復元することもできます。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

以下のセキュリティの問題と CVE の修正:

マネージド型の環境での処理を微調整するための修正およびその他の機能強化。以下の CVE の追加の修正:

- 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

全般的な機能強化:

- 24XL インスタンスクラスでのデータベース IO レイテンシー低減のために、パフォーマンスの最適化を追加しました。
- ECDHE SSL 暗号のサポートを追加しました。これらの SSL 暗号を使用するためのクライアントの設定方法については、MySQL ドキュメントの「接続暗号構成」を参照してください。
- Amazon S3、Amazon ML、および AWS Lambda などの他の AWS サービスと、Aurora MySQL との統合に関連する、セキュリティ上の問題を修正しました
- データベースに概ね 1GB を超えるユーザーと特権の組み合わせがある場合、データベースインスタンスの再起動が失敗する不具合を修正しました。

- 範囲述語を含む GROUP BY 句と WHERE 句を使用してクエリを実行した際、データベースがグループ化やソート順序として誤った結果を返す可能性がある、Parallel Query の問題を修正しました。
- Aurora MySQL 1.x (MySQL 5.6 互換) から Aurora MySQL 2.x (MySQL 5.7 互換) への、メジャーバージョンに関するインプレースアップグレード後に、general_log テーブルと slow_log テーブルにアクセスできなくなる不具合を修正しました。
- データベースへのワークロードが重い状態で innodb_trx、innodb_locks、または innodb_lockwaits テーブルがクエリされた際、まれにデータベースインスタンスが再起動する不具合を修正しました。こういったテーブルは、Performance Insights などのモニタリングツールによりクエリされる場合があります。
- 次の条件がすべて満たされ際に、既存の行の TIMESTAMP 列の値が最新のタイムスタンプに更新される問題を修正しました。

  1. テーブルのトリガーが存在する。
  2. ON DUPLICATE KEY UPDATE 句があるテーブルに対して INSERT が実行される。
  3. 挿入された行が、UNIQUE インデックスまたは PRIMARY KEY で重複値違反を引き起こす。
  4. 1 つ以上の列が TIMESTAMP データ型であり、デフォルト値は CURRENT_TIMESTAMP です。

- まれに、バイナリログが有効なインスタンスに対し、バイナリログレプリカが接続されないことがある問題を修正しました。
- バイナリログが有効になっているインスタンスで実行した場合に、まれにトランザクションがコミット不能となる問題を修正しました。
- バイナリロが有効になっているインスタンスに対し、新しい接続を確立できないという問題を修正しました。
- ゼロダウンタイムのパッチの適用後に再起動することで、ローカルストレージを使い切った状態になると、過剰な内部ログが発生する問題を修正しました。
- 特定の DDL および DCL ステートメントをレプリケートした際、バイナリログレプリカが HA_ERR_FOUND_DUPP_KEY エラーを発生して停止する問題を修正しました。この問題は出典インスタンスが MIXED バイナリログ記録形式と READ COMMITTED または READ UNCOMMITTED 分離レベルで構成されている場合に発生します。
- マルチスレッドレプリケーションが有効になっている場合、バイナリログのレプリケーション I/O スレッドがプライマリインスタンスに追いつくことができなくなる問題を修正しました
- データベースインスタンスに対し大量のアクティブな接続がある場合、まれに、CloudWatch CommitLatency メトリクスが誤って報告される可能性がある問題を修正しました。

- LOAD FROM S3、または SELECT INTO S3 の実行時に Graviton インスタンス上のローカルストレージを使い切ってしまう問題を修正しました。
- 外部キーを使用してテーブルをクエリする際に、以下の条件の両方が満たされていると、誤ったクエリ結果が返される問題を修正しました。
  1. クエリキャッシュが有効になっている
  2. そのテーブルでカスケード削除 (または更新) を含むトランザクションがロールバックされている
- まれに Aurora リーダーインスタンスが再起動する問題を修正しました。この問題が発生する可能性は、トランザクションでのロールバック数が増加するとともに高まります。
- セッションを開いて閉じると、Performance Schema でミューテックスの「LOCK_epoch_id_master」の発生回数が増える問題を修正しました。
- ワークロードで、多数のトランザクションが同じ行のセットを同時に更新すると、そのワークロードにおいてデッドロック数が増加する問題を修正しました。
- データベースボリュームが増加し 160GB の倍数に到達すると、まれにインスタンスが再起動する問題を修正しました。
- LIMIT 句を使用して SQL 文を実行した際にデータベースの再起動を引き起こすことがある、Parallel Query の不具合を修正しました。
- READ COMMITTED 分離レベルを使用して XA トランザクションを実行した際、まれにデータベースインスタンスが再起動する問題を修正しました。
- Aurora Read インスタンスを再起動する際に重い DDL ワークロードが存在すると、そのインスタンスが再び起動し直すことがある問題を修正しました。
- Aurora リーダーレプリケーションラグの誤ったレポートの問題を修正しました。
- メモリ内のデータ整合性チェックが失敗した際、ライターインスタンスがまれに再起動する不具合を修正しました。
- Performance Insights (PI) セッションで、その処理が終了してアイドル状態であるにも関わらず、CPU がアクティブに使用されている場合、まれに「Database Load (データベースロード)」グラフが誤って表示される問題を修正しました。
- Parallel Query を使用してクエリを処理すると、まれにデータベースサーバーが再起動する問題を修正しました。
- Global Database のレプリケーション中の競合状態が原因で、まれにプライマリ Global Database クラスターのライターインスタンスが再起動する問題を修正しました。
- データベースインスタンスの再起動中に発生し、再起動を 1 回以上引き起こすことがある問題を修正しました。

MySQL Community Edition でのバグ修正の統合

- テーブル統計に関連するコードでのエラーが、dict0stats.cc ソースファイルでアサーションを発生させるという、InnoDB の問題を修正しました。(バグ #24585978)
- インデックスがオンラインで構築された際に、仮想列上のセカンダリインデックスが破損する問題を修正しました。UPDATE ステートメントでは、この問題を次のように修正します。インデックスレコードの仮想列に NULL 値がに設定されている場合は、クラスターのインデックスレコードからこの値を自動生成します。(バグ #30556595))
- InnoDB のマークされた行を削除する際に、部分的なロールバックが完了する前に外部読み取りロックを取得できてしまう問題を修正しました。部分的なロールバック中、外部読み取りロックにより、暗黙的なロックから明示的なロックへの変換が阻害され、アサーションの失敗を引き起こしていました。(バグ #29195848)
- アカウント内に空のホスト名があることで、サーバーの誤動作の原因となる問題を修正しました。(バグ #28653104)
- InnoDB で、ロック待機中のクエリの中断が原因となりエラーが発生する問題を修正しました。(バグ #28068293)
- レプリケーションにおいて、トランザクション分離レベルが REPEATABLE READ に設定されている場合、インターリーブトランザクションがスレーブアプライヤをデッドロックすることがある問題を修正しました。(バグ #25040331)
- ロック待機のタイムアウトが原因で、バイナリログレプリカの処理が低速になる問題を修正しました。(バグ #27189701)

Aurora MySQL バージョン 1 との比較

以下の Amazon Aurora MySQL 機能は Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。

- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2021-10-21 (バージョン 2.10.1) (廃止)

バージョン: 2.10.1

Aurora MySQL 2.10.1 は一般公開されています。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

既存の Aurora MySQL 2.* データベースクラスタを Aurora MySQL 2.10.0 にアップグレードできます。Aurora MySQL バージョン 1 を実行しているクラスターの場合、既存の Aurora MySQL 1.23 以降のクラスターを 2.10.0 に直接アップグレードできます。現在サポートされている Aurora MySQL リリースから取得したスナップショットを Aurora MySQL 2.10.0 で復元することもできます。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

以下のセキュリティの問題と CVE の修正:

マネージド型の環境での処理を微調整するための修正およびその他の機能強化。以下の CVE の追加の修正:

- 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

可用性の向上:

- 将来のメジャーバージョンアップグレードのためにクラスターをクリーンにシャットダウンする機能が追加されました。

全般的な機能強化:

- 内部診断ログファイル内の過剰な情報メッセージのログが原因で、リーダーインスタンスの CPU 使用率が高くなる問題を修正しました。
- 次の条件がすべて満たされ際に、既存の行の TIMESTAMP 列の値が最新のタイムスタンプに更新される問題を修正しました。
  1. テーブルのトリガーが存在する。
  2. ON DUPLICATE KEY UPDATE 句があるテーブルに対して INSERT が実行される。
  3. 挿入された行が、UNIQUE インデックスまたは PRIMARY KEY で重複値違反を引き起こす。
  4. 1 つ以上の列が TIMESTAMP データ型であり、デフォルト値は CURRENT_TIMESTAMP です。
- バージョン 2.10.0 で確認された、json_merge 関数を使用すると特定のケースでエラーコードが起きる問題を修正しました。具体的には、生成された列を含む DDL で json_merge 関数が使用された際、エラーコード 1305 を返すこ問題です。
- リードレプリカ上のトランザクションのリードビューに対してラージオブジェクトの更新履歴が検証されている際に、まれにリードレプリカが再起動する問題を修正しました。
- メモリ内データの整合性チェックが失敗した際、まれにライターインスタンスが再起動する問題を修正しました。

MySQL Community Edition バグ修正の統合

- CURRENT_TIMESTAMP は、トリガーでゼロを生成します。(バグ #25209512)

Aurora MySQL バージョン 1 との比較

以下の Amazon Aurora MySQL 機能は Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。

- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2021-05-25 (バージョン 2.10.0) (廃止)

バージョン: 2.10.0

Aurora MySQL 2.10.0 は一般公開されています。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリース
は、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

既存の Aurora MySQL 2.* データベースクラスタを Aurora MySQL 2.10.0 にアップグレードできます。Aurora MySQL バージョン 1 を実行しているクラスターの場合、既存の Aurora MySQL 1.23 以降のクラスターを 2.10.0 に直接アップグレードできます。現在サポートされている Aurora MySQL リリースから取得したスナップショットを Aurora MySQL 2.10.0 で復元することもできます。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS Support AWS から サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

以下のセキュリティの問題と CVE の修正:

マネージド型の環境での処理を微調整するための修正およびその他の機能強化。以下の CVE の追加の修正:

- 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

新機能:

- `db.t3.large` インスタンスクラスが Aurora MySQL のためにサポートされるようになりました。
- バイナリログレプリケーション:
  - ライタースレッドとダンプスレッド間の競合を減らすことによって、バイナリログのパフォーマンスを向上させるために、バイナリログ I/O キャッシュを導入しました。詳細については、「Amazon Aurora ユーザーガイド」の「バイナリログのレプリケーションの最適化」を参照してください。
  - Aurora MySQL バージョン 2.08 では、バイナリログ (binlog) 処理が改善され、非常に大きなトランザクションが関与している場合に、クラッシュリカバリ時間とコミット時間のレイテンシーが短縮されました。GTID が有効になっているクラスターでは、これらの改善がサポートされるようになりました。
- リーダーインスタンスの可用性の向上:
  - これまでは、ライターインスタンスが再起動すると、Aurora MySQL クラスター内のすべてのリーダーインスタンスも再起動していました。本日のリリースにより、リージョン内のリーダーインスタンスはライターインスタンスの再起動中も引き続き読み取りリクエストを処理するようになります。これにより、クラスターでの読み取りの可用性が向上します。詳細については、「Amazon Aurora ユーザーガイド」の「Rebooting an Aurora MySQL cluster (version 2.10 and higher)」を参照してください。

    > **Important**
    >
    > Aurora MySQL 2.10 にアップグレードした後、ライターインスタンスを再起動してもクラスター全体の再起動は実行されません。クラスター全体を再起動する場合は、ライターインスタンスを再起動した後に、クラスター内のリーダーインスタンスを再起動します。

- Logical read ahead (LRA) 技術によってリクエストされた事前読み取りページの読み取りのパフォーマンスが改善されました。これは、Aurora ストレージに送信された単一のリクエストで複数のページ読み取りをバッチ処理することによって行われました。その結果、LRA 最適化を使用するクエリは、最大 3 倍速く実行されます。
- ダウンタイムのない再起動とパッチ適用:
  - ダウンタイムのない再起動 (ZDR) とダウンタイムのないパッチ適用 (ZDP) が改善され、バイナリログ記録が有効になっている場合の追加サポートなど、幅広いシナリオで ZDR および ZDP が有効になりました。また、ZDR および ZDP イベントの可視性が向上しました。詳細については、「Amazon Aurora ユーザーガイド」の「ダウンタイムのない再起動 (ZDR) (Amazon Aurora MySQL 用)」および「ダウンタイムのないパッチ適用の使用」を参照してください。

可用性の向上:

- 以前に中断された DDL アクティビティ中に作成されたテンポラリインデックスおよびテーブルが多数データベースに存在する場合に、起動を高速化するための改善。
- 中断された DDL オペレーションのクラッシュリカバリ中に再起動を繰り返すことに関連する複数の問題を修正しました。これには、`DROP TRIGGER` や `ALTER TABLE` が含まれるほか、特に、テーブル内のパーティショニングのタイプまたはパーティションの数を変更する `ALTER TABLE` が含まれます。
- Database Activity Streams (DAS) のログ処理中にサーバーが再起動される状態を引き起こし得る問題を修正しました。
- システムテーブルでの `ALTER` クエリの処理中にエラーメッセージが出力される問題を修正しました。

全般的な機能強化:

- クエリキャッシュがリーダーインスタンスで古い結果を返す状態を引き起こし得る問題を修正しました。
- システム可変 `innodb_flush_log_at_trx_commit` が 0 または 2 に設定されているときに、一部の Aurora コミットメトリクスが更新されない問題を修正しました。
- クエリキャッシュに格納されたクエリ結果がマルチステートメントトランザクションによって更新されない問題を修正しました。
- バイナリログファイルの最終変更タイムスタンプが正しく更新されない状態を引き起こし得る問題を修正しました。これにより、お客様が設定した保存期間に達する前に、バイナリログファイルが早期に消去される可能性があります。

- クラッシュリカバリ後に InnoDB から不正確に報告されたバイナリログファイル名と位置を修正しました。
- `binlog_checksum` パラメータが `NONE` に設定されている場合、大規模なトランザクションが誤ったバイナリログイベントを生成する可能性がある問題を修正しました。
- レプリケートされたトランザクションに DDL ステートメントと多数の行変更が含まれている場合に、バイナリログレプリカがエラーで停止する問題を修正しました。
- テーブルをドロップするときにリーダーインスタンスで再起動する問題を修正しました。
- 大きなトランザクションでバイナリログファイルを消費しようとする場合に、オープンソースのコネクタが失敗する状況を引き起こす問題を修正しました。
- 大きなジオメトリの値を持つテーブル上で空間インデックスを作成した後に、大きなジオメトリ列で誤ったクエリ結果を発生させ得る問題を修正しました。
- データベースは再起動中にテンポラリテーブルスペースを再作成し、関連付けられたストレージ領域の解放と再利用を可能にすることができるようになりました。
- Aurora リーダーインスタンスで `performance_schema` テーブルが切り捨てられない問題を修正しました。
- バイナリログレプリカが `HA_ERR_KEY_NOT_FOUND` エラーで停止する問題を修正しました。
- `FLUSH TABLES WITH READ LOCK` ステートメントの実行時にデータベースが再起動する問題を修正しました。
- Aurora リーダーインスタンスでユーザーレベルのロック機能を使用できない状態を引き起こす問題を修正しました。

MySQL Community Edition バグ修正の統合

- トランザクション分離レベルが REPEATABLE READ に設定されている場合、インターリーブトランザクションがレプリカアプライヤをデッドロックすることがありました。(バグ #25040331)
- ストアドプロシージャに、あるビューを参照するステートメントであって、代わりに別のビューを参照したものが含まれていた場合、プロシージャを複数回正常に呼び出すことができませんでした。(バグ #87858、バグ #26864199)
- 多くの `OR` 条件を持つクエリでは、オプティマイザのメモリ効率がより高くなり、システム可変 range_optimizer_max_mem_size によって課されるメモリ制限を超える可能性が低くなりました。さらに、その可変のデフォルト値は 1,536,000 から 8,388,608 に引き上げられました。(バグ #79450、バグ #22283790)

- レプリケーション: リレーログから次のイベントを読み込むためにレプリカの SQL スレッドによって呼び出される next_event() 関数では、(例えば、クローズされたリレーログを原因として) SQL スレッドが実行されてエラーになったときに取得した relaylog.log_lock を解放しなかったため、他のすべてのスレッドが、終了するためにリレーログのロックを取得するまで待機する状態を生じさせました。この修正により、SQL スレッドがその状況下で関数を離れる前にロックが解放されます。(バグ #21697821)
- 仮想列を使用した ALTER TABLE のメモリ破損を修正しました。(バグ #24961167、バグ #24960450)
- レプリケーション: マルチスレッドレプリカは、そのサイズより大きいトランザクションを処理する必要があった場合、slave_pending_jobs_size_max を使用してより小さなキューサイズで設定できませんでした。slave_pending_jobs_size_max より大きいパケットは、slave_max_allowed_packet で設定された制限よりもパケットが小さい場合でも、エラー ER_MTS_EVENT_BIGGER_PENDING_JOBS_SIZE_MAX で拒否されました。今回の修正により、slave_pending_jobs_size_max がハード制限ではなくソフト制限になります。パケットのサイズが slave_pending_jobs_size_max を超えるが、slave_max_allowed_packet より小さい場合、すべてのレプリカワーカーが空のキューを持つまでトランザクションが保持されてから処理されます。後続のすべてのトランザクションは、大規模なトランザクションが完了するまで保持されます。したがって、レプリカワーカーのキューサイズは制限されますが、時折発生するより大きなトランザクションは引き続き許可されます。(バグ #21280753、バグ #77406)
- レプリケーション: マルチスレッドレプリカを使用する場合、適用元エラーで、Performance Schemaのレプリケーションテーブルで外部化されたデータと一致しないワーカー ID データが表示されました。(バグ #25231367)
- レプリケーション： -gtid-mode=ON、-log-bin=OFF および -slave-skip-errors で実行されている GTID ベースのレプリケーションレプリカで、無視するエラーが発生したため、正しく更新Exec_Master_Log_Posされず、 との同期Exec_Master_Log_Posが切断されましたRead_master_log_pos。GTID_NEXT が指定されていない場合、レプリカは、単一のステートメントトランザクションからロールバックするときに GTID 状態を更新しません。トランザクションが終了しても、GTID 状態は別途表示されるので、Exec_Master_Log_Pos は更新されません。この修正により、GTID_NEXT が指定された場合にのみトランザクションがロールバックされるときに GTID 状態の更新の制約がなくなります。(バグ #22268777)
- レプリケーション: バイナリログ記録が無効になっているときに、部分的に失敗したステートメントが、自動生成または指定された GTID を正しく消費しませんでした。この修正により、部分的に失敗した DROP TABLE、部分的に失敗した DROP USER、または部分的に失敗した DROP VIEW がそれぞれ関連する GTID を消費することと、バイナリログ記録が無効である場合は、それ

を @@GLOBAL.GTID_EXECUTED および mysql.gtid_executed テーブルに保存することが保証されます。(バグ #21686749)

- レプリケーション: MySQL 5.7 を実行しているレプリカは、MySQL 5.5 の一部ではない server_uuid の取得エラーにより、MySQL 5.5 ソースに接続できませんでした。これは、server_uuid の取得方法が変更されたことが原因となって生じました。(バグ #22748612)
- レプリケーション: 以前に実行された GTID トランザクションをサイレントにスキップする GTID トランザクションのスキップメカニズムは、XA トランザクションで正常に動作しませんでした。(バグ #25041920)
- 渡されたトランザクション ID が正しくないせいで失敗した ">XA ROLLBACK ステートメントが、正しいトランザクション ID でバイナリログに記録され、レプリケーションレプリカによって処理される可能性があります。バイナリログ記録が実行される前にエラー状況をチェックし、失敗した XA ROLLBACK ステートメントはログに記録されなくなりました。(バグ #26618925)
- レプリケーション： ソースログファイル名とソースログの位置を指定しない CHANGE MASTER TO ステートメントを使用してレプリカを設定した場合、START SLAVE が発行される前にシャットダウンし、 オプション -relay-log-recovery 設定で再起動しました。レプリケーションは開始されませんでした。これは、リレーログの復旧が試行される前にレシーバースレッドがスタートされておらず、ソースログファイル名とソースログの位置を提供するログローテーションイベントがリレーログで使用できないために発生しました。この場合、レプリカはリレーログの復旧をスキップし、警告をログに記録し、レプリケーションをスタートします。(バグ #28996606、バグ #93397)
- レプリケーション: 行ベースのレプリケーションでは、utf8mb3 列を持つテーブルから、列が utf8mb4 文字セットで定義されている同じ定義のテーブルにレプリケートするときに、フィールド文字数を誤って表示するメッセージが返されました。(バグ #25135304、バグ #83918)
- レプリケーション: GTID が使用中のレプリケーションレプリカで RESET SLAVE ステートメントが発行されると、既存のリレーログファイルは消去されましたが、チャネル用の受信した GTID のセットがクリアされる前に、代わりの新しいリレーログファイルが生成されました。したがって、以前の GTID セットは PREVIOUS_GTIDS イベントとして新しいリレーログファイルに書き込まれ、両方のサーバーの gtid_executed セットが空であっても、レプリカがソースよりも多くの GTID を持つことを示す致命的なエラーがレプリケーションで発生しました。ここで、RESET SLAVE が発行されると、新しいリレーログファイルが生成される前に受信した GTID のセットがクリアされるため、この状況は発生しません。(バグ #27411175)
- レプリケーション: レプリケーションのために GTID が使用されているときに、分析エラー (ER_PARSE_ERROR) を引き起こしたステートメントを含むトランザクションを、同じ GTID で空のトランザクションまたは代替トランザクションの挿入のための推奨される方法によって、手動でスキップできませんでした。このアクションにより、レプリカは既に使用されている GTID を識別し、GTID を共有した不要なトランザクションをスキップします。ただし、分析エラーの場

合、GTID をスキップする必要があるかどうかを確認する前にステートメントが分析されたため、いかなる場合でもトランザクションをスキップする意図があったにもかかわらず、分析エラーによりレプリケーション適用元スレッドが停止しました。この修正により、GTID が既に使用されていたため、関連するトランザクションをスキップする必要がある場合に、レプリケーションの適用元スレッドが分析エラーを無視するようになりました。この動作の変更は、`mysqlbinlog` によって生成されるバイナリログ出力で構成されるワークロードの場合には適用されないことに注意してください。そのような状況では、スキップされたトランザクションの直後に実行される、分析エラーがあるトランザクションも、エラーを発生させるべきときに警告なしでスキップされるリスクがあります。(バグ #27638268)

- レプリケーション: GTID に対する SQL スレッドが部分的なトランザクションをスキップできるようにします。(バグ #25800025)
- レプリケーション: 負のタイムアウトパラメータまたは小数のタイムアウトパラメータが `WAIT_UNTIL_SQL_THREAD_AFTER_GTIDS()` に提供されると、サーバーは予期しない動作をしました。この修正によりもたらされる結果は次のとおりです。
  - 小数のタイムアウト値はそのまま読み込まれ、丸められることはありません。
  - サーバーが厳密な SQL モードの場合、負のタイムアウト値はエラーで拒否されます。サーバーが厳密な SQL モードでない場合、この値は、関数が待機せずに、即時に `NULL` を返し、その後に警告を発行するようにします。(バグ #24976304、バグ #83537)
- レプリケーション: `WAIT_FOR_EXECUTED_GTID_SET()` 関数が小数部 (1.5 など) を含むタイムアウト値とともに使用された場合、キャストロジックのエラーは、タイムアウトが最も近い 1 秒に切り捨てられ、1 秒未満の値 (0.1 など) はゼロに切り捨てられたことを意味していました。キャストロジックが修正され、タイムアウト値が当初指定されたとおりに、丸められることなく、適用されるようになりました。Dirkjan Bussink の貢献に感謝の意を表します。(バグ #29324564、バグ #94247)
- GTID を有効にすると、複数ステートメントトランザクション内の切断された XA トランザクションで XA COMMIT がアサーションを発生させました。(バグ #22173903)
- レプリケーション: gtid_next 値が手動で設定されたときに、不明なトランザクション識別子に対して XA ROLLBACK ステートメントが発行された場合、デバッグ構築でアサーションが発生しました。XA `ROLLBACK` ステートメントがエラーで失敗した場合、サーバーは GTID 状態の更新を試行しなくなりました。(バグ #27928837、バグ #90640)
- 複数の `CASE` 関数が `ORDER BY` 句で使用されている際に発生する誤ったソート順の問題を修正しました (Bug#22810883)。
- 順序付けを使用する一部のクエリが、初期化されていない列に最適化中にアクセスし、サーバーを終了させる可能性があります。(バグ #27389294)

Aurora MySQL バージョン 1 との比較

以下の Amazon Aurora MySQL 機能は Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- 関数を同期的に呼び出すためのネイティブ AWS Lambda 関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2021-11-12 (バージョン 2.09.3) (廃止)

バージョン: 2.09.3

Aurora MySQL 2.09.3 は一般公開されています。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

既存の Aurora MySQL 2.* データベースクラスタを Aurora MySQL 2.10.0 にアップグレードできます。Aurora MySQL バージョン 1 を実行しているクラスターの場合、既存の Aurora MySQL 1.23 以降のクラスターを 2.10.0 に直接アップグレードできます。スナップショットは、現在サポートされている Aurora MySQL リリースから Aurora MySQL 2.10.0 に復元することもできます。

古いバージョンの Aurora MySQL を使用してクラスターを作成するには AWS Management Console、AWS CLI、または Amazon RDS API を使用してエンジンバージョンを指定します。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

セキュリティの修正内容:

マネージド型の環境での処理を微調整するための修正およびその他の機能強化。以下の CVE の追加の修正:

- 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

可用性の向上:

- information_schema 内のテーブルで実行されるクエリの競合を減らすための最適化が導入されました。
- ECDHE SSL 暗号のサポートを追加します。

全般的な機能強化

- メモリ内のデータ整合性チェックが失敗した際、ライターインスタンスがまれに再起動する不具合を修正しました。
- バイナリログ記録が有効になっている間にクラスターボリュームが拡張している際、データベースインスタンスがまれに再起動する不具合を修正しました。
- データベースインスタンスの再起動中、まれに再起動を1回以上引き起こす競合状態を修正しました。
- データベースに多数のユーザーと特権の組み合わせがある場合、データベースインスタンスの再起動が失敗する不具合を修正しました。

- LIMIT 句で SQL 文を実行した際、データベースが再起動するパラレルクエリの不具合を修正しました。
- Aurora レプリケーションラグの誤ったレポートの問題を修正しました。
- Aurora-MySQL 1.x (MySQL 5.6 に基づく) から Aurora-MySQL 2.x (MySQL 5.7 に基づく) へのインプレースメジャーバージョンアップグレード後に general_log テーブルと slow_log テーブルにアクセスできなくなる不具合を修正しました。
- データベースへのワークロードが高い状態で innodb_trx、innodb_locks、または innodb_lockwaits テーブルがクエリされた際、まれにデータベースインスタンスが再起動する不具合を修正しました。パフォーマンスインサイト などのモニタリングツールや機能は、そのようなテーブルをクエリする場合があります。
- 「FLUSH TABLES WITH READ LOCK」SQL ステートメントが実行された際、データベースインスタンスが再起動する問題を修正しました。
- リーダーインスタンスの削除中に履歴リストの長さが一時的に増加し、InnoDB のパージプロセスが一時停止しする問題を修正しました。
- 仮想列を含むテーブルに対して SQL 文を実行した際、データベースの再起動を引き起こすパラレルクエリの問題を修正しました。
- 「GROUP BY」 句と範囲述語を含む 「WHERE」 句を使用してクエリを実行した際、データベースが誤ったグループ化やソート順序を返す可能性があるパラレルクエリの問題を修正しました。
- JSON 関数で SQL ステートメントを実行すると、まれにデータベースの再起動を引き起こすパラレルクエリの問題を修正しました。
- グローバルデータベースレプリケーション中の競合状態が原因で、まれにプライマリグローバルデータベースクラスターのライターインスタンスが再起動する問題を修正しました。
- 特定の DDL および DCL ステートメントをレプリケートした際、バイナリログレプリカが HA_ERR_FOUND_DUPP_KEY エラーで停止する問題を修正しました。この問題は出典インスタンスが MIXED バイナリログ記録形式と READ COMMITTED または READ UNCOMMITTED 分離レベルで構成されている場合に発生します。
- READ COMMITTED 分離レベルで XA トランザクションを使用した際、まれにデータベースインスタンスが再起動する問題を修正しました。
- 以下の条件がすべて満たされた場合、既存の行の TIMESTAMP 列の値が、最新のタイムスタンプに更新されてしまう問題を修正しました。1. テーブルにトリガーが存在している。2. ON DUPLICATE KEY UPDATE 句が含まれるテーブルに対して INSERT が実行されている。3. 挿入された行が UNIQUE インデックスまたは PRIMARY KEY で値の重複違反を引き起こす可能性がある。4. TIMESTAMP データ型の列が 1 つ以上存在し、そこにデフォルト値の CURRENT_TIMESTAMP 書き込まれている。

- 誤ったチェック処理が原因で、まれにリーダーインスタンスが再起動する不具合を修正しました。
- ライターインスタンスがデータベースボリュームを特定のボリュームサイズの境界を越えるまで拡大させた際、リーダーインスタンスが再起動する問題を修正しました。
- クローンクラスターボリュームを使用してのデータベースインスタンスの再起動時間が長くなる問題を修正しました。
- ライターインスタンスで TRUNCATE TABLE 操作を実行した後、データベースインスタンスの再起動が 1 回以上失敗することがある不具合を修正しました。
- まれにデータベースインスタンスが再起動する問題を修正しました。
- データベースボリュームが 160 GB の倍数に達すると、まれにライターインスタンスが再起動することがある問題を修正しました。

MySQL Community Edition バグ修正の統合

- Bug #23533396 - 新しいインデックスを追加した際、サーバーは内部的に定義された外部キーインデックスを削除し、仮想生成カラムに定義されたセカンダリインデックスを外部キーインデックスとして使用しようとしたため、サーバーの終了を引き起こした。InnoDB は、外部キー制約が仮想生成カラムで定義されたセカンダリインデックスを参照することを許可するようになりました。
- バグ #29550513 - レプリケーション: WAIT_FOR_EXECUTED_GTID_SET () 関数のロックの問題により、特定の状況でサーバーがハングする可能性がある。この問題はすでに修正されました。

Aurora MySQL バージョン 1 との比較

以下の Amazon Aurora MySQL 機能は Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。

- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2021-02-26 (バージョン 2.09.2) (廃止)

バージョン: 2.09.2

Aurora MySQL 2.09.2 は一般公開されています。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリース
は、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

既存の Aurora MySQL 2.* データベースクラスターを Aurora MySQL 2.09.2 にアップグレードできます。Aurora MySQL バージョン 1 を実行しているクラスターの場合、既存の Aurora MySQL 1.23 以降のクラスターを 2.09.2 に直接アップグレードできます。スナップショットを、現在サポートされている Aurora MySQL リリースから Aurora MySQL 2.09.2 に復元することもできます。

古いバージョンの Aurora MySQL を使用してクラスターを作成するには AWS Management Console、AWS CLI、または Amazon RDS API を使用してエンジンバージョンを指定します。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

新機能:

- Aurora MySQL クラスターで、Arm ベースの AWS Graviton2 プロセッサを搭載した `r6g.large`、`r6g.xlarge`、`r6g.2xlarge`、`r6g.4xlarge`、`r6g.8xlarge`、`r6g.12xlarge`、および `r6g.16xlarge` の EC2 R6g インスタンスのサポートをスタートしました。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora DB インスタンスクラス」を参照してください。

セキュリティの修正内容:

マネージド型の環境での処理を微調整するための修正およびその他の機能強化。以下の CVE の追加の修正:

- 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

可用性の向上:

- クラスターストレージボリュームのスケーリング中に書き込みレイテンシーが上昇する可能性がある、2.09.0 で導入された問題を修正しました。
- 動的サイズ変更機能で、Aurora リードレプリカが再起動する可能性がある問題を修正しました。
- 1.23.* から 2.09.* へのアップグレード中にダウンタイムが長くなる可能性がある問題を修正しました。
- ページのプリフェッチリクエスト中に DDL または DML がエンジンを再起動する問題を修正しました。
- レプリケートされたトランザクションに DDL ステートメントと多数の行変更が含まれている場合に、バイナリログレプリカがエラーで停止する問題を修正しました。
- mysql `time_zone` テーブルで DDL イベントをレプリケートしているときに、バイナリログレプリカとして機能するデータベースが再起動することがある問題を修正しました。
- `binlog_checksum` パラメータが `NONE` に設定されている場合、大規模なトランザクションが誤ったバイナリログイベントを生成する可能性がある問題を修正しました。
- バイナリログレプリカが `HA_ERR_KEY_NOT_FOUND` エラーで停止する問題を修正しました。
- 全体的な安定性が向上しました。

Aurora MySQL バージョン 1 との比較

次の Amazon Aurora MySQL 機能は、Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。

- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2020-12-11 (バージョン 2.09.1) (廃止)

バージョン: 2.09.1

Aurora MySQL 2.09.1 は一般公開されています。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

既存の Aurora MySQL 2.* データベースクラスターを Aurora MySQL 2.09.1 にアップグレードできます。Aurora MySQL バージョン 1 を実行しているクラスターの場合、既存の Aurora MySQL 1.23 以降のクラスターを 2.09.1 に直接アップグレードできます。スナップショットを、現在サポートされている Aurora MySQL リリースから Aurora MySQL 2.09.1 に復元することもできます。

古いバージョンの Aurora MySQL を使用してクラスターを作成するには AWS Management Console、AWS CLI、または RDS API を使用してエンジンバージョンを指定してください。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

セキュリティの修正内容:

マネージド型の環境での処理を微調整するための修正およびその他の機能強化。以下の CVE の追加の修正:

- 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

互換性のない変更:

このバージョンでは、mysqldump コマンドの動作に影響するアクセス許可の変更が導入されています。ユーザーは、PROCESS テーブルにアクセスする INFORMATION_SCHEMA.FILES 特権を有している必要があります。変更せずに mysqldump コマンドを実行するには、PROCESS コマンドが接続するデータベースユーザーに mysqldump 特権を付与します。mysqldump オプションを指定して --no-tablespaces コマンドを実行することもできます。このオプションを使用すると、mysqldump 出力に CREATE LOGFILE GROUP または CREATE TABLESPACE ステートメントは含まれません。この場合、mysqldump コマンドは INFORMATION_SCHEMA.FILES テーブルにアクセスしないため、PROCESS アクセス許可を付与する必要はありません。

可用性の向上:

- ネットワークへの読み取りまたは書き込み中にデータベースエンジンでエラーが発生すると、クライアントセッションがハングする可能性がある問題を修正しました。
- 2.09.0 で導入された動的サイズ変更機能のメモリリークを修正しました。

グローバルデータベース:

- プライマリリージョンのライターを古いリリースバージョンで使用している場合に、リリース 2.09.0 にアップグレードすると、グローバルデータベースのセカンダリリージョンのレプリカが再起動する可能性がある複数の問題を修正しました。

MySQL Community Edition バグ修正の統合

- レプリケーション: トランザクション分離レベルが REPEATABLE READ に設定されている場合、インターリーブトランザクションがスレーブアプライヤをデッドロックすることがありました。(バグ #25040331)
- デフォルト値の CURRENT_TIMESTAMP が書き込まれた TIMESTAMP または DATETIME 列を持つテーブルで、そのテーブルに BEFORE INSERT トリガーがある場合は、これらの列が 0000-00-00 00:00:00 に初期化されることがあります。(バグ #25209512、バグ #84077)
- VALUES リストが結合を含むサブクエリを使用して 2 行目以降の値を生成した INSERT ステートメントの場合、必要な権限の解決に失敗するとサーバーが終了する可能性があります。(バグ #23762382)

Aurora MySQL バージョン 1 との比較

次の Amazon Aurora MySQL 機能は、Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空

間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2020-09-17 (バージョン 2.09.0) (廃止)

バージョン 2.09.0

Aurora MySQL 2.09.0 は一般公開されています。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

Aurora MySQL 1.23.* から取得したスナップショットを Aurora MySQL2.09.0 で復元できます。既存の Aurora MySQL 2.* データベースクラスターを Aurora MySQL 2.09.0 にアップグレードすることもできます。既存の Aurora MySQL 1.23.* クラスターは直接 2.09.0 にアップグレードできませんが、そのスナップショットは Aurora MySQL 2.09.0 に復元できます。

> **Important**
>
> このバージョンでの Aurora ストレージの改善により、利用可能なアップグレードパスは Aurora MySQL 1.* から Aurora MySQL 2.09 に制限されています。Aurora MySQL 1.* クラス

ターを 2.09 にアップグレードする場合、Aurora MySQL 1.23 からアップグレードする必要があります。

古いバージョンの Aurora MySQL を使用してクラスターを作成するには AWS Management Console、AWS CLI、または RDS API を使用してエンジンバージョンを指定してください。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

新機能:

- このリリースでは、最大 128 テビバイト (TiB) のストレージを備えた Amazon Aurora MySQL データベースインスタンスを作成できます。新しいストレージ制限は、以前の 64 TiB から引き上げられています。128 TiB のストレージサイズは、より大きなデータベースに対応します。この機能は、スモールインスタンスサイズ (db.t2 または db.t3) ではサポートされていません。1 つのテーブルスペースは、16 KB のページサイズという InnoDB の制限があるため、64 TiB を超えて拡張することはできません。

  Aurora は、クラスターボリュームサイズが 128 TiB に近い場合にアラートを表示し、サイズ制限に達する前にアクションを実行できるようにします。このアラートは、AWS Management Console の mysql ログと RDS イベントに表示されます。
- DB クラスターパラメータの値を変更することで、既存のクラスターのパラレルクエリのオンとオフを切り替えることができるようになりました `aurora_parallel_query`。クラスターを作成するときに、`parallelquery` パラメータの `--engine-mode` 設定を使用する必要はありません。

  パラレルクエリが拡張され、Aurora MySQL が利用できるすべてのリージョンで利用できるようになりました。

Aurora クラスタ内のパラレルクエリをアップグレードおよび有効化するための手順には、他にも多くの機能拡張や変更があります。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL のパラレルクエリの使用」を参照してください。

- Aurora は、クラスターのストレージ領域のサイズを動的に変更します。動的サイズ変更では、Aurora DB クラスターからデータを削除すると、DB クラスターのストレージ領域が自動的に減少します。詳細については、「Amazon Aurora ユーザーガイド」の「ストレージのスケーリング」を参照してください。

> **Note**
>
> 動的サイズ変更機能は、Aurora が利用可能な AWS リージョンで段階的にデプロイされています。クラスターを使用するリージョンによっては、この機能はまだ利用できない場合があります。詳細については、新しい発表を参照してください。

優先度の高い修正:

- コミュニティバグのバックポート #27659490: SELECT USING DYNAMIC RANGE AND INDEX MERGE USE TOO MUCH MEMORY (OOM)
- バグ #26881508: MYSQL #1: DISABLE_ABORT_ON_ERROR IN AUTH_COMMON.H
- コミュニティバグのバックポート #24437124: POSSIBLE BUFFER OVERFLOW ON CREATE TABLE
- バグのバックポート #27158030: INNODB ONLINE ALTER CRASHES WITH CONCURRENT DML
- バグ #29770705: SERVER CRASHED WHILE EXECUTING SELECT WITH SPECIFIC WHERE CLAUSE
- バグのバックポート #26502135: MYSQLD SEGFAULTS IN MDL_CONTEXT::TRY_ACQUIRE_LOCK_IMPL
- バグ #26935001: ALTER TABLE AUTO_INCREMENT TRIES TO READ INDEX FROM DISCARDED TABLESPACE
- バグ #28491099: [FATAL] MEMORY BLOCK IS INVALID | INNODB: ASSERTION FAILURE: UT0UT.CC:670
- バグ #30499288: GCC 9.2.1 REPORTS A NEW WARNING FOR OS_FILE_GET_PARENT_DIR
- バグ #29952565 where MYSQLD GOT SIGNAL 11 WHILE EXECUTING A QUERY(UNION + ORDER BY + SUB-QUERY)

- バグ #30628268: OUT OF MEMORY CRASH
- バグ #30441969: BUG #29723340: MYSQL SERVER CRASH AFTER SQL QUERY WITH DATA ? AST
- バグ #30569003: 5.7 REPLICATION BREAKAGE WITH SYNTAX ERROR WITH GRANT MANAGEMENT
- バグ #29915479: RUNNING COM_REGISTER_SLAVE WITHOUT COM_BINLOG_DUMP CAN RESULTS IN SERVER EXIT
- バグ #30569003: 5.7 REPLICATION BREAKAGE WITH SYNTAX ERROR WITH GRANT MANAGEMENT
- バグ #29915479: RUNNING COM_REGISTER_SLAVE WITHOUT COM_BINLOG_DUMP CAN RESULTS IN SERVER EXIT
- バグ #20712046: SHOW PROCESSLIST AND PERFORMANCE_SCHEMA TABLES DO NOT MASK PASSWORD FROM QUERY
- バックポートバグ #18898433: EXTREMELY SLOW PERFORMANCE WITH OUTER JOINS AND JOIN BUFFER (fixed in 5.7.21)。結合バッファリングが使用されている場合、多くの左結合を持つクエリが低速になります (ブロックネストされたループアルゴリズムを使用する場合など)。(バグ #18898433、バグ #72854)」
- バックポートバグ #26402045: MYSQLD CRASHES ON QUERY (MySQL 5.7.23 で修正)。サブクエリのマテリアル化の特定のケースで、サーバーが終了する可能性があります。これらのクエリは、マテリアル化が無効であることを示すエラーを生成するようになりました。(バグ #26402045)
- [MySQL からのバックポート] rdsadmin 以外のユーザーは、リーダーレプリカの pfs テーブルを更新できません。
- リーダーレプリカで顧客が perfschema を更新できない問題を修正します。
- バグ #26666274: INFINITE LOOP IN PERFORMANCE SCHEMA BUFFER CONTAINER
- バグ #26997096: relay_log_space 値は同期された方法で更新されないため、その値はリレーログで使用される実際のディスク領域よりもはるかに大きくなることがあります。
- バグ #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

可用性の向上:

- ロックマネージャー ABA 修正をデフォルトで有効にします。
- 競合状態により、2 つのトランザクションでロックが共有され、データベースが再起動することがあるロックマネージャーの問題を修正しました。
- 圧縮された行フォーマットでテンポラリテーブルを作成すると再起動する場合がある問題を修正しました。
- 16XL および 24XL インスタンスの `table_open_cache` のデフォルト値を修正しました。このデフォルト値では、ラージインスタンスクラス (R4/R5-16XL、R5-12XL、R5-24XL) でフェイルオーバーが繰り返され、CPU 使用率が高くなる場合があります。これは 2.07.x に影響しました。
- S3 バックアップに `mysql.host` テーブルが含まれていない場合、Amazon S3 から Aurora MySQL バージョン 2.08.0 へのクラスターの復元に予想以上に時間がかかる問題を修正しました。
- セカンダリインデックスを使用した仮想列の更新により、フェールオーバーが繰り返し発生する可能性がある問題を修正しました。
- 長時間実行される書き込みトランザクションによりデータベースが再起動される、トランザクションロックメモリ管理に関する問題を修正しました。
- パッチ適用のための安全ポイントのチェックにおいて、ダウンタイムのないパッチ適用中にエンジンがクラッシュすることがある複数の問題を修正しました。
- 以前クラッシュの原因となっていたテンポラリテーブルの redo ログをスキップする問題を修正しました。

- 接続/クエリの強制終了とセッションの強制終了の間のロックマネージャーの競合状態を修正しました。
- binlog レプリカで、MySQL time_zone テーブルで DDL イベントを受け取ると、データベースがクラッシュする可能性がある問題を修正しました。

グローバルデータベース:

- セカンダリリージョンの MySQL INFORMATION_SCHEMA.REPLICA_HOST_STATUS ビューに、そのリージョンに属するレプリカのエントリが表示されるようになりました。
- プライマリリージョンとセカンダリリージョン間のテンポラリネットワーク接続の問題の後に、グローバル DB セカンダリリージョンで発生する可能性がある予期しないクエリの失敗が修正されました。
-

パラレルクエリ:

- Parallel Query クエリの EXPLAIN プランを修正しました。単純な単一テーブルクエリでは正しくありません。
- Parallel Query が有効な場合に発生する可能性があるセルフデッドラッチが修正されました。

全般的な機能強化

- S3 へのエクスポートで ENCRYPTION キーワードがサポートされるようになりました。
- これで、aurora_binlog_replication_max_yield_seconds パラメータの最大値が 36,000 になりました。以前の最大許容値は 45 でした。このパラメータ aurora_binlog_use_large_read_buffer は、パラメータが 1 に設定されている場合にのみ機能します。
- MIXED 実行時の動作を binlog_format ではなく ROW STATEMENT を LOAD DATA FROM INFILE | S3 にマッピングするように変更しました。
- Aurora MySQL バイナリログプライマリに接続されたバイナリログレプリカで、プライマリが LOAD DATA FROM S3 を実行し、binlog_format が STATEMENT に設定されているときに、不完全なデータが表示される可能性がある問題を修正しました。
- 監査システム可変 server_audit_incl_users および server_audit_excl_users の最大許容長を、1024 バイトから 2000 バイトに引き上げました。

- 現在の接続が設定されている値よりも大きい場合に、パラメータグループの `max_connections` パラメータを下げると、ユーザーがデータベースにアクセスできなくなる問題を修正しました。
- データアクティビティストリーミングで、一重引用符とバックスラッシュが正しくエスケープされない問題を修正しました。

MySQL Community Edition バグ修正の統合

- バグ #27659490: SELECT USING DYNAMIC RANGE AND INDEX MERGE USE TOO MUCH MEMORY(OOM)
- バグ #26881508: MYSQL #1: DISABLE_ABORT_ON_ERROR IN AUTH_COMMON.H
- バグ #24437124: POSSIBLE BUFFER OVERFLOW ON CREATE TABLE
- バグ #27158030: INNODB ONLINE ALTER CRASHES WITH CONCURRENT DML
- バグ #29770705: SERVER CRASHED WHILE EXECUTING SELECT WITH SPECIFIC WHERE CLAUSE
- バグ #26502135: MYSQLD SEGFAULTS IN MDL_CONTEXT::TRY_ACQUIRE_LOCK_IMPL
- バグ #26935001: ALTER TABLE AUTO_INCREMENT TRIES TO READ INDEX FROM DISCARDED TABLESPACE
- バグ #28491099: [FATAL] MEMORY BLOCK IS INVALID | INNODB: ASSERTION FAILURE: UT0UT.CC:670
- バグ #30499288: GCC 9.2.1 REPORTS A NEW WARNING FOR OS_FILE_GET_PARENT_DIR
- バグ #29952565: where MYSQLD GOT SIGNAL 11 WHILE EXECUTING A QUERY(UNION + ORDER BY + SUB-QUERY)
- バグ #30628268: OUT OF MEMORY CRASH
- バグ #30441969: BUG #29723340: MYSQL SERVER CRASH AFTER SQL QUERY WITH DATA ? AST
- バグ #30569003: 5.7 REPLICATION BREAKAGE WITH SYNTAX ERROR WITH GRANT MANAGEMENT
- バグ #29915479: RUNNING COM_REGISTER_SLAVE WITHOUT COM_BINLOG_DUMP CAN RESULTS IN SERVER EXIT
- バグ #30569003: 5.7 REPLICATION BREAKAGE WITH SYNTAX ERROR WITH GRANT MANAGEMENT
- バグ #29915479: RUNNING COM_REGISTER_SLAVE WITHOUT COM_BINLOG_DUMP CAN RESULTS IN SERVER EXIT

- バグ #20712046: SHOW PROCESSLIST AND PERFORMANCE_SCHEMA TABLES DO NOT MASK PASSWORD FROM QUERY
- バグ #18898433: EXTREMELY SLOW PERFORMANCE WITH OUTER JOINS AND JOIN BUFFER (fixed in 5.7.21)
- バグ #26402045: MYSQLD CRASHES ON QUERY (fixed in MySQL 5.7.23)
- バグ #23103937: PS_TRUNCATE_ALL_TABLES() がSUPER_READ_ONLY モードで機能しない
- バグ #26666274: INFINITE LOOP IN PERFORMANCE SCHEMA BUFFER CONTAINER
- バグ #26997096: relay_log_space 値は同期された方法で更新されないため、その値はリレーログで使用される実際のディスク領域よりもはるかに大きくなることがあります。(https://github.com/mysql/mysql-server/commit/78f25d2809ad457e81f90342239c9bc32a36cdfa)
- バグ #25082593: FOREIGN KEY VALIDATION DOESN'T NEED TO ACQUIRE GAP LOCK IN READ COMMITTED
- バグ #24764800: REPLICATION FAILING ON SLAVE WITH XAER_RMFAIL ERROR。
- バグ #81441: WARNING ABOUT LOCALHOST WHEN USING SKIP-NAME-RESOLVE。

Aurora MySQL バージョン 1 との比較

次の Amazon Aurora MySQL 機能は、Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2022-01-06 (バージョン 2.08.4) (廃止)

バージョン: 2.08.4

Aurora MySQL 2.08.4 は一般公開されています。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

既存の Aurora MySQL 2.* データベースクラスタを Aurora MySQL 2.10.0 にアップグレードできます。Aurora MySQL バージョン 1 を実行しているクラスターの場合、既存の Aurora MySQL 1.23 以降のクラスターを 2.10.0 に直接アップグレードできます。スナップショットは、現在サポートされている Aurora MySQL リリースから Aurora MySQL 2.10.0 に復元することもできます。

古いバージョンの Aurora MySQL を使用してクラスターを作成するには AWS Management Console、AWS CLI、または Amazon RDS API を使用してエンジンバージョンを指定します。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

セキュリティに関する修正および一般的な機能強化:

- Amazon S3、Amazon ML、および AWS Lambda などの他の AWS サービスと、Aurora MySQL との統合に関連する、セキュリティ上の問題を修正しました
- 以下の条件がすべて満たされた場合、既存の行の TIMESTAMP 列の値が、最新のタイムスタンプに更新されてしまう問題を修正しました。1. テーブルにトリガーが存在している。2. ON DUPLICATE KEY UPDATE 句が含まれるテーブルに対して INSERT が実行されている。3. 挿入された行が UNIQUE インデックスまたは PRIMARY KEY で値の重複違反を引き起こす可能性がある。4. TIMESTAMP データ型の列が 1 つ以上存在し、そこにデフォルト値の CURRENT_TIMESTAMP 書き込まれている。
- メモリ内データの整合性チェックが失敗した際、まれにライターインスタンスが再起動する問題を修正しました。
- LIMIT 句を使用して SQL 文を実行した際にデータベースの再起動を引き起こすことがある、Parallel Query の不具合を修正しました。

Aurora MySQL バージョン 1 との比較

次の Amazon Aurora MySQL 機能は、Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2020-11-12 (バージョン 2.08.3) (廃止)

バージョン: 2.08.3

Aurora MySQL 2.08.3 は一般公開されています。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

既存の Aurora MySQL 2.* データベースクラスターは、Aurora MySQL 2.08.3 に直接アップグレードできます。既存の Aurora MySQL 1.* クラスターを 2.07.3 以降に直接アップグレードしてから、2.08.3 に直接アップグレードできます。

古いバージョンの Aurora MySQL を使用してクラスターを作成するには AWS Management Console、AWS CLI、または RDS API を使用してエンジンバージョンを指定してください。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

セキュリティの修正内容:

マネージド型の環境での処理を微調整するための修正およびその他の機能強化。以下の CVE の追加の修正:

- 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

互換性のない変更:

このバージョンでは、mysqldump コマンドの動作に影響するアクセス許可の変更が導入されています。ユーザーは、PROCESS テーブルにアクセスする INFORMATION_SCHEMA.FILES 特権を有している必要があります。変更せずに mysqldump コマンドを実行するには、PROCESS コマンドが接続するデータベースユーザーに mysqldump 特権を付与します。mysqldump オプションを指定して --no-tablespaces コマンドを実行することもできます。このオプションを使用すると、mysqldump 出力に CREATE LOGFILE GROUP または CREATE TABLESPACE ステートメントは含まれません。この場合、mysqldump コマンドは INFORMATION_SCHEMA.FILES テーブルにアクセスしないため、PROCESS アクセス許可を付与する必要はありません。

MySQL Community Edition バグ修正の統合

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

Aurora MySQL バージョン 1 との比較

次の Amazon Aurora MySQL 機能は、Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。

- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2020-08-28 (バージョン 2.08.2) (廃止)

バージョン: 2.08.2

Aurora MySQL 2.08.2 は一般公開されています。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリース
は、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

現在サポートされている Aurora MySQL リリースから取得したスナップショットを Aurora MySQL 2.08.2 で復元できます。既存の Aurora MySQL 2.* データベースクラスターは、Aurora MySQL 2.08.2 にアップグレードすることもできます。既存の Aurora MySQL 1.* クラスターは直接 2.08.2 にアップグレードできませんが、そのスナップショットは Aurora MySQL 2.08.2 に復元できます。スナップショットの復元に関する詳細については、「Amazon Aurora ユーザーガイド」の「DB クラスターのスナップショットからの復元」を参照してください。

古いバージョンの Aurora MySQL を使用してクラスターを作成するには AWS Management Console、AWS CLI、または RDS API を使用してエンジンバージョンを指定してください。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

重要な修正:

- 予期しない停止を引き起こし、データベースの可用性に影響を与える可能性がある問題を修正しました。

可用性の修正:

- binlog レプリカで、mysql `time_zone` テーブルで DDL イベントをレプリケートすると、Aurora MySQL データベースが再起動することがある問題を修正しました。

Aurora MySQL バージョン 1 との比較

次の Amazon Aurora MySQL 機能は、Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード

- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- CREATE TABLESPACE SQL ステートメント

Aurora MySQL データベースエンジンの更新 2020-06-18 (バージョン 2.08.1) (廃止)

バージョン: 2.08.1

Aurora MySQL 2.08.1 は一般公開されています。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

現在サポートされている Aurora MySQL リリースから取得したスナップショットを Aurora MySQL 2.08.1 で復元できます。既存の Aurora MySQL 2.* データベースクラスターは、Aurora MySQL 2.08.1 にアップグレードすることもできます。既存の Aurora MySQL 1.* クラスターは直接 2.08.1 にアップグレードできませんが、そのスナップショットは Aurora MySQL 2.08.1 に復元できます。

古いバージョンの Aurora MySQL を使用してクラスターを作成するには AWS Management Console、AWS CLI、または RDS API を使用してエンジンバージョンを指定してください。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

新機能:

- グローバルデータベースの書き込み転送。Aurora Global Database で、セカンダリクラスターへの接続中に、DML ステートメントなどの特定の書き込み操作を実行できるようになりました。書き込み操作はプライマリクラスターに転送され、変更はすべてセカンダリクラスターにレプリケートされます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora Global Database の書き込み転送を使用する」を参照してください。

全般の安定性に関する修正:

- S3 バックアップに `mysql.host` テーブルが含まれていない場合、クラスターを Amazon S3 から Aurora MySQL バージョン 2.08.0 に復元すると、予想以上に時間がかかっていた問題を修正しました。

Aurora MySQL バージョン 1 との比較

次の Amazon Aurora MySQL 機能は、Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2020-06-02 (バージョン 2.08.0) (廃止)

バージョン: 2.08.0

Aurora MySQL 2.08.0 は一般公開されています。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

現在サポートされている Aurora MySQL リリースから取得したスナップショットを Aurora MySQL 2.08.0 で復元できます。既存の Aurora MySQL 2.* データベースクラスターは、Aurora MySQL 2.08.0 にアップグレードすることもできます。既存の Aurora MySQL 1.* クラスターは直接 2.08.0 にアップグレードできませんが、そのスナップショットは Aurora MySQL 2.08.0 に復元できます。

古いバージョンの Aurora MySQL を使用してクラスターを作成するには AWS Management Console、AWS CLI、または RDS API を使用してエンジンバージョンを指定してください。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

新機能:

- バイナリログ (binlog) 処理が改善され、非常に大きなトランザクションが関与している場合に、クラッシュリカバリ時間とコミット時間のレイテンシーが短縮されました。
- Aurora MySQL のデータベースアクティビティストリーミング (DAS) 機能の起動。この機能を使用すると、リレーショナルデータベース内のデータベースアクティビティのデータストリーミングをほぼリアルタイムで取得することができ、アクティビティのモニタリングに役立ちます。詳細については、「Amazon Aurora ユーザーガイド」の「データベースアクティビティストリームを使用した Amazon Aurora のモニタリング」を参照してください。
- 最新のブラジルのタイムゾーンの変更をサポートするようにタイムゾーンファイルを更新しました。
- 特定のテーブルまたは内部テーブル (またはその両方) のハッシュ結合機能を実行するために SQL に新しいキーワードが導入されました。`HASH_JOIN`、`HASH_JOIN_PROBING`、`HASH_JOIN_BUILDING`。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL のヒント」を参照してください。
- MySQL 8.0 機能をバックポートすることにより、Aurora MySQL 5.7 で結合順序のヒントのサポートが導入されました。新しいヒントは、`JOIN_FIXED_ORDER`、`JOIN_ORDER`、`JOIN_PREFIX`、および `JOIN_SUFFIX` です。結合順序ヒントのサポートの詳細については、「WL #9158: 結合順序のヒント」を参照してください。

- Aurora 機械学習では、戻り値の型としての MEDIUMINT を持つユーザー定義関数がサポートされるようになりました。
- lambda_async() ストアドプロシージャがすべての MySQL utf8 文字をサポートするようになりました。

優先度の高い修正:

- 書き込み DB インスタンスで INFORMATION_SCHEMA.INNODB_SYS_TABLES テーブルがクエリされた後、読み取り DB インスタンスが FTS クエリに対して不完全な結果を返す問題を修正しました。
- CVE-2019-5443
- CVE-2019-3822

可用性の向上:

- クエリキャッシュを有効にして複数のテーブルまたはデータベースにアクセスするマルチクエリステートメントを実行した後、データベースが再起動する問題を修正しました。
- トランザクションのロールバック中にデータベースの再起動またはフェイルオーバーが発生する、ロックマネージャーの競合状態を修正しました。
- 複数の接続でフルテキスト検索インデックスを使用して同じテーブルを更新しようとすると、データベースの再起動またはフェイルオーバーがトリガーされる問題を修正しました。
- kill session コマンド中にデータベースの再起動またはフェイルオーバーがトリガーされる問題を修正しました。この問題が発生した場合は、AWS サポートに連絡し、インスタンスでこの修正を有効にしてください。
- 複数の SELECT ステートメントを含むマルチステートメントトランザクションと、AUTOCOMMIT が有効になっている書き込み DB インスタンスで書き込みワークロードが高い場合に、読み取り DB インスタンスが再起動する問題を修正しました。
- 書き込み DB インスタンスが OLTP 書き込みワークロードの負荷が高い場合に、長時間実行クエリを実行した後に読み取り DB インスタンスが再起動する問題を修正しました。

全般的な機能強化:

- バイナリログが有効になっている場合、長時間実行されるトランザクションのデータベースのリカバリ時間とコミットレイテンシーが改善されました。

- アルゴリズムを改善し、データ分布が歪んだ列を含む、インデックス付きの列で明確な値のカウントを推定するためのより正確な統計情報が生成されるようになりました。
- MyISAM テンポラリテーブルにアクセスする統合クエリの応答時間と CPU 使用率が削減され、結果がローカルストレージに残るようになりました。
- スペースを含むデータベース名またはテーブル名を含む Aurora MySQL 5.6 スナップショットが新しい Aurora MySQL 5.7 クラスターに復元されない問題を修正しました。
- `show engine innodb status` でデッドロックが解決されたときに対象トランザクション情報を含めるようになりました。
- 複数の異なるバージョンのクライアントが同じデータベースに接続され、クエリキャッシュにアクセスしているときに接続が停止する問題を修正しました。
- データベースインスタンスの存続期間を通して、ゼロダウンタイムパッチ (ZDP) またはゼロダウンタイム再起動 (ZDR) ワークフローの複数回呼び出しに起因するメモリリークが修正されました。
- 自動コミットフラグがオフの場合、最後のトランザクションが中止されたことを誤って示すゼロダウンタイムパッチ (ZDP) またはゼロダウンタイム再起動 (ZDR) オペレーションのエラーメッセージを修正しました。
- ゼロダウンタイムパッチ (ZDP) オペレーションで、新しいデータベースプロセスでユーザーセッション可変を復元するときにサーバー障害エラーメッセージが表示される問題を修正しました。
- パッチ適用中に長時間実行されるクエリがある場合に、断続的なデータベース障害を引き起こす可能性がある ZDP (ZDP) オペレーションの問題が修正されました。
- Amazon SageMaker や Amazon Comprehend などの機械学習サービスからのエラー応答が正しく処理されなかったために、Aurora 機械学習関数を含むクエリが空のエラーメッセージを返していた問題を修正しました。
- `table_definition_cache` パラメータのカスタム値が採用されないメモリ不足モニタリング機能の問題を修正しました。
- Aurora 機械学習のクエリが中断されると、エラーメッセージ「クエリの実行が中断されました」が返されます。以前は、汎用メッセージ「ML リクエストの処理中に内部エラーが発生しました」が代わりに返されていました。
- `slave_net_timeout` パラメータが `aurora_binlog_replication_max_yield_seconds` パラメータより小さく、バイナリログマスタークラスターのワークロードが低い場合に、バイナリログワーカーで接続タイムアウトが発生することがある問題を修正しました。
- 1 分あたり 1 メッセージの頻度でエラーログに情報メッセージを出力することにより、バイナリログリカバリの進行状況のモニタリングが改善されました。

- `SHOW ENGINE INNODB STATUS` クエリによってアクティブなトランザクションが報告されない問題を修正しました。

MySQL Community Edition バグ修正の統合

- バグ #25289359: フルテキストキャッシュサイズがフルテキストキャッシュサイズの制限を超えた場合、データの同期時に行われるフルテキストキャッシュロックが解放されませんでした。
- バグ #29138644: MySQL サーバーの実行中にシステム時刻を手動で変更すると、ページクリーナーのスレッドの遅延が発生しました。
- バグ #25222337: 仮想インデックスの仮想列フィールド名が NULL の場合、外部キー制約の影響を受ける仮想列への入力中に発生するフィールド名の比較中にサーバー終了が発生しました。
- バグ #25053286: ビューにアクセスしたクエリを含むストアドプロシージャを実行すると、セッションが終了するまで解放されなかったメモリが割り当てられました。
- バグ #25586773: 特定の SELECT ステートメントの内容からテーブルを作成したステートメントを含むストアドプロシージャを実行すると、メモリリークが発生する可能性があります。
- バグ #28834208: ログアプリケーション中、OPTIMIZE TABLE オペレーション後に、InnoDB が仮想列のインデックスの更新をチェックする前に仮想列を挿入しませんでした。
- バグ #26666274: 32 ビット符号なし整数オーバーフローが原因で、Performance Schemaバッファコンテナに無限ループが発生しました。

Aurora MySQL バージョン 1 との比較

次の Amazon Aurora MySQL 機能は、Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。

- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2023-08-15 (バージョン 2.07.10、MySQL 5.7.12 互換)

バージョン: 2.07.10

Aurora MySQL 2.07.10 は一般公開されています。Aurora MySQL 2.07 バージョンは、MySQL 5.7.12 と互換性があります。コミュニティ版の変更点の詳細については、「Changes in MySQL 5.7.12 (2016-04-11, General Availability)」を参照してください。

> **Note**
>
> このバージョンは、長期サポート (LTS) リリースとして指定されています。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL 長期サポート (LTS) リリース」を参照してください。

現在サポートされている Aurora MySQL リリースは、2.07.*、2.11.*、3.01.*、3.02.*、3.03.*、3.04.* です。

現在サポートされている Aurora MySQL リリースから取得したスナップショットを Aurora MySQL 2.07.10 で復元できます。既存の Aurora MySQL 2.* データベースクラスターを Aurora MySQL 2.07.10 にアップグレードすることもできます。Aurora MySQL 1.* クラスターから Aurora MySQL 2.* へのインプレースアップグレードが可能です (「Aurora MySQL 1.x から 2.x へのアップグレード」を参照)。また、Aurora MySQL 2.* クラスターから Aurora MySQL 3.* にもインプレースアップグレードできます (「Aurora MySQL 2.x から 3.x へのアップグレード」を参照)。

Aurora MySQL 2.07.10 へのエンジンバージョンのインプレースアップグレードが実行された直後に、db.r4、db.r5、db.t2、db.t3 の DB インスタンスクラスで影響を受けるすべてのインスタンスにオペレーティングシステムのアップグレードが自動的に適用されます (インスタンスが古いオペレーティングシステムバージョンを実行している場合)。マルチ AZ DB クラスターでは、まず、すべてのリーダーインスタンスがオペレーティングシステムのアップグレードを適用します。最初のリーダーインスタンスでオペレーティングシステムのアップグレードが完了すると、フェイルオーバーが発生し、以前のライターインスタンスがアップグレードされます。

> **Note**
>
> メジャーバージョンのアップグレード中は、Aurora グローバルデータベースにはオペレーティングシステムのアップグレードは自動適用されません。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改善点

以下のセキュリティの問題と CVE の修正:

マネージド型の環境での処理を微調整するための修正およびその他の機能強化。以下の CVE の追加の修正:

- CVE-2023-21963
- CVE-2023-21912
- CVE-2023-0215
- CVE-2022-43551
- CVE-2022-37434
- 監査ログのローテーション中に報告されたイベントが監査ログに書き込まれない場合がある問題を修正しました。
- Aurora MySQL が使用するデフォルトの SSL 暗号を更新し、安全性の低い DES-CBC3-SHA 値を SSL_CIPHER データベースパラメータから除外しました。DES-CBC3-SHA 暗号の削除により SSL 接続に問題が生じた場合は、適切かつ安全な暗号を使用してください。詳細については、「Aurora MySQL DB クラスターへの接続用暗号スイートを設定する」を参照してください。
- OpenSSL がバージョン 1.0.2zh に更新されました。

全般的な機能強化:

- 小さいサイズの暗号化キーを使用する ECDHE-RSA SSL 暗号のサポートが追加されました。
- クエリでハッシュ結合を実行する際のメモリ管理の問題を修正しました。

Aurora MySQL バージョン 2 ではサポートされていない機能

以下の機能は、現時点では Aurora MySQL バージョン 2 (MySQL 5.7 互換) ではサポートされていません。

- Asynchronous Key Prefetch (AKP)。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- `CREATE TABLESPACE` SQL ステートメント
- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- X プロトコル

Aurora MySQL データベースエンジンの更新 2023-05-04 (バージョン 2.07.9、MySQL 5.7.12 互換)

バージョン: 2.07.9

Aurora MySQL 2.07.9 は一般公開されています。Aurora MySQL 2.07 バージョンは、MySQL 5.7.12 と互換性があります。コミュニティ版の変更点の詳細については、「Changes in MySQL 5.7.12 (2016-04-11, General Availability)」を参照してください。

> **Note**
>
> このバージョンは、長期サポート (LTS) リリースとして指定されています。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL 長期サポート (LTS) リリース」を参照してください。

現在サポートされている Aurora MySQL リリースは、2.07.*、2.11.*、3.01.*、3.02.*、3.03.* です。

現在サポートされている Aurora MySQL リリースから取得したスナップショットを Aurora MySQL 2.07.9 で復元できます。既存の Aurora MySQL 2.* データベースクラスターを Aurora MySQL 2.07.9 にアップグレードすることもできます。Aurora MySQL 1.* クラスターから Aurora MySQL 2.* へのインプレースアップグレードが可能です (「Aurora MySQL 1.x から 2.x へのアップグレード」を参照)。また、Aurora MySQL 2.* クラスターから Aurora MySQL 3.* にもインプレースアップグレードできます (「Aurora MySQL 2.x から 3.x へのアップグレード」を参照)。

Aurora MySQL 2.07.9 へのエンジンバージョンのインプレースアップグレードが実行された直後に、db.r4、db.r5、db.t2、db.t3 の DB インスタンスクラスで影響を受けるすべてのインスタンスにオペレーティングシステムのアップグレードが自動的に適用されます (インスタンスが古いオペレーティングシステムバージョンを実行している場合)。マルチ AZ DB クラスターでは、まず、すべてのリーダーインスタンスがオペレーティングシステムのアップグレードを適用します。最初のリーダーインスタンスでオペレーティングシステムのアップグレードが完了すると、フェイルオーバーが発生し、以前のライターインスタンスがアップグレードされます。

> **Note**
>
> メジャーバージョンのアップグレード中は、Aurora グローバルデータベースにはオペレーティングシステムのアップグレードは自動適用されません。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改善点

以下のセキュリティの問題と CVE の修正:

マネージド型の環境での処理を微調整するための修正およびその他の機能強化。以下の CVE の追加の修正:

- CVE-2022-32221

可用性の向上:

- 解放可能なメモリが高度な監査ログのローテーションによって減少し、DB インスタンスの再起動につながる可能性がある問題を修正しました。
- データベースの再起動中に問題が発生し、データベースが長時間正常に起動しなくなる場合がありましたが、この問題を修正しました。

全般的な機能強化:

- データベースボリュームが増加し 160GB の倍数に到達すると、まれにインスタンスが再起動する問題を修正しました。

Aurora MySQL バージョン 2 ではサポートされていない機能

以下の機能は、現時点では Aurora MySQL バージョン 2 (MySQL 5.7 互換) ではサポートされていません。

- Asynchronous Key Prefetch (AKP)。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更

- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- CREATE TABLESPACE SQL ステートメント

Aurora MySQL データベースエンジンの更新 2022-06-16 (バージョン 2.07.8) (廃止)

バージョン: 2.07.8

Aurora MySQL 2.07.8 は一般公開されています。Aurora MySQL 2.* バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.* バージョンは MySQL 5.6 と互換性があります。

> **Note**
>
> このバージョンは、長期サポート (LTS) リリースとして指定されています。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL 長期サポート (LTS) リリース」を参照してください。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

現在サポートされている Aurora MySQL リリースから取得したスナップショットを Aurora MySQL 2.07.8 で復元できます。既存の Aurora MySQL 2.* データベースクラスターを Aurora MySQL 2.07.8 にアップグレードすることもできます。Aurora MySQL 1.* クラスターから Aurora MySQL 2.* へのインプレースアップグレードが可能です (「Aurora MySQL 1.x から 2.x へのアップグレード」を参照)。また、Aurora MySQL 2.* クラスターから Aurora MySQL 3.* にもインプレースアップグレードできます (「Aurora MySQL 2.x から 3.x へのアップグレード」を参照)。

古いバージョンの Aurora MySQL を使用してクラスターを作成するには AWS Management Console、AWS CLI、または RDS API を使用してエンジンバージョンを指定してください。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

セキュリティの修正内容:

マネージド型の環境での処理を微調整するための修正およびその他の機能強化。以下の CVE の追加の修正:

- CVE-2022-21245
- CVE-2021-36222
- CVE-2021-22926

全般的な機能強化:

- 競合状態が原因でデッドロック検出スレッドが停止状態になると、まれにデータベースサーバーが再起動することがある問題を修正しました。

MySQL Community Edition バグ修正の統合

- プライマリキーが 1024 バイト超の一時テーブルを UPDATE が必要とし、そのテーブルが InnoDB を使用して作成された場合、サーバーが終了する可能性があります。(バグ #25153670)

Aurora MySQL バージョン 1 との比較

次の Amazon Aurora MySQL 機能は、Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。

- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2021-11-24 (バージョン 2.07.7) (廃止)

バージョン: 2.07.7

Aurora MySQL 2.07.7 は一般公開されています。Aurora MySQL 2.* バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.* バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

現在サポートされている Aurora MySQL リリースから取得したスナップショットを Aurora MySQL 2.07.7 で復元できます。既存の Aurora MySQL 2.* データベースクラスターを Aurora MySQL 2.07.7 にアップグレードすることもできます。既存の Aurora MySQL 1.* クラスターを直接 2.07.7 にアップグレードすることはできませんが、そのスナップショットを Aurora MySQL 2.07.7 で復元することはできます。

古いバージョンの Aurora MySQL を使用してクラスターを作成するには AWS Management Console、AWS CLI、または RDS API を使用してエンジンバージョンを指定してください。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

セキュリティの修正内容:

マネージド型の環境での処理を微調整するための修正およびその他の機能強化。以下の CVE の追加の修正:

- CVE-2019-17543

- CVE-2019-2960

全般的な機能強化

- Amazon S3、Amazon ML、Lambda などの他の AWS サービスと Aurora MySQL の統合に関連するセキュリティ問題を修正しました
- Aurora レプリケーションラグの誤ったレポートの問題を修正しました。
- データベースに多数のユーザーと特権の組み合わせがある場合、データベースインスタンスの再起動が失敗する不具合を修正しました。
- Aurora MySQL 1.x (MySQL 5.6 に基づく) から Aurora MySQL 2.x (MySQL 5.7 に基づく) へのインプレースメジャーバージョンアップグレード後に、 general_log テーブルと slow_log テーブルにアクセスできなくなる不具合を修正しました。
- 誤ったチェック処理が原因で、まれにリーダーインスタンスが再起動する不具合を修正しました。
- セッションの処理が終了してアイドル状態であっても、CPU をアクティブに使用していると Performance Insights (PI) セッションの 「データベースロード」 グラフに表示される問題を修正しました。
- LIMIT 句で SQL 文を実行した際、データベースの再起動を引き起こすパラレルクエリの不具合を修正しました。
- 次の条件がすべて満たされ際に、既存の行の TIMESTAMP 列の値が最新のタイムスタンプに更新される問題を修正しました: 1. テーブルに対してトリガーが存在している。 2. ON DUPLICATE KEY UPDATE 句を持つテーブルに対して INSERT が実行されている。 3. 挿入された行によって、UNIQUE インデックスまたは PRIMARY KEY に重複値違反が発生する可能性がある。 4. 一つ以上の列が TIMESTAMP データ型で、デフォルト値が CURRENT_TIMESTAMP になっている。
- READ COMMITTED 分離レベルで XA トランザクションを使用した際、まれにデータベースインスタンスが再起動する問題を修正しました。

Aurora MySQL バージョン 1 との比較

次の Amazon Aurora MySQL 機能は、Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。

- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2021-09-02 (バージョン 2.07.6) (廃止)

バージョン: 2.07.6

Aurora MySQL 2.07.6 は一般公開されています。Aurora MySQL 2.* バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.* バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

現在サポートされている Aurora MySQL リリースから取得したスナップショットを Aurora MySQL 2.07.6 で復元できます。既存の Aurora MySQL 2.* データベースクラスターを Aurora MySQL 2.07.6 にアップグレードすることもできます。既存の Aurora MySQL 1.* クラスターを直接 2.07.6 にアップグレードできませんが、そのスナップショットを Aurora MySQL 2.07.6 に復元することはできます。

古いバージョンの Aurora MySQL を使用してクラスターを作成するには AWS Management Console、AWS CLI、または RDS API を使用してエンジンバージョンを指定してください。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

MySQL Community Edition バグ修正の統合

- INSERTING 64K SIZE RECORDS TAKE TOO MUCH TIME. (64K サイズのレコードの挿入に時間がかかりすぎています) (バグ #23031146)

Aurora MySQL バージョン 1 との比較

次の Amazon Aurora MySQL 機能は、Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。

- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2021-07-06 (バージョン 2.07.5) (廃止)

バージョン 2.07.5

Aurora MySQL 2.07.5 は一般公開されています。Aurora MySQL 2.* バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.* バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリース
は、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

現在サポートされている Aurora MySQL リリースから取得したスナップショットを Aurora MySQL 2.07.5 で復元できます。既存の Aurora MySQL 2.* データベースクラスターは、Aurora MySQL 2.07.5 にアップグレードすることができます。既存の Aurora MySQL 1.* クラスターは直接 2.07.5 にアップグレードできませんが、そのスナップショットは Aurora MySQL 2.07.5 に復元できます。

古いバージョンの Aurora MySQL を使用してクラスターを作成するには AWS Management Console、AWS CLI、または RDS API を使用してエンジンバージョンを指定してください。

> **Note**
>
> このバージョンは、長期サポート (LTS) リリースとして指定されています。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL 長期サポート (LTS) リリース」を参照してください。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

可用性の向上:

- Aurora レプリカでユーザーレベルのロックが許可されない問題を修正しました。
- `READ COMMITTED` の分離レベルで XA トランザクションを使用した際に、データベースの再起動が発生する可能性がある問題を修正しました。
- `server_audit_incl_users` および `server_audit_excl_users` のグローバルパラメータで最大許容長を 2000 に拡張しました。

Aurora MySQL バージョン 1 との比較

次の Amazon Aurora MySQL 機能は、Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2021-03-04 (バージョン 2.07.4) (廃止)

バージョン: 2.07.4

Aurora MySQL 2.07.4 は一般公開されています。Aurora MySQL 2.* バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.* バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリース
は、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

現在サポートされている Aurora MySQL リリースから取得したスナップショットを Aurora MySQL 2.07.4 で復元できます。既存の Aurora MySQL 2.* データベースクラスターを Aurora MySQL 2.07.4 にアップグレードすることもできます。既存の Aurora MySQL 1.* クラスターは直接 2.07.4 にアップグレードできませんが、そのスナップショットは Aurora MySQL 2.07.4 に復元できます。

古いバージョンの Aurora MySQL を使用してクラスターを作成するには AWS Management Console、AWS CLI、または RDS API を使用してエンジンバージョンを指定してください。

> **Note**
>
> このバージョンは、長期サポート (LTS) リリースとして指定されています。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL 長期サポート (LTS) リリース」を参照してください。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

セキュリティの修正内容:

- 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

可用性の向上:

- ネットワークパケットの読み取りまたは書き込み中にネットワークエラーが発生した場合にクライアントがハングすることがある問題を修正しました。
- DDL が中断された後のエンジン再起動時間を改善しました。
- ページのプリフェッチリクエスト中に DDL または DML がエンジンを再起動する問題を修正しました。
- Aurora リードレプリカでテーブル/インデックスのリバーススキャンを実行中にレプリカが再起動することがある問題を修正しました。
- クローンクラスター操作で、クローンに時間がかかることがある問題を修正しました。
- 地理空間列に対してパラレルクエリ最適化を使用するときに、データベースが再起動することがある問題を修正しました。
- バイナリログレプリカが `HA_ERR_KEY_NOT_FOUND` エラーで停止する問題を修正しました。

MySQL Community Edition バグ修正の統合

- ' ' (スペース)、'%'、',' を含むトークンを処理するときにフルテキスト ngram パーサーで発生する問題を修正しました。ngram パーサーを使用する場合は、FTS インデックスを再構築する必要があります。(バグ #25873310)
- ネストされた SQL ビューでクエリを実行しているときにエンジンを再起動することがある問題を修正しました。(バグ #27214153、バグ #26864199)

Aurora MySQL バージョン 1 との比較

次の Amazon Aurora MySQL 機能は、Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2020-11-10 (バージョン 2.07.3) (廃止)

バージョン: 2.07.3

Aurora MySQL 2.07.3 は一般公開されています。Aurora MySQL 2.* バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.* バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

現在サポートされている Aurora MySQL リリースから取得したスナップショットを Aurora MySQL 2.07.3 で復元できます。既存の Aurora MySQL 2.* データベースクラスターを Aurora MySQL 2.07.3 にアップグレードすることもできます。既存の Aurora MySQL 1.* クラスターは直接 2.07.3 にアップグレードできませんが、そのスナップショットは Aurora MySQL 2.07.3 に復元できます。

古いバージョンの Aurora MySQL を使用してクラスターを作成するには AWS Management Console、AWS CLI、または RDS API を使用してエンジンバージョンを指定してください。

> **Note**
>
> このバージョンは、長期サポート (LTS) リリースとして指定されています。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL 長期サポート (LTS) リリース」を参照してください。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

セキュリティの修正内容:

マネージド型の環境での処理を微調整するための修正およびその他の機能強化。

- 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

互換性のない変更:

このバージョンでは、mysqldump コマンドの動作に影響するアクセス許可の変更が導入されています。ユーザーは、PROCESS テーブルにアクセスする INFORMATION_SCHEMA.FILES 特権を有している必要があります。変更せずに mysqldump コマンドを実行するには、PROCESS コマンドが接続するデータベースユーザーに mysqldump 特権を付与します。mysqldump オプションを指定して --no-tablespaces コマンドを実行することもできます。このオプションを使用すると、mysqldump 出力に CREATE LOGFILE GROUP または CREATE TABLESPACE ステートメントは含まれません。この場合、mysqldump コマンドは INFORMATION_SCHEMA.FILES テーブルにアクセスしないため、PROCESS アクセス許可を付与する必要はありません。

可用性の向上:

- 接続/クエリの強制終了とセッション終了の間のロックマネージャの競合状態が修正され、データベースが再起動しました。
- クエリキャッシュを有効にして複数のテーブルまたはデータベースにアクセスするマルチクエリステートメントを実行した後、データベースが再起動する問題を修正しました。
- セカンダリインデックスを使用した仮想列の更新により、再起動が繰り返し発生する可能性がある問題を修正しました。

MySQL Community Edition バグ修正の統合

- InnoDB: マスターの XA 準備段階に正常に実行された同時 XA トランザクションは、スレーブで再生されたときに競合し、アプライヤスレッドでロック待機タイムアウトが発生しました。競合は、トランザクションがスレーブ上で連続して再生された際のギャップロック範囲が異なることによって発生します。このタイプの競合を防ぐために、READ COMMITTED の分離レベルの XA トランザクションによる ギャップロックは、XA トランザクションが準備段階に達したときに解放される (継承されなくなる) ようになりました。(バグ #27189701、バグ #25866046)
- InnoDB: READ COMMITTED の分離レベルの使用中、外部キーの検証で不要なギャップロックが実行されました。(バグ #25082593)
- レプリケーション: XA トランザクションを使用中、レプリケーションスレーブ上の applier (SQL) スレッドでロック待機タイムアウトまたはデッドロックが発生した場合、自動再試行は機能しません。これは、SQL スレッドがロールバックを行う間、XA トランザクションをロールバックしないことが原因でした。トランザクションが再試行された際の初期のイベントは XA START であり、XA トランザクションがすでに進行中だったために無効となり、XAER_RMFAIL エラーが発生したということです。(バグ #24764800)
- レプリケーション: トランザクション分離レベルが REPEATABLE READ に設定されている場合、インターリーブトランザクションがスレーブアプライヤをデッドロックすることがありました。(バグ #25040331)
- レプリケーション: すべての既存のリレーログファイル (Relay_Log_Space) の合計サイズに対する SHOW SLAVE STATUS ステートメントによって返される値は、リレーログファイルによって使用される実際のディスク容量よりもはるかに大きくなる可能性があります。I/O スレッドが値を更新する間に可変をロックしなかったため、SQL スレッドはリレーログファイルを自動的に削除し、I/O スレッドが値の更新を完了する前に低減された値を書き込むことができました。その後 I/O スレッドは元のサイズ計算を書き込み、SQL スレッドの更新を無視して削除したファイルの容量を再追加しました。同時更新を防ぎ正確に計算するため、更新中の Relay_Log_Space の値はロックされます。(バグ #26997096、バグ #87832)
- VALUES リストが結合を含むサブクエリを使用して 2 行目以降の値を生成した INSERT ステートメントの場合、必要な権限の解決に失敗するとサーバーが終了する可能性があります。(バグ #23762382)
- デフォルト値の CURRENT_TIMESTAMP が書き込まれた TIMESTAMP または DATETIME 列を持つテーブルで、そのテーブルに `BEFORE INSERT` トリガーがある場合は、これらの列が `0000-00-00 00:00:00` に初期化されることがあります。(バグ #25209512、バグ #84077)
- メタデータのPerformance Schemaオブジェクトの登録と登録解除を複数のスレッドが同時に試行すると、サーバーが終了する可能性があります。(バグ #26502135)

- 特定の SELECT ステートメントの内容からテーブルを作成したステートメントを含むストアドプロシージャを実行すると、メモリリークが発生する可能性があります。(バグ #25586773)
- ビューにアクセスしたクエリを含むストアドプロシージャを実行すると、セッションが終了するまで解放されなかったメモリが割り当てられる可能性があります。(バグ #25053286)
- サブクエリのマテリアル化の特定のケースで、サーバーが終了する可能性があります。これらのクエリは、マテリアル化が無効であることを示すエラーを生成するようになりました。(バグ #26402045)
- 結合バッファリングが使用されている場合、多くの左結合を持つクエリが低速になります (ブロックネストされたループアルゴリズムを使用する場合など)。(バグ #18898433、バグ #72854)
- オプティマイザは、2 番目の列に対して LIKE の句との内部結合を実行する際、複合インデックスの 2 番目の列をスキップしました。(バグ #28086754)

Aurora MySQL バージョン 1 との比較

次の Amazon Aurora MySQL 機能は、Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空

間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2020-04-17 (バージョン 2.07.2) (廃止)

バージョン: 2.07.2

Aurora MySQL 2.07.2 は一般利用可能です。Aurora MySQL 2.* バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.* バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

スナップショットは、現在サポートされている Aurora MySQL リリースから Aurora MySQL 2.07.2 に復元できます。既存の Aurora MySQL 2.* データベースクラスターは、Aurora MySQL 2.07.2 にアップグレードすることができます。既存の Aurora MySQL 1.* クラスターは直接 2.07.2 にアップグレードできませんが、そのスナップショットは Aurora MySQL 2.07.2 に復元できます。

古いバージョンの Aurora MySQL でクラスターを作成するには、、 AWS CLI、または RDS API を使用してエンジンバージョンを指定してください。 AWS Management Console

> **Note**
>
> このバージョンは、長期サポート (LTS) リリースとして指定されています。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL 長期サポート (LTS) リリース」を参照してください。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

セキュリティの修正内容:

- CVE-2016-8287
- CVE-2016-5634

優先度の高い修正:

- 書き込み負荷が高い一部のデータベースクラスターでクローン作成に時間がかかる問題を修正しました。
- リーダー DB インスタンスに対するクエリの実行プランでセカンダリインデックスを使用した場合に、クエリからコミットされていないデータが返されるという問題を修正しました。この問題は、プライマリまたはセカンダリインデックスのキー列を変更するデータ操作言語 (DML) のオペレーションによって影響を受けるデータに限定されます。

全般的な機能強化:

- FTS (全文検索) インデックスが含まれている Aurora 1.x DB クラスターを Aurora 2.x DB クラスターに復元する際に、復元速度が遅くなる問題を修正しました。
- テーブル名に特殊文字が含まれているパーティショニングテーブルがある Aurora 1.x データベーススナップショットを Aurora 2.x DB クラスターに復元する際に、復元速度が遅くなる問題を修正しました。
- リーダー DB インスタンスで低速クエリログと一般ログをクエリするときにエラーが発生する問題を修正しました。

MySQL Community Edition バグ修正の統合

- バグ #23104498: メモリ使用量の合計を報告する際のPerformance Schema の問題を修正しました。(https://github.com/mysql/mysql-server/commit/20b6840df5452f47313c6f9a6ca075bfbc00a96b)
- バグ #22551677: オフラインにしようとするとデータベースエンジンがクラッシュする可能性があったPerformance Schemaの問題を修正しました。(https://github.com/mysql/mysql-server/commit/05e2386eccd32b6b444b900c9f8a87a1d8d531e9)
- バグ #23550835、バグ #23298025、バグ #81464: 内部バッファの容量を超えたことが原因でデータベースエンジンがクラッシュしていたPerformance Schemaの問題を修正しました。(https://github.com/mysql/mysql-server/commit/b4287f93857bf2f99b18fd06f555bbe5b12debfc)

Aurora MySQL バージョン 1 との比較

次の Amazon Aurora MySQL 機能は、Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空

間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2019-12-23 (バージョン 2.07.1) (廃止)

バージョン 2.07.1

Aurora MySQL 2.07.1 は一般利用可能です。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリース
は、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

スナップショットを現在サポートされている Aurora MySQL リリースから Aurora MySQL 2.07.1 に復元できます。既存の Aurora MySQL 2.* データベースクラスターは、Aurora MySQL 2.07.1 にアップグレードすることができます。既存の Aurora MySQL 1.* クラスターは直接 2.07.1 にアップグレードできませんが、そのスナップショットは Aurora MySQL 2.07.1 に復元できます。

古いバージョンの Aurora MySQL でクラスターを作成するには、、 AWS CLI、または RDS API を使用してエンジンバージョンを指定してください。 AWS Management Console

> **Note**
>
> このバージョンは現在、 AWS GovCloud (米国東部) [us-gov-east-1]、(米国西部) [us-gov-west-1]、中国 AWS GovCloud (寧夏) [cn-northwest-1]、アジアパシフィック (香港) [ap-east-1]、および中東 (バーレーン) [me-south-1] ではご利用いただけません。 AWS ご利用可能になりましたら、別途お知らせします。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

優先度の高い修正:

- Aurora 固有のデータベーストレースおよびログサブシステムで、確保できるメモリを不足させる低速メモリリークを修正しました。

全般の安定性に関する修正:

- 内部的に中間テーブルを使用するマルチテーブル結合や集約を伴う複雑なクエリの実行中に発生するクラッシュを修正しました。

Aurora MySQL バージョン 1 との比較

次の Amazon Aurora MySQL 機能は、Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2019-11-25 (バージョン 2.07.0) (廃止)

バージョン: 2.07.0

Aurora MySQL 2.07.0 は一般利用可能です。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリース
は、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

スナップショットを現在サポートされている Aurora MySQL リリースから Aurora MySQL 2.07.0 に復元できます。既存の Aurora MySQL 2.* データベースクラスターは、Aurora MySQL 2.07.0 にアップグレードすることができます。既存の Aurora MySQL 1.* クラスターは直接 2.07.0 にアップグレードできませんが、そのスナップショットは Aurora MySQL 2.07.0 に復元できます。

古いバージョンの Aurora MySQL でクラスターを作成するには、、 AWS CLI、または RDS API を使用してエンジンバージョンを指定してください。 AWS Management Console

> **Note**
>
> このバージョンは現在、 AWS GovCloud (米国東部) [us-gov-east-1]、(米国西部) [us-gov-west-1]、 AWS GovCloud 中国 (寧夏) [cn-northwest-1]、アジア太平洋 (香港) [ap-east-1]、中東 (バーレーン) [me-south-1]、南米 (サンパウロ) [sa-east-1] ではご利用いただけません。AWS ご利用可能になりましたら、別途お知らせします。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

新機能:

- グローバルデータベースでは、米国東部 (バージニア北部) [us-east-1]、米国東部 (オハイオ) [us-east-2]、米国西部 (北カリフォルニア) [us-west-1]、米国西部 (オレゴン) [us-west-2]、ヨーロッパ

(アイルランド) [eu-west-1]、ヨーロッパ (ロンドン) [eu-west-2] のリージョンに展開されているデータベースクラスターに、読み取り専用のセカンダリレプリカリージョンを追加できるようになりました。、ヨーロッパ (パリ) [eu-west-3]、アジアパシフィック (東京) [ap-northeast-1]、アジアパシフィック (ソウル) [ap-northeast-2]、アジアパシフィック (シンガポール) [ap-southeast-1]、アジアパシフィック (シドニー) [ap-southeast-2]、カナダ (中部) [ca-central-1]、ヨーロッパ (フランクフルト) [AWS eu-central-1]、およびアジアパシフィック (ムンバイ) [ap-south-1]。

- Amazon Aurora 機械学習は、Aurora MySQL AWS データベースと機械学習 (ML) サービスを高度に最適化して統合したものです。Aurora 機械学習によって開発者は、すでにデータベース開発に使用している慣れ親しんだ SQL プログラミング言語を使用して ML モデルを呼び出すことにより、さまざまな ML ベースの予測をデータベースアプリケーションに追加することができます。カスタム統合を構築したり別個のツールを学習する必要はありません。詳細については、「Amazon Aurora での機械学習 (ML) 機能の使用」を参照してください。
- リードレプリカでの ANSI `READ COMMITTED` 分離レベルのサポートが追加されました。この分離レベルによって、ライターノードでの高い書き込みスループットに影響を与えることなく、リードレプリカに対する長期実行クエリが可能になります。詳細については、「Aurora MySQL の分離レベル」を参照してください。

重要な修正:

- CVE-2019-2922
- CVE-2019-2923
- CVE-2019-2924
- CVE-2019-2910

優先度の高い修正:

- 長期のデータベースダウンタイムが発生していた DDL 復旧の問題を修正しました。複数テーブルの drop ステートメント (例えば、`DROP TABLE t1, t2, t3`) の実行後に使用できなくなるクラスターは、このバージョンに更新する必要があります。
- 長期のデータベースダウンタイムが発生していた DDL 復旧の問題を修正しました。`INPLACE ALTER TABLE` DDL ステートメントの実行後に使用できなくなるクラスターは、このバージョンに更新する必要があります。

全般の安定性に関する修正:

- `information_schema.replica_host_status` テーブルで不整合データが生成された問題を修正しました。

MySQL Community Edition バグ修正の統合

- バグ #26251621: INCORRECT BEHAVIOR WITH TRIGGER AND GCOL
- バグ #22574695: ASSERTION `!TABLE || (!TABLE->READ_SET || BITMAP_IS_SET(TABLE->READ_SET, FIEL
- バグ #25966845: INSERT ON DUPLICATE KEY GENERATE A DEADLOCK
- バグ #23070734: CONCURRENT TRUNCATE TABLES CAUSE STALL
- バグ #26191879: FOREIGN KEY CASCADES USE EXCESSIVE MEMORY
- バグ #20989615: INNODB AUTO_INCREMENT PRODUCES SAME VALUE TWICE

Aurora MySQL バージョン 1 との比較

次の Amazon Aurora MySQL 機能は、Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

Aurora MySQL 2.07.0 は、MySQL 5.7 とのワイヤー互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

Aurora MySQL 2.07.0 では、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2019-11-22 (バージョン 2.06.0) (廃止)

バージョン: 2.06.0

Aurora MySQL 2.06.0 は一般利用可能です。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリースは、1.14.*、1.15.*、1.16.*、1.17.*、1.18.*、1.19.*、2.01.*、2.02.*、2.03.*、2.04.*、2.05.*、および 2.06.* です。

スナップショットを現在サポートされている Aurora MySQL リリースから Aurora MySQL 2.06.0 に復元できます。既存の Aurora MySQL 2.* データベースクラスターは、Aurora MySQL 2.06.0 にアップグレードすることができます。既存の Aurora MySQL 1.* クラスターは直接 2.06.0 にアップグレードできませんが、そのスナップショットは Aurora MySQL 2.06.0 に復元できます。

古いバージョンの Aurora MySQL でクラスターを作成するには、、 AWS CLI、または RDS API を使用してエンジンバージョンを指定してください。 AWS Management Console

> **Note**
>
> このバージョンは現在、 AWS GovCloud (米国東部) [us-gov-east-1]、(米国西部) [us-gov-west-1]、中国 AWS GovCloud (寧夏) [cn-northwest-1]、アジアパシフィック (香港) [ap-

east-1]、および中東 (バーレーン) [me-south-1] ではご利用いただけません。 AWS ご利用可能になりましたら、別途お知らせします。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

新機能:

- Aurora MySQL クラスターで、インスタンスタイプ db.r5.8xlarge、db.r5.16xlarge、および db.r5.24xlarge がサポートされるようになりました。Aurora MySQL クラスターのインスタンスタイプの詳細については、「Amazon Aurora ユーザーガイド」の「Aurora DB インスタンスクラス」を参照してください。
- ハッシュ結合機能が一般利用可能になりました。Aurora ラボモード設定をオンにする必要はありません。等価結合を使用して大量のデータを結合する必要がある場合は、この機能によりクエリのパフォーマンスが向上することがあります。この機能の使い方については、「Amazon Aurora ユーザーガイド」の「Aurora Serverless の Data API の使用」を参照してください。
- ホット行の競合機能が一般利用可能になりました。Aurora ラボモード設定をオンにする必要はありません。この機能により、同じページの列で多くのトランザクションが競合しているワークロードのスループットが大幅に向上します。
- Aurora MySQL 2.06 以降では、バックアップからデータを復元しないで、DB クラスターを特定の時刻に「巻き戻し」することができます。この機能はバックトラックと呼ばれ、間違ったテーブルのドロップや、意図しない行の削除など、ユーザーエラーをすばやく復元することができます。バックトラックは、大きなデータベースの場合でも数秒で完了します。AWS 概要についてはブログを読み、詳細については Amazon Aurora ユーザーガイドの「Aurora DB クラスターのバックトラッキング」を参照してください。

- Aurora 2.06 以降では、 AWS Lambda ネイティブ関数による同期呼び出しがサポートされています。`lambda_sync()`また、ネイティブ関数 `lambda_async()` もサポートするようになり、Lambda の非同期呼び出しのための既存のストアードプロシージャの代わりに使用できます。Lambda 関数の呼び出しについては、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。

重要な修正:

なし。

優先度の高い修正:

セキュリティの修正内容

- 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

接続の処理

- データベースの可用性が向上し、1 つ以上の DDL を実行している際のクライアント接続での急増に適切に対応できるようになりました。必要に応じて追加のスレッドを一時的に作成することによって処理されます。DDL の処理中に接続が急増した後、データベースが応答しなくなる場合は、アップグレードすることをお勧めします。

エンジンの再起動

- エンジンの再起動中に長期にわたって利用不可になる問題を修正しました。これは、バッファープール初期化の問題に対処しています。この問題が発生することはまれですが、サポートされているリリースに影響する可能性があります。
- バイナリログ (binlog) のマスターとして設定されたデータベースが、負荷の高い書き込みワークロードの実行中に再起動する問題を修正しました。

全般の安定性に関する修正:

- キャッシュされていないデータにアクセスするクエリの実行が通常より遅くなることがあった点が改善されました。キャッシュされていないデータへのアクセス中に原因不明の読み取りレイテンシーが増えているお客様は、この問題が発生している場合があるため、アップグレードすることをお勧めします。
- パーティショニングされたテーブルをデータベーススナップショットから復元できなかった問題を修正しました。Aurora MySQL 1.* データベースのスナップショットから復元されたデータベースにあるパーティショニングされたテーブルへのアクセス時にエラーが発生するお客様は、このバージョンを使用することをお勧めします。
- 書き込み DB インスタンスでの DDL クエリの実行中に発生する、読み取りクエリを処理するスレッドとスキーマの変更を適用するスレッドとのロック競合を修正することによって、Aurora レプリカの安定性が向上しました。
- DDL 操作によってトリガーされる `mysql.innodb_table_stats` テーブルの更新に関する安定性の問題を修正しました。
- ネストされたクエリが Aurora レプリカのテンポラリテーブルに対して実行されるとき、間違って `ERROR 1836` が報告される問題を修正しました。

パフォーマンスの拡張:

- クエリキャッシュがバイナリログワーカーで無効になっている場合にキャッシュへの不要な API コールを防止することによって、バイナリログのレプリケーションパフォーマンスが向上しました。

Aurora MySQL バージョン 1 との比較

次の Amazon Aurora MySQL 機能は、Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

Aurora MySQL 2.06.0 は、MySQL 5.7 とのワイヤー互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

Aurora MySQL 2.06.0 では、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2019-11-11 (バージョン 2.05.0) (廃止)

バージョン: 2.05.0

Aurora MySQL 2.05.0 は一般利用可能です。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリースは、1.14.*、1.15.*、1.16.*、1.17.*、1.18.*、1.19.*、2.01.*、2.02.*、2.03.*、および 2.04.* です。

スナップショットを現在サポートされている Aurora MySQL リリースから Aurora MySQL 2.05.0 に復元できます。既存の Aurora MySQL 2.* データベースクラスターは、最大 2.04.6 から Aurora MySQL 2.05.0 にアップグレードすることができます。既存の Aurora MySQL 1.* クラスターは直接 2.05.0 にアップグレードできませんが、そのスナップショットは Aurora MySQL 2.05.0 に復元できます。

古いバージョンの Aurora MySQL でクラスターを作成するには、、 AWS CLI、または RDS API を使用してエンジンバージョンを指定してください。 AWS Management Console

> **Note**
>
> このバージョンは現在、 AWS GovCloud (米国東部) [us-gov-east-1]、(米国西部) [us-gov-west-1]、中国 AWS GovCloud (寧夏) [cn-northwest-1]、アジア太平洋 (香港) [ap-east-1]、ヨーロッパ (ストックホルム) [eu-north-1]、および中東 (バーレーン) [me-south-1] ではご利用いただけません。 AWS ご利用可能になりましたら、別途お知らせします。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

重要な修正:

- 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
- パラメータ sync_binlog の値が 1 に設定されていなかった場合に、マスターの現在のバイナリログファイルのイベントがワーカーでレプリケートされなかった問題を修正しました。

優先度の高い修正:

- 64 テビバイト (TiB) に近いサイズのデータベースを使用するお客様は、Aurora ストレージの制限に近いボリュームに影響する安定性のバグが引き起こすダウンタイムを防止するため、このバージョンにアップグレードすることを強くお勧めします。
- バイナリログのマスターのフォアグラウンドクエリのパフォーマンスを優先してレプリケーションラグの増加を防ぐために、パラメータ aurora_binlog_replication_max_yield_seconds のデフォルト値がゼロに変更されました。

MySQL バグ修正の統合

- バグ #23054591: PURGE BINARY LOGS TO がバイナリログファイル全体を読み取っているため MySql 、処理が停止する

Aurora MySQL バージョン 1 との比較

次の Amazon Aurora MySQL 機能は、Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。

- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- 関数を同期的に呼び出すためのネイティブ関数。 AWS Lambda 詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

Aurora MySQL 2.05.0 は、MySQL 5.7 とのワイヤー互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

Aurora MySQL 2.05.0 では、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2020-08-14 (バージョン 2.04.9) (廃止)

バージョン: 2.04.9

Aurora MySQL 2.04.9 は一般利用可能です。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリース
は、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

古いバージョンの Aurora MySQL でクラスターを作成するには、、 AWS CLI、または RDS API を使用してエンジンバージョンを指定してください。 AWS Management Console

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> このバージョンは現在、(AWS 米国東部) [-1]、 AWS GovCloud (米国西部) [us-gov-east-1]、アジア太平洋 AWS GovCloud (香港) [us-gov-westap-east-1]、および中東 (バーレーン) [me-south-1] ではご利用いただけません。ご利用可能になりましたら、別途お知らせします。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

優先度の高い修正:

重要な修正:

- 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

可用性の向上:

- `kill session` コマンドの実行により、データベースの再起動またはフェイルオーバーが発生する可能性がある問題を修正しました。この問題が発生した場合は、サポートに連絡して、インスタンスでこの修正を有効にしてください。 AWS
- 内部的に中間テーブルを使用する複数テーブル結合および集約を含む複雑なクエリの実行中に、データベースが再起動する問題を修正しました。
- 複数のテーブルで `DROP TABLE` が中断されたためにデータベースが再起動する問題を修正しました。
- データベースのリカバリ中にデータベースのフェールオーバーが発生する問題を修正しました。
- 監査ログと低速クエリログが有効な場合に threads_running の誤った報告が原因で発生したデータベースの再起動が修正されました。
- 実行中に `kill query` コマンドがスタックすることがある問題を修正しました。
- トランザクションのロールバック中にデータベースの再起動またはフェイルオーバーが発生する、ロックマネージャーの競合状態を修正しました。
- 複数の接続でフルテキスト検索インデックスを使用して同じテーブルを更新しようとすると、データベースの再起動またはフェイルオーバーがトリガーされる問題を修正しました。
- インデックスのパージ時にデッドラッチが発生し、フェイルオーバーまたは再起動につながる可能性がある問題を修正しました。

全般的な機能強化:

- リードレプリカのクエリで、コミットされていないトランザクションのデータが使用される問題を修正しました。この問題は、データベースの再起動直後にスタートされるトランザクションに限定されます。
- トリガーが定義されているテーブルで `INPLACE ALTER TABLE` 中、および DDL に `RENAME` 句が含まれていない場合に発生していた問題を修正しました。
- 書き込み負荷が高い一部のデータベースクラスターでクローン作成に時間がかかる問題を修正しました。
- パーティション化されたテーブルで名前にスペースが埋め込まれている場合に、アップグレード中に発生する問題を修正しました。
- リードレプリカが、ライターで最近コミットされたトランザクションの結果の一部を一時的に表示することがある問題を修正しました。
- FTS テーブルに対するリードレプリカのクエリで、古い結果が生成されることがある問題を修正しました。これは、リードレプリカの FTS クエリが、ライターの同じ FTS テーブルに対する

INFORMATION_SCHEMA.INNODB_SYS_TABLES のクエリに密接に従っている場合にのみ発生します。

- FTS (全文検索) インデックスが含まれている Aurora 1.x データベースクラスターを Aurora 2.x データベースクラスターに復元する際に、復元速度が遅くなる問題を修正しました。
- server_audit_incl_users および server_audit_excl_users グローバルパラメータの最大許容長を 2000 に拡張しました。
- Aurora 1.x から Aurora 2.x への復元の完了に長い時間がかかることがある問題を修正しました。
- ストアドプロシージャによる lambda_async 呼び出しが Unicode で機能しない問題を修正しました。
- 空間インデックスがオフレコードジオメトリ列を正しく処理しない場合に発生する問題を修正しました。
- リーダー DB インスタンスでクエリが失敗し、「Operation terminated (internal error)」という InternalFailureException エラーが表示される問題を修正しました。

MySQL バグ修正の統合

- バグ #23070734、バグ #80060: Concurrent TRUNCATE TABLEs cause stalls
- バグ #23103937: PS_TRUNCATE_ALL_TABLES() がSUPER_READ_ONLY モードで機能しない
- バグ #22551677: サーバーをオフラインにすると、Performance Schema内の競合状態により、サーバーが終了する可能性があります。
- バグ #27082268: 無効な FTS の同期。
- バグ #12589870: クエリキャッシュが有効なときにマルチクエリステートメントで再起動が発生する問題を修正しました。
- バグ #26402045: サブクエリのマテリアル化の特定のケースで、サーバーが終了する可能性があります。これらのクエリは、マテリアル化が無効であることを示すエラーを生成するようになりました。
- バグ #18898433: 結合バッファリングが使用されている場合 (ブロックネストされたループアルゴリズムを使用する場合など)、多くの左結合を持つクエリが低速になります。
- バグ #25222337: 仮想インデックスの仮想列フィールド名が NULL の場合、外部キー制約の影響を受ける仮想列への入力中に発生するフィールド名の比較中にサーバー終了が発生しました。 (https://github.com/mysql/mysql-server/commit/273d5c9d7072c63b6c47dbef6963d7dc491d5131)

- バグ #25053286: ビューにアクセスしたクエリを含むストアドプロシージャを実行すると、セッションが終了するまで解放されなかったメモリが割り当てられました。(https://github.com/mysql/mysql-server/commit/d7b37d4d141a95f577916448650c429f0d6e193d)
- バグ #25586773: 特定の SELECT (https://dev.mysql.com/doc/refman/5.7/en/select.html) ステートメントの内容からテーブルを作成したステートメントを含むストアドプロシージャを実行すると、メモリリークが発生する可能性があります。(https://github.com/mysql/mysql-server/commit/88301e5adab65f6750f66af284be410c4369d0c1)
- バグ #26666274: Performance Schemaバッファコンテナ内の無限ループ
- バグ #23550835、バグ #23298025、バグ #81464: 内部バッファがいっぱいになったときに SELECT Performance Schemaテーブルによってサーバーが終了する可能性があります。

Aurora MySQL バージョン 1 との比較

次の Amazon Aurora MySQL 機能は、Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

Aurora MySQL 2.04.9 は、MySQL 5.7 とのワイヤー互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空

間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

Aurora MySQL 2.04.9 では、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2019-11-20 (バージョン 2.04.8) (廃止)

バージョン: 2.04.8

Aurora MySQL 2.04.8 は一般利用可能です。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリース
は、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

2.* Aurora MySQL リリースのすべてのスナップショットを Aurora MySQL 2.04.8 に復元できます。既存の Aurora MySQL 2.* データベースクラスターは、Aurora MySQL 2.04.8 にアップグレードすることができます。

古いバージョンの Aurora MySQL でクラスターを作成するには、、 AWS CLI、または RDS API を使用してエンジンバージョンを指定してください。 AWS Management Console

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> このバージョンは現在、(AWS 米国東部) [-1]、 AWS GovCloud (米国西部) [us-gov-east-1]、アジア太平洋 AWS GovCloud (香港) [us-gov-westap-east-1]、および中東 (バーレーン) [me-south-1] ではご利用いただけません。ご利用可能になりましたら、別途お知らせします。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

新機能:

- リードレプリカの改善:
  - Aurora DB クラスター内の読み取りインスタンスへデータを効率的に送信することによって、書き込みインスタンスからのネットワークトラフィックを削減しました。この改善は、レプリカが遅延して再起動する事態を回避できるため、デフォルトで有効化されます。この機能のためのパラメータは `aurora_enable_repl_bin_log_filtering` です。
  - 圧縮を使用して、Aurora DB クラスターにおける書き込みインスタンスから読み取りインスタンスへのネットワークトラフィックを削減しました。この改善は、8xlarge および 16xlarge インスタンスクラスに対してのみデフォルトで有効化されます。これらのインスタンスは、圧縮のための追加 CPU オーバーヘッドを許容できるためです。この機能のためのパラメータは `aurora_enable_replica_log_compression` です。

優先度の高い修正:

- Aurora 書き込みインスタンスのメモリ管理が向上しました。これによって、Aurora DB クラスター内に読み取りインスタンスがある状態で負荷の高いワークロードを実行中にメモリ不足の問題によって発生する書き込み機能の再起動を防止します。
- Performance Schemaのオブジェクトに同時にアクセスしているときにエンジンの再起動が発生するスケジューラの非決定性の条件を修正しました。

Aurora MySQL バージョン 1 との比較

次の Amazon Aurora MySQL 機能は、Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

Aurora MySQL 2.04.8 は、MySQL 5.7 とのワイヤー互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

Aurora MySQL 2.04.8 では、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン

- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- CREATE TABLESPACE SQL ステートメント

Aurora MySQL データベースエンジンの更新 2019-11-14 (バージョン 2.04.7) (廃止)

バージョン: 2.04.7

Aurora MySQL 2.04.7 は一般利用可能です。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

スナップショットを現在サポートされている Aurora MySQL リリースから Aurora MySQL 2.04.7 に復元できます。既存の Aurora MySQL 2.* データベースクラスターは、Aurora MySQL 2.04.7 にアップグレードすることができます。既存の Aurora MySQL 1.* クラスターは直接 2.04.7 にアップグレードできませんが、そのスナップショットは Aurora MySQL 2.04.7 に復元できます。

古いバージョンの Aurora MySQL でクラスターを作成するには、、 AWS CLI、または RDS API を使用してエンジンバージョンを指定してください。 AWS Management Console

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> このバージョンは現在、(AWS 米国東部) [-1]、 AWS GovCloud (米国西部) [us-gov-east-1]、アジア太平洋 AWS GovCloud (香港) [us-gov-westap-east-1]、および中東 (バーレーン) [me-south-1] ではご利用いただけません。ご利用可能になりましたら、別途お知らせします。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

優先度の高い修正:

接続の処理

- データベースの可用性が向上し、1 つ以上の DDL を実行している際のクライアント接続での急増に適切に対応できるようになりました。必要に応じて追加のスレッドを一時的に作成することによって処理されます。DDL の処理中に接続が急増した後、データベースが応答しなくなる場合は、アップグレードすることをお勧めします。
- `Threads_running` グローバルステータス可変の値が正しくない問題を修正しました。

エンジンの再起動

- エンジンの再起動中に長期にわたって利用不可になる問題を修正しました。これは、バッファープール初期化の問題に対処しています。この問題が発生することはまれですが、サポートされているリリースに影響する可能性があります。

全般の安定性に関する修正:

- キャッシュされていないデータにアクセスするクエリの実行が通常より遅くなることがあった点が改善されました。キャッシュされていないデータへのアクセス中に原因不明の読み取りレイテンシーが増えているお客様は、この問題が発生している場合があるため、アップグレードすることをお勧めします。

Aurora MySQL バージョン 1 との比較

次の Amazon Aurora MySQL 機能は、Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

Aurora MySQL 2.04.7 は、MySQL 5.7 とのワイヤー互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

Aurora MySQL 2.04.7 では、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2019-09-19 (バージョン 2.04.6) (廃止)

バージョン: 2.04.6

Aurora MySQL 2.04.6 は一般利用可能です。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリース
は、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

既存の Aurora MySQL 2.* データベースクラスターは、Aurora MySQL 2.04.6 にアップグレードすることができます。Aurora MySQL 1.* クラスターをインプレースアップグレードすることはできません。この制限は、以降の Aurora MySQL 2.* リリースで解除されます。Aurora MySQL 1.14.*、1.15.*、1.16.*、1.17.*、1.18.*、1.19.*、2.01.*、2.02.*、2.03.*、および 2.04.* のスナップショットは Aurora MySQL 2.04.6 に復元することができます。

古いバージョンの Aurora MySQL を使用するには、、、または Amazon RDS API を使用してエンジンバージョンを指定して AWS Management Console、新しいデータベースクラスターを作成できます。 AWS CLI

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> このバージョンは現在、ヨーロッパ (ロンドン) [eu-west-2]、(米国東部) [-1]、 AWS GovCloud (米国西部) [us-gov-east-1]、 AWS GovCloud 中国 (寧夏) [cn-northwest-1]、およびアジア太平洋 (香港) [us-gov-westap-east-1] ではご利用いただけません。 AWS ご利用可能になりましたら、別途お知らせします。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

- パラメータ sync_binlog の値が 1 に設定されていなかった場合に、マスターの現在のバイナリログファイルのイベントがワーカーでレプリケートされなかった問題を修正しました。
- バイナリログのマスターのフォアグラウンドクエリのパフォーマンスを優先してレプリケーションラグの増加を防ぐために、パラメータ aurora_binlog_replication_max_yield_seconds のデフォルト値がゼロに変更されました。

MySQL バグ修正の統合

- バグ #23054591: バイナリログを消去するとバイナリログファイル全体が読み込まれ、処理が停止する MySql

Aurora MySQL バージョン 1 との比較

次の Amazon Aurora MySQL 機能は、Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- 関数を同期的に呼び出すためのネイティブ関数。 AWS Lambda 詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

Aurora MySQL 2.04.6 は、MySQL 5.7 とのワイヤー互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

Aurora MySQL 2.04.6 では、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2019-07-08 (バージョン 2.04.5) (廃止)

バージョン: 2.04.5

Aurora MySQL 2.04.5 は一般利用可能です。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

既存の Aurora MySQL 2.* データベースクラスターは、Aurora MySQL 2.04.5 にアップグレードすることができます。Aurora MySQL 1.* クラスターをインプレースアップグレードすることはできません。この制限は、以降の Aurora MySQL 2.* リリースで解除されます。Aurora MySQL

1.14.*、1.15.*、1.16.*、1.17.*、1.18.*、1.19.*、2.01.*、2.02.*、2.03.*、および 2.04.* のスナップショットを Aurora MySQL 2.04.5 に復元することができます。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

セキュリティの修正内容:

- CVE-2016-3518

一般的な修正:

- データベースを再起動する原因となった、ストレージボリュームの拡大中の競合状態を修正しました。
- データベースを再起動する原因となったボリュームオープン中の内部通信障害を修正しました。
- パーティションされたテーブルの `ALTER TABLE ALGORITHM=INPLACE` の DDL 復元サポートを追加しました。
- データベースを再起動する原因となった `ALTER TABLE ALGORITHM=COPY` の DDL 復元の問題を修正しました。
- ライターの重度な削除ワークロードの Aurora レプリカの安定性が向上しました。
- 全文検索インデックスの同期を実行しているスレッドと、辞書キャッシュから全文検索テーブルの削除を実行しているスレッドとの間のデッドラッチが原因でデータベースが再起動される問題を修正しました。
- バイナリログマスターへの接続が不安定なときに、DDL レプリケーション中に発生するバイナリログワーカーの安定性の問題を修正しました。
- out-of-memory データベースが再起動する原因となっていた全文検索コードの問題を修正しました。

- 64 テビバイト (TiB) のボリューム全体を使用したときに再起動する原因となった Aurora ライターの問題を修正しました。
- データベースを再起動させるPerformance Schema機能の競合状態を修正しました。
- ネットワークプロトコル管理のエラーの処理中に接続が中止される問題を修正。

Aurora MySQL バージョン 1 との比較

次の Amazon Aurora MySQL 機能は、Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

Aurora MySQL 2.04.5 は、MySQL 5.7 とのワイヤー互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

Aurora MySQL 2.04.5 では、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加

- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- CREATE TABLESPACE SQL ステートメント

Aurora MySQL データベースエンジンの更新 2019-05-29 (バージョン 2.04.4) (廃止)

バージョン: 2.04.4

Aurora MySQL 2.04.4 が一般公開されました。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

新しい Aurora MySQL DB クラスターを作成する場合 (スナップショットの復元を含む)、MySQL 5.7 または MySQL 5.6 との互換性を選択できます。Aurora MySQL 1.* クラスターのインプレースアップグレードや Aurora MySQL 1.* クラスターの Amazon S3 バックアップからの Aurora MySQL 2.04.4. への復元を行うことはできません。これらの制限は、今後の Aurora MySQL 2.* リリースで削除する予定です。

Aurora MySQL 1.14.*、1.15.*、1.16.*、1.17.*、1.18.*、1.19.*、2.01.*、2.02.*、2.03.* および 2.04.* のスナップショットは Aurora MySQL 2.04.4 に復元することができます。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> このバージョンは現在、 AWS GovCloud (米国西部) [us-gov-west-1]、ヨーロッパ (ストックホルム) [eu-north-1]、中国 (寧夏) [cn-northwest-1]、およびアジアパシフィック (香港) [ap-

east-1] リージョンではご利用いただけません。 AWS ご利用可能になりましたら、別途お知らせします。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

- データを S3 から Aurora にロードする場合にエラーが発生する問題を修正。
- データを Aurora から S3 にアップロードする場合にエラーが発生する問題を修正。
- ネットワークプロトコル管理のエラーの処理中に接続が中止される問題を修正。
- パーティショニングされたテーブルを使用する際にクラッシュする問題を修正。
- 一部のリージョンで Performance Insights 機能が利用できない問題を修正。

Aurora MySQL バージョン 1 との比較

次の Amazon Aurora MySQL 機能は、Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。

- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

Aurora MySQL 2.04.4 は、MySQL 5.7 とのワイヤー互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

Aurora MySQL 2.04.4 では、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2019-05-09 (バージョン 2.04.3) (廃止)

バージョン 2.04.3

Aurora MySQL 2.04.3 は一般利用可能です。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

新しい Aurora MySQL DB クラスターを作成する場合 (スナップショットの復元を含む)、MySQL 5.7 または MySQL 5.6 との互換性を選択できます。Aurora MySQL 1.* クラスターのインプレースアップグレードや Aurora MySQL 1.* クラスターの Amazon S3 バックアップからの Aurora MySQL 2.04.3. への復元を行うことはできません。これらの制限は、今後の Aurora MySQL 2.* リリースで削除する予定です。

Aurora MySQL 1.14.*、1.15.*、1.16.*、1.17.*、1.18.*、1.19.*、2.01.*、2.02.*、2.03.* および 2.04.* のスナップショットを Aurora MySQL MySQL 2.04.3 に復元することができます。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> このバージョンは現在、 AWS GovCloud (米国西部) [us-gov-west-1] リージョンと中国 (寧夏) [cn-northwest-1] リージョンではご利用いただけません。 AWS ご利用可能になりましたら、別途お知らせします。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

- バイナリログワーカーとして設定された Aurora インスタンスで問題が発生する可能性のあるバイナリログレプリケーションのバグを修正。
- 大きなストアドルーチンを処理する際の問題が修正されました。 out-of-memory
- 特定の種類の `ALTER TABLE` コマンドを処理する際のエラーを修正。
- ネットワークプロトコル管理のエラーによる接続の失敗に関する問題を修正。

Aurora MySQL バージョン 1 との比較

次の Amazon Aurora MySQL 機能は、Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

Aurora MySQL 2.04.3 は、MySQL 5.7 とのワイヤー互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

Aurora MySQL 2.04.3 では、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン

- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- CREATE TABLESPACE SQL ステートメント

Aurora MySQL データベースエンジンの更新 2019-05-02 (バージョン 2.04.2) (廃止)

バージョン 2.04.2

Aurora MySQL 2.04.2 は一般利用可能です。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリース
は、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

新しい Aurora MySQL DB クラスターを作成する場合 (スナップショットの復元を含む)、MySQL 5.7 または MySQL 5.6 との互換性を選択できます。Aurora MySQL 1.* クラスターのインプレースアップグレードや Aurora MySQL 1.* クラスターの Amazon S3 バックアップからの Aurora MySQL 2.04.2. への復元を行うことはできません。これらの制限は、今後の Aurora MySQL 2.* リリースで削除する予定です。

Aurora MySQL 1.14.*、1.15.*、1.16.*、1.17.*、1.18.*、1.19.*、2.01.*、2.02.*、2.03.*、2.04.0 および 2.04.1 のスナップショットは Aurora MySQL 2.04.2 に復元することができます。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> このバージョンは現在、 AWS GovCloud (米国西部) [us-gov-west-1] リージョンと中国 (寧夏) [cn-northwest-1] リージョンではご利用いただけません。 AWS ご利用可能になりましたら、別途お知らせします。

> **Note**
>
> Aurora MySQL データベースクラスターをアップグレードする方法については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

- カスタム証明書を使用した SSL binlog レプリケーションのサポートを追加。Aurora MySQL の SSL binlog レプリケーションの使用については、「mysql_rds_import_binlog_ssl_material」を参照してください。
- 全文検索インデックスを持つテーブルが最適化されている場合に発生する Aurora プライマリインスタンスのデッドロックを修正。
- `SELECT(*)` を使用した特定のクエリのパフォーマンスがセカンダリインデックスを持つテーブルに影響を及ぼす可能性があるという Aurora レプリカの問題を修正。
- エラー 1032 が投稿される原因となった条件を修正。
- Aurora Replica の安定性を向上させるために複数のデッドロックを修正。

MySQL バグ修正の統合

- Bug #24829050 - INDEX_MERGE_INTERSECTION OPTIMIZATION CAUSES WRONG QUERY RESULTS

Aurora MySQL バージョン 1 との比較

次の Amazon Aurora MySQL 機能は、Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。

- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

Aurora MySQL 2.04.2 は、MySQL 5.7 とのワイヤー互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

Aurora MySQL 2.04.2 では、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2019-03-25 (バージョン 2.04.1) (廃止)

バージョン: 2.04.1

Aurora MySQL 2.04.1 は一般利用可能です。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリース
は、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

新しい Aurora MySQL DB クラスターを作成する場合 (スナップショットの復元を含む)、MySQL 5.7 または MySQL 5.6 との互換性を選択できます。Aurora MySQL 1.* クラスターのインプレースアップグレードや Aurora MySQL 1.* クラスターの Amazon S3 バックアップからの Aurora MySQL 2.04.1. への復元を行うことはできません。これらの制限は、今後の Aurora MySQL 2.* リリースで削除する予定です。

Aurora MySQL 1.14.*、1.15.*、1.16.*、1.17.*、1.18.*、1.19.*、2.01.*、2.02.*、2.03.*、2.04.0 のスナップショットを Aurora MySQL 2.04.1 に復元することができます。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> このバージョンは現在 AWS GovCloud (米国西部) [us-gov-west-1] リージョンではご利用いただけません。ご利用可能になりましたら、別途お知らせします。

> **Note**
>
> DB クラスターのアップグレード手順が変わりました。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

- 1.16 以前のバージョン用の Aurora MySQL 5.6 スナップショットを最新の Aurora MySQL 5.7 クラスターに復元できなかった問題を修正。

Aurora MySQL バージョン 1 との比較

次の Amazon Aurora MySQL 機能は、Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン

- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- CREATE TABLESPACE SQL ステートメント

Aurora MySQL データベースエンジンの更新 2019-03-25 (バージョン 2.04.0) (廃止)

バージョン: 2.04

Aurora MySQL 2.04 は一般利用可能です。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

新しい Aurora MySQL DB クラスターを作成する場合 (スナップショットの復元を含む)、MySQL 5.7 または MySQL 5.6 との互換性を選択できます。Aurora MySQL 1.* クラスターのインプレースアップグレードや Aurora MySQL 1.* クラスターの Amazon S3 バックアップからの Aurora MySQL 2.04.0. への復元を行うことはできません。これらの制限は、今後の Aurora MySQL 2.* リリースで削除する予定です。

Aurora MySQL 1.19.*、2.01.*、2.02.*、2.03.* のスナップショットを Aurora MySQL 2.04.0 に復元することができます。Aurora MySQL 1.14* 以前、1.15*、1.16*、1.17*、1.18.* のスナップショットを Aurora MySQL 2.04.0 に復元することはできません。この制限は、Aurora MySQL 2.04.1 では廃止されています。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> このバージョンは現在 AWS GovCloud (米国西部) [us-gov-west-1] リージョンではご利用いただけません。ご利用可能になりましたら、別途お知らせします。

> **Note**
>
> DB クラスターのアップグレード手順が変わりました。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

- GTID ベースのレプリケーションをサポート。Aurora MySQL での GTID ベースのレプリケーションの使用については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL の GTID ベースレプリケーションを使用する」を参照してください。
- テンポラリテーブル内の行を削除または更新するステートメントに InnoDB サブクエリが含まれている場合に Aurora Replica で `Running in read-only mode` エラーが誤ってスローされる問題を修正。

MySQL バグ修正の統合

- Bug #26225783: MYSQL CRASH ON CREATE TABLE (REPRODUCEABLE) -> INNODB: ALONG SEMAPHORE WAIT。

Aurora MySQL バージョン 1 との比較

次の Amazon Aurora MySQL 機能は、Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。

- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

この Aurora MySQL バージョンは MySQL 5.7 とワイヤ互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

この Aurora MySQL バージョンでは、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント

Aurora MySQL データベースエンジンの更新 2019-02-07 (バージョン 2.03.4) (廃止)

(バージョン 2.03.4)

Aurora MySQL 2.03.4 は一般利用可能です。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

新しい Aurora MySQL DB クラスターを作成する場合は、スナップショットから復元する場合も含めて、MySQL 5.7 または MySQL 5.6 との互換性を選択できます。

Aurora MySQL 1.* クラスターを Aurora MySQL 2.03.4 にインプレースアップグレードしたり、Amazon S3 バックアップから Aurora MySQL 2.03.4 に復元したりすることはできません。これらの制限は、今後の Aurora MySQL 2.* リリースで削除する予定です。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> このバージョンは現在、 AWS GovCloud (米国西部) [us-gov-west-1] リージョンと中国 (北京) [cn-north-1] リージョンではご利用いただけません。ご利用可能になりましたら、別途お知らせします。

> **Note**
>
> DB クラスターのアップグレード手順が変わりました。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

- UTF8MB 4 Unicode 9.0 のアクセントを区別し、大文字と小文字を区別しない照合 `utf8mb4_0900_as_ci` をサポートしました。

Aurora MySQL バージョン 1 との比較

以下の Amazon Aurora MySQL 機能は Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。

- 関数を同期的に呼び出すためのネイティブ関数。 AWS Lambda 詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

Aurora MySQL 2.03.4 は、MySQL 5.7 とのワイヤー互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

Aurora MySQL 2.03.4 では、現在、MySQL 5.7 の次の機能はサポートされていません。

- グローバルトランザクション ID (GTID) Aurora MySQL では、バージョン 2.04 以降の GTID をサポートしています。
- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント
- X プロトコル

Aurora MySQL データベースエンジンの更新 2019-01-18 (バージョン 2.03.3) (廃止)

(バージョン 2.03.3)

Aurora MySQL 2.03.3 は一般利用可能です。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

新しい Aurora MySQL DB クラスターを作成する場合は、スナップショットから復元する場合も含めて、MySQL 5.7 または MySQL 5.6 との互換性を選択できます。

Aurora MySQL 1.* クラスターを Aurora MySQL 2.03.3 にインプレースアップグレードしたり、Amazon S3 バックアップから Aurora MySQL 2.03.3 に復元したりすることはできません。これらの制限は、今後の Aurora MySQL 2.* リリースで削除する予定です。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> このバージョンは現在、 AWS GovCloud (米国西部) [us-gov-west-1] リージョンと中国 (北京) [cn-north-1] リージョンではご利用いただけません。ご利用可能になりましたら、別途お知らせします。

> **Note**
>
> DB クラスターのアップグレード手順が変わりました。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

CVE の修正

- CVE-2016-5436

重要な修正:

- インデックスに対してバックワードスキャンを実行すると Aurora Replica がデッドラッチになる問題を修正しました。
- Aurora プライマリインスタンスのパーティションテーブルでインプレース DDL オペレーションが実行されると、Aurora Replica が再起動する問題を修正しました。
- Aurora プライマリインスタンスで DDL オペレーションを実行後、クエリキャッシュが無効化されている間に Aurora Replica が再起動する問題を修正しました。
- Aurora プライマリインスタンスのそのテーブルで切り捨てが行われている間に、Aurora Replica がテーブルの `SELECT` クエリ中に再起動する問題を修正しました。
- インデックス作成された列のみアクセスされる MyISAM の一部テーブルに関する誤った結果の問題を修正しました。
- 約 40,000 回のクエリ後に `query_time` と `lock_time` で大きな値が定期的に誤って生成される低速ログの問題を修正しました。
- 「tmp」という名前のスキーマが原因で、RDS for MySQL から Aurora MySQL への移行がスタックする問題を修正しました。
- ログの更新中に、イベントが監査ログに見つからない問題を修正しました。
- 高速 DDL 機能のラボモードが有効になっている場合に、Aurora 5.6 スナップショットから復元した Aurora プライマリインスタンスが再起動する場合があるという問題を修正しました。
- ディクショナリの統計スレッドが原因で CPU 使用率が 100% になる問題を修正しました。
- `CHECK TABLE` ステートメントの実行中に Aurora Replica が再起動する問題を修正しました。

MySQL バグ修正の統合

- Bug #25361251: INCORRECT BEHAVIOR WITH INSERT ON DUPLICATE KEY IN SP
- Bug #26734162: INCORRECT BEHAVIOR WITH INSERT OF BLOB + ON DUPLICATE KEY UPDATE
- Bug #27460607: INCORRECT BEHAVIOR OF IODKU WHEN INSERT SELECT's SOURCE TABLE IS EMPTY
- `DISTINCT` または `GROUP BY` 句を使用したクエリでは、誤った結果が返される可能性があります。(MySQL 5.7 バグ #79591、バグ #22343910)
- `DELETE` 句の派生テーブルを使用する結合テーブルからの `WHERE` は、エラー 1093 (バグ #23074801) によって失敗します。

- GCOLS: INCORRECT BEHAVIOR WITH CHARSET CHANGES (Bug #25287633)。

Aurora MySQL バージョン 1 との比較

以下の Amazon Aurora MySQL 機能は Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- 関数を同期的に呼び出すためのネイティブ関数。 AWS Lambda 詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

Aurora MySQL 2.03.3 は、MySQL 5.7 とのワイヤー互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

Aurora MySQL 2.03.3 では、現在、MySQL 5.7 の次の機能はサポートされていません。

- グローバルトランザクション ID (GTID) Aurora MySQL では、バージョン 2.04 以降の GTID をサポートしています。
- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード

- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント
- X プロトコル

Aurora MySQL データベースエンジンの更新 2019-01-09 (バージョン 2.03.2) (廃止)

(バージョン 2.03.2)

Aurora MySQL 2.03.2 は一般利用可能です。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

新しい Aurora MySQL DB クラスターを作成する場合は、スナップショットから復元する場合も含めて、MySQL 5.7 または MySQL 5.6 との互換性を選択できます。

Aurora MySQL 1.* クラスターを Aurora MySQL 2.03.2 にインプレースアップグレードしたり、Amazon S3 バックアップから Aurora MySQL 2.03.2 に復元したりすることはできません。これらの制限は、今後の Aurora MySQL 2.* リリースで削除する予定です。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> このバージョンは現在、 AWS GovCloud (米国西部) [us-gov-west-1] リージョンと中国 (北京) [cn-north-1] リージョンではご利用いただけません。ご利用可能になりましたら、別途お知らせします。

> **Note**
>
> DB クラスターのアップグレード手順が変わりました。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

- Aurora バージョンセレクタ - Aurora MySQL 2.03.2 以降、 AWS Management Consoleで複数のバージョンの MySQL 5.7 互換 Aurora から選択できます。詳細については、「Amazon Aurora ユーザーガイド」の「AWSによる Aurora MySQL エンジンバージョンの確認または指定」を参照してください。

重要な修正:

- CVE-2016-3495

Aurora MySQL バージョン 1 との比較

以下の Amazon Aurora MySQL 機能は Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- 関数を同期的に呼び出すためのネイティブ関数。 AWS Lambda 詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

Aurora MySQL 2.03.2 は、MySQL 5.7 とのワイヤー互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

Aurora MySQL 2.03.2 では、現在、MySQL 5.7 の次の機能はサポートされていません。

- グローバルトランザクション ID (GTID) Aurora MySQL では、バージョン 2.04 以降の GTID をサポートしています。
- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント
- X プロトコル

Aurora MySQL データベースエンジンの更新 2018-10-24 (バージョン 2.03.1) (廃止)

バージョン: 2.03.1

Aurora MySQL 2.03.1 は一般利用可能です。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

新しい Aurora MySQL DB クラスターを作成する場合は、MySQL 5.7 または MySQL 5.6 との互換性を選択できます。MySQL 5.6 互換スナップショットを復元する場合は、MySQL 5.7 または MySQL 5.6 との互換性を選択できます。

Aurora MySQL 1.14.*、1.15.*、1.16.*、1.17.*、1.18.*、2.01.*、2.02.*、2.03 のスナップショットを Aurora MySQL 2.03.1 に復元することができます。

Aurora MySQL 1.* クラスターを Aurora MySQL 2.03.1 にインプレースアップグレードしたり、Amazon S3 バックアップから Aurora MySQL 2.03.1 に復元したりすることはできません。これらの制限は、今後の Aurora MySQL 2.* リリースで削除する予定です。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> このバージョンは現在、 AWS GovCloud (米国西部) [us-gov-west-1] リージョンと中国 (北京) [cn-north-1] リージョンではご利用いただけません。ご利用可能になりましたら、別途お知らせします。

改良点

- トランザクションデッドロック検出の実行中に Aurora ライターが再起動する問題を修正しました。

Aurora MySQL バージョン 1 との比較

以下の Amazon Aurora MySQL 機能は Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- 関数を同期的に呼び出すためのネイティブ関数。 AWS Lambda 詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。

- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

Aurora MySQL 2.03.1 は、MySQL 5.7 とのワイヤー互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

Aurora MySQL 2.03.1 では、現在、MySQL 5.7 の次の機能はサポートされていません。

- グローバルトランザクション ID (GTID) Aurora MySQL では、バージョン 2.04 以降の GTID をサポートしています。
- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント
- X プロトコル

Aurora MySQL データベースエンジンの更新 2018-10-11 (バージョン 2.03) (廃止)

バージョン: 2.03

Aurora MySQL 2.03 は一般利用可能です。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

新しい Aurora MySQL DB クラスターを作成する場合は、MySQL 5.7 または MySQL 5.6 との互換性を選択できます。MySQL 5.6 互換スナップショットを復元する場合は、MySQL 5.7 または MySQL 5.6 との互換性を選択できます。

Aurora MySQL 1.14.*、1.15.*、1.16.*、1.17.*、1.18.*、2.01.*、2.02.* のスナップショットを Aurora MySQL 2.03 に復元することができます。

Aurora MySQL 1.* クラスターを Aurora MySQL 2.03 にインプレースアップグレードしたり、Amazon S3 バックアップから Aurora MySQL 2.03 に復元したりすることはできません。これらの制限は、今後の Aurora MySQL 2.* リリースで削除する予定です。

> **Note**
>
> このバージョンは現在、 AWS GovCloud (米国西部) [us-gov-west-1] リージョンと中国 (北京) [cn-north-1] リージョンではご利用いただけません。ご利用可能になりましたら、別途お知らせします。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

- Performance Schemaが利用可能です。
- 「強制終了」状態のゾンビセッションがより多くの CPU を消費するような問題を修正しました。
- 読み取り専用トランザクションが Aurora ライター上のレコードのロックを取得しているときのデッドラッチの問題を修正しました。
- 顧客のワークロードがない Aurora レプリカで CPU 使用率が高くなるような問題を修正しました。
- Aurora レプリカまたは Aurora ライターの再起動を発生させるような問題に関する複数の修正。
- ディスクのスループット制限に達したときに診断ログをスキップする機能が追加されました。
- Aurora ライターでバイナリログが有効になっているときのメモリリークの問題を修正しました。

MySQL Community Edition バグ修正の統合

- パーティショニングされたテーブルの逆スキャンが、ICP を行う - ORDER BY DESC (バグ #24929748)。
- JSON_OBJECT は無効な JSON コードを作成する (バグ #26867509)。
- 大規模な JSON データを挿入すると、膨大な時間がかかる (バグ #22843444)。
- パーティショニングされたテーブルが 5.6 より 5.7 でより多くのメモリーを使用する (バグ #25080442)。

Aurora MySQL バージョン 1 との比較

以下の Amazon Aurora MySQL 機能は Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

Aurora MySQL 2.03 は、MySQL 5.7 とのワイヤー互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

Aurora MySQL 2.03 では、現在、MySQL 5.7 の次の機能はサポートされていません。

- グローバルトランザクション ID (GTID) Aurora MySQL では、バージョン 2.04 以降の GTID をサポートしています。
- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント
- X プロトコル

Aurora MySQL データベースエンジンの更新 2018-10-08 (バージョン 2.02.5) (廃止)

バージョン: 2.02.5

Aurora MySQL 2.02.5 は一般利用可能です。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

新しい Aurora MySQL DB クラスターを作成する場合は、MySQL 5.7 または MySQL 5.6 との互換性を選択できます。MySQL 5.6 互換スナップショットを復元する場合は、MySQL 5.7 または MySQL 5.6 との互換性を選択できます。

Aurora MySQL 1.14.*、1.15.*、1.16.*、1.17.*、1.18.*、2.01.*、2.02.* のスナップショットを Aurora MySQL 2.02.5 に復元することができます。Aurora MySQL 2.01.* または 2.02.* から Aurora MySQL 2.02.5 へのインプレースアップグレードを実行することもできます。

Aurora MySQL 1.* クラスターを Aurora MySQL 2.02.5 にインプレースアップグレードしたり、Amazon S3 バックアップから Aurora MySQL 2.02.5 に復元したりすることはできません。これらの制限は、今後の Aurora MySQL 2.* リリースで削除する予定です。

今回の Aurora MySQL 5.7 のリリースでは、Performance Schemaは無効になっています。Performance Schemaのサポートのために Aurora 2.03 にアップグレードします。

> **Note**
>
> このバージョンは現在、 AWS GovCloud (米国西部) [us-gov-west-1] リージョンと中国 (北京) [cn-north-1] リージョンではご利用いただけません。ご利用可能になりましたら、別途お知らせします。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

- テーブルの逆スキャンの実行時に Aurora レプリカが再起動するような問題を修正しました。

Aurora MySQL バージョン 1 との比較

以下の Amazon Aurora MySQL 機能は Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

Aurora MySQL 2.02.5 は、MySQL 5.7 とのワイヤー互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

Aurora MySQL 2.02.5 では、現在、MySQL 5.7 の次の機能はサポートされていません。

- グローバルトランザクション ID (GTID) Aurora MySQL では、バージョン 2.04 以降の GTID をサポートしています。
- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント
- X プロトコル

Aurora MySQL データベースエンジンの更新 2018-09-21 (バージョン 2.02.4) (廃止)

バージョン: 2.02.4

Aurora MySQL 2.02.4 は一般利用可能です。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

新しい Aurora MySQL DB クラスターを作成する場合は、MySQL 5.7 または MySQL 5.6 との互換性を選択できます。MySQL 5.6 互換スナップショットを復元する場合は、MySQL 5.7 または MySQL 5.6 との互換性を選択できます。

Aurora MySQL 1.14.*、1.15.*、1.16.*、1.17.*、1.18.*、2.01.*、2.02.* のスナップショットを Aurora MySQL 2.02.4 に復元することができます。Aurora MySQL 2.01.* または 2.02.* から Aurora MySQL 2.02.4 へのインプレースアップグレードを実行することもできます。

Aurora MySQL 1.* クラスターを Aurora MySQL 2.02.4 にインプレースアップグレードしたり、Amazon S3 バックアップから Aurora MySQL 2.02.4 に復元したりすることはできません。これらの制限は、今後の Aurora MySQL 2.* リリースで削除する予定です。

今回の Aurora MySQL 5.7 のリリースでは、Performance Schemaは無効になっています。Performance Schemaのサポートのために Aurora 2.03 にアップグレードします。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

- Aurora MySQL 5.6 スナップショットから復元されたテーブルの全文検索インデックスに関する安定性の問題を修正しました。

MySQL Community Edition バグ修正の統合

- 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

Aurora MySQL バージョン 1 との比較

以下の Amazon Aurora MySQL 機能は Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

MySQL 5.7 の互換性

Aurora MySQL 2.02.4 は、MySQL 5.7 とのワイヤー互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

Aurora MySQL 2.02.4 では、現在、MySQL 5.7 の次の機能はサポートされていません。

- グローバルトランザクション ID (GTID) Aurora MySQL では、バージョン 2.04 以降の GTID をサポートしています。
- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント
- X プロトコル

Aurora MySQL データベースエンジンの更新 2018-08-23 (バージョン 2.02.3) (廃止)

バージョン: 2.02.3

Aurora MySQL 2.02.3 は一般利用可能です。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

新しい Aurora MySQL DB クラスターを作成する場合は、MySQL 5.7 または MySQL 5.6 との互換性を選択できます。MySQL 5.6 互換スナップショットを復元する場合は、MySQL 5.7 または MySQL 5.6 との互換性を選択できます。

Aurora MySQL 1.14.*、1.15.*、1.16.*、1.17.*、2.01.*、2.02.* のスナップショットを Aurora MySQL 2.02.3 に復元することができます。Aurora MySQL 2.01.* または 2.02.* から Aurora MySQL 2.02.3 へのインプレースアップグレードを実行することもできます。

Aurora MySQL 1.* クラスターを Aurora MySQL 2.02.3 にインプレースアップグレードしたり、Amazon S3 バックアップから Aurora MySQL 2.02.3 に復元したりすることはできません。これらの制限は、今後の Aurora MySQL 2.* リリースで削除する予定です。

今回の Aurora MySQL 5.7 のリリースでは、Performance Schemaは無効になっています。Performance Schemaのサポートのために Aurora 2.03 にアップグレードします。

> **Note**
>
> このバージョンは現在、 AWS GovCloud (米国西部) [us-gov-west-1] リージョンと中国 (北京) [cn-north-1] リージョンではご利用いただけません。ご利用可能になりましたら、別途お知らせします。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

Aurora MySQL バージョン 1 との比較

以下の Amazon Aurora MySQL 機能は Aurora MySQL バージョン 1 (MySQL 5.6 互換) でサポートされていますが、Aurora MySQL バージョン 2 (MySQL 5.7 互換) では現在サポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

現在、Aurora MySQL 2.01 は、Aurora MySQL のバージョン 1.16 以降で追加された機能をサポートしていません。Aurora MySQL のバージョン 1.16 の詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。

MySQL 5.7 の互換性

Aurora MySQL 2.02.3 は、MySQL 5.7 とのワイヤー互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

Aurora MySQL 2.02.3 では、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グローバルトランザクション ID (GTID) Aurora MySQL では、バージョン 2.04 以降の GTID をサポートしています。
- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント
- X プロトコル

Aurora MySQL 2.x と Aurora MySQL 1.x の CLI の違い

- Aurora MySQL 2.x のエンジン名は `aurora-mysql` で、Aurora MySQL 1.x のエンジン名は引き続き `aurora` です。
- Aurora MySQL 2.x のデフォルトのパラメータグループは `default.aurora-mysql5.7` で、Aurora MySQL 1.x のデフォルトのパラメータグループは引き続き `default.aurora5.6` です。
- Aurora MySQL 2.x の DB クラスターパラメータグループファミリー名は `aurora-mysql5.7` で、Aurora MySQL 1.x の DB クラスターパラメータグループファミリー名は引き続き `aurora5.6` になります。

CLI コマンドのフルセットと Aurora MySQL 2.x および Aurora MySQL 1.x の相違点については、Aurora のドキュメントを参照してください。

改良点

- レコードの読み取り中にオプティミスティックカーソル復元を使用すると、Aurora レプリカが再起動することがある問題を修正しました。
- インデックス統計を改善するために、パラメータ `innodb_stats_persistent_sample_pages` のデフォルト値を 128 に更新しました。
- Aurora プライマリインスタンスで同時に変更されている小さなテーブルにアクセスすると、Aurora レプリカが再起動するような問題を修正しました。
- `ANALYZE TABLE` を修正して、テーブル定義キャッシュのフラッシュを停止しました。
- 地理空間のポイントクエリを検索範囲に変換するときに Aurora のプライマリインスタンスまたは Aurora レプリカが再起動するような問題を修正しました。

Aurora MySQL データベースエンジンの更新 2018-06-04 (バージョン 2.02.2) (廃止)

バージョン: 2.02.2

Aurora MySQL 2.02.2 は一般利用可能です。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

新しい Aurora MySQL DB クラスターを作成する場合は、MySQL 5.7 または MySQL 5.6 との互換性を選択できます。MySQL 5.6 互換スナップショットを復元する場合は、MySQL 5.7 または MySQL 5.6 との互換性を選択できます。

Aurora MySQL 1.14*、1.15*、1.16*、1.17*、2.01*、2.02 のスナップショットを Aurora MySQL 2.02.2 に復元することができます。Aurora MySQL 2.01* または 2.02 から Aurora MySQL 2.02.2 へのインプレースアップグレードを実行することもできます。

Aurora MySQL 1.* クラスターを Aurora MySQL 2.02.2 にインプレースアップグレードしたり、Amazon S3 バックアップから Aurora MySQL 2.02.2 に復元したりすることはできません。これらの制限は、今後の Aurora MySQL 2.* リリースで削除する予定です。

今回の Aurora MySQL 5.7 のリリースでは、Performance Schemaは無効になっています。Performance Schemaのサポートのために Aurora 2.03 にアップグレードします。

> **Note**
>
> このバージョンは現在、 AWS GovCloud (米国西部) [us-gov-west-1] リージョンと中国 (北京) [cn-north-1] リージョンではご利用いただけません。ご利用可能になりましたら、別途お知らせします。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

重要な修正:

- CVE-2016-3486

Aurora MySQL 5.6 との比較

次の Amazon Aurora MySQL 機能は Aurora MySQL 5.6 でサポートされていますが、これらの機能は現在 Aurora MySQL 5.7 ではサポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

現在、Aurora MySQL 2.01 は、Aurora MySQL のバージョン 1.16 以降で追加された機能をサポートしていません。Aurora MySQL のバージョン 1.16 の詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。

MySQL 5.7 の互換性

Aurora MySQL 2.02.2 は、MySQL 5.7 とのワイヤー互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

Aurora MySQL 2.02.2 では、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グローバルトランザクション ID (GTID) Aurora MySQL では、バージョン 2.04 以降の GTID をサポートしています。
- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント
- X プロトコル

Aurora MySQL 2.x と Aurora MySQL 1.x の CLI の違い

- Aurora MySQL 2.x のエンジン名は `aurora-mysql` で、Aurora MySQL 1.x のエンジン名は引き続き `aurora` です。
- Aurora MySQL 2.x のデフォルトのパラメータグループは `default.aurora-mysql5.7` で、Aurora MySQL 1.x のデフォルトのパラメータグループは引き続き `default.aurora5.6` です。

- Aurora MySQL 2.x の DB クラスターパラメータグループファミリー名は `aurora-mysql5.7` で、Aurora MySQL 1.x の DB クラスターパラメータグループファミリー名は引き続き `aurora5.6` になります。

CLI コマンドのフルセットと Aurora MySQL 2.x および Aurora MySQL 1.x の相違点については、Aurora のドキュメントを参照してください。

改良点

- Aurora ライターが Aurora レプリカの進行状況の追跡中に再起動することがある問題を修正しました。
- パーティショニングされたテーブルがインデックス作成の実行後にアクセスされた後、または Aurora ライターのテーブルにステートメントをドロップした後に、Aurora レプリカが再起動またはエラーをスローする問題を修正しました。
- Aurora ライターで ALTER table ADD/DROP column ステートメントを実行することで発生した変更を適応している間、Aurora レプリカのテーブルにアクセスできない問題を修正しました。

Aurora MySQL データベースエンジンの更新 2018-05-03 (バージョン 2.02) (廃止)

バージョン: 2.02

Aurora MySQL 2.02 は一般利用可能です。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

新しい Aurora MySQL DB クラスターを作成する場合は、MySQL 5.7 または MySQL 5.6 との互換性を選択できます。MySQL 5.6 互換スナップショットを復元する場合は、MySQL 5.7 または MySQL 5.6 との互換性を選択できます。

Aurora MySQL 1.14*、1.15*、1.16*、1.17*、2.01* のスナップショットを Aurora MySQL 2.02 に復元することができます。Aurora MySQL 2.01* から Aurora MySQL 2.02 へのインプレースアップグレードを実行することもできます。

Aurora MySQL 1.x クラスターを Aurora MySQL 2.02 にインプレースアップグレードすることや、Amazon S3 バックアップから Aurora MySQL 2.02 に復元することはできません。これらの制限は、今後の Aurora MySQL 2.x リリースで削除する予定です。

今回の Aurora MySQL 5.7 のリリースでは、Performance Schemaは無効になっています。Performance Schemaのサポートのために Aurora 2.03 にアップグレードします。

質問や懸念がある場合は、 AWS コミュニティフォーラムやSupport AWS を通じてSupport を受けることができます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

Aurora MySQL 5.6 との比較

次の Amazon Aurora MySQL 機能は Aurora MySQL 5.6 でサポートされていますが、これらの機能は現在 Aurora MySQL 5.7 ではサポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- AWS Lambda 関数を同期的に呼び出すためのネイティブ関数。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

現在、Aurora MySQL 2.01 は、Aurora MySQL のバージョン 1.16 以降で追加された機能をサポートしていません。Aurora MySQL のバージョン 1.16 の詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。

MySQL 5.7 の互換性

Aurora MySQL 2.02 は、MySQL 5.7 とのワイヤー互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

Aurora MySQL 2.02 では、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グローバルトランザクション ID (GTID) Aurora MySQL では、バージョン 2.04 以降の GTID をサポートしています。
- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント
- X プロトコル

Aurora MySQL 2.x と Aurora MySQL 1.x の CLI の違い

- Aurora MySQL 2.x のエンジン名は `aurora-mysql` で、Aurora MySQL 1.x のエンジン名は引き続き `aurora` です。
- Aurora MySQL 2.x のデフォルトのパラメータグループは `default.aurora-mysql5.7` で、Aurora MySQL 1.x のデフォルトのパラメータグループは引き続き `default.aurora5.6` です。
- Aurora MySQL 2.x の DB クラスターパラメータグループファミリー名は `aurora-mysql5.7` で、Aurora MySQL 1.x の DB クラスターパラメータグループファミリー名は引き続き `aurora5.6` になります。

CLI コマンドのフルセットと Aurora MySQL 2.x および Aurora MySQL 1.x の相違点については、Aurora のドキュメントを参照してください。

改良点

- INSERT ステートメントを実行して高速挿入最適化を利用すると Aurora ライターが再起動する問題を修正しました。

- Aurora レプリカで ALTER DATABASE ステートメントを実行すると Aurora レプリカが再起動する問題を修正しました。
- Aurora ライターにドロップされた直後のテーブルでクエリを実行すると Aurora レプリカが再起動する問題を修正しました。
- Aurora レプリカで `innodb_adaptive_hash_index` を `OFF` に設定すると Aurora レプリカが再起動する問題を修正しました。
- Aurora ライターで TRUNCATE TABLE クエリを実行すると Aurora レプリカが再起動する問題を修正しました。
- INSERT ステートメントを実行すると、特定の状況下で Aurora ライターがフリーズする問題を修正しました。マルチノードクラスターでは、これによりフェイルオーバーとなることがあります。
- セッション可変の設定に関係するメモリリークを修正しました。
- 生成した列があるテーブルの undo のクリアに関係した特定の状況下で Aurora ライターがフリーズする問題を修正しました。
- バイナリログ記録を有効にすると時折 Aurora ライターが再起動することがある問題を修正しました。

MySQL バグ修正の統合

- 左結合は外部側で不正な結果を返します (バグ #22833364)。

Aurora MySQL データベースエンジンの更新 2018-03-13 (バージョン 2.01.1) (廃止)

バージョン: 2.01.1

Aurora MySQL 2.01.1 は一般利用可能です。Aurora MySQL 2.x バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.x バージョンは MySQL 5.6 と互換性があります。

新しい Aurora MySQL DB クラスターを作成する場合は、MySQL 5.7 または MySQL 5.6 との互換性を選択できます。MySQL 5.6 互換スナップショットを復元する場合は、MySQL 5.7 または MySQL 5.6 との互換性を選択できます。

Aurora MySQL 1.14*、1.15*、1.16* および 1.17* のスナップショットを Aurora MySQL 2.01.1 に復元することができます。

Aurora MySQL 1.x クラスターを Aurora MySQL 2.01.1 にインプレースアップグレードすることや、Amazon S3 バックアップから Aurora MySQL 2.01.1 に復元することはできません。これらの制限は、今後の Aurora MySQL 2.x リリースで削除する予定です。

今回の Aurora MySQL 5.7 のリリースでは、Performance Schemaは無効になっています。Performance Schemaのサポートのために Aurora 2.03 にアップグレードします。

Aurora MySQL 5.6 との比較

次の Amazon Aurora MySQL 機能は Aurora MySQL 5.6 でサポートされていますが、これらの機能は現在 Aurora MySQL 5.7 ではサポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- 関数を同期的に呼び出すためのネイティブ関数。 AWS Lambda 詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。
- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

現在、Aurora MySQL 2.01.1 は、Aurora MySQL のバージョン 1.16 以降で追加された機能をサポートしていません。Aurora MySQL のバージョン 1.16 の詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。

MySQL 5.7 の互換性

Aurora MySQL 2.01.1 は、MySQL 5.7 とのワイヤー互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

Aurora MySQL 2.01.1 では、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グローバルトランザクション ID (GTID) Aurora MySQL では、バージョン 2.04 以降の GTID をサポートしています。
- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント
- X プロトコル

Aurora MySQL 2.x と Aurora MySQL 1.x の CLI の違い

- Aurora MySQL 2.x のエンジン名は `aurora-mysql` で、Aurora MySQL 1.x のエンジン名は引き続き `aurora` です。
- Aurora MySQL 2.x のデフォルトのパラメータグループは `default.aurora-mysql5.7` で、Aurora MySQL 1.x のデフォルトのパラメータグループは引き続き `default.aurora5.6` です。
- Aurora MySQL 2.x の DB クラスターパラメータグループファミリー名は `aurora-mysql5.7` で、Aurora MySQL 1.x の DB クラスターパラメータグループファミリー名は引き続き `aurora5.6` になります。

CLI コマンドのフルセットと Aurora MySQL 2.x および Aurora MySQL 1.x の相違点については、Aurora のドキュメントを参照してください。

改良点

- MySQL 5.6 互換のスナップショットが MySQL 5.7 との互換性で復元されると Aurora 固有のデータベース権限が不正に作成される、スナップショットの復元に関する問題を修正しました。

- 1.17 のスナップショットの復元に対するサポートを追加しました。

Aurora MySQL データベースエンジンの更新 2018-02-06 (バージョン 2.01) (廃止)

バージョン: 2.01

Aurora MySQL 2.01 は一般利用可能です。今後、Aurora MySQL 2.x バージョンは MySQL 5.7 と、Aurora MySQL 1.x バージョンは MySQL 5.6 との互換性を備えます。

新しい Aurora MySQL DB クラスターを作成する場合は、スナップショットから復元する場合も含めて、MySQL 5.7 または MySQL 5.6 との互換性を選択できます。

Aurora MySQL 1.14*、1.15*、および 1.16* のスナップショットを Aurora MySQL 2.01 に復元することができます。

Aurora MySQL 1.x クラスターを Aurora MySQL 2.01 にインプレースアップグレードすることや、Amazon S3 バックアップから Aurora MySQL 2.01 に復元することはできません。これらの制限は、今後の Aurora MySQL 2.x リリースで削除する予定です。

今回の Aurora MySQL 5.7 のリリースでは、Performance Schemaは無効になっています。Performance Schemaのサポートのために Aurora 2.03 にアップグレードします。

Aurora MySQL 5.6 との比較

次の Amazon Aurora MySQL 機能は Aurora MySQL 5.6 でサポートされていますが、これらの機能は現在 Aurora MySQL 5.7 ではサポートされていません。

- Asynchronous Key Prefetch (AKP)。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。
- ハッシュ結合。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- 関数を同期的に呼び出すためのネイティブ関数。 AWS Lambda 詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL ネイティブ関数を使用した Lambda 関数の呼び出し」を参照してください。
- スキャンバッチ処理。詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。

- Amazon S3 バケットを使用した MySQL からのデータ移行。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットを使用した MySQL からのデータ移行」を参照してください。

現在、Aurora MySQL 2.01 は、Aurora MySQL のバージョン 1.16 以降で追加された機能をサポートしていません。Aurora MySQL のバージョン 1.16 の詳細については、「Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)」を参照してください。

MySQL 5.7 の互換性

Aurora MySQL 2.01 は、MySQL 5.7 とのワイヤー互換性があり、JSON のサポート、空間インデックス、列生成などの機能が含まれています。Aurora MySQL は、z オーダーカーブを使用した空間インデックス作成のネイティブ実装を使用して、空間データセットにおいて、MySQL 5.7 と比較して 20 倍以上の書き込みパフォーマンスと 10 倍以上の読み取りパフォーマンスを実現します。

Aurora MySQL 2.01 では、現在、MySQL 5.7 の以下の機能はサポートされていません。

- グローバルトランザクション ID (GTID) Aurora MySQL では、バージョン 2.04 以降の GTID をサポートしています。
- グループのレプリケーションプラグイン
- ページサイズの増加
- 起動時の InnoDB バッファープールのロード
- InnoDB フルテキストパーサープラグイン
- マルチソースレプリケーション
- オンラインバッファープールのサイズ変更
- パスワード検証プラグイン
- クエリ書き換えプラグイン
- レプリケーションフィルタリング
- `CREATE TABLESPACE` SQL ステートメント
- X プロトコル

Aurora MySQL 2.x と Aurora MySQL 1.x の CLI の違い

- Aurora MySQL 2.x のエンジン名は `aurora-mysql` で、Aurora MySQL 1.x のエンジン名は引き続き `aurora` です。

- Aurora MySQL 2.x のデフォルトのパラメータグループは `default.aurora-mysql5.7` で、Aurora MySQL 1.x のデフォルトのパラメータグループは引き続き `default.aurora5.6` です。
- Aurora MySQL 2.x の DB クラスターパラメータグループファミリー名は `aurora-mysql5.7` で、Aurora MySQL 1.x の DB クラスターパラメータグループファミリー名は引き続き `aurora5.6` になります。

CLI コマンドのフルセットと Aurora MySQL 2.x および Aurora MySQL 1.x の相違点については、Aurora のドキュメントを参照してください。

Amazon Aurora MySQL バージョン 1 のデータベースエンジンの更新 (廃止)

Amazon Aurora バージョン 1 データベースエンジンの更新は次のとおりです。

- Aurora MySQL データベースエンジンの更新 2021-09-30 (バージョン 1.23.4) (廃止)
- Aurora MySQL データベースエンジンの更新 2021-06-28 (バージョン 1.23.3) (廃止)
- Aurora MySQL データベースエンジンの更新 2021-03-18 (バージョン 1.23.2) (廃止)
- Aurora MySQL データベースエンジンの更新 2020-11-24 (バージョン 1.23.1) (廃止)
- Aurora MySQL データベースエンジンの更新 2020-09-02 (バージョン 1.23.0) (廃止)
- Aurora MySQL データベースエンジンの更新 2021-06-03 (バージョン 1.22.5) (廃止)
- Aurora MySQL データベースエンジンの更新 2021-03-04 (バージョン 1.22.4) (廃止)
- Aurora MySQL データベースエンジンの更新 2020-11-09 (バージョン 1.22.3) (廃止)
- Aurora MySQL データベースエンジンの更新 2020-03-05 (バージョン 1.22.2) (廃止)
- Aurora MySQL データベースエンジンの更新 2019-12-23 (バージョン 1.22.1) (廃止)
- Aurora MySQL データベースエンジンの更新 2019-11-25 (バージョン 1.22.0) (廃止)
- Aurora MySQL データベースエンジンの更新 2019-11-25 (バージョン 1.21.0) (廃止)
- Aurora MySQL データベースエンジンの更新 2020-03-05 (バージョン 1.20.1) (廃止)
- Aurora MySQL データベースエンジンの更新 2019-11-11 (バージョン 1.20.0) (廃止)
- Aurora MySQL データベースエンジンの更新 2020-03-05 (バージョン 1.19.6) (廃止)
- Aurora MySQL データベースエンジンの更新 2019-09-19 (バージョン 1.19.5) (廃止)
- Aurora MySQL データベースエンジンの更新 2019-06-05 (バージョン 1.19.2) (廃止)
- Aurora MySQL データベースエンジンの更新 2019-05-09 (バージョン 1.19.1) (廃止)
- Aurora MySQL データベースエンジンの更新 2019-02-07 (バージョン 1.19.0) (廃止)
- Aurora MySQL データベースエンジンの更新 2018-09-20 (バージョン 1.18.0) (廃止)
- Aurora MySQL データベースエンジンの更新 2020-03-05 (バージョン 1.17.9) (廃止)
- Aurora MySQL データベースエンジンの更新 2019-01-17 (バージョン 1.17.8) (廃止)
- Aurora MySQL データベースエンジンの更新 2018-10-08 (バージョン 1.17.7) (廃止)
- Aurora MySQL データベースエンジンの更新 2018-09-06 (バージョン 1.17.6) (廃止)

- Aurora MySQL データベースエンジンの更新 2018-08-14 (バージョン 1.17.5) (廃止)
- Aurora MySQL データベースエンジンの更新 2018-08-07 (バージョン 1.17.4) (廃止)
- Aurora MySQL データベースエンジンの更新 2018-06-05 (バージョン 1.17.3) (廃止)
- Aurora MySQL データベースエンジンの更新 2018-04-27 (バージョン 1.17.2) (廃止)
- Aurora MySQL データベースエンジンの更新 2018-03-23 (バージョン 1.17.1) (廃止)
- Aurora MySQL データベースエンジンの更新 2018-03-13 (バージョン 1.17) (廃止)
- Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)
- Aurora MySQL データベースエンジンの更新 2017-11-20 (バージョン 1.15.1) (廃止)
- Aurora MySQL データベースエンジンの更新 2017-10-24 (バージョン 1.15) (廃止)
- Aurora MySQL データベースエンジンの更新: 2018-03-13 (バージョン 1.14.4) (廃止)
- Aurora MySQL データベースエンジンの更新: 2017-09-22 (バージョン 1.14.1) (廃止)
- Aurora MySQL データベースエンジンの更新: 2017-08-07 (バージョン 1.14) (廃止)
- Aurora MySQL データベースエンジンの更新: 2017-05-15 (バージョン 1.13) (廃止)
- Aurora MySQL データベースエンジンの更新: 2017-04-05 (バージョン 1.12) (廃止)
- Aurora MySQL データベースエンジンの更新: 2017-02-23 (バージョン 1.11) (廃止)
- Aurora MySQL データベースエンジンの更新: 2017-01-12 (バージョン 1.10.1) (廃止)
- Aurora MySQL データベースエンジンの更新: 2016-12-14 (バージョン 1.10) (廃止)
- Aurora MySQL データベースエンジンの更新: 2016-11-10 (バージョン 1.9.0、1.9.1) (廃止)
- Aurora MySQL データベースエンジンの更新: 2016-10-26 (バージョン 1.8.1) (廃止)
- Aurora MySQL データベースエンジンの更新: 2016-10-18 (バージョン 1.8) (廃止)
- Aurora MySQL データベースエンジンの更新: 2016-09-20 (バージョン 1.7.1) (廃止)
- Aurora MySQL データベースエンジンの更新: 2016-08-30 (バージョン 1.7.0) (廃止)
- Aurora MySQL データベースエンジンの更新: 2016-06-01 (バージョン 1.6.5) (廃止)
- Aurora MySQL データベースエンジンの更新: 2016-04-06 (バージョン 1.6) (廃止)
- Aurora MySQL データベースエンジンの更新: 2016-01-11 (バージョン 1.5) (廃止)
- Aurora MySQL データベースエンジンの更新: 2015-12-03 (バージョン 1.4) (廃止)
- Aurora MySQL データベースエンジンの更新: 2015-10-16 (バージョン 1.2、1.3) (廃止)
- Aurora MySQL データベースエンジンの更新: 2015-08-24 (バージョン 1.1) (廃止)

Aurora MySQL データベースエンジンの更新 2021-09-30 (バージョン 1.23.4) (廃止)

バージョン: 1.23.4

Aurora MySQL 1.23.4 は一般公開されています。Aurora MySQL 2.* バージョンは MySQL 5.7 と互換性があり、Aurora MySQL 1.* バージョンは MySQL 5.6 と互換性があります。

このエンジンバージョンは 2023 年 2 月 28 日に非推奨となる予定です。詳細については、「Amazon Aurora MySQL 互換エディションバージョン 1 のサポート終了に向けて準備する」を参照してください。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

古いバージョンの Aurora MySQL を使用してクラスターを作成するには、RDS コンソール、AWS CLI、または Amazon RDS API を使用してエンジンバージョンを指定してください。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

全般的な機能強化:

- 内部診断ログファイル内の過剰な情報メッセージのログが原因で、リーダーインスタンスの CPU 使用率が高くなる問題を修正しました。

優先度の高い修正:

- CVE-2021-2307
- CVE-2021-2226
- CVE-2021-2160
- CVE-2021-2154
- CVE-2021-2060
- CVE-2021-2032
- CVE-2021-2001

Aurora MySQL データベースエンジンの更新 2021-06-28 (バージョン 1.23.3) (廃止)

バージョン 1.23.3

Aurora MySQL 1.23.3 は一般公開されています。Aurora MySQL 1.* バージョンは MySQL 5.6 と互換性があり、Aurora MySQL 2.* バージョンは MySQL 5.7 と互換性があります。

このエンジンバージョンは 2023 年 2 月 28 日に非推奨となる予定です。詳細については、「Amazon Aurora MySQL 互換エディションバージョン 1 のサポート終了に向けて準備する」を参照してください。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

古いバージョンの Aurora MySQL を使用してクラスターを作成するには、RDS コンソール、AWS CLI、または Amazon RDS API を使用してエンジンバージョンを指定してください。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

安定性と可用性全般に関する機能強化

セキュリティの修正内容:

- CVE-2021-23841
- CVE-2021-3449
- CVE-2020-28196

Aurora MySQL データベースエンジンの更新 2021-03-18 (バージョン 1.23.2) (廃止)

バージョン: 1.23.2

Aurora MySQL 1.23.2 は一般公開されています。Aurora MySQL 1.* バージョンは MySQL 5.6 と互換性があり、Aurora MySQL 2.* バージョンは MySQL 5.7 と互換性があります。

このエンジンバージョンは 2023 年 2 月 28 日に非推奨となる予定です。詳細については、「Amazon Aurora MySQL 互換エディションバージョン 1 のサポート終了に向けて準備する」を参照してください。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

古いバージョンの Aurora MySQL を使用してクラスターを作成するには、RDS コンソール、AWS CLI、または Amazon RDS API を使用してエンジンバージョンを指定してください。

> **Note**
>
> このバージョンは現在、次のリージョンでは使用できません。AWS GovCloud (米国東部) [us-gov-east-1]、AWS GovCloud (米国西部) [us-gov-west-1] ご利用可能になりましたら、別途お知らせします。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

優先度の高い修正:

- CVE-2020-14867
- CVE-2020-14812
- CVE-2020-14769
- CVE-2020-14765
- CVE-2020-14793
- CVE-2020-14672
- CVE-2020-1971
- CVE-2018-3143

可用性の向上:

- 動的なクラスターストレージのサイズ変更機能で、リーダー DB インスタンスが再起動する可能性がある問題を修正しました。

- RESET QUERY CACHE ステートメントでの競合状態によるフェイルオーバーの問題を修正しました。
- クエリキャッシュを使用したネストされたストアドプロシージャ呼び出しのクラッシュを修正しました。
- パーティション化されたテーブルまたはサブパーティション化されたテーブルの不完全な切り捨てから回復するときに、mysqld の再起動が繰り返されないように問題を修正しました。
- オンプレミスまたは RDS for MySQL から Aurora MySQL への移行が失敗する可能性がある問題を修正しました。
- ストレージボリュームのスケーリング中にデータベースを再起動できる稀な競合状態を修正しました。
- 競合状態により、2 つのトランザクションでロックが共有され、データベースが再起動することがあるロックマネージャーの問題を修正しました。
- 長時間実行される書き込みトランザクションによりデータベースが再起動される、トランザクションロックメモリ管理に関する問題を修正しました。
- トランザクションのロールバック中にデータベースの再起動またはフェイルオーバーが発生する、ロックマネージャーの競合状態を修正しました。
- 5.6 のラボモードでテーブルの Fast Online DDL が有効になっている場合に、5.6 から 5.7 にアップグレードする際の問題を修正しました。
- パッチ適用のためのデータベースアクティビティの休止ポイントのチェックにおいて、ダウンタイムのないパッチ適用中にエンジンが再起動することがある複数の問題を修正しました。
- DROP TRIGGER、ALTER TABLE などの DDL 操作、特にテーブル内のパーティションのタイプや数を変更する ALTER TABLE が中断された場合に、再起動が繰り返されることに関連する複数の問題を修正しました。
- 16XL および 24XL インスタンスの table_open_cache のデフォルト値を更新しました。これにより、ラージインスタンスクラス (R4/R5-16XL、R5-12XL、R5-24XL) で再起動が繰り返され、CPU 使用率が高くなるのを防ぎます。この影響があるのは、1.21.x と 1.22.x のリリースです。
- バイナリログレプリカが HA_ERR_KEY_NOT_FOUND エラーで停止する問題を修正しました。

MySQL Community Edition バグ修正の統合

- レプリケーション: SHOW BINLOG EVENTS 文の実行中に、パラレルトランザクションがブロックされました。この修正により、SHOW BINLOG EVENTS プロセスはファイルの終了位置を計算す

る間だけロックを取得するため、パラレルトランザクションが長期間ブロックされることはありません。(バグ #76618、バグ #20928790)

Aurora MySQL データベースエンジンの更新 2020-11-24 (バージョン 1.23.1) (廃止)

バージョン: 1.23.1

Aurora MySQL 1.23.1 は一般公開されています。Aurora MySQL 1.* バージョンは MySQL 5.6 と互換性があり、Aurora MySQL 2.* バージョンは MySQL 5.7 と互換性があります。

このエンジンバージョンは 2023 年 2 月 28 日に非推奨となる予定です。詳細については、「Amazon Aurora MySQL 互換エディションバージョン 1 のサポート終了に向けて準備する」を参照してください。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

古いバージョンの Aurora MySQL を使用してクラスターを作成するには、RDS コンソール、AWS CLI、または Amazon RDS API を使用してエンジンバージョンを指定してください。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

セキュリティの修正内容:

マネージド型の環境での処理を微調整するための修正およびその他の機能強化。以下の CVE の追加の修正:

- CVE-2020-14559
- CVE-2020-14539

互換性のない変更:

このバージョンでは、`mysqldump` コマンドの動作に影響するアクセス許可の変更が導入されています。ユーザーは、`PROCESS` テーブルにアクセスする `INFORMATION_SCHEMA.FILES` 特権を

有している必要があります。変更せずに mysqldump コマンドを実行するには、PROCESS コマンドが接続するデータベースユーザーに mysqldump 特権を付与します。mysqldump オプションを指定して --no-tablespaces コマンドを実行することもできます。このオプションを使用すると、mysqldump 出力に CREATE LOGFILE GROUP または CREATE TABLESPACE ステートメントは含まれません。この場合、mysqldump コマンドは INFORMATION_SCHEMA.FILES テーブルにアクセスしないため、PROCESS アクセス許可を付与する必要はありません。

可用性の向上:

- 1.23.0 を実行しているグローバルデータベースのセカンダリクラスターの Aurora リーダーインスタンスが繰り返し再起動する問題を修正しました。
- 古いリリースバージョンのプライマリリージョンのライターをリリース 1.23.0 にアップグレードすると、グローバルデータベースのセカンダリリージョンのレプリカが再起動する可能性がある問題を修正しました。
- Aurora MySQL 1.23.0 で導入された動的サイズ変更機能のメモリリークを修正しました。
- パラレルクエリ機能を使用すると、クエリの実行中にサーバーが再起動する可能性がある問題を修正しました。
- ネットワークへの読み取りまたは書き込み中にデータベースエンジンでエラーが発生すると、クライアントセッションがハングする可能性がある問題を修正しました。

Aurora MySQL データベースエンジンの更新 2020-09-02 (バージョン 1.23.0) (廃止)

バージョン: 1.23.0

Aurora MySQL 1.23.0 は一般公開されています。Aurora MySQL 1.* バージョンは MySQL 5.6 と互換性があり、Aurora MySQL 2.* バージョンは MySQL 5.7 と互換性があります。

このエンジンバージョンは 2023 年 2 月 28 日に非推奨となる予定です。詳細については、「Amazon Aurora MySQL 互換エディションバージョン 1 のサポート終了に向けて準備する」を参照してください。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

Aurora MySQL 1.* データベースのスナップショットを Aurora MySQL 1.23.0 に復元できます。

> **Important**
>
> このバージョンでの Aurora ストレージの改良により、利用可能なアップグレードパスが Aurora MySQL 1.23 から Aurora MySQL 2.* に制限されています。Aurora MySQL 1.23 クラスターを 2.* にアップグレードする場合は、Aurora MySQL 2.09.0 以降にアップグレードする必要があります。

古いバージョンの Aurora MySQL を使用してクラスターを作成するには、RDS コンソール、AWS CLI、または Amazon RDS API を使用してエンジンバージョンを指定してください。

> **Note**
>
> このバージョンは現在、次のリージョンでは使用できません。AWS GovCloud (米国東部) [us-gov-east-1]、AWS GovCloud (米国西部) [us-gov-west-1] ご利用可能になりましたら、別途お知らせします。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

新機能:

- DB クラスターパラメータの値を変更することで、既存のクラスターのパラレルクエリのオンとオフを切り替えることができるようになりました `aurora_parallel_query`。クラスターを作成するときに、`parallelquery` パラメータの `--engine-mode` 設定を使用する必要はありません。

  パラレルクエリが拡張され、Aurora MySQL が利用できるすべてのリージョンで利用できるようになりました。

  Aurora クラスタ内のパラレルクエリをアップグレードおよび有効化するための手順には、他にも多くの機能拡張や変更があります。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL のパラレルクエリの使用」を参照してください。
- このリリースでは、最大 128 テビバイト (TiB) のストレージを備えた Amazon Aurora MySQL データベースインスタンスを作成できます。新しいストレージ制限は、以前の 64 TiB から引き上げられています。128 TiB のストレージサイズは、より大きなデータベースに対応します。この

機能は、スモールインスタンスサイズ (db.t2 または db.t3) ではサポートされていません。1 つのテーブルスペースは、16 KB のページサイズという InnoDB の制限があるため、64 TiB を超えて拡張することはできません。

Aurora は、クラスターボリュームサイズが 128 TiB に近い場合にアラートを表示し、サイズ制限に達する前にアクションを実行できるようにします。このアラートは、AWS Management Console の mysql ログと RDS イベントに表示されます。

- バイナリログ (binlog) 処理が改善され、非常に大きなトランザクションが関与している場合に、クラッシュリカバリ時間とコミット時間のレイテンシーが短縮されました。
- Aurora は、クラスターのストレージ領域のサイズを動的に変更します。動的サイズ変更では、Aurora DB クラスターからデータを削除すると、DB クラスターのストレージ領域が自動的に減少します。詳細については、「Amazon Aurora ユーザーガイド」の「ストレージのスケーリング」を参照してください。

> **Note**
>
> 動的サイズ変更機能は、Aurora が利用可能な AWS リージョンで段階的にデプロイされています。クラスターを使用するリージョンによっては、この機能はまだ利用できない場合があります。詳細については、新しい発表を参照してください。

優先度の高い修正:

- CVE-2019-2911
- CVE-2019-2537
- CVE-2018-2787
- CVE-2018-2784
- CVE-2018-2645
- CVE-2018-2640

可用性の向上:

- 競合状態により、2 つのトランザクションでロックが共有され、データベースが再起動することがあるロックマネージャーの問題を修正しました。
- 長時間実行される書き込みトランザクションによりデータベースが再起動される、トランザクションロックメモリ管理に関する問題を修正しました。

- トランザクションのロールバック中にデータベースの再起動またはフェイルオーバーが発生する、ロックマネージャーの競合状態を修正しました。
- Fast DDL が有効になっているテーブルで innodb_file_format が変更された場合の、5.6 から 5.7 へのアップグレード時の問題を修正しました。
- パッチ適用のためのデータベースアクティビティの休止ポイントのチェックにおいて、ダウンタイムのないパッチ適用中にエンジンが再起動することがある複数の問題を修正しました。
- 中断された DROP TRIGGER オペレーションの復旧中に DB インスタンスの再起動に影響する、DDL の復旧に関連する問題を修正しました。
- 特定のパーティショニングオペレーションの実行中にクラッシュが発生すると、データベースが使用できなくなるバグを修正しました。具体的には、パーティション化のタイプまたはテーブル内のパーティションの数を変更する ALTER TABLE オペレーションの中断です。
- 16XL および 24XL インスタンスの table_open_cache のデフォルト値を修正しました。このデフォルト値では、ラージインスタンスクラス (R4/R5-16XL、R5-12XL、R5-24XL) でフェイルオーバーが繰り返され、CPU 使用率が高くなる場合があります。この影響があるのは、1.21.x と 1.22.x です。

グローバルデータベース:

- Aurora Global Database のプライマリおよびセカンダリの AWS リージョンの MySQL INFORMATION_SCHEMA.REPLICA_HOST_STATUS ビューに、不足しているデータを入力します。
- プライマリリージョンとセカンダリリージョン間のテンポラリネットワーク接続の問題発生後、プライマリリージョンの UNDO レコードのガベージコレクションが原因で、グローバル DB セカンダリリージョンで発生する可能性がある予期しないクエリエラーを修正しました。

パラレルクエリ:

- パラレルクエリが原因で、実行時間が長いクエリで空の結果が返されることがある問題を修正しました。
- Aurora リードレプリカの小さなテーブルのクエリに 1 秒以上かかることがある問題を修正しました。
- パラレルクエリと DML ステートメントがワークロード負荷の高い状態で同時に実行されている場合に、再起動が発生する可能性がある問題を修正しました。

全般的な機能強化:

- 既存の大きな空間値を持つテーブルで空間インデックスが作成された場合、空間インデックスを使用するクエリで部分的な結果が返されることがある問題を修正しました。
- 監査システム可変 `server_audit_incl_users` および `server_audit_excl_users` の最大許容長を、1024 バイトから 2000 バイトに引き上げました。
- Aurora MySQL バイナリログプライマリが `statement` `binlog_format` で S3 からデータをロードしたときに、Aurora MySQL バイナリログプライマリに接続されたバイナリログレプリカが不完全なデータを表示することがある問題を修正しました。
- データをロードするための `mixed` ではなく、`row` binlog_format を `statement` にマッピングするためのコミュニティの動作に準拠します。
- ユーザーが接続を閉じ、セッションがテンポラリテーブルを使用しているときに、バイナリログのレプリケーションが機能しなくなる問題を修正しました。
- MyISAM テンポラリテーブルを含むクエリの応答時間が改善されました。
- バイナリログワーカーがネイティブ Lambda 関数を実行する際のアクセス許可の問題を修正しました。
- 低速ログまたは一般ログのクエリまたはローテーションを試みたときの Aurora リードレプリカに関する問題を修正しました。
- マスターとレプリカで `binlog_checksum` パラメータに異なる値が設定されている場合に、論理的なレプリケーションが停止される問題を修正しました。
- リードレプリカが、ライターで最近コミットされたトランザクションの結果の一部を一時的に表示することがある問題を修正しました。
- デッドロックが解決されたときに、ロールバックされたトランザクションのトランザクション情報を `show engine innodb status` に含めます。

MySQL Community Edition バグ修正の統合

- `ALTER TABLE ADD COLUMN ALGORITHM=QUICK` のバイナリログイベントは、コミュニティ版と互換性を持つように `ALGORITHM=DEFAULT` として書き換えられます。
- バグ #22350047: IF CLIENT KILLED AFTER ROLLBACK TO SAVEPOINT PREVIOUS STMTS COMMITTED
- バグ #29915479: RUNNING COM_REGISTER_SLAVE WITHOUT COM_BINLOG_DUMP CAN RESULTS IN SERVER EXIT

- バグ #30441969: BUG #29723340: MYSQL SERVER CRASH AFTER SQL QUERY WITH DATA ? AST
- バグ #30628268: OUT OF MEMORY CRASH
- バグ #27081349: UNEXPECTED BEHAVIOUR WHEN DELETE WITH SPATIAL FUNCTION
- バグ #27230859: UNEXPECTED BEHAVIOUR WHILE HANDLING INVALID POLYGON"
- バグ #27081349: UNEXPECTED BEHAVIOUR WHEN DELETE WITH SPATIAL"
- バグ #26935001: ALTER TABLE AUTO_INCREMENT TRIES TO READ INDEX FROM DISCARDED TABLESPACE
- バグ #29770705: SERVER CRASHED WHILE EXECUTING SELECT WITH SPECIFIC WHERE CLAUSE
- バグ #27659490: SELECT USING DYNAMIC RANGE AND INDEX MERGE USE TOO MUCH MEMORY(OOM)
- バグ #24786290: REPLICATION BREAKS AFTER BUG #74145 HAPPENS IN MASTER
- バグ #27703912: EXCESSIVE MEMORY USAGE WITH MANY PREPARE
- バグ #20527363: TRUNCATE TEMPORARY TABLE CRASH: !DICT_TF2_FLAG_IS_SET(TABLE, DICT_TF2_TEMPORARY)
- バグ #23103937: PS_TRUNCATE_ALL_TABLES() が SUPER_READ_ONLY モードで機能しない
- バグ #25053286: USE VIEW WITH CONDITION IN PROCEDURE CAUSES INCORRECT BEHAVIOR (fixed in 5.6.36)
- バグ #25586773: INCORRECT BEHAVIOR FOR CREATE TABLE SELECT IN A LOOP IN SP (fixed in 5.6.39)
- バグ #27407480: AUTOMATIC_SP_PRIVILEGES REQUIRES NEED THE INSERT PRIVILEGES FOR MYSQL.USER TABLE
- バグ #26997096: `relay_log_space` 値は同期された方法で更新されないため、その値はリレーログで使用される実際のディスク領域よりもはるかに大きくなることがあります。
- バグ#15831300 SLAVE_TYPE_CONVERSIONS=ALL_NON_LOSSY NOT WORKING AS EXPECTED
- SSL バグのバックポート: バグ #17087862、バグ #20551271
- バグ #16894092: PERFORMANCE REGRESSION IN 5.6.6+ FOR INSERT INTO ..。SELECT ..。FROM (5.6.15 で修正)。
- `SLAVE_TYPE_CONVERSIONS` に関連するバグ修正を移植します。

Aurora MySQL データベースエンジンの更新 2021-06-03 (バージョン 1.22.5) (廃止)

バージョン: 1.22.5

Aurora MySQL 1.22.5 は一般公開されています。Aurora MySQL 1.* バージョンは MySQL 5.6 と互換性があり、Aurora MySQL 2.* バージョンは MySQL 5.7 と互換性があります。

このエンジンバージョンは 2023 年 2 月 28 日に非推奨となる予定です。詳細については、「Amazon Aurora MySQL 互換エディションバージョン 1 のサポート終了に向けて準備する」を参照してください。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

古いバージョンの Aurora MySQL を使用してクラスターを作成するには、RDS コンソール、AWS CLI、または Amazon RDS API を使用してエンジンバージョンを指定してください。

> **Note**
>
> このバージョンは、長期サポート (LTS) リリースとして指定されています。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL 長期サポート (LTS) リリース」を参照してください。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

可用性の向上:

- 内部クリーンアップスレッド間の同時実行の競合により、データベースが停止し、その後に再起動またはフェイルオーバーする状況を引き起こし得る問題を解決しました。
- XA トランザクションを準備完了の状態のままデータベースが再起動し、それらのトランザクションがコミットまたはロールバックされる前に再起動する場合に、クラスターが使用不可となる状態を引き起こし得る問題を修正しました。この修正の前に、クラスターを初期の再起動前の時点に復元することで、この問題に対処できます。

- DDL ステートメントの処理中にデータベースが再起動すると、InnoDB 消去がブロックされる状況を引き起こし得る問題を修正しました。その結果、InnoDB の履歴リストが長くなり、クラスターストレージボリュームがフルになるまで増加し続け、データベースが使用できなくなります。この修正の前に、データベースをもう一度再起動して消去のブロックを解除することで、問題を軽減できます。

Aurora MySQL データベースエンジンの更新 2021-03-04 (バージョン 1.22.4) (廃止)

バージョン: 1.22.4

Aurora MySQL 1.22.4 は一般公開されています。Aurora MySQL 1.* バージョンは MySQL 5.6 と互換性があり、Aurora MySQL 2.* バージョンは MySQL 5.7 と互換性があります。

このエンジンバージョンは 2023 年 2 月 28 日に非推奨となる予定です。詳細については、「Amazon Aurora MySQL 互換エディションバージョン 1 のサポート終了に向けて準備する」を参照してください。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

古いバージョンの Aurora MySQL を使用してクラスターを作成するには、RDS コンソール、AWS CLI、または Amazon RDS API を使用してエンジンバージョンを指定してください。

> **Note**
>
> このバージョンは、長期サポート (LTS) リリースとして指定されています。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL 長期サポート (LTS) リリース」を参照してください。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

セキュリティの修正内容:

マネージド型の環境での処理を微調整するための修正およびその他の機能強化。以下の CVE の追加の修正:

- CVE-2020-14867
- CVE-2020-14812
- CVE-2020-14793
- CVE-2020-14769
- CVE-2020-14765
- CVE-2020-14672
- CVE-2020-1971

可用性の向上:

- `kill session` コマンド中にデータベースの再起動またはフェイルオーバーがトリガーされる問題を修正しました。この問題が発生した場合は、AWS サポートに連絡し、インスタンスでこの修正を有効にしてください。
- バイナリログ記録が改善され、大きなトランザクションが関与している場合に、クラッシュリカバリ時間とコミットのレイテンシーが短縮されました。
- バイナリログレプリカが `HA_ERR_KEY_NOT_FOUND` エラーで停止する問題を修正しました。

Aurora MySQL データベースエンジンの更新 2020-11-09 (バージョン 1.22.3) (廃止)

バージョン: 1.22.3

Aurora MySQL 1.22.3 は一般公開されています。Aurora MySQL 1.* バージョンは MySQL 5.6 と互換性があり、Aurora MySQL 2.* バージョンは MySQL 5.7 と互換性があります。

このエンジンバージョンは 2023 年 2 月 28 日に非推奨となる予定です。詳細については、「Amazon Aurora MySQL 互換エディションバージョン 1 のサポート終了に向けて準備する」を参照してください。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

古いバージョンの Aurora MySQL を使用してクラスターを作成するには、RDS コンソール、AWS CLI、または Amazon RDS API を使用してエンジンバージョンを指定してください。

> **Note**
>
> このバージョンは、長期サポート (LTS) リリースとして指定されています。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL 長期サポート (LTS) リリース」を参照してください。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

セキュリティの修正内容:

マネージド型の環境での処理を微調整するための修正およびその他の機能強化。以下の CVE の追加の修正:

- CVE-2020-14559
- CVE-2020-14539
- CVE-2020-2579
- CVE-2020-2812
- CVE-2020-2780
- CVE-2020-2763

互換性のない変更:

このバージョンでは、mysqldump コマンドの動作に影響するアクセス許可の変更が導入されています。ユーザーは、PROCESS テーブルにアクセスする INFORMATION_SCHEMA.FILES 特権を有している必要があります。変更せずに mysqldump コマンドを実行するには、PROCESS コマンドが接続するデータベースユーザーに mysqldump 特権を付与します。mysqldump オプションを指定して --no-tablespaces コマンドを実行することもできます。このオプションを使用すると、mysqldump 出力に CREATE LOGFILE GROUP または CREATE TABLESPACE ステートメント

は含まれません。この場合、mysqldump コマンドは INFORMATION_SCHEMA.FILES テーブルにアクセスしないため、PROCESS アクセス許可を付与する必要はありません。

可用性の向上:

- コミットされていない DDL ステートメントの復旧中にサーバーを再起動する可能性がある問題を修正しました。
- サーバーを再起動する可能性があるロックマネージャーの競合状態を修正しました。
- 大規模なトランザクションの復旧中に、モニタリングエージェントがサーバを再起動する可能性がある問題を修正しました。

全般的な機能強化:

- MIXED 実行時の動作を binlog_format ではなく ROW STATEMENT を LOAD DATA FROM INFILE | S3 にマッピングするように変更しました。
- Aurora MySQL バイナリログプライマリに接続されたバイナリログレプリカで、プライマリが LOAD DATA FROM S3 を実行し、binlog_format が STATEMENT に設定されているときに、不完全なデータが表示される可能性がある問題を修正しました。

MySQL Community Edition バグ修正の統合

- バグ #26654685: 外部キーチェック中に破損したインデックス ID が検出され、アサーションが発生しました
- バグ #15831300: デフォルトでは、整数をマスター上の小さいタイプからスレーブ上のより大きなタイプ (例えば、マスターの SMALLINT 列からスレーブの BIGINT 列へ) に昇格すると、昇格された値は署名付きのように処理されます。このような場合、サーバーシステム可変 slave_type_converss に指定された値のセットで、ALL_SIGNED および ALL_UNSIGNED の一方または両方を使用してこの動作を変更またはオーバーライドすることができます。詳細については、「行ベースのレプリケーション: 属性の昇格と降格」、および可変についての説明を参照してください。
- バグ #17449901: foreign_key_checks=0 で、InnoDB は外部キー制約で必要なインデックスを削除することを許可し、テーブルを不整合に配置したため、テーブルのロード時に発生する外部キーチェックに失敗しました。InnoDB で、foreign_key_checks=0 であっても、外部キーの制約により必要なインデックスが削除できなくなりました。外部キーの制約は、外部キーのインデックスを削除するよりも前に削除する必要があります。

- バグ #20768847: ALTER TABLE ... 外部キーの依存関係を持つテーブルでの DROP INDEX のオペレーションによってアサーションが発生しました。

Aurora MySQL データベースエンジンの更新 2020-03-05 (バージョン 1.22.2) (廃止)

バージョン: 1.22.2

Aurora MySQL 1.22.2 は一般利用可能です。Aurora MySQL 1.* バージョンは MySQL 5.6 と互換性があり、Aurora MySQL 2.* バージョンは MySQL 5.7 と互換性があります。

このエンジンバージョンは 2023 年 2 月 28 日に非推奨となる予定です。詳細については、「Amazon Aurora MySQL 互換エディションバージョン 1 のサポート終了に向けて準備する」を参照してください。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

古いバージョンの Aurora MySQL を使用してクラスターを作成するには、RDS コンソール、AWS CLI、または Amazon RDS API を使用してエンジンバージョンを指定してください。

> **Note**
>
> このバージョンは現在、次のリージョンでは使用できません。AWS GovCloud (米国東部) [us-gov-east-1]、AWS GovCloud (米国西部) [us-gov-west-1] ご利用可能になりましたら、別途お知らせします。
> このバージョンは、長期サポート (LTS) リリースとして指定されています。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL 長期サポート (LTS) リリース」を参照してください。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

優先度の高い修正:

- 証明書のローテーション後に接続が断続的に失敗する問題を修正しました。
- 書き込み負荷が高い一部のデータベースクラスターでクローン作成に時間がかかる問題を修正しました。
- マスターとレプリカで `binlog_checksum` パラメータに異なる値が設定されている場合に、論理的なレプリケーションが停止される問題を修正しました。
- リードレプリカでスローログと全般ログが正しくローテーションされない問題を修正しました。
- ANSI の Read Committed 分離レベルの動作の問題を修正しました。

Aurora MySQL データベースエンジンの更新 2019-12-23 (バージョン 1.22.1) (廃止)

バージョン: 1.22.1

Aurora MySQL 1.22.1 は一般利用可能です。Aurora MySQL 1.* バージョンは MySQL 5.6 と互換性があり、Aurora MySQL 2.* バージョンは MySQL 5.7 と互換性があります。

このエンジンバージョンは 2023 年 2 月 28 日に非推奨となる予定です。詳細については、「Amazon Aurora MySQL 互換エディションバージョン 1 のサポート終了に向けて準備する」を参照してください。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

古いバージョンの Aurora MySQL を使用してクラスターを作成するには AWS Management Console、AWS CLI、または RDS API を使用してエンジンバージョンを指定してください。既存の Aurora MySQL 1.* データベースクラスターは、Aurora MySQL 1.22.1 にアップグレードすることができます。

> **Note**
>
> このバージョンは、現在以下の AWS リージョンでは使用できません。AWS GovCloud (米国東部) [us-gov-east-1]、AWS GovCloud (米国西部) [us-gov-west-1]、中国 (寧夏) [cn-northwest-1]、アジアパシフィック (香港) [ap-east-1]、および 中東 (バーレーン) [me-south-1]。ご利用可能になりましたら、別途お知らせします。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> DB クラスターのアップグレード手順が変わりました。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

重要な修正:

- テーブルロックとテンポラリテーブルを伴うエンジンリカバリが妨げていた問題を修正しました。
- テンポラリテーブルを使用するときのバイナリログの安定性を改善しました。

優先度の高い修正:

- Aurora 固有のデータベーストレースおよびログサブシステムで、確保できるメモリを不足させる低速メモリリークを修正しました。

Aurora MySQL データベースエンジンの更新 2019-11-25 (バージョン 1.22.0) (廃止)

バージョン: 1.22.0

Aurora MySQL 1.22.0 は一般利用可能です。Aurora MySQL 1.* バージョンは MySQL 5.6 と互換性があり、Aurora MySQL 2.* バージョンは MySQL 5.7 と互換性があります。

このエンジンバージョンは 2023 年 2 月 28 日に非推奨となる予定です。詳細については、「Amazon Aurora MySQL 互換エディションバージョン 1 のサポート終了に向けて準備する」を参照してください。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

古いバージョンの Aurora MySQL を使用してクラスターを作成するには AWS Management Console、AWS CLI、または RDS API を使用してエンジンバージョンを指定してください。既存の Aurora MySQL 1.* データベースクラスターは、Aurora MySQL 1.22.0 にアップグレードすることができます。

> **Note**
>
> このバージョンは、現在以下の AWS リージョンでは使用できません。AWS GovCloud (米国東部) [us-gov-east-1]、AWS GovCloud (米国西部) [us-gov-west-1]、中国 (寧夏) [cn-northwest-1]、アジアパシフィック (香港) [ap-east-1]、中東 (バーレーン) [me-south-1] および 南米 (サンパウロ) [sa-east-1]。ご利用可能になりましたら、別途お知らせします。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> DB クラスターのアップグレード手順が変わりました。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

新機能:

- Aurora MySQL クラスターで、インスタンスタイプ r5.8xlarge、r5.16xlarge、および r5.24xlarge がサポートされるようになりました。
- バイナリログに、非常に大きなトランザクション処理時のコミットのレイテンシーを改善するための新しい機能強化があります。
- Aurora MySQL に、大きなトランザクションがコミット時にバイナリログに書き込まれる時間枠を最小化するメカニズムが備わりました。これは実際には、その時間枠にデータベースのクラッシュが発生したとき、長期のオフライン復旧を招かないようにします。この機能は、バイナリログコミット時に大きなトランザクションが小さなトランザクションをブロックする問題も修正します。この機能はデフォルトでオフになっていますが、ワークロードに応じて必要な場合はサービスチー

ムによって有効化できます。有効化した場合、トランザクションサイズが 500 MB を超えるとトリガーされます。

- リードレプリカでの ANSI `READ COMMITTED` 分離レベルのサポートが追加されました。この分離レベルによって、ライターノードでの高い書き込みスループットに影響を与えることなく、リードレプリカに対する長期実行クエリが可能になります。詳細については、「Aurora MySQL の分離レベル」を参照してください。
- グローバルデータベースで、以下の AWS リージョンにデプロイされたデータベースクラスター用に、セカンダリの読み取り専用レプリカリージョンを追加できるようになりました。米国東部 (バージニア北部) [us-east-1]、米国東部 (オハイオ) [us-east-2]、米国西部 (北カリフォルニア) [us-west-1]、米国西部 (オレゴン) [us-west-2]、欧州 (アイルランド) [eu-west-1]、欧州 (ロンドン) [eu-west-2]、欧州 (パリ) [eu-west-3]、アジアパシフィック (東京) [ap-northeast-1]、アジアパシフィック (ソウル) [ap-northeast-2]、アジアパシフィック (シンガポール) [ap-southeast-1]、アジアパシフィック (シドニー) [ap-southeast-2]、カナダ (中部) [ca-central-1]、欧州 (フランクフルト) [eu-central-1] および アジアパシフィック (ムンバイ) [ap-south-1]。
- ホット行の競合機能が一般利用可能になりました。Aurora ラボモード設定をオンにする必要はありません。この機能により、同じページの列で多くのトランザクションが競合しているワークロードのスループットが大幅に向上します。
- このバージョンでは、新しいクラスター用に最新のブラジルのタイムゾーン更新をサポートするように、タイムゾーンファイルが更新されています。

重要な修正:

- CVE-2019-2922
- CVE-2019-2923
- CVE-2019-2924
- CVE-2019-2910

優先度の高い修正:

- 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
- 長期のデータベースダウンタイムが発生していた DDL 復旧コンポーネントの問題を修正しました。TRUNCATE TABLE 列のあるテーブルへの AUTO_INCREMENT クエリの実行後に使用できなくなるクラスターは、更新する必要があります。
- 長期のデータベースダウンタイムが発生していた DDL 復旧コンポーネントの問題を修正しました。複数のテーブルへの DROP TABLE クエリのパラレル実行後に使用できなくなるクラスターは、更新する必要があります。

全般の安定性に関する修正:

- 長時間実行トランザクション中にリードレプリカが再起動する問題を修正しました。解放可能なメモリでの高速ドロップと同時にレプリカが再起動しているお客様は、このバージョンへのアップグレードを検討する必要があります。
- ネストされたクエリがリードレプリカのテンポラリテーブルに対して実行されるとき、間違って ERROR 1836 が報告される問題を修正しました。
- Aurora 書き込みインスタンスで負荷の高い書き込みワークロードを実行しているときの Aurora 読み取りインスタンスのパラレルクエリ中止エラーを修正しました。
- バイナリログのマスターとして設定されたデータベースが、負荷の高い書き込みワークロードの実行中に再起動する問題を修正しました。
- エンジンの再起動中に長期にわたって利用不可になる問題を修正しました。これは、バッファープール初期化の問題に対処しています。この問題が発生することはまれですが、サポートされているリリースに影響する可能性があります。
- information_schema.replica_host_status テーブルで不整合データが生成された問題を修正しました。
- リーダーノードが断続的に再起動する、パラレルクエリとスタンダード実行パスの間の競合状態を修正しました。
- クライアント接続の数が max_connections パラメータ値を超えるときのデータベースの安定性が向上しました。
- サポートされない DDL および LOAD FROM S3 クエリをブロックすることによって、読み取りインスタンスの安定性が向上しました。

MySQL Community Edition バグ修正の統合

- バグ #16346241 - SERVER CRASH IN ITEM_PARAM::QUERY_VAL_STR
- バグ #17733850 - NAME_CONST() CRASH IN ITEM_NAME_CONST::ITEM_NAME_CONST()
- バグ #20989615 - INNODB AUTO_INCREMENT PRODUCES SAME VALUE TWICE
- バグ #20181776 - ACCESS CONTROL DOESN'T MATCH MOST SPECIFIC HOST WHEN IT CONTAINS WILDCARD
- バグ #27326796 - MYSQL CRASH WITH INNODB ASSERTION FAILURE IN FILE PARS0PARS.CC
- バグ #20590013 - IF YOU HAVE A FULLTEXT INDEX AND DROP IT YOU CAN NO LONGER PERFORM ONLINE DDL

Aurora MySQL データベースエンジンの更新 2019-11-25 (バージョン 1.21.0) (廃止)

バージョン: 1.21.0

Aurora MySQL 1.21.0 は一般利用可能です。Aurora MySQL 1.* バージョンは MySQL 5.6 と互換性があり、Aurora MySQL 2.* バージョンは MySQL 5.7 と互換性があります。

現在サポートされている Aurora MySQL リリースは、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.* および 2.07.* です。古いバージョンの Aurora MySQL を使用してクラスターを作成するには AWS Management Console、AWS CLI、または RDS API を使用してエンジンバージョンを指定してください。既存の Aurora MySQL 1.* データベースクラスターは、Aurora MySQL 1.21.0 にアップグレードすることができます。

> **Note**
>
> このバージョンは、現在以下の AWS リージョンでは使用できません。AWS GovCloud (米国東部) [us-gov-east-1]、AWS GovCloud (米国西部) [us-gov-west-1]、中国 (寧夏) [cn-northwest-1]、アジアパシフィック (香港) [ap-east-1]、欧州 (ストックホルム) [eu-north-1] および 中東 (バーレーン) [me-south-1]。ご利用可能になりましたら、別途お知らせします。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> DB クラスターのアップグレード手順が変わりました。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

重要な修正:

- CVE-2018-0734
- CVE-2019-2534
- CVE-2018-2612
- CVE-2017-3599
- CVE-2018-2562
- CVE-2017-3329
- CVE-2018-2696
- CVE-2015-4737

優先度の高い修正:

- 64 テビバイト (TiB) に近いサイズのデータベースを使用するお客様は、Aurora ストレージの制限に近いボリュームに影響する安定性のバグが引き起こすダウンタイムを防止するため、このバージョンにアップグレードすることを強くお勧めします。

全般の安定性に関する修正:

- Aurora 書き込みインスタンスで重度の書き込みワークロードの実行しているときの Aurora 読み取りインスタンスのパラレルクエリ中止エラーを修正しました。

- 書き込みインスタンスの重度のトランザクションコミットトラフィックがあるときに、トランザクションが長時間稼働した際に空きメモリを減少させた Aurora 読み取りインスタンスの問題を修正しました。
- データベースの再起動またはホストの置き換えの後で、パラメータ `aurora_disable_hash_join` の値は保持されます。
- Aurora インスタンスがメモリ不足となる原因となったフルテキスト検索キャッシュに関する問題を修正しました。フルテキスト検索を使用するお客様は、アップグレードする必要があります。
- ハッシュ結合機能が有効になっており、インスタンスのメモリが低いときのデータベースの安定性を改善しました。ハッシュ結合を使用するお客様は、アップグレードする必要があります。
- 「接続が多すぎます」エラーが再起動を引き起こす可能性のあるクエリキャッシュの問題を修正しました。
- 不要な再起動を防止するため、スワップメモリ領域を含めるように T2 インスタンスの空きメモリの計算を修正しました。

MySQL Community Edition バグ修正の統合

- バグ #19929406: HANDLE_FATAL_SIGNAL (SIG=11) IN __MEMMOVE_SSSE3_BACK FROM STRING::COPY
- バグ #17059925: UNION ステートメントに、行検証の値が正しく計算されませんでした。これは、Performance Schemaステートメントテーブル (events_statements_current など) の `ROWS_EXAMINED` 列に対して大きすぎる値として現れていました。
- バグ #11827369: `SELECT ... FROM DUAL` ネストされたサブクエリのいくつかのクエリでアサーションが発生しました。
- バグ #16311231: `IN` 句の XOR オペレーションを含む `WHERE` 句でクエリがサブクエリを含む場合に、正しくない結果が返されました。

Aurora MySQL データベースエンジンの更新 2020-03-05 (バージョン 1.20.1) (廃止)

バージョン: 1.20.1

Aurora MySQL 1.20.1 は一般利用可能です。Aurora MySQL 1.* バージョンは MySQL 5.6 と互換性があり、Aurora MySQL 2.* バージョンは MySQL 5.7 と互換性があります。

現在サポートされている Aurora MySQL リリースは、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.* および 2.07.* です。Aurora MySQL 1.* データベースのスナップショットを Aurora MySQL 1.20.1 に復元できます。

古いバージョンの Aurora MySQL を使用してクラスターを作成するには、RDS コンソール、AWS CLI、または Amazon RDS API を使用してエンジンバージョンを指定してください。

> **Note**
>
> このバージョンは現在、次のリージョンでは使用できません。AWS GovCloud (米国東部) [us-gov-east-1]、AWS GovCloud (米国西部) [us-gov-west-1] ご利用可能になりましたら、別途お知らせします。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

優先度の高い修正:

- 証明書のローテーション後に接続が断続的に失敗する問題を修正しました。
- ワークロードが高い場合にフェイルオーバーが発生する接続の終了の同時実行に関する問題を修正しました。

全般の安定性に関する修正:

- 内部的に中間テーブルを使用する複数のテーブルの結合や集計を伴う複雑なクエリの実行中に発生するクラッシュを修正しました。

Aurora MySQL データベースエンジンの更新 2019-11-11 (バージョン 1.20.0) (廃止)

バージョン: 1.20.0

Aurora MySQL 1.20.0 は一般利用可能です。Aurora MySQL 1.* バージョンは MySQL 5.6 と互換性があり、Aurora MySQL 2.* バージョンは MySQL 5.7 と互換性があります。

現在サポートされている Aurora MySQL リリースは、1.14.*、1.15.*、1.16.*、1.17.*、1.18.*、1.19.*、1.20.*、2.01.*、2.02.*、2.03.*、および 2.04.* です。古いバージョンの Aurora MySQL を使用してクラスターを作成するには AWS Management Console、AWS CLI、または RDS API を使用してエンジンバージョンを指定してください。既存の Aurora MySQL 1.* データベースクラスターは、最大 1.19.5 から Aurora MySQL 1.20.0 にアップグレードすることができます。

> **Note**
>
> このバージョンは、現在以下の AWS リージョンでは使用できません。AWS GovCloud (米国東部) [us-gov-east-1]、AWS GovCloud (米国西部) [us-gov-west-1]、中国 (寧夏) [cn-northwest-1]、アジアパシフィック (香港) [ap-east-1]、欧州 (ストックホルム) [eu-north-1] および 中東 (バーレーン) [me-south-1]。ご利用可能になりましたら、別途お知らせします。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> DB クラスターのアップグレード手順が変わりました。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

重要な修正:

- CVE-2018-0734
- CVE-2019-2534
- CVE-2018-2612
- CVE-2017-3599

- CVE-2018-2562
- CVE-2017-3329
- CVE-2018-2696
- CVE-2015-4737

優先度の高い修正:

- 64 テビバイト (TiB) に近いサイズのデータベースを使用するお客様は、Aurora ストレージの制限に近いボリュームに影響する安定性のバグが引き起こすダウンタイムを防止するため、このバージョンにアップグレードすることを強くお勧めします。

全般の安定性に関する修正:

- Aurora 書き込みインスタンスで重度の書き込みワークロードの実行しているときの Aurora 読み取りインスタンスのパラレルクエリ中止エラーを修正しました。
- 書き込みインスタンスの重度のトランザクションコミットトラフィックがあるときに、トランザクションが長時間稼働した際に空きメモリを減少させた Aurora 読み取りインスタンスの問題を修正しました。
- データベースの再起動またはホストの置き換えの後で、パラメータ `aurora_disable_hash_join` の値は保持されます。
- Aurora インスタンスがメモリ不足となる原因となったフルテキスト検索キャッシュに関する問題を修正しました。フルテキスト検索を使用するお客様は、アップグレードする必要があります。
- ハッシュ結合機能が有効になっており、インスタンスのメモリが低いときのデータベースの安定性を改善しました。ハッシュ結合を使用するお客様は、アップグレードする必要があります。
- 「接続が多すぎます」エラーが再起動を引き起こす可能性のあるクエリキャッシュの問題を修正しました。
- 不要な再起動を防止するため、スワップメモリ領域を含めるように T2 インスタンスの空きメモリの計算を修正しました。

MySQL Community Edition バグ修正の統合

- バグ #19929406: HANDLE_FATAL_SIGNAL (SIG=11) IN __MEMMOVE_SSSE3_BACK FROM STRING::COPY

- バグ #17059925: UNION ステートメントに、行検証の値が正しく計算されませんでした。これは、Performance Schemaステートメントテーブル (events_statements_current など) の ROWS_EXAMINED 列に対して大きすぎる値として現れていました。
- バグ #11827369: SELECT ... FROM DUAL ネストされたサブクエリのいくつかのクエリでアサーションが発生しました。
- バグ #16311231: IN 句の XOR オペレーションを含む WHERE 句でクエリがサブクエリを含む場合に、正しくない結果が返されました。

Aurora MySQL データベースエンジンの更新 2020-03-05 (バージョン 1.19.6) (廃止)

バージョン: 1.19.6

Aurora MySQL 1.19.6 は一般利用可能です。Aurora MySQL 1.* バージョンは MySQL 5.6 と互換性があり、Aurora MySQL 2.* バージョンは MySQL 5.7 と互換性があります。

このエンジンバージョンは 2023 年 2 月 28 日に非推奨となる予定です。詳細については、「Amazon Aurora MySQL 互換エディションバージョン 1 のサポート終了に向けて準備する」を参照してください。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

Aurora MySQL 1.* データベースのスナップショットを Aurora MySQL 1.19.6 に復元できます。

古いバージョンの Aurora MySQL を使用してクラスターを作成するには、RDS コンソール、AWS CLI、または Amazon RDS API を使用してエンジンバージョンを指定してください。

> **Note**
>
> このバージョンは現在、次のリージョンでは使用できません。AWS GovCloud (米国東部) [us-gov-east-1]、AWS GovCloud (米国西部) [us-gov-west-1] ご利用可能になりましたら、別途お知らせします。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

優先度の高い修正:

- 証明書のローテーション後に接続が断続的に失敗する問題を修正しました。

Aurora MySQL データベースエンジンの更新 2019-09-19 (バージョン 1.19.5) (廃止)

バージョン 1.19.5

Aurora MySQL 1.19.5 は一般利用可能です。Aurora MySQL 1.* バージョンは MySQL 5.6 と互換性があり、Aurora MySQL 2.* バージョンは MySQL 5.7 と互換性があります。

このエンジンバージョンは 2023 年 2 月 28 日に非推奨となる予定です。詳細については、「Amazon Aurora MySQL 互換エディションバージョン 1 のサポート終了に向けて準備する」を参照してください。

現在サポートされている Aurora MySQL リリースは、1.19.5、1.19.6、1.22.*、1.23.*、2.04.*、2.07.*、2.08.*、2.09.*、2.10.*、3.01.*、3.02.* です。

既存のデータベースクラスターは、Aurora MySQL 1.19.5 にアップグレードすることができます。Aurora MySQL 1.14.*、1.15.*、1.16.*、1.17.*、1.18.*、1.19.1、および 1.19.2 のスナップショットは Aurora MySQL 1.19.5 に復元することができます。

Aurora MySQL の古いバージョンを使用するには AWS Management Console、AWS CLI、または RDS API を通してエンジンバージョンを指定することにより、新しいデータベースクラスターを作成します。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> このバージョンは、現在以下の AWS リージョンでは使用できません。欧州 (ロンドン) [eu-west-2]、AWS GovCloud (米国東部) [us-gov-east-1]、AWS GovCloud (米国西部) [us-gov-west-1]、中国 (寧夏) [cn-northwest-1] および アジアパシフィック (香港) [ap-east-1]。ご利用可能になりましたら、別途お知らせします。

> **Note**
>
> DB クラスターのアップグレード手順が変わりました。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

- 書き込みインスタンスの重度のトランザクションコミットトラフィックがあるときに、トランザクションが長時間稼働した際に空きメモリを減少させた Aurora 読み取りインスタンスの問題を修正しました。
- Aurora 書き込みインスタンスで重度の書き込みワークロードの実行しているときの Aurora 読み取りインスタンスのパラレルクエリ中止エラーを修正しました。
- データベースの再起動またはホストの置き換えの後で、パラメータ `aurora_disable_hash_join` の値は保持されます。
- Aurora インスタンスがメモリ不足となる原因となったフルテキスト検索キャッシュに関する問題を修正しました。
- フェイルオーバーなしに復旧ワークフローが完了するための 160 GB のスペースを予約することで、ボリュームサイズが 64 テビバイト (TiB) ボリュームの制限に近いときのデータベースの安定性を向上させました。
- ハッシュ結合機能が有効になっており、インスタンスのメモリが低いときのデータベースの安定性を改善しました。
- 途中で再起動する原因となった T2 インスタンスのスワップメモリ領域を含むために、空きメモリの計算を修正しました。
- 「接続が多すぎます」エラーが再起動を引き起こす可能性のあるクエリキャッシュの問題を修正しました。

MySQL Community Edition バグ修正の統合

- CVE-2018-2696
- CVE-2015-4737
- バグ #19929406: HANDLE_FATAL_SIGNAL (SIG=11) IN __MEMMOVE_SSSE3_BACK FROM STRING::COPY

- バグ #17059925: UNION ステートメントに、行検証の値が正しく計算されませんでした。これは、Performance Schemaステートメントテーブル (events_statements_current など) の ROWS_EXAMINED 列に対して大きすぎる値として現れていました。
- バグ #11827369: SELECT ... FROM DUAL ネストされたサブクエリのいくつかのクエリでアサーションが発生しました。
- バグ #16311231: IN 句の XOR オペレーションを含む WHERE 句でクエリがサブクエリを含む場合に、正しくない結果が返されました。

Aurora MySQL データベースエンジンの更新 2019-06-05 (バージョン 1.19.2) (廃止)

バージョン: 1.19.2

Aurora MySQL 1.19.2 が一般公開されました。MySQL 5.6 との互換性を持つ新しい Aurora MySQL データベースクラスターはすべて、スナップショットから復元されるものも含め、1.17.8、1.19.0、1.19.1、または 1.19.2 で作成できます。既存のデータベースクラスターは Aurora MySQL 1.19.2 にアップグレードできます。ただし必須ではありません。古いバージョンを使用する場合は、新しいデータベースクラスターを Aurora MySQL 1.14.4、Aurora MySQL 1.15.1、Aurora MySQL 1.16、Aurora MySQL 1.17.8、または Aurora MySQL 1.18 で作成できます。このためには、AWS CLI または Amazon RDS API を使用して、エンジンバージョンを指定します。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> このバージョンは現在、AWS GovCloud (米国西部) [us-gov-west-1]、欧州 (ストックホルム) [eu-north-1]、中国 (寧夏) [cn-northwest-1]、および アジアパシフィック (香港) [ap-east-1] の、各 AWS リージョンでは使用できません。ご利用可能になりましたら、別途お知らせします。

> **Note**
>
> DB クラスターのアップグレード手順が変わりました。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

- データを Amazon S3 から Aurora にロードする場合にエラーが発生する問題を修正。
- データを Aurora から Amazon S3 にアップロードする場合にエラーが発生する問題を修正。
- ゾンビセッションが強制終了状態のままになる問題を修正。
- ネットワークプロトコル管理のエラーの処理中に接続が中止される問題を修正。
- パーティショニングされたテーブルを使用する際にクラッシュする問題を修正。
- トリガーの作成の binlog レプリケーションに関する問題を修正。

Aurora MySQL データベースエンジンの更新 2019-05-09 (バージョン 1.19.1) (廃止)

バージョン: 1.19.1

Aurora MySQL 1.19.1 は一般利用可能です。MySQL 5.6 との互換性を持つ新しい Aurora MySQL データベースクラスターはすべて、スナップショットから復元されるものも含め、1.17.8、1.19.0、または 1.19.1 で作成できます。既存のデータベースクラスターは Aurora MySQL 1.19.1 にアップグレードできます。ただし必須ではありません。古いバージョンを使用する場合は、新しいデータベースクラスターを Aurora MySQL 1.14.4、Aurora MySQL 1.15.1、Aurora MySQL 1.16、Aurora MySQL 1.17.8、または Aurora MySQL 1.18 で作成できます。このためには、AWS CLI または Amazon RDS API を使用して、エンジンバージョンを指定します。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> このバージョンは現在、AWS GovCloud (米国西部) [us-gov-west-1] リージョンおよび 中国 (北京) [cn-north-1] リージョンでは使用できません。ご利用可能になりましたら、別途お知らせします。

> **Note**
>
> DB クラスターのアップグレード手順が変わりました。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

改良点

- バイナリログワーカーとして設定された Aurora インスタンスで問題が発生する可能性のあるバイナリログレプリケーションのバグを修正。
- 特定の種類の `ALTER TABLE` コマンドを処理する際のエラーを修正。
- ネットワークプロトコル管理のエラーによる接続の失敗に関する問題を修正。

Aurora MySQL データベースエンジンの更新 2019-02-07 (バージョン 1.19.0) (廃止)

バージョン: 1.19.0

Aurora MySQL 1.19.0 は一般利用可能です。MySQL 5.6 との互換性を持つ新しい Aurora MySQL データベースクラスターはすべて、スナップショットから復元されるものも含め、1.17.8 または 1.19.0 で作成できます。既存のデータベースクラスターは Aurora MySQL 1.19.0 にアップグレードできます。ただし必須ではありません。古いバージョンを使用する場合は、新しいデータベースクラスターを Aurora MySQL 1.14.4、Aurora MySQL 1.15.1、Aurora MySQL 1.16、Aurora MySQL 1.17.8、Aurora MySQL 1.18.0 で作成できます。このためには、AWS CLI または Amazon RDS API を使用して、エンジンバージョンを指定します。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> このバージョンは現在、AWS GovCloud (米国西部) [us-gov-west-1] リージョンおよび 中国 (北京) [cn-north-1] リージョンでは使用できません。ご利用可能になりましたら、別途お知らせします。

> **Note**
>
> DB クラスターのアップグレード手順が変わりました。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのマイナーバージョンまたはパッチレベルのアップグレード」を参照してください。

機能

- Aurora バージョンセレクタ - Aurora MySQL 1.19.0 以降、Amazon RDS で複数のバージョンの MySQL 5.6 互換 Aurora から選択できます。詳細については、「Amazon Aurora ユーザーガイド」の「AWS による Aurora MySQL エンジンバージョンの確認または指定」を参照してください。

改良点

- Aurora Replica の `CHECK TABLE` クエリに関する安定性の問題を修正しました。
- ハッシュ結合を無効にするための新しいグローバルユーザー可変 `aurora_disable_hash_join` を追加しました。
- 複数テーブルのハッシュ結合中に出力行を生成する際に生じる安定性の問題を修正しました。
- ハッシュ結合の適用可能性チェック中の計画変更が原因で誤った結果が返る問題を修正しました。
- 長時間実行トランザクションでのダウンタイムのないパッチ適用がサポートされました。この機能強化は、バージョン 1.19 から上位のバージョンにアップグレードすると有効になります。

- binlog が有効になっている場合のダウンタイムのないパッチ適用のサポートがスタートされました。この機能強化は、バージョン 1.19 から上位のバージョンにアップグレードすると有効になります。
- ワークロードとは無関係に Aurora Replica の CPU 使用率が急上昇する問題を修正しました。
- データベースの再起動につながるロックマネージャーの競合状態を修正しました。
- Aurora インスタンスの安定性を向上させるためにロックマネージャーコンポーネントの競合状態を修正しました。
- ロックマネージャーコンポーネント内のデッドロックディテクターの安定性が向上しました。
- INSERTインデックスが破損されていることが InnoDB によって検出された場合、テーブルの オペレーションは禁止されます。
- 高速 DDL における安定性の問題を修正しました。
- 単一行のサブクエリのスキャンバッチ処理におけるメモリ消費を削減することで、Aurora の安定性は向上しました。
- システム可変 `foreign_key_checks` が 0 に設定されている場合に外部キーが削除されると発生する安定性の問題を修正しました。
- ユーザーによる `table_definition_cache` 値の変更を誤って上書きしてしまうメモリー不足 (OOM) 回避機能の問題を修正しました。
- メモリ不足回避機能の安定性の問題を修正しました。
- `query_time` の `lock_time` と `slow_query_log` がガベージ値に設定される問題を修正しました。
- 文字列の照合が内部で不適切に処理されることによって生じるパラレルクエリの安定性の問題を修正しました。
- セカンダリインデックスの検索によってトリガーされるパラレルクエリの安定性の問題を修正しました。
- マルチテーブルの更新によってトリガーされるパラレルクエリの安定性の問題を修正しました。

MySQL Community Edition バグ修正の統合

- BUG #32917: DETECT ORPHAN TEMP-POOL FILES, AND HANDLE GRACEFULLY
- BUG #63144 CREATE TABLE IF NOT EXISTS METADATA LOCK IS TOO RESTRICTIVE

Aurora MySQL データベースエンジンの更新 2018-09-20 (バージョン 1.18.0) (廃止)

(バージョン 1.18.0)

Aurora MySQL 1.18.0 は一般利用可能です。MySQL 5.6 との互換性を持つすべての新しい Aurora MySQL パラレルクエリクラスターは、スナップショットから復元されるものも含めて、Aurora MySQL 1.18.0 で作成されます。既存のパラレルクエリクラスターは Aurora MySQL 1.18.0 にアップグレードできます (必須ではありません)。新しい DB クラスターは、Aurora MySQL 1.14.4、Aurora MySQL 1.15.1、 Aurora MySQL 1.16、または Aurora MySQL 1.17.6 で作成できます。このためには、AWS CLI または Amazon RDS API を使用して、エンジンバージョンを指定します。

Aurora MySQL のバージョン 1.18.0 では、クラスターパッチ適用モデルが使用されており、Aurora DB クラスターのすべてのノードに同時にパッチが適用されます。

> **Important**
>
> Aurora MySQL 1.18.0 は、Aurora パラレルクエリクラスターにのみ適用されます。プロビジョニングされた 5.6.10a クラスターをアップグレードすると、結果のバージョンは 1.17.8 になります。パラレルクエリ 5.6.10a クラスターをアップグレードすると、結果のバージョンは 1.18.0 になります。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

機能

- このリリースでは、パラレルクエリを、新しいクラスターと復元されたスナップショットに対し使用できます。Aurora MySQL パラレルクエリは、データ集約的なクエリの処理に関連する I/O と計算の一部をパラレル処理する最適化です。パラレル処理される作業には、ストレージから行を取得し、列の値を抽出して、`WHERE` 句と結合句の条件に一致する行を判別することが含まれます。このデータ集約型の作業は、Aurora 分散ストレージレイヤー内の複数のノードに委譲されます (データベース最適化の用語では、プッシュダウンされます)。並行クエリがないと、各クエリはすべてのスキャンされたデータを Aurora MySQL クラスター (ヘッドノード) 内の単一のノードに持ち込み、そこですべてのクエリ処理を実行します。

- パラレルクエリ機能を有効にすると、Aurora MySQL エンジンは、ヒントやテーブル属性などの SQL の変更を必要とせずに、クエリがいつ利点を得られるかを自動的に判断します。

  詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL のパラレルクエリの使用」を参照してください。

- OOM 回避: この機能はシステムメモリをモニタリングし、データベースのさまざまなコンポーネントによって消費されるメモリを追跡します。システムのメモリが不足すると、さまざまなトラッキング対象コンポーネントからメモリを解放し、データベースがメモリ不足 (OOM) にならないようにするためのアクションのリストを実行し、データベースの再起動を回避します。このベストエフォート型機能は、デフォルトで t2 インスタンスで有効になっており、`aurora_oom_response` という名前の新しいインスタンスパラメータを使用して他のインスタンスクラスで有効にすることができます。このインスタンスパラメータには、メモリが少ないときにインスタンスによって実行されるべきカンマ区切りの文字列のアクションを取ります。有効なアクションには、「print」、「tune」、「decline」、「kill_query」、またはこれらの任意の組み合わせが含まれます。空の文字列はアクションが実行されないことを意味し、実質的にこの機能が無効になります。この機能のデフォルトのアクションは「print、tune」であることに注意してください。使用例:

  - 「print」 - 大量のメモリを使用するクエリのみを出力します。
  - 「tune」 - 内部テーブルキャッシュを調整して、メモリをシステムに戻します。
  - 「decline」 - インスタンスがメモリ不足になったら、新しいクエリを拒否します。
  - 「kill_query」 - インスタンスメモリが低いしきい値を超えるまで、メモリ消費の降順でクエリを強制終了します。データ定義言語 (DDL) ステートメントは強制終了されません。
  - 「print、tune」 -「print」と「tune」の両方について記述されたアクションを実行します。
  - 「tune、decline、kill_query」 -「tune」、「decline」、「kill_query」について記述されたアクションを実行します。

  メモリ不足の問題の処理やその他のトラブルシューティングのアドバイスについては、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL のメモリ不足の問題」を参照してください。

Aurora MySQL データベースエンジンの更新 2020-03-05 (バージョン 1.17.9) (廃止)

(バージョン 1.17.9)

Aurora MySQL 1.17.9 は一般利用可能です。Aurora MySQL 1.* バージョンは MySQL 5.6 と互換性があり、Aurora MySQL 2.* バージョンは MySQL 5.7 と互換性があります。

現在サポートされている Aurora MySQL リリースは、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.* および 2.07.* です。Aurora MySQL 1.* データベースのスナップショットを Aurora MySQL 1.17.9 に復元できます。

古いバージョンの Aurora MySQL を使用してクラスターを作成するには、RDS コンソール、AWS CLI、または Amazon RDS API を使用してエンジンバージョンを指定してください。

> **Note**
>
> このバージョンは現在、次のリージョンでは使用できません。AWS GovCloud (米国東部) [us-gov-east-1]、AWS GovCloud (米国西部) [us-gov-west-1] ご利用可能になりましたら、別途お知らせします。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

優先度の高い修正:

- 証明書のローテーション後に接続が断続的に失敗する問題を修正しました。

Aurora MySQL データベースエンジンの更新 2019-01-17 (バージョン 1.17.8) (廃止)

(バージョン 1.17.8)

Aurora MySQL 1.17.8 は一般利用可能です。MySQL 5.6 との互換性を持つすべての新しい Aurora MySQL データベースクラスターは、スナップショットから復元されるものも含めて、Aurora MySQL 1.17.8 で作成されます。既存のデータベースクラスターは Aurora MySQL 1.17.8 にアップグレードできます。ただし必須ではありません。古いバージョンを使用する場合は、新しいデータベー

スクラスターを Aurora MySQL 1.14.4、1.15.1、1.16、または 1.17.7 で作成できます。このためには、AWS CLI または Amazon RDS API を使用して、エンジンバージョンを指定します。

Aurora MySQL のバージョン 1.17.8 では、クラスターパッチ適用モデルが使用されており、Aurora DB クラスターのすべてのノードに同時にパッチが適用されます。

> **Note**
>
> このバージョンは現在、AWS GovCloud (米国西部) [us-gov-west-1] リージョンおよび 中国 (北京) [cn-north-1] リージョンでは使用できません。ご利用可能になりましたら、別途お知らせします。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

- 再起動後の Aurora レプリカで CPU 使用率が高くなるというパフォーマンス問題を修正しました。
- ハッシュ結合を使用した SELECT クエリの安定性の問題を修正しました。

MySQL Community Edition バグ修正の統合

- バグ #13418638: CREATE TABLE IF NOT EXISTS METADATA LOCK IS TOO RESTRICTIVE

Aurora MySQL データベースエンジンの更新 2018-10-08 (バージョン 1.17.7) (廃止)

(バージョン 1.17.7)

Aurora MySQL 1.17.7 は一般利用可能です。MySQL 5.6 との互換性を持つすべての新しい Aurora MySQL データベースクラスターは、スナップショットから復元されるものも含めて、Aurora MySQL 1.17.7 で作成されます。既存のデータベースクラスターは Aurora MySQL 1.17.7 にアップグレードできます。ただし必須ではありません。古いバージョンを使用する場合は、新しいデータベー

スクラスターを Aurora MySQL 1.14.4、1.15.1、1.16、1.17.6 で作成できます。このためには、AWS CLI または Amazon RDS API を使用して、エンジンバージョンを指定します。

Aurora MySQL のバージョン 1.17.7 では、クラスターパッチ適用モデルが使用されており、Aurora DB クラスターのすべてのノードに同時にパッチが適用されます。

> **Note**
>
> このバージョンは現在、AWS GovCloud (米国西部) [us-gov-west-1] リージョンおよび 中国 (北京) [cn-north-1] リージョンでは使用できません。ご利用可能になりましたら、別途お知らせします。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

- InnoDB ステータス可変 `innodb_buffer_pool_size` は、顧客が変更することができるように公開されています。
- フェイルオーバー時に発生した Aurora クラスターの安定性の問題を修正しました。
- 失敗した `TRUNCATE` 操作の後に発生した DDL 回復の問題を修正して、クラスターの可用性を向上させました。
- DDL 操作によってトリガーされる `mysql.innodb_table_stats` テーブルの更新に関する安定性の問題を修正しました。
- DDL 操作後のクエリキャッシュの無効化中にトリガーされた Aurora レプリカの安定性の問題が修正されました。
- バックグラウンドで定期的なディクショナリキャッシュの削除時に無効なメモリアクセスによってトリガーされた安定性の問題を修正しました。

MySQL Community Edition バグ修正の統合

- バグ #16208542: 外部キー列のインデックスを削除すると、テーブルが見つからなくなります。
- バグ #76349: add_derived_key() のメモリーリーク。

- バグ #16862316: パーティショニングされたテーブルの場合、インデックスマージが使用されたかどうかによって、クエリが異なる結果を返すことがありました。
- バグ #17588348: HASH でパーティショニングされたテーブルに対して実行すると、index_merge 最適化 (インデックスマージの最適化) を使用するクエリが無効な結果を返す可能性があります。

Aurora MySQL データベースエンジンの更新 2018-09-06 (バージョン 1.17.6) (廃止)

(バージョン 1.17.6)

Aurora MySQL 1.17.6 は一般利用可能です。MySQL 5.6 との互換性を持つすべての新しい Aurora MySQL データベースクラスターは、スナップショットから復元されるものも含めて、Aurora MySQL 1.17.6 で作成されます。既存のデータベースクラスターは Aurora MySQL 1.17.6 にアップグレードできます。ただし必須ではありません。古いバージョンを使用する場合は、新しいデータベースクラスターを Aurora MySQL 1.14.4、1.15.1、1.16、1.17.5 で作成できます。このためには、AWS CLI または Amazon RDS API を使用して、エンジンバージョンを指定します。

Aurora MySQL のバージョン 1.17.6 では、クラスターパッチ適用モデルが使用されており、Aurora DB クラスターのすべてのノードに同時にパッチが適用されます。

> **Note**
>
> このバージョンは現在、AWS GovCloud (米国西部) [us-gov-west-1] リージョンおよび 中国 (北京) [cn-north-1] リージョンでは使用できません。ご利用可能になりましたら、別途お知らせします。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

- Aurora ライターが同じテーブルに対して DDL 操作を実行している間に、SELECT クエリに対する Aurora リーダーの安定性の問題を修正しました。
- ヒープ/メモリエンジンを使用するテンポラリテーブルの DDL ログの作成と削除によって生じる安定性の問題を修正しました。

- バイナリログマスターへの接続が不安定なときに、DDL ステートメントのレプリケーション中に発生するバイナリログワーカーの安定性の問題を修正しました。
- 低速クエリログへの書き込み中に発生した安定性の問題を修正しました。
- 誤った Aurora リーダーのラグ情報が表示されるレプリカステータステーブルの問題を修正しました。

MySQL Community Edition バグ修正の統合

- BINARY 列のデフォルト値の名称変更または変更した ALTER TABLE ステートメントの場合、変更はテーブルコピーを使用して実行されており、適切ではありません。(バグ #67141、バグ #14735373、バグ #69580、バグ #17024290)
- 暗黙的にグループ化された通常のテーブルと派生したテーブルの間の外部結合によって、サーバーが終了する可能性があります。(バグ #16177639)

Aurora MySQL データベースエンジンの更新 2018-08-14 (バージョン 1.17.5) (廃止)

(バージョン 1.17.5)

Aurora MySQL 1.17.5 は一般利用可能です。MySQL 5.6 との互換性を持つすべての新しい Aurora MySQL データベースクラスターは、スナップショットから復元されるものも含めて、Aurora MySQL 1.17.5 で作成されます。既存のデータベースクラスターは Aurora MySQL 1.17.5 にアップグレードできます。ただし必須ではありません。古いバージョンを使用する場合は、新しいデータベースクラスターを Aurora MySQL 1.14.4、1.15.1、1.16、1.17.4 で作成できます。このためには、AWS CLI または Amazon RDS API を使用して、エンジンバージョンを指定します。

Aurora MySQL のバージョン 1.17.5 では、クラスターパッチ適用モデルが使用されており、Aurora DB クラスターのすべてのノードに同時にパッチが適用されます。

> **Note**
>
> このバージョンは現在、AWS GovCloud (米国西部) [us-gov-west-1] リージョンおよび 中国 (北京) [cn-north-1] リージョンでは使用できません。ご利用可能になりましたら、別途お知らせします。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

- ゼロダウンタイムパッチ機能を使用して Aurora クラスターにパッチを適用した後、Aurora ライターが再起動する可能性がある問題を修正しました。

Aurora MySQL データベースエンジンの更新 2018-08-07 (バージョン 1.17.4) (廃止)

(バージョン 1.17.4)

Aurora MySQL 1.17.4 は一般利用可能です。MySQL 5.6 との互換性を持つすべての新しい Aurora MySQL データベースクラスターは、スナップショットから復元されるものも含めて、Aurora MySQL 1.17.4 で作成されます。既存のデータベースクラスターは Aurora MySQL 1.17.4 にアップグレードできます。ただし必須ではありません。古いバージョンを使用する場合は、新しいデータベースクラスターを Aurora MySQL 1.14.4、1.15.1、1.16、1.17.3 で作成できます。このためには、AWS CLI または Amazon RDS API を使用して、エンジンバージョンを指定します。

Aurora MySQL のバージョン 1.17.4 では、クラスターパッチ適用モデルが使用されており、Aurora DB クラスターのすべてのノードに同時にパッチが適用されます。

> **Note**
>
> このバージョンは現在、AWS GovCloud (米国西部) [us-gov-west-1] リージョンおよび 中国 (北京) [cn-north-1] リージョンでは使用できません。ご利用可能になりましたら、別途お知らせします。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

- レプリケーションの改善:

  - ネットワークトラフィックを減らすために、クラスターレプリカへのバイナリログの送信を停止します。この改善は、デフォルトで有効になります。
  - ネットワークトラフィックを減らすためにレプリケーションメッセージを圧縮します。この改善は、8xlarge と 16xlarge のインスタンスクラスに対してデフォルトで有効になります。このような大規模なインスタンスは、大量の書き込みトラフィックに対応できるため、結果としてレプリケーションメッセージのネットワークトラフィックは膨大な量になります。
  - レプリカのクエリキャッシュを修正しました。
- `ORDER BY LOWER(col_name)` 照合の使用時に `utf8_bin` で不正な順序が生成される問題を修正しました。
- DDL ステートメント (特に `TRUNCATE TABLE`) が Aurora レプリカで障害 (不安定性やテーブルの欠如など) を起こす問題を修正しました。
- ストレージノードの再起動時にソケットが半開きの状態になる問題を修正しました。
- 以下の新しい DB クラスターパラメータが利用可能です。
  - `aurora_enable_zdr` - Aurora レプリカで開かれた接続を、レプリカの再起動時にアクティブなままにします。
  - `aurora_enable_replica_log_compression` - レプリケーションペイロードの圧縮を有効にして、マスターと Aurora レプリカ間のネットワーク帯域幅使用率を向上させます。
  - `aurora_enable_repl_bin_log_filtering` - マスターの Aurora レプリカで使用できないレプリケーションレコードのフィルタリングを有効にします。

Aurora MySQL データベースエンジンの更新 2018-06-05 (バージョン 1.17.3) (廃止)

(バージョン 1.17.3)

Aurora MySQL 1.17.3 は一般利用可能です。MySQL 5.6 との互換性を持つすべての新しい Aurora MySQL データベースクラスターは、スナップショットから復元されるものも含めて、Aurora MySQL 1.17.3 で作成されます。既存のデータベースクラスターは Aurora MySQL 1.17.3 にアップグレードできます。ただし必須ではありません。新しいデータベースクラスターは、Aurora MySQL 1.14.4、Aurora MySQL 1.15.1、または Aurora MySQL 1.16 で作成できます。このためには、AWS CLI または Amazon RDS API を使用して、エンジンバージョンを指定します。

Aurora MySQL のバージョン 1.17.3 では、クラスターパッチ適用モデルが使用されており、Aurora DB クラスターのすべてのノードに同時にパッチが適用されます。

> **Note**
>
> このバージョンは現在、AWS GovCloud (米国西部) [us-gov-west-1] リージョンおよび 中国 (北京) [cn-north-1] リージョンでは使用できません。ご利用可能になりましたら、別途お知らせします。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

- レコードの読み取り中にオプティミスティックカーソル復元を使用すると、Aurora レプリカが再起動することがある問題を修正しました。
- Performance Schemaが有効の状態で MySQL セッションをキルする (「*<session id>*」のキル) と、Aurora ライターが再起動する問題を修正しました。
- ガベージコレクションのしきい値を計算すると Aurora が再起動する問題を修正しました。
- Aurora ライターが Aurora レプリカの進行状況をログアプリケーションで追跡中に再起動することがある問題を修正しました。
- 自動コミットがオフの場合のクエリキャッシュで stale リードが発生する可能性がある問題を修正しました。

Aurora MySQL データベースエンジンの更新 2018-04-27 (バージョン 1.17.2) (廃止)

(バージョン 1.17.2)

Aurora MySQL 1.17.2 は一般利用可能です。MySQL 5.6 との互換性を持つすべての新しい Aurora MySQL データベースクラスターは、スナップショットから復元されるものも含めて、Aurora MySQL 1.17.2 で作成されます。既存のデータベースクラスターは Aurora MySQL 1.17.2 にアップグレードできます。ただし必須ではありません。新しいデータベースクラスターは、Aurora MySQL 1.14.4、Aurora MySQL 1.15.1、または Aurora MySQL 1.16 で作成できます。このためには、AWS CLI または Amazon RDS API を使用して、エンジンバージョンを指定します。

Aurora MySQL のバージョン 1.17.2 では、クラスターパッチ適用モデルが使用されており、Aurora DB クラスターのすべてのノードに同時にパッチが適用されます。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

- 特定の DDL パーティション操作中の再起動の原因となっていた問題を修正しました。
- ネイティブな Aurora MySQL 関数を介した AWS Lambda 関数呼び出しのサポートを無効にする原因となっていた問題を修正しました。
- Aurora レプリカでの再起動の原因となっていたキャッシュの無効化に関する問題を修正しました。
- 再起動の原因となっていたロックマネージャーにおける問題を修正しました。

Aurora MySQL データベースエンジンの更新 2018-03-23 (バージョン 1.17.1) (廃止)

(バージョン 1.17.1)

Aurora MySQL 1.17.1 は一般利用可能です。スナップショットから復元されたデータベースを含む、すべての新しいデータベースクラスターは、Aurora MySQL 1.17.1 で作成されます。既存のデータベースクラスターは Aurora MySQL 1.17.1 にアップグレードできます。ただし必須ではありません。新しい DB クラスターは、Aurora MySQL 1.15.1、Aurora MySQL 1.16、または Aurora MySQL 1.17 で作成することができます。このためには、AWS CLI または Amazon RDS API を使用して、エンジンバージョンを指定します。

Aurora MySQL のバージョン 1.17.1 では、クラスターパッチ適用モデルが使用されており、Aurora DB クラスターのすべてのノードに同時にパッチが適用されます。このリリースでは、既知の一部のエンジン問題やデグレードが修正されています。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

> **Note**
>
> Aurora MySQL エンジンの最新バージョンに問題があります。1.17.1 にアップグレードした後、エンジンのバージョンは 1.17 として間違って報告されます。1.17.1 にアップグレードした場合は、AWS Management Console の DB クラスターの [Maintenance (メンテナンス)] 列を確認してアップグレードを確認できます。none と表示されている場合、エンジンは 1.17.1 にアップグレードされています。

改良点

- バイナリログインデックスファイルのサイズが大きいと復旧時間が長くなるバイナリログ復旧の問題を修正しました。この状況はバイナリログが頻繁にローテーションする場合に発生する可能性があります。
- パーティショニングされたテーブルで非効率的なクエリプランを生成したクエリオプティマイザの問題を修正しました。
- データベースエンジンの再起動の原因となる範囲のクエリによるクエリオプティマイザの問題を修正しました。

Aurora MySQL データベースエンジンの更新 2018-03-13 (バージョン 1.17) (廃止)

(バージョン 1.17)

Aurora MySQL 1.17 は一般利用可能です。Aurora MySQL1.x バージョンは MySQL 5.6 とのみ互換性があり、MySQL 5.7 とは互換性がありません。すべての新しい 5.6 との互換性のあるデータベースクラスターは、スナップショットから復元されるものも含めて、Aurora 1.17 で作成されます。既存のデータベースクラスターは Aurora 1.17. にアップグレードできます。ただし必須ではありません。新しい DB クラスターは、Aurora 1.14.1、Aurora 1.15.1、または Aurora 1.16 で作成できます。作成するには、AWS CLI または Amazon RDS API を使用して、エンジンバージョンを指定します。

Aurora のバージョン 1.17 では、クラスターパッチ適用モデルが使用されており、Aurora DB クラスターのすべてのノードに同時にパッチが適用されます。ダウンタイムのないパッチ適用に対応しています。この方法では、パッチ適用プロセス中のクライアント接続を維持するために、ベストエフォートに基づいて動作します。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。

ダウンタイムのないパッチ適用

ダウンタイムのないパッチ適用 (ZDP) 機能では、ベストエフォートに基づいて、エンジンパッチ中のクライアント接続を維持するよう試みます。ZDP の詳細については、「Amazon Aurora ユーザーガイド」の「ダウンタイムのないパッチ適用の使用」を参照してください。

新機能

- Aurora MySQL では、ロックの圧縮がサポートされるようになりました。この機能により、ロックマネージャーのメモリ使用量が最適化されます。バージョン 1.17 からは、ラボモードを有効にせずにこの機能を使用できます。

改良点

- コア数が少ないインスタンスで主に見られる、データベースがアイドル状態であってもシングルコアの CPU 使用率が 100% になる問題を修正しました。
- Aurora クラスターからバイナリログを取得する際のパフォーマンスが向上しました。
- Aurora レプリカでテーブル統計が永続的ストレージに書き込まれ、クラッシュする問題を修正しました。
- Aurora レプリカでクエリキャッシュが想定どおりに機能しない問題を修正しました。
- エンジンの再起動の原因となるロックマネージャーの競合状態を修正しました。
- 読み取り専用の自動コミットトランザクションによるロックが原因でエンジンが再起動する問題を修正しました。
- 一部のクエリが監査ログに書き込まれない問題を修正しました。
- フェイルオーバー時の特定のパーティションメンテナンスオペレーションの復旧に関する問題を修正しました。

MySQL バグ修正の統合

- レプリケーションフィルタが使用されていると LAST_INSERT_ID が不正にレプリケートされる (バグ #69861)
- クエリは、INDEX_MERGE 設定の有無に応じて異なる結果を返す (バグ #16862316)

- ストアドルーチン、非効率なクエリプランのクエリ処理の再実行 (バグ #16346367)
- INNODB FTS : FTS_CACHE_APPEND_DELETED_DOC_IDS のアサート (バグ #18079671)
- ALTER TABLE CHANGE COLUMN の RBT_EMPTY(INDEX_CACHE->WORDS) のアサート (バグ #17536995)
- 保存ポイントが関わる場合に InnoDB 全文検索でレコードが見つからない (バグ #70333、バグ #17458835)

Aurora MySQL データベースエンジンの更新 2017-12-11 (バージョン 1.16) (廃止)

(バージョン 1.16)

Aurora MySQL 1.16 は一般利用可能です。スナップショットから復元されたデータベースを含む、すべての新しいデータベースクラスターは、Aurora 1.16 で作成されます。既存のデータベースクラスターは Aurora 1.16. にアップグレードできます。ただし必須ではありません。新しい DB クラスターは、 Aurora 1.14.1 または Aurora 1.15.1 で作成できます。このためには、AWS CLI または Amazon RDS API を使用して、エンジンバージョンを指定します。

Aurora のバージョン 1.16 では、クラスターパッチ適用モデルが使用されており、Aurora DB クラスターのすべてのノードに同時にパッチが適用されます。パッチ適用プロセス中のクライアント接続を維持するために、ベストエフォートに基づいて動作するダウンタイムのないパッチ適用を有効にしています。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。

ダウンタイムのないパッチ適用

ダウンタイムのないパッチ適用 (ZDP) 機能では、ベストエフォートに基づいて、エンジンパッチ中のクライアント接続を維持するよう試みます。ZDP の詳細については、「Amazon Aurora ユーザーガイド」の「ダウンタイムのないパッチ適用の使用」を参照してください。

新機能

- Aurora MySQL で、ネイティブ関数 `lambda_sync()` による AWS Lambda の同期呼び出しをサポートするようになりました。また、ネイティブ関数 `lambda_async()` もサポートするようにな

り、Lambda の非同期呼び出しのための既存のストアードプロシージャの代わりに使用できます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL DB クラスターからの Lambda 関数の呼び出し」を参照してください。

- Aurora MySQL で、等価結合クエリを高速化するためにハッシュ結合をサポートするようになりました。Aurora のコストベースのオプティマイザにより、いつハッシュ結合を使用するか自動的に決定し、クエリプランでその使用を強制できます。詳細については、「Amazon Aurora ユーザーガイド」の「ハッシュ結合を使用した大規模な Aurora MySQL 結合クエリの最適化」を参照してください。
- Aurora MySQL は、スキャンバッチ処理をサポートするようになり、インメモリ、スキャン指向のクエリが大幅に高速になりました。この機能を使用すると、バッチ処理によりテーブルフルスキャン、インデックスフルスキャン、インデックス範囲スキャンのパフォーマンスが向上します。

改良点

- マスターにドロップされた直後のテーブルでクエリを実行すると、リードレプリカがクラッシュする問題を修正しました。
- FULLTEXT インデックスが大量にあるデータベースクラスターでライターを再起動すると、復旧が想定よりも長くかかる問題を修正しました。
- バイナリログをフラッシュすると binlog イベントで LOST_EVENTS インシデントが発生する問題を修正しました。
- Performance Schemaが有効なときのスケジューラの安定性の問題を修正しました。
- テンポラリテーブルを使用するサブクエリが部分的な結果を返すことがある問題を修正しました。

MySQL バグ修正の統合

なし

Aurora MySQL データベースエンジンの更新 2017-11-20 (バージョン 1.15.1) (廃止)

バージョン: 1.15.1

Aurora MySQL 1.15.1 は一般利用可能です。スナップショットから復元されたデータベースを含む、すべての新しいデータベースクラスターは、Aurora 1.15.1 で作成されます。既存の DB クラスター

は Aurora 1.15.1 にアップグレードできます。ただし必須ではありません。新しい DB クラスターは、 Aurora 1.14.1 で作成できます。このためには、AWS CLI または Amazon RDS API を使用して、エンジンバージョンを指定します。

Aurora のバージョン 1.15.1 では、クラスターパッチ適用モデルが使用されており、Aurora DB クラスターのすべてのノードに同時にパッチが適用されます。パッチ適用プロセス中のクライアント接続を維持するために、ベストエフォートに基づいて動作するダウンタイムのないパッチ適用を有効にしています。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

ダウンタイムのないパッチ適用

ダウンタイムのないパッチ適用 (ZDP) 機能では、ベストエフォートに基づいて、エンジンパッチ中のクライアント接続を維持するよう試みます。ZDP の詳細については、「Amazon Aurora ユーザーガイド」の「ダウンタイムのないパッチ適用の使用」を参照してください。

改良点

- 適応型セグメントセレクタの読み取りリクエストの問題を修正しました。これにより、同じセグメントを 2 回選択しても、特定の条件下で読み取りレイテンシーが急に長くなることはなくなりました。
- スレッドスケジューラに合わせた Aurora MySQL の最適化に起因する問題を修正しました。この問題はスローログへの書き込み中の偽のエラーとして発生しますが、関連するクエリ自体は正常に実行されます。
- 大容量 (5 TB 超) のボリュームでのリードレプリカの安定性の問題を修正しました。
- 偽の未処理接続数のためにワーカースレッド数が増加し続ける問題を修正しました。
- 挿入のワークロード中に長いセマフォ待機になるテーブルロックの問題を修正しました。
- Aurora MySQL 1.15 に含まれている以下の MySQL バグ修正を元に戻しました。
  - MySQL インスタンスで停止される “doing SYNC index” (バグ #73816)
  - ALTER TABLE CHANGE COLUMN の RBT_EMPTY(INDEX_CACHE->WORDS) のアサート (バグ #17536995)
  - 保存ポイントが関わる場合に InnoDB 全文検索でレコードが見つからない (バグ #70333)

MySQL バグ修正の統合

なし

Aurora MySQL データベースエンジンの更新 2017-10-24 (バージョン 1.15) (廃止)

バージョン: 1.15

Aurora MySQL 1.15 は一般利用可能です。スナップショットから復元されたデータベースを含む、すべての新しいデータベースクラスターは、Aurora 1.15 で作成されます。既存の DB クラスターは Aurora 1.15 にアップグレードできます。ただし必須ではありません。新しい DB クラスターは、Aurora 1.14.1 で作成できます。このためには、AWS CLI または Amazon RDS API を使用して、エンジンバージョンを指定します。

Aurora のバージョン 1.15 では、クラスターパッチ適用モデルが使用されており、Aurora DB クラスターのすべてのノードに同時にパッチが適用されます。更新では、データベースを再起動する必要があるため、20～30 秒間のダウンタイムが発生します。その後、DB クラスターの使用を再開できます。DB クラスターで現在 Aurora 1.14 または Aurora 1.14.1, を実行している場合は、Aurora MySQL のダウンタイムゼロのパッチ機能により、Aurora MySQL プライマリインスタンスへのクライアント接続が、ワークロードに応じて、アップグレードが終わるまで維持されます。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

ダウンタイムのないパッチ適用

ダウンタイムのないパッチ適用 (ZDP) 機能では、ベストエフォートに基づいて、エンジンパッチ中のクライアント接続を維持するよう試みます。ZDP の詳細については、「Amazon Aurora ユーザーガイド」の「ダウンタイムのないパッチ適用の使用」を参照してください。

新機能

- Asynchronous Key Prefetch - Asynchronous key prefetch (AKP) は、キーを実際の必要時点よりも前にメモリにプリフェッチすることで、キャッシュされないインデックス結合のパフォーマンスを向上させることを目的とした機能です。AKP の対象となるプライマリユースケースは、小さい外部テーブルと (インデックスの選択性が高い) 大きい内部テーブルの間のインデックス結合です。また、Multi-Range Read (MRR) インターフェイスが有効になっている場合、AKP はセカンダリ

からプライマリへのインデックスのルックアップに利用されます。メモリに制限がある小さいインスタンスでは、キーのカーディナリティが正しい場合、状況に応じて AKP を利用できます。詳細については、「Amazon Aurora ユーザーガイド」の「Asynchronous Key Prefetch を使用した Aurora MySQL インデックス付き結合クエリの最適化」を参照してください。

- 高速 DDL - Aurora 1.13 でリリースされた機能が、デフォルト値を含むオペレーションに拡張されました。この拡張により、高速 DDL は、デフォルト値の有無にかかわらず、テーブルの末尾に null が許容される列を追加するオペレーションに適用可能です。この機能は Aurora ラボモードに残ります。詳細については、「Amazon Aurora ユーザーガイド」の「高速 DDL を使用して Amazon Aurora のテーブルを変更する」を参照してください。

改良点

- WITHIN/CONTAINS 空間クエリの最適化で空の結果セットが生成されるというエラーを修正しました。
- `SHOW VARIABLE` コマンドを修正し、パラメータグループで `innodb_buffer_pool_size` パラメータが変更されるたびに更新された値が表示されるようにしました。
- 適応型ハッシュインデックスが無効で、挿入するレコードがページの初期のレコードであるときに、高速 DDL を使用して変更したテーブルへの一括挿入時のプライマリインスタンスの安定性を改善しました。
- ユーザーが [server_audit_events] DB クラスターパラメータ値を **`default`** に設定を試みて Aurora の安定性が改善されました。
- Aurora プライマリインスタンスで実行した ALTER TABLE ステートメントのデータベース文字セットの変更が、Aurora レプリカが再起動されるまでレプリケートされないという問題を修正しました。
- プライマリインスタンスでの競合状態を修正して安定性を改善しました。以前は、プライマリインスタンス自体のボリュームがクローズされた場合でも Aurora レプリカを登録できました。
- ロックプロトコルを変更してインデックス構築中のデータ操作言語 (DML) ステートメントの同時実行を可能にすることで、大きいテーブルでのインデックス作成時のプライマリインスタンスのパフォーマンスを改善しました。
- ALTER TABLE RENAME クエリ時の InnoDB メタデータの不整合を修正して安定性を改善しました。例: 同じ ALTER ステートメント内でテーブル t1(c1, c2) の列名を周期的に t1(c2, c3) に変更する場合。
- Aurora レプリカにアクティブなワークロードがなく、プライマリインスタンスが応答しない場合のシナリオの Aurora レプリカの安定性を改善しました。

- Aurora レプリカがテーブルの明示的なロックを保持し、レプリケーションスレッドがプライマリインスタンスから受け取った DDL の変更を適用することをブロックする場合のシナリオの Aurora レプリカの可用性を改善しました。
- 外部キーおよび列が 2 つの個別のセッションから同時にテーブルに追加されるときに高速 DDL が有効になっている場合のプライマリインスタンスの安定性を改善しました。
- 元に戻すレコードの切り捨てをレコードがパージされるまでブロックすることで、高負荷の書き込みワークロード中のプライマリインスタンスでのパージスレッドの安定性を改善しました。
- テーブルを削除するトランザクションのコミットプロセス時のロックリリース順を修正することで安定性を改善しました。
- DB インスタンスで起動を完了できず、ポート 3306 が使用中であると報告される Aurora レプリカの不具合を修正しました。
- 特定の information_schema テーブル (innodb_trx、innodb_lock、innodb_lock_waits) で実行された SELECT クエリによってクラスターの不安定性が増すという競合状態を修正しました。

MySQL バグ修正の統合

- CREATE USER はプラグインおよびパスワードハッシュを受け入れるが、パスワードハッシュを無視する (バグ #78033)
- パーティションエンジンは、読み取りビットのセットにフィールドを追加し、パーティションインデックスからソートされたエントリを返せるようにします。これにより、結合バッファでは不要なフィールドまで読み取ろうとします。すべてのパーティションフィールドを read_set に追加するのではなく、read_set の設定済みのプレフィックスフィールドでのみソートするように修正しました。DBUG_ASSERT を追加し、key_cmp を行う場合、少なくとも初期のフィールドが必ず読み取られるようにしました (バグ #16367691)
- MySQL インスタンスで停止される “doing SYNC index” (バグ #73816)
- ALTER TABLE CHANGE COLUMN の RBT_EMPTY(INDEX_CACHE->WORDS) のアサート (バグ #17536995)
- 保存ポイントが関わる場合に InnoDB 全文検索でレコードが見つからない (バグ #70333)

Aurora MySQL データベースエンジンの更新: 2018-03-13 (バージョン 1.14.4) (廃止)

バージョン: 1.14.4

Aurora MySQL 1.14.4 は一般利用可能です。新しい DB クラスターを Aurora 1.14.4 で作成することが可能です。その際は、AWS CLI または Amazon RDS API を使用してエンジンバージョンを指定します。既存の 1.14.x DB データベースクラスターは Aurora 1.14.4 にアップグレードできます。ただし必須ではありません。

Aurora のバージョン 1.14.4 では、クラスターパッチ適用モデルが使用されており、Aurora DB クラスターのすべてのノードに同時にパッチが適用されます。ダウンタイムのないパッチ適用に対応しています。この方法では、パッチ適用プロセス中のクライアント接続を維持するために、ベストエフォートに基づいて動作します。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

ダウンタイムのないパッチ適用

ダウンタイムのないパッチ適用 (ZDP) 機能では、ベストエフォートに基づいて、エンジンパッチ中のクライアント接続を維持するよう試みます。ZDP の詳細については、「Amazon Aurora ユーザーガイド」の「ダウンタイムのないパッチ適用の使用」を参照してください。

新機能

- Aurora MySQL で、db.r4 インスタンスクラスがサポートされるようになりました。

改良点

- 大きなバイナリログイベントを書き込む際に `LOST_EVENTS` が生成される問題を修正しました。

MySQL バグ修正の統合

- 無視できるイベントが動作せず、テストされない (バグ #74683)
- NEW->OLD ASSERT FAILURE 'GTID_MODE > 0' (バグ #20436436)

Aurora MySQL データベースエンジンの更新: 2017-09-22 (バージョン 1.14.1) (廃止)

バージョン: 1.14.1

Aurora MySQL 1.14.1 は一般利用可能です。スナップショットから復元されたデータベースを含む、すべての新しいデータベースクラスターは、Aurora MySQL 1.14.1 で作成されます。Aurora MySQL 1.14.1 は、また、既存の Aurora MySQL DB クラスターの必須アップグレードです。詳細については、AWS デベロッパーフォーラムのウェブサイトでお知らせ: Amazon Aurora の必須アップグレードスケジュールの延長を参照してください。

Aurora MySQL のバージョン 1.14.1 では、クラスターパッチ適用モデルが使用されており、Aurora MySQL DB クラスターのすべてのノードに同時にパッチが適用されます。更新では、データベースを再起動する必要があるため、20～30 秒間のダウンタイムが発生します。その後、DB クラスターの使用を再開できます。DB クラスターが現在バージョン 1.13 以上を実行している場合、Aurora MySQL のダウンタイムゼロのパッチ機能により、Aurora MySQL プライマリインスタンスへのクライアント接続が、ワークロードに応じて、アップグレードが終わるまで維持されます。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。

改良点

- 挿入とパージに関連する競合条件を修正して、高速 DDL 機能の安定性を向上させました。この機能は、Aurora MySQL ラボモードに残ります。

Aurora MySQL データベースエンジンの更新: 2017-08-07 (バージョン 1.14) (廃止)

バージョン: 1.14

Aurora MySQL 1.14 は一般利用可能です。スナップショットから復元されたデータベースを含む、すべての新しいデータベースクラスターは、Aurora MySQL 1.14 で作成されます。Aurora MySQL 1.14 は、また、既存の Aurora MySQL DB クラスターの必須アップグレードです。Aurora MySQL の以前のバージョンを廃止するためのタイムラインについては、別途お知らせします。

Aurora MySQL のバージョン 1.14 では、クラスターパッチ適用モデルが使用されており、Aurora DB クラスターのすべてのノードに同時にパッチが適用されます。更新では、データベースを再起

動する必要があるため、20 ~ 30 秒間のダウンタイムが発生します。その後、DB クラスターの使用を再開できます。DB クラスターが現在バージョン 1.13 を実行している場合、Aurora のダウンタイムゼロのパッチ機能により、Aurora プライマリインスタンスへのクライアント接続が可能になり、ワークロードに応じて、アップグレードを継続します。

ご質問やご不明点がございましたら、コミュニティフォーラムや AWS サポートから AWS サポートにお問い合わせください。

ダウンタイムのないパッチ適用

ダウンタイムのないパッチ適用 (ZDP) 機能では、ベストエフォートに基づいて、エンジンパッチ中のクライアント接続を維持するよう試みます。ZDP の詳細については、「Amazon Aurora ユーザーガイド」の「ダウンタイムのないパッチ適用の使用」を参照してください。

改良点

- プライマリインデックスではなくセカンダリインデックスでレコードが見つかった場合、誤った「レコードが見つかりません」というエラーが修正されました。
- 個々の書き込みが 32 ページを超えている場合に、強すぎる防御アサーション (1.12 で追加) のために発生する可能性のある安定性の問題を修正しました。このような状況は、例えば BLOB 値が大きい場合などに発生します。
- テーブルスペースキャッシュとディクショナリキャッシュ間の不整合による安定性の問題が修正されました。
- プライマリインスタンスへの最大接続試行回数を超えた後、Aurora レプリカが応答しなくなっていた問題が修正されました。アイドル期間がプライマリインスタンスによるヘルスチェックに使用されるハートビート時間よりも長い場合、Aurora レプリカが再起動します。
- 1 つの接続が `ALTER TABLE` などのコマンドを発行中に排他的なメタデータロック (MDL) を取得しようとするときに非常に高い同時実行性で発生する可能性のあるライブロックを修正しました。
- 論理/パラレル先読みがある場合の Aurora リードレプリカの安定性の問題を修正しました。
- 2 つの方法での `LOAD FROM S3` の向上。
  1. 既存の再試行に加えて SDK の再試行を使用した Amazon S3 タイムアウトエラーの処理の向上。
  2. クライアントの状態をキャッシュして再利用することによる、非常に大きなファイルや多数のファイルをロードするときのパフォーマンスの最適化。
- `ALTER TABLE` オペレーションに対する高速 DDL の安定性の次の問題を修正しました。

1. ALTER TABLE ステートメントに複数の ADD COLUMN コマンドがあり、列名が昇順でない場合。
2. 更新される列の名前文字列と内部システムテーブルから取得される対応する名前文字列が NULL 終了文字 (/0) によって異なる場合。
3. 特定の B-tree 分割操作の下。
4. テーブルに可変長のプライマリキーがある場合。

- 全文検索 (FTS) インデックスキャッシュをプライマリインスタンスのものを一貫させるのに時間がかかりすぎる場合の Aurora レプリカの安定性の問題を修正しました。これは、プライマリインスタンス上の新しく作成された FTS インデックスエントリの大部分がまだディスクにフラッシュされていない場合に発生する可能性があります。
- インデックス作成中に発生する可能性がある安定性の問題を修正しました。
- メモリー不足 (OOM) 回避戦略を構築するために使用される接続および関連するテレメトリごとのメモリ消費を追跡する新しいインフラストラクチャ。
- Aurora レプリカで ANALYZE TABLE が誤って許可されていた問題を修正しました。これは現在ブロックされています。
- 論理的な先読みとパージ間の競合状態のために、まれなデッドロックが原因で発生した安定性の問題を修正しました。

MySQL バグ修正の統合

- 派生テーブル (FROM 句のサブクエリ) と結合された全文検索では、サーバーが終了しました。ここで、全文操作が派生テーブルに依存する場合、サーバーは、マテリアライズされたテーブルで全文検索を実行できないことを示すエラーを生成します。(バグ #68751、バグ #16539903)

Aurora MySQL データベースエンジンの更新: 2017-05-15 (バージョン 1.13) (廃止)

バージョン: 1.13

> **Note**
>
> Aurora MySQL バージョン 1.13 初回リリースの後、新しい機能 SELECT INTO OUTFILE S3 が有効になりました。この変更を反映するためにリリースノートを更新しました。

Aurora MySQL 1.13 は一般利用可能です。スナップショットから復元されたデータベースを含む、すべての新しいデータベースクラスターは、Aurora MySQL 1.13 で作成されます。既存のデータベースクラスターは Aurora MySQL 1.13 にアップグレードできます。ただし必須ではありません。Aurora のバージョン 1.13 では、クラスターパッチ適用モデルが使用されており、Aurora DB クラスターのすべてのノードに同時にパッチが適用されます。パッチ適用プロセス中のクライアント接続を維持するために、ベストエフォートに基づいて動作するダウンタイムのないパッチ適用を有効にしています。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

ダウンタイムのないパッチ適用

ダウンタイムのないパッチ適用 (ZDP) 機能では、ベストエフォートに基づいて、エンジンパッチ中のクライアント接続を維持するよう試みます。ZDP の詳細については、「Amazon Aurora ユーザーガイド」の「ダウンタイムのないパッチ適用の使用」を参照してください。

新機能:

- SELECT INTO OUTFILE S3 - Aurora MySQL では、Amazon S3 バケットの 1 つ以上のファイルにクエリ結果をアップロードできるようになりました。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL DB クラスターから Amazon S3 バケット内のテキストファイルへのデータの保存」を参照してください。

改良点:

- 復元時間が長引かないようにエンジン起動時の CSV 形式のログファイルを切り捨てるようにしました。`general_log_backup`、`general_log`、`slow_log_backup`、および `slow_log` の各テーブルは、データベースの再起動後に保持されなくなりました。
- **`test`** という名前のデータベースの移行が失敗する問題を修正しました。
- 正しいロックセグメントを再利用することで、ロックマネージャーのガベージコレクターの安定性を改善しました。
- デッドロック検出アルゴリズムでの無効なアサーションを削除することで、ロックマネージャーの安定性を改善しました。
- 非同期レプリケーションを再有効化し、負荷のないワークロードまたは読み取り専用ワークロードで不正なレプリカラグがレポートされる関連問題を修正しました。バージョン 1.10 で導入されたレプリケーションパイプラインの改善。これらの改善は、Aurora レプリカのバッファキャッ

シュにログストリーミング更新を適用するために導入されました。Aurora レプリカの読み取りパフォーマンスと安定性を向上させるのに役立ちます。

- autocommit=OFF により、スケジュールされたイベントがブロックされ、サーバーが再起動するまで長いトランザクションが開いたままになる問題を修正しました。
- 非同期コミットで処理されたクエリを一般クエリログ、監査クエリログ、およびスロークエリログで記録できない問題を修正しました。
- 論理的な先読み (LRA) 機能のパフォーマンスを最大 2.5 倍まで改善しました。これは B ツリーの中間ページ全体でプリフェッチを継続して行うことを許可することで実現しました。
- 不要なスペースをトリミングするために監査可変にパラメータ検証を追加しました。
- Aurora MySQL バージョン 1.11 で導入された回帰を修正しました。以前は、SQL_CALC_FOUND_ROWS オプションを使用して FOUND_ROWS() 関数を呼び出したときにクエリから不正な結果が返される場合がありました。
- メタデータのロックリストの不正な構成で生じる安定性の問題を修正しました。
- sql_mode を PAD_CHAR_TO_FULL_LENGTH に設定してコマンド `SHOW FUNCTION STATUS WHERE Db='string'` を実行した場合の安定性を改善しました。
- ボリュームの整合性チェックエラーにより、Aurora バージョンのアップグレード後にインスタンスが表示されないという、まれなケースを修正しました。
- ユーザーが多数のテーブルを持っているときに Aurora ライターのパフォーマンスが軽減されるという、Aurora MySQL バージョン 1.12 で生じた問題を修正しました。
- Aurora ライターをバイナリログワーカーとして設定し、接続数が 16,000 に近づいたときの安定性の問題を改善しました。
- Aurora マスターで DDL を実行し、メタデータのロックを待っている間に接続がブロックされると、Aurora レプリカが再起動するという、まれな問題を修正しました。

MySQL バグ修正の統合

- 空の InnoDB テーブルでは、テーブルが空であっても、ALTER TABLE ステートメントを使用して auto_increment 値を減らすことはできません。(バグ #69882)
- 一致 () ..。AGAINST() の引数として長い文字列を使用する AGAINST クエリは、全文検索インデックスを持つ InnoDB テーブルで実行すると エラーになる可能性があります。(バグ #17640261)
- SQL_CALC_FOUND_ROWS を ORDER BY および LIMIT と組み合わせて処理すると、FOUND_ROWS() の結果が不正になる場合あります。(バグ #68458、バグ #16383173)
- 外部キーがある場合、ALTER TABLE では列の NULL 値の変更が許可されません。(バグ #77591)

Aurora MySQL データベースエンジンの更新: 2017-04-05 (バージョン 1.12) (廃止)

バージョン: 1.12

新しい DB クラスターの作成 (スナップショットからの復元を含む) の推奨バージョンは、Aurora MySQL 1.12 になりました。

既存のクラスターのアップグレードについては、強制ではありません。フリート全体のパッチを 1.11 に完了した後、既存のクラスターをバージョン 1.12 にアップグレードするオプションがあります (Aurora 1.11 リリースノートおよび対応する フォーラムでのお知らせを参照)。Aurora のバージョン 1.12 では、クラスターパッチ適用モデルが使用されており、Aurora DB クラスターのすべてのノードに同時にパッチが適用されます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

新機能

- 高速 DDL - Aurora MySQL では、ALTER TABLE tbl_name ADD COLUMN col_name column_definition オペレーションをほぼ即座に実行できるようになりました。このオペレーションでは、テーブルをコピーする必要はありません。他の DML ステートメントに実質的な影響を及ぼすこともありません。テーブルのコピーにはテンポラリストレージを使用しないため、スモールインスタンスクラスの大きなテーブルに対しても、DDL ステートメントが現実的になります。高速 DDL は現在、デフォルトの値を持たない、null が許容される列を、表の末尾に追加する際にのみ使用できます。この機能は現在、Aurora ラボモードで使用できます。詳細については、「Amazon Aurora ユーザーガイド」の「高速 DDL を使用して Amazon Aurora のテーブルを変更する」を参照してください。
- ボリュームステータスを表示 - 新しいモニタリングコマンド、SHOW VOLUME STATUS が追加されました。このコマンドを使用して、ボリューム内のノードやディスクの数を表示できます。詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL DB クラスターのボリュームステータスの表示」を参照してください。

改良点

- ロックオブジェクトごとに割り当てられたメモリをさらに削減するために、圧縮をロックするよう変更が導入されました。この実装はラボモードで使用できます。
- データベースがアイドル状態のときでも、`trx_active_transactions` メトリクスが急速に減少する問題を修正しました。

- ディスクおよびノードの障害をシミュレートする際の、障害挿入クエリ構文に関する無効なエラーメッセージを修正しました。
- ロックマネージャーの競合状態やデッドラッチに関する複数の問題を修正しました。
- クエリオプティマイザでバッファのオーバーフローを起こす問題を修正しました。
- 基礎となるストレージノードで使用可能なメモリが不足している場合の Aurora リードレプリカの安定性の問題を修正しました。
- アイドル接続が `wait_timeout` パラメータ設定を超えて持続する問題を修正しました。
- インスタンスの再起動後、`query_cache_size` が予期せぬ値を返す問題を修正しました。
- 書き込みがストレージに移行していない場合に、ネットワークが頻繁にプローブされる診断スレッドによるパフォーマンスの問題を修正しました。

MySQL バグ修正の統合

- 空のときに削除されたテーブルをリロードすると、AUTO INCREMENT 値がリセットされました。(バグ #21454472、バグ #77743)
- purge_node_t 構造の不一致により、ロールバック時にインデックスレコードが見つかりませんでした。この不一致により、「秒インデックスエントリの更新でエラーが発生しました」、「レコードをパージできませんでした」、「削除用にマークされていない秒インデックスエントリをパージしようとしました」などの警告やエラーメッセージが表示されました。(バグ #19138298、バグ #70214、バグ #21126772、バグ #21065746)
- qsort オペレーションのスタックサイズの計算が正しくないと、スタックのオーバーフローが発生します。(バグ #73979)
- ロールバック時にインデックスにレコードが見つかりませんでした。(バグ #70214、バグ #72419)
- CURRENT_TIMESTAMP 更新時に ALTER TABLE の列 TIMESTAMP を追加すると、ZERO-datas が挿入されます (バグ #17392)

Aurora MySQL データベースエンジンの更新: 2017-02-23 (バージョン 1.11) (廃止)

バージョン: 1.11

すべての Aurora MySQL DB クラスターには、リリース後まもなく、最新バージョンのパッチが適用されます。DB クラスターへのパッチの適用にはレガシー手順を使用され、適用中には約 5 ~ 30 秒のダウンタイムが発生します。

パッチの適用は、データベースインスタンスごとに指定したシステムメンテナンスウィンドウ中に行われます。このウィンドウは、AWS Management Console を使用して表示したり変更したりできます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

または、AWS Management Console ですぐにパッチを適用するには、DB クラスターを選択し、[Cluster Actions (クラスターアクション)] を選択してから [Upgrade Now (今すぐアップグレード)] を選択します 。

Aurora MySQL のバージョン 1.11 では、クラスターパッチ適用モデルが使用されており、Aurora DB クラスターのすべてのノードに同時にパッチが適用されます。

新機能

- LOAD DATA FROM S3 の MANIFEST オプション - LOAD DATA FROM S3 はバージョン 1.8 でリリースされました。このコマンドのオプションが拡張され、マニフェストファイルを使用して Amazon S3 からの Aurora DB クラスターにロードするファイルのリストを指定できるようになりました。これにより、1 つ以上の場所にある特定のファイルからのデータのロードが簡単になります。これは、FILE オプションを使用して 1 つのファイルからデータをロードしたり、PREFIX オプションを使用して場所およびプレフィックスが同じ複数のファイルからデータをロードしたりする方法とは対照的です。マニフェストファイルの形式は、Amazon Redshift で使用されるものと同じです。LOAD DATA FROM S3 で MANIFEST オプションを使用する方法については、「Amazon Aurora ユーザーガイド」の「マニフェストを使用して、ロードするデータファイルを指定する」を参照してください。
- 空間インデックスのデフォルトでの有効化 - この機能は、バージョン 1.10 のラボモードでリリースされ、現在、デフォルトで有効になっています。空間インデックスでは、空間的データを使用するクエリにおける大きなデータセットのクエリパフォーマンスが向上します。空間インデックスの使用については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL と空間データ」を参照してください。
- 高度な監査のタイミング精度の変更 - この機能は、データベースのアクティビティの高度な監査機能を提供するために、バージョン 1.10.1 でリリースされました。このリリースでは、監査ログのタイムスタンプの精度が 1 秒から 1 マイクロ秒に変更されました。より高い精度のタイムスタンプにより、監査イベントがいつ発生したかをより正確に把握できます。監査の詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL DB クラスターでのアドバンストな監査の使用」を参照してください。

改良点

- thread_handling パラメータを変更して、Aurora のスレッドプールでサポートされている唯一のモデルである **multiple-connections-per-thread** 以外のモデルに設定できないようにしました。
- buffer_pool_size または query_cache_size のいずれかのパラメータを DB クラスターの合計メモリより大きく設定した場合に発生する問題を修正しました。この状況では、Aurora は変更されたパラメータをデフォルト値に設定するため、DB クラスターは起動してクラッシュすることはありません。
- テーブルが別のトランザクションで無効化された場合にトランザクションが古い読み取り結果を取得するクエリキャッシュの問題を修正しました。
- 削除のマークが付けられたバイナリログファイルがすぐにではなく少し遅れて削除される問題を修正しました。
- **tmp** という名前で作成されたデータベースが、エフェメラルストレージに保存されるシステムデータベースとして処理され、Aurora 分散ストレージに保持されない問題を修正しました。
- SHOW TABLES の動作を変更して、特定の内部システムテーブルを対象から除外しました。この変更により、mysqldump が SHOW TABLES で一覧表示されるすべてのファイルをロックすることで発生する不要なフェイルオーバーが回避され、内部システムテーブルへの書き込みを禁止することでのみフェイルオーバーが発生します。
- 引数が InnoDB テーブルの列である関数を呼び出すクエリからテンポラリテーブルを作成すると、Aurora レプリカが誤って再起動する問題を修正しました。
- Aurora レプリカノードでのメタデータのロック競合により、Aurora レプリカがプライマリ DB クラスターで遅延して、最終的に再起動する問題を修正しました。
- リーダーノードのレプリケーションパイプラインのデッドラッチにより、Aurora レプリカが遅延して、最終的に再起動する問題を修正しました。
- Aurora レプリカが 1 テラバイト (TB) を超える暗号化ボリュームで遅延しすぎる問題を修正しました。
- システムクロック時間を読み取る方法を改良することで、Aurora レプリカのデッドラッチ検出を改良しました。
- Aurora レプリカがライターの登録解除後に 1 回ではなく 2 回再起動することがある問題を修正しました。
- 一時的な統計により一意でないインデックス列で統計の不一致が生じると Aurora レプリカでのクエリパフォーマンスが低下する問題を修正しました。

- Aurora レプリカで DDL ステートメントがレプリケートされているときに、同時に、Aurora レプリカで関連するクエリが処理されていると、Aurora レプリカがクラッシュすることがある問題を修正しました。
- バージョン 1.10 で導入されたレプリケーションパイプラインの改良点がデフォルトで有効にならずに無効になるように変更しました。これらの改良点は、Aurora レプリカのバッファキャッシュにログストリーミングの更新を適用するために導入されました。この機能により、Aurora レプリカの読み取りパフォーマンスと安定性が向上しますが、特定のワークロードでレプリカラグが長くなります。
- 同じテーブルで進行中の DDL ステートメントと保留中のパラレル先読みが同時にあると、DDL トランザクションのコミットフェーズ中にアサーションエラーが発生する問題を修正しました。
- 全般ログとスロークエリログを拡張して、DB クラスターの再起動後にも保持されるようにしました。
- ACL モジュールのメモリ消費量を減らすことで、特定の長時間実行されるクエリのメモリ不足の問題を修正しました。
- テーブルに非空間インデックスがあり、クエリに空間述語があり、プランナーが非空間インデックスの使用を選択したが、誤って空間条件をインデックスにプッシュしたときに発生する、再起動の問題を修正しました。
- 外部 (LOB など) に保存された巨大な地理空間オブジェクトの削除、更新、またはパージがあると DB クラスターが再起動する問題を修正しました。
- ALTER SYSTEM SIMULATE を使用した障害シミュレーションが正常に動作しない問題を修正しました。FOR INTERVAL を使用した障害シミュレーションが正しく動作しない問題を解決しました。
- ロックマネージャーの誤った不変条件の無効なアサーションにより発生する安定性の問題を修正しました。
- バージョン 1.10 で導入された InnoDB 全文検索の以下の 2 つの改良点を無効にしました。これは、厳しいワークロードにより安定性の問題が発生したためです。
  - 全文検索インデックスキャッシュのレプリケート速度を向上させるために、Aurora レプリカへの読み取りリクエスト後にのみキャッシュを更新する。
  - FTS キャッシュのディスクへの同期中に MySQL のクエリが長時間停止することを回避するために、キャッシュのサイズが合計サイズの 10% を超えたらすぐに、別のスレッドにキャッシュ同期タスクをオフロードする。(バグ #22516559、#73816)

MySQL バグ修正の統合

- 別の DROP オペレーションと同時に ALTER テーブルの DROP 外部キーを実行すると、テーブルがなくなる。(バグ #16095573)
- ORDER BY を使用した一部の INFORMATION_SCHEMA クエリで、以前と同じように filesort 最適化が使用されない。(バグ #16423536)
- FOUND_ROWS () によって返されるテーブルの行数が正しくない。(バグ #68458)
- 開いているテンポラリテーブルが多すぎると、エラーが発生する代わりに、サーバーに障害が発生する。(バグ #18948649)

Aurora MySQL データベースエンジンの更新: 2017-01-12 (バージョン 1.10.1) (廃止)

バージョン: 1.10.1

Aurora MySQL のバージョン 1.10.1 は任意のバージョンであるため、データベースインスタンスへのパッチ適用には使用されません。新しい Aurora インスタンスの作成および既存のインスタンスのアップグレードに使用できます。Amazon RDS コンソールでクラスターを選択し、[クラスターアクション]、[今すぐアップグレード] の順に選択することで、パッチを適用できます。パッチを適用するには、データベースを再起動する必要があり、ダウンタイムが通常 5 ~ 30 秒続きますが、その後 DB クラスターを使用して再開できます。このパッチは、Aurora クラスター内のすべてのノードが同時にパッチ適用されるクラスターパッチ適用モデルを使用しています。

新機能

- 高度な監査 - Aurora MySQL には、データベースアクティビティの監査に使用できる、高パフォーマンスの高度な監査機能が用意されています。高度な監査の有効化と使用の詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL DB クラスターでのアドバンストな監査の使用」を参照してください。

改良点

- 同じステートメントで列を作成してインデックスを追加するときの、空間インデックスの問題を修正しました。
- DB クラスターの再起動時に空間統計が保持されない問題を修正しました。

Aurora MySQL データベースエンジンの更新: 2016-12-14 (バージョン 1.10) (廃止)

バージョン: 1.10

新機能

- ダウンタイムのないパッチ - この機能は、DB インスタンスがダウンタイムなしにパッチ適用できるようにします。つまり、データベースのアップグレードは、クライアントアプリケーションの切断、またはデータベースの再起動をせずに実行されます。このアプローチにより、メンテナンス期間中の Aurora DB クラスターの可用性が向上します。Performance Schemaのデータなどのテンポラリデータはアップグレードプロセス中にリセットされますのでご注意ください。この機能は、メンテナンス期間中のサービス提供パッチ、およびユーザーがスタートしたパッチに適用されます。

  パッチがスタートされると、サービスは、オープンなロック、トランザクションまたはテンポラリテーブルがないことを確認し、データベースにパッチを適用し再起動できる適切なタイミングを待ちます。パッチの進行中、スループットが一時的に (5 秒ほど) 低下しますが、アプリケーションセッションは保持されます。適切な時間枠が利用できない場合、パッチ適用はスタンダードのパッチ動作に戻ります。

  ダウンタイムなしのパッチはベストエフォート型で実行され、以下で説明するようないくつかの制約があります。

  - 現在この機能は、単一ノード DB クラスター、または、複数ノード DB クラスターの書き込みインスタンスへのパッチに適用されます。
  - この機能で SSL 接続はサポートされていません。アクティブな SSL 接続がある場合、Amazon Aurora MySQL はダウンタイムなしのパッチを実行せず、代わりに SSL 接続が終了したかどうかを確認するため定期的に再試行します。停止している場合、ダウンタイムなしのパッチが実行されます。SSL 接続が数秒以上続く場合、ダウンタイムのあるスタンダードのパッチが実行されます。
  - この機能は Aurora リリース 1.10 以降で利用できます。今後、ダウンタイムなしのパッチ適用の機能を使用して適用することができないリリースまたはパッチを確認していきます。
  - バイナリログに基づくレプリケーションが有効の場合、この機能は使用できません。
- 空間インデックス - 空間インデックスでは、空間的データを使用するクエリにおける大きなデータセットのクエリパフォーマンスが向上します。空間インデックスの使用については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL と空間データ」を参照してください。

この機能はデフォルトでは無効になっており、Aurora のラボモードを有効化すると有効になります。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL ラボモード」を参照してください。

- レプリケーションのパイプラインの改良 - Aurora MySQL では、Aurora レプリカのバッファキャッシュにログストリーミングの更新を適用するためのメカニズムが改善されました。この機能は、マスターでの書き込み負荷が多いとき、または Aurora レプリカでの読み取り負荷が大きいときに、レプリカの読み取りパフォーマンスを向上させます。この機能は、デフォルトでご利用になれます。
- キャッシュ読み取りのワークロードにおけるスループットの向上 - Aurora MySQL では、読み取りビューの実行にラッチのない同時アルゴリズムを使用することで、バッファキャッシュから提供された読み取りクエリのスループットが向上しています。このような改善の結果、1 秒あたりの読み取り数が、MySQL 5.7 では SysBench の SELECT-only のワークロードで 164 K のスループットであるのに対し、Amazon Aurora MySQL では最大 625 K のスループットを実現できます。
- ホット行の競合があるワークロードにおけるスループットの向上 - Aurora MySQL は新しいロックリリースアルゴリズムを使用し、特にホットページの競合がある場合 (同じページの列で多くのトランザクションが競合している場合) のパフォーマンスが向上しています。TPC-C ベンチマークのテストでは、1 分あたりのスループットが MySQL 5.7 に比べ 16 倍に向上しました。この機能はデフォルトでは無効になっており、Aurora のラボモードを有効化すると有効になります。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL ラボモード」を参照してください。

改良点

- Aurora レプリカへの読み取りリクエストの後にキャッシュを更新することで、全文検索インデックスキャッシュレプリケーションが高速化されました。この方法により、レプリケーションスレッドがディスクから読み込むことを回避します。
- データベース名またはテーブル名に特殊文字があるテーブルで Aurora レプリカがディクショナリキャッシュを無効化できない問題を修正しました。
- ストレージの温度管理が有効になっている場合の分散ストレージノードのデータ移行中に発生する `STUCK IO` 問題を修正しました。
- トランザクションのロールバックまたはコミットの準備中にトランザクションロック待機スレッドのアサーションチェックが失敗するロックマネージャーの問題を修正しました。
- ディクショナリテーブルのエントリで正しくリファレンス数を更新することで、破損ディクショナリテーブルを開くときの問題を修正しました。

- DB クラスターの最小読み取りポイントが低速な Aurora レプリカによって保持されるバグを修正しました。
- クエリキャッシュで発生する可能性のあるメモリリークを修正しました。
- クエリがストアドプロシージャの IF ステートメントで使用されると、Aurora レプリカがテーブルに列単位のロックをするバグを修正しました。

MySQL バグ修正の統合

- 派生テーブルの UNION は、「1=0/false」句のある不正な結果を返します。(バグ #69471)
- サーバーはストアドプロシージャの 2 回目の実行の際、ITEM_FUNC_GROUP_CONCAT::FIX_FIELDS でクラッシュします。(バグ #20755389)
- キャッシュのサイズが合計サイズの 10% を超えたらすぐに、別のスレッドにキャッシュ同期タスクをオフロードすることで、FTS キャッシュのディスクへの同期中に MySQL のクエリが長時間停止することを回避できます。(バグ #22516559、#73816)

Aurora MySQL データベースエンジンの更新: 2016-11-10 (バージョン 1.9.0、1.9.1) (廃止)

バージョン: 1.9.0、1.9.1

新機能

- インデックス作成の改良 - セカンダリインデックスの作成の実装は、ボトムアップ式にインデックスを作成するようになり、不要なページ分割がなくなります。これにより、インデックスの作成またはテーブルを再構築に必要な時間を最大 75% 短縮できます (db.r3.8xlarge DB インスタンスクラスに基づく)。この機能は、Aurora MySQL バージョン 1.7 のラボモードのもので、Aurora バージョン 1.9 以降ではデフォルトで有効化されています。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL ラボモード」を参照してください。
- ロックの圧縮 (ラボモード) - この実装により、ロックマネージャーが使用するメモリの量が大幅に (最大 66%) 減少します。ロックマネージャーは、メモリ不足例外に遭遇せずに、より多くの行ロックを取得できます。この機能はデフォルトでは無効になっており、Aurora のラボモードを有効化すると有効になります。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL ラボモード」を参照してください。

- Performance Schema - Aurora MySQL でパフォーマンスへの影響を最小限に抑えるPerformance Schemaがサポートされました。SysBench を使用したテストでは、Performance Schemaを有効にすると、MySQL のパフォーマンスが最大 60% 低下しました。

  Aurora DB クラスターの SysBench テストでは、MySQL に比べてパフォーマンスへの影響が 4 倍少ないという結果が出ました。`db.r3.8xlarge` DB インスタンスのクラスを実行すると、Performance Schemaを有効にした状態で、100K の SQL 書き込み / 秒と 550K 以上の読み取り / 秒が発生しました。
- ホット行競合の改善 - この機能により CPU 使用率が低下し、少数のホット行に大量の接続がアクセスした場合のスループットが向上します。この機能は、ホット行の競合がある場合の `error 188` を減らします。
- メモリ不足処理の改善 - 必須でないロック SQL ステートメントを実行し、その予約メモリプールが不足した場合、Aurora は SQL ステートメントを強制的にロールバックします。この機能は空きメモリを増やし、メモリ不足例外によるエンジンのクラッシュを防止します。
- スマートリードセレクタ - この実装により、すべての読み取りのさまざまなセグメントの間で最適なストレージセグメントを選択することで、読み取りのレイテンシーが改善され、読み取りのスループットが向上します。SysBench テストでは、書き込みワークロードで 27% のパフォーマンスの向上が示されました 。

改良点

- Aurora レプリカでエンジンの起動時に共有ロックが発生する問題を修正しました。
- パージシステムの読み取りビューポインタが NULL の場合に、Aurora レプリカがクラッシュする可能性があるのを修正しました。

Aurora MySQL データベースエンジンの更新: 2016-10-26 (バージョン 1.8.1) (廃止)

バージョン: 1.8.1

改良点

- AWS Lambda プロシージャを起動するトリガーを使用する一括挿入が失敗する問題を修正しました。
- 自動コミットがグローバルに OFF の場合にカタログの移行が失敗する問題を修正しました。

- SSL と改善された Diffie-Hellman グループを使用して LogJam 攻撃に対処するときに、Aurora への接続が失敗する問題を解決しました。

MySQL バグ修正の統合

- OpenSSL は、LogJam の問題により Diffie-Hellman のキー長パラメータを変更しました。(バグ #18367167)

Aurora MySQL データベースエンジンの更新: 2016-10-18 (バージョン 1.8) (廃止)

バージョン: 1.8

新機能

- AWS Lambda の統合 - `mysql.lambda_async` プロシージャを使用して Aurora DB DB クラスターから AWS Lambda 関数を非同期的に呼び出すことができるようになりました。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL DB クラスターからの Lambda 関数の呼び出し」を参照してください。
- Amazon S3 からのデータのロード - `LOAD DATA FROM S3` コマンドまたは `LOAD XML FROM S3` コマンドを使用して、Amazon S3 バケットからテキストファイルや XML ファイルを Aurora DB クラスター内にロードできるようになりました。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon S3 バケットのテキストファイルから Amazon Aurora MySQL DB クラスターへのデータのロード」を参照してください。
- カタログの移行 - Aurora では、バージョニングをサポートするためにクラスターボリュームにカタログメタデータが保持されるようになりました。これにより、バージョン間や復元間でカタログをシームレスに移行できます。
- クラスターレベルのメンテナンスとパッチ適用 - Aurora で DB クラスター全体のメンテナンス更新が管理されるようになりました。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora DB クラスターのメンテナンス」を参照してください。

改良点

- 転送中の DDL テーブルにメタデータのロックが許可されない場合に、Aurora レプリカがクラッシュする問題を修正しました。

- `log_output=TABLE` である場合に、Aurora レプリカから InnoDB 以外のテーブルを変更し、スローログと一般ログの CSV ファイルのローテーションを行うことを許可しました。
- プライマリインスタンスから Aurora レプリカに統計を更新するときの遅延を修正しました。この修正を適用しない場合、Aurora レプリカの統計はプライマリインスタンスの統計と同期しなくなり、Aurora レプリカのクエリプランは別の (おそらくはパフォーマンスが低い) ものになります。
- Aurora レプリカがロックを取得しないようにする競合状態を修正しました。
- Aurora レプリカをプライマリインスタンスに登録または登録解除するときに、まれに失敗する状況を修正しました。
- ボリュームを開くか閉じるときに `db.r3.large` インスタンスのデッドロックにつながる可能性のある競合状態を修正しました。
- 大量の書き込みワークロードと Aurora Distributed Storage サービスでの障害が重なったときに、メモリ不足になる問題を修正しました。
- 実行時間の長いトランザクションがあると消去スレッドがスピンして CPU の使用率が高くなる問題を修正しました。
- 大きな負荷がかかっているときに、情報スキーマのクエリを実行してロックに関する情報を入手する際に発生する問題を修正しました。
- Aurora からストレージノードへの書き込みが、まれに停止して再起動/ファイルオーバーする場合がある診断プロセスの問題を修正しました。
- `CREATE TABLE [if not exists]` ステートメントの処理中にクラッシュが発生した場合、クラッシュ回復中に、正常に作成されたテーブルが削除される場合がある状況を修正しました。
- カタログの移行を使用して一般ログとスローログをディスクに保存していない場合、ログのローテーション手順が破綻するケースを修正しました。
- ユーザー定義関数内にテンポラリテーブルを作成した後で、そのユーザー定義関数をクエリの SELECT リストで使用したときに発生するクラッシュを修正しました。
- GTID イベントの再生時に発生するクラッシュを修正しました。GTID は Aurora MySQL でサポートされません。

MySQL バグ修正の統合:

- 複数のインデックスがある列ですべてのインデックスを削除する場合、外部キーの制約に基づいてインデックスが必要なときに、InnoDB は DROP INDEX オペレーションをブロックできませんでした。(バグ #16896810)
- 外部キーの制約に伴うクラッシュの解決策を追加しました。(バグ #16413976)

- ストアドプロシージャでカーソルを取得し、同時にテーブルを分析またはフラッシュするときに発生するクラッシュを修正しました。(バグ #18158639)
- テーブルを変更して AUTO_INCREMENT の値を自動インクリメント列の最大値より小さくしたときに発生する自動インクリメントバグを修正しました。(バグ #16310273)

Aurora MySQL データベースエンジンの更新: 2016-09-20 (バージョン 1.7.1) (廃止)

バージョン: 1.7.1

改良点

- InnoDB 全文検索のキャッシュがフルの場合、Aurora レプリカがクラッシュする問題を修正。
- スレッドプールのワーカースレッドが自身を待っている場合、データベースエンジンがクラッシュする問題を修正。
- テーブルのメタデータのロックによってデッドロックが発生した場合に Aurora レプリカがクラッシュする問題を修正。
- スレッドプールの 2 つのワーカースレッド間における競合状態のためにデータベースエンジンがクラッシュする問題を修正。
- モニタリングエージェントで分散ストレージサブシステムへの書き込みオペレーションの進行が検出されない場合に、高負荷時に不要なフェイルオーバーが発生する問題を修正。

Aurora MySQL データベースエンジンの更新: 2016-08-30 (バージョン 1.7.0) (廃止)

バージョン: 1.7.0

新機能

- NUMA 対応スケジューラ - Aurora MySQL エンジンのためのタスクスケジューラが NUMA (Non-Uniform Memory Access) 対応となりました。これにより、クロス CPU ソケット競合が最小化され、`db.r3.8xlarge` DB インスタンスクラスのスループットパフォーマンスが向上します。
- パラレル先読みのバックグラウンド非同期動作 - パラレル先読みは、専用スレッドを使用してスレッド競合を減らし、パフォーマンスを向上するように改良されました。

- インデックス作成の改良 (ラボモード) - セカンダリインデックスの作成の実装は、ボトムアップ式にインデックスを作成するようになり、不要なページ分割がなくなります。これによりインデックスの作成またはテーブルの再構築のための時間を短縮できます。この機能はデフォルトでは無効になっており、Aurora のラボモードを有効化すると有効になります。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL ラボモード」を参照してください。

改良点

- インスタンスへの接続リクエストの数が急増した場合に、接続の確立に長い時間がかかる問題を修正しました。
- InnoDB を使用せずにパーティショニングされたテーブルで ALTER TABLE を実行した場合にクラッシュが発生していた問題を修正しました。
- 高負荷の書き込みワークロードによりフェイルオーバーが発生する場合があった問題を修正しました。
- パーティションテーブルで RENAME TABLE を実行した場合にエラーが発生する、誤ったアサーションを修正しました。
- 高負荷の挿入ワークロード時にトランザクションがロールバックされるときの安定性が向上しました。
- Aurora レプリカで全文検索インデックスが動作しなかった問題を修正しました。

MySQL バグ修正の統合

- LOCK_grant ロックのパーティショニングにより安定性を向上しました。(ポート WL #8355)
- ストアドプロシージャの SELECT でカーソルを開くとセグメンテーション違反が発生します。(ポートのバグ #16499751)
- MySQL のいくつかの特別な使用法で、誤った結果となります。(バグ #11751794)
- GET_SEL_ARG_FOR_KEYPART でのクラッシュ - バグ #11751794 のパッチにより発生します。(バグ #16208709)
- GROUP BY を使った単純なクエリで誤った結果となります。(バグ #17909656)
- 範囲の述語を使った半結合 (semi-join) クエリで、余分な行が返されます。(バグ #16221623)
- IN サブクエリの後に ORDER BY 句を追加すると、重複列が返されます。(バグ #16308085)
- MyISAM で GROUP BY によるルーススキャンのクエリで EXPLAIN を使うとクラッシュします。(バグ #16222245)

- 引用符付きの INT 述語を使ったルースインデックススキャンでランダムなデータが返されます。(バグ #16394084)
- オプティマイザがルースインデックススキャンを使用した場合、サーバーがテンポラリテーブルを作成しようとするときに終了する場合があります。(バグ #16436567)
- COUNT(DISTINCT) は NULL 値をカウントするべきではありませんが、オプティマイザがルースインデックススキャンを使用したときにカウントされます。(バグ #17222452)
- クエリが MIN()/MAX() の両方と集計関数 (DISTINCT) (例えば SUM(DISTINCT)) を含む場合に、ルースインデックススキャンを使って実行されると、MIN()/MAX() の結果の値が正しく設定されません。(バグ #17217128)

Aurora MySQL データベースエンジンの更新: 2016-06-01 (バージョン 1.6.5) (廃止)

バージョン: 1.6.5

新機能

- バイナリログの効率的なストレージ - バイナリログの効率的なストレージは、すべての Aurora MySQL DB クラスターでデフォルトで有効になりました。設定することはできません。バイナリログの効率的なストレージは、2016 年 4 月の更新で導入されました。詳細については、「Aurora MySQL データベースエンジンの更新: 2016-04-06 (バージョン 1.6) (廃止)」を参照してください。

改良点

- プライマリインスタンスの負荷が高いときの Aurora レプリカの安定性が向上しました。
- パーティショニングされたテーブルと、テーブル名に特殊文字が含まれるテーブルでクエリを実行するときの Aurora レプリカの安定性が向上しました。
- セキュアな接続を使用する場合の接続の問題が修正されました。

MySQL バグ修正の統合

- SLAVE CAN'T CONTINUE REPLICATION AFTER MASTER'S CRASH RECOVERY (Port Bug #17632285)

Aurora MySQL データベースエンジンの更新: 2016-04-06 (バージョン 1.6) (廃止)

バージョン: 1.6

この更新には以下の改良点が含まれています。

新機能

- パラレル先読み - パラレル先読みは、すべての Aurora MySQL DB クラスターでデフォルトで有効になりました。設定することはできません。パラレル先読みは、2015 年 12 月の更新で導入されました。詳細については、「Aurora MySQL データベースエンジンの更新: 2015-12-03 (バージョン 1.4) (廃止)」を参照してください。

  パラレル先読みがデフォルトで有効になったことに加えて、このリリースではパラレル先読みに以下の機能強化が追加されています。

  - パラレル先読みの積極性が低くなるようにロジックが改善されました。これは、DB クラスターで多くのパラレルワークロードが生じた場合に役立ちます。
  - 小さいテーブルの安定性が向上しました。
- バイナリログの効率的なストレージ (ラボモード) - MySQL バイナリログファイルが、Aurora MySQL により効率的に保存されるようになりました。新しいストレージ実装により、バイナリログファイルを早期に削除できるようになり、バイナリログレプリケーションマスターとなっている Aurora MySQL DB クラスターにおけるインスタンスのシステムパフォーマンスが向上します。

  バイナリログの効率的なストレージを有効にするには、プライマリインスタンスまたは Aurora レプリカのパラメータグループで、`aurora_lab_mode` パラメータを `1` に設定します。`aurora_lab_mode` パラメータはインスタンスレベルのパラメータであり、デフォルトでは `default.aurora5.6` クラスターパラメータグループにあります。DB パラメータグループの変更については、「Amazon Aurora ユーザーガイド」の「「DB パラメータグループのパラメータの変更」を参照してください。パラメータグループと Aurora MySQL の詳細については、「Amazon Aurora ユーザーガイド」の「Aurora MySQL 設定パラメータ」を参照してください。

  バイナリログの効率的なストレージは、MySQL バイナリログレプリケーションマスターインスタンスとなっている、Aurora MySQL DB クラスター内のインスタンスでのみ有効にしてください。
- AURORA_VERSION システム可変 - `AURORA_VERSION` システム可変のクエリを実行することで、Aurora MySQL DB クラスターの Aurora バージョンを取得できるようになりました。

Aurora バージョンを取得するには、次のいずれかのクエリを使用します。

```
select AURORA_VERSION();
select @@aurora_version;
show variables like '%version';
```

Aurora バージョンは、DB クラスターの変更時に AWS Management Console で確認できます。さらに、AWS CLI コマンドの describe-db-engine-versions 、または API オペレーションの DescribeDBEngineVersions を呼び出して確認することもできます。

- ロックマネージャーのメモリ使用量メトリクス - ロックマネージャーのメモリ使用量に関する情報がメトリクスとして入手できるようになりました。

  ロックマネージャーのメモリ使用量メトリクスを取得するには、次のいずれかのクエリを使用します。

  ```
  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');
  ```

改良点

- バイナリログと XA トランザクションの回復時に安定性が向上しました。
- 接続数が多いことによるメモリの問題が修正されました。
- メトリクス `Read Throughput`、 `Read IOPS`、`Read Latency`、`Write Throughput`、`Write IOPS`、`Write Latency`、`Disk Queue Depth` の精度が向上しました。
- クラッシュ後に大きいインスタンスの起動が遅くなる安定性の問題が修正されました。
- 同期メカニズムとキャッシュの削除に関連するデータディクショナリの同時性が強化されました。
- Aurora レプリカの安定性とパフォーマンスの向上:
  - プライマリインスタンスの書き込みワークロードが大きい場合や急増したときの Aurora レプリカの安定性の問題が修正されました。
  - db.r3.4xlarge インスタンスと db.r3.8xlarge インスタンスのレプリカラグが強化されました。
  - ログレコードの適用と Aurora レプリカでの同時読み取りの間の競合を減らすことにより、パフォーマンスが向上しました。

- 新たに作成された統計または更新された統計について、Aurora レプリカにおける統計の更新時の問題が修正されました。
- プライマリインスタンスにトランザクションが多く、Aurora レプリカにおいて同じデータの同時読み取りが行われる場合に、Aurora レプリカの安定性が向上しました。
- UPDATE ステートメントを使用して DELETE および JOIN ステートメントを実行するときの Aurora レプリカの安定性が向上しました。
- INSERT ... SELECT ステートメントを実行するときの Aurora レプリカの安定性が向上しました。

MySQL バグ修正の統合

- アサーション `!M_ORDERED_REC_BUFFER' 失敗のバグ #18694052 修正を 5.6 にバックポート (ポートのバグ #18305270)
- MEMCPY()、HA_PARTITION::POSITION のセグメンテーション違反 (ポートのバグ # 18383840)
- PARTITIONING,INDEX_MERGE AND NO PK の結果が正しくない (ポートのバグ # 18167648)
- EXPORT: ASSERTION IN HA_PARTITION::EXTRA の FLUSH TABLES (ポートのバグ # 16943907)
- 仮想 HA_ROWS HANDLER::MULTI_RANGE_READ_INFO_CONST におけるサーバークラッシュ (ポートのバグ # 16164031)
- 範囲オプティマイザが SEL_ARG::RB_INSERT() でクラッシュする (ポートのバグ # 16241773)

Aurora MySQL データベースエンジンの更新: 2016-01-11 (バージョン 1.5) (廃止)

バージョン: 1.5

この更新には以下の改良点が含まれています。

改良点

- Aurora ストレージのデプロイメント時にアイドル状態のインスタンスに対する書き込みオペレーションが 10 秒間停止する問題を修正しました。

- `innodb_file_per_table` を `No` に設定した場合の論理的な先読みが機能するようになりました。論理的な先読み機能の詳細については、「Aurora MySQL データベースエンジンの更新: 2015-12-03 (バージョン 1.4) (廃止)」を参照してください。
- Aurora レプリカがプライマリインスタンスに再接続する問題を修正しました。この改善により、Aurora レプリカのテストで `quantity` パラメータに大きな値を指定した場合に、障害挿入クエリの使用に失敗する問題も修正されます。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora ユーザーガイド」の「Aurora レプリカの障害のテスト」を参照してください。
- Aurora レプリカの遅延と再起動のモニタリングを改善しました。
- Aurora レプリカが遅延し、登録解除され、再起動する原因となっていた問題を修正しました。
- デッドロックで `show innodb status` コマンドを実行する際の問題を修正しました。
- 高書き込みスループット時の、より大きなインスタンスに対するフェイルオーバーの問題を修正しました。

MySQL バグ修正の統合

- MySQL の全文検索が、データベース名が数字で始まるテーブルに影響する件を一部修正しました。(ポートのバグ #17607956)

Aurora MySQL データベースエンジンの更新: 2015-12-03 (バージョン 1.4) (廃止)

バージョン: 1.4

この更新には以下の改良点が含まれています。

新機能

- 高速挿入 - プライマリキーによってソートされたパラレル挿入を加速します。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL パフォーマンスの拡張」を参照してください。
- 大きなデータセットの読み取りパフォーマンス - Aurora MySQL は IO にかかる大きな負荷を自動的に検出し、DB クラスターのパフォーマンスを向上させるためにより多くのスレッドを起動します。Aurora スケジューラは IO アクティビティを参照して、システム内のスレッド数を動的に最適化し、オーバーヘッドを抑えながら IO の負荷と CPU の負荷を迅速に調整します。

- パラレル先読み - プライマリインスタンスまたは Aurora レプリカで使用可能なメモリに対して大きすぎる B-Tree スキャンのパフォーマンスを改善します (範囲クエリを含む)。パラレル先読みはページの読み取りパターンを検出し、必要な場合に備えて事前にページをバッファキャッシュに取得します。パラレル先読みは同じトランザクション内で同時に複数のテーブルで機能します。

改良点:

- Aurora ストレージのデプロイメント時に Aurora データベースの可用性が短い問題を修正しました。
- `max_connection` 制限を正しく強制します。
- Aurora がバイナリログのマスターであり、大きなデータをロードした後にデータベースが再起動する際のバイナリログの破棄を改善しました。
- テーブルキャッシュのメモリ管理の問題を修正しました。
- 高速復旧のために共有メモリのバッファキャッシュでの膨大なページのサポートを追加しました。
- スレッドローカルストレージが初期化されない問題を修正しました。
- 16K 接続をデフォルトで許可します。
- IO が膨大なワークロード用の動的スレッドプール。
- クエリキャッシュで UNION を含めた表示の正常な無効化の問題を修正しました。
- ディクショナリの統計スレッドでの安定性の問題を修正しました。
- キャッシュの削除に関連したディクショナリサブシステムのメモリリークを修正しました。
- マスターで書き込み負荷がきわめて低いときに Aurora レプリカで大きな読み取りレイテンシーが発生する問題を修正しました。
- マスターで ALTER TABLE ... REORGANIZE PARTITION などの DDL でパーティショニングされたテーブルでオペレーションを実行する場合の Aurora レプリカの安定性の問題を 修正しました。
- ボリューム増加時の Aurora レプリカの安定性の問題を修正しました。
- Aurora レプリカでクラスター化されていないインデックスをスキャンする際のパフォーマンスの問題を修正しました。
- Aurora レプリカでラグを発生させ、場合によっては登録解除され再起動される安定性の問題を修正しました。

MySQL バグ修正の統合

- FTSPARSE() のセグメンテーション違反。(バグ #16446108)

- InnoDB データディクショナリが列名を変更しながら更新されない。(バグ #19465984)
- 別のデータベースにテーブル名を変更した後に FTS がクラッシュする。(バグ #16834860)
- 削除されたテーブルでトリガーの準備が失敗すると エラー 1054 が発生する。(バグ #18596756)
- メタデータを変更するとトリガーの実行で問題が発生する場合がある。(バグ #18684393)
- 長い UTF8 VARCHAR フィールドに対してマテリアル化が選択されない。(バグ #17566396)
- ORDER BY で制限を X にした場合に実行プランのパフォーマンスが悪い (バグ #16697792)。
- バグ #11765744 を 5.1、5.5 および 5.6 にバックポート。(バグ #17083851)
- SQL/SQL_SHOW.CC が SIG6 になるミューテックスの問題。ソースが頻繁に FILL_VARIABLES になる。(バグ #20788853)
- バグ #18008907 を 5.5 以上のバージョンにバックポート。(バグ #18903155)
- MySQL 5.7 にスタックのオーバーフローエラーの修正を適応。(バグ #19678930)

Aurora MySQL データベースエンジンの更新: 2015-10-16 (バージョン 1.2、1.3) (廃止)

バージョン: 1.2、1.3

この更新には以下の改良点が含まれています。

修正内容

- 新しいロックマネージャーで実行時間の長いトランザクションの際のメモリー不足の問題を解決しました。
- 非 RDS for MySQL データベースでレプリケートする際のセキュリティの脆弱性を解決しました。
- ストレージ障害時にクォーラム書き込みが確実に正しく再試行されるように更新しました。
- レプリカの遅延をより正確にレポートするように更新しました。
- 多くの競合トランザクションが同じ列を変更しようとしているとき接続を減らすことでパフォーマンスを改善しました。
- 2 つのテーブルを結合して作成されたビューでのクエリキャッシュの無効化を解決しました。
- UNCOMMITTED_READ 分離を使用したトランザクションのクエリキャッシュを無効化しました。

改良点

- ウォームキャッシュでのカタログクエリが遅い問題のパフォーマンスがよくなりました。
- ディクショナリ統計での同時実行を改善しました。
- 新規クエリのキャッシュリソースマネージャー、拡張管理、Amazon Aurora スマートストレージに保存するファイル、およびログ記録のバッチ書き込みの安定性が向上しました。

MySQL バグ修正の統合

- innodb 内でクエリを強制するとアサーションとともにクラッシュすることがある。(バグ #1608883)
- イベントスケジューラ、イベント実行、または新規接続で新規スレッドの作成に失敗すると、エラーログにメッセージが書き込まれない。(バグ #16865959)
- 1 つの接続がデフォルトのデータベースを変更すると同時に別の接続が SHOW PROCESSLIST を実行した場合、2 番目の接続が 1 番目の接続のデフォルトのデータベースメモリを表示しようとすると無効なメモリにアクセスする。(バグ #11765252)
- 設計によるバイナリログの消去が使用中またはアクティブなバイナリログファイルを削除せず、これが発生したときに通知しない。(バグ #13727933)
- 一部のステートメントで、オプティマイザが不必要なサブクエリ句を削除するとメモリリークが発生することがある。(バグ #15875919)
- シャットダウン時に、サーバーが初期化されていないミューテックスをロックする場合がある。(バグ #16016493)
- 複数の列に名前をつける GROUP_CONCAT() および ORDER BY 句を使用した準備済みステートメントが、サーバーを終了させる場合がある。(バグ #16075310)
- Performance Schemaの計測がレプリカワーカースレッドで見つからない。(バグ #16083949)
- `STOP SLAVE` は、1 つ以上のステータス可変 `Slave_retried_transactions`、`Slave_heartbeat_period`、`Slave_received_heartbeats`、`S` または `Slave_running` の値を取得する SHOW STATUS などのステートメントと同時に発行されると、デッドロックが発生する可能性がある。(バグ #16088188)
- 検索する用語が引用句で囲まれた語句の場合に、ブールモードを使用した全文クエリの結果がゼロになることがある。(バグ #16206253)
- サブクエリの結合の ON 句でサブクエリを使用して準備済みステートメントを実行する際に、オプティマイザが冗長サブクエリ句を削除しようとすると、アサーションが立ち上がる。(バグ #16318585)

- GROUP_CONCAT が不安定、ITEM_SUM::CLEAN_UP_AFTER_REMOVAL でクラッシュが発生する。(バグ #16347450)
- デフォルトの InnoDB 全文検索 (FTS) ストップワードリストを、INFORMATION_SCHEMA.INNODB_FT_DEFAULT_STOPWORD と同じ構造を持つ InnoDB テーブルを作成することで置き換えようとすると、エラーになる。(バグ #16373868)
- ワーカーのクライアントスレッドが FLUSH TABLES WITH READ LOCK を実行した後、続いてマスターで更新があると、SHOW SLAVE STATUS を実行したときにワーカーがハングアップする。(バグ #16387720)
- 全文検索で「abc-def」などの区切り検索文字列を分析する場合、InnoDB で MyISAM と同じ単語区切り記号を使用するようになりました。(バグ #16419661)
- FTS_AST_TERM_SET_WILDCARD がクラッシュする。(バグ #16429306)
- FTS RQG テストの FTS_AST_VISIT() のセグメンテーション違反。(バグ #16435855)
- デバッグを構築するために、オプティマイザがサブクエリを指す Item_ref を削除すると、サーバーが終了する。(バグ #16509874)
- InnoDB テーブルの全文検索が、リテラル句を + または - 演算子と組み合わせた検索に失敗する。(バグ #16516193)
- START SLAVEオプション --master-info-repository=TABLE relay-log-info-repository=TABLE を付け、autocommit を 0 に設定し、 を付けてサーバーを起動すると、 が失敗する。--skip-slave-start(バグ #16533802)
- 非常に大きい InnoDB 全文検索 (FTS) 結果でメモリの消費が大きすぎる。(バグ #16625973)
- デバッグの構築で、検索文字列に直接バイナリを使用すると、バイナリに NULL バイトや意味のない文字が含まれている場合があるため、OPT_CHECK_ORDER_BY でアサーションが発生する。(バグ #16766016)
- 一部のステートメントで、オプティマイザが不必要なサブクエリ句を削除するとメモリリークが発生することがある。(バグ #16807641)
- ワーカーへの新規接続で STOP SLAVE を発行した後に元の接続を使用して SHOW SLAVE STATUS を発行した場合、FLUSH TABLES WITH READ LOCK を発行した後にデッドロックが発生する可能性がある。(バグ #16856735)
- 無効な区切り記号を使用した GROUP_CONCAT でサーバーが終了する場合がある。(バグ #16870783)
- サーバーが SHOW STATUS LIKE 'pattern' ステートメントの LOCK_active_mi および active_mi-> rli-> data_lock ミューテックスを過度にロックする。パターンがミューテックス (Slave_heartbeat_period、Slave_last_heartbeat、Slave_received_heartbeats、Slave_ を使用するステータス可変と一致しなかった場合でも、同じ結果になる。(バグ #16904035)

- ブールモード修飾子を使用した全文検索がアサーションを伴う失敗になる。(バグ #16927092)
- + ブール演算子を使用した検索で InnoDB テーブルの全文検索が失敗する。(バグ #17280122)
- 4 ウェイデッドロック: ゾンビ、バイナリログの破棄、プロセスリストの表示、バイナリログの表示。(バグ #17283409)
- コミットロックを待っている SQL スレッドが強制終了され再びスタートされると、トランザクションがワーカーにスキップされる。(バグ #17450876)
- 「unended」トークンが原因で InnoDB 全文検索失敗が発生する。文字列および文字列長を、文字列比較に渡す必要があります。(バグ #17659310)
- 多数のパーティショニングされた InnoDB テーブルを MySQL 5.6 または 5.7 で使用すると、MySQL サーバーの以前のリリースで同じテーブルを使用したときよりも多くのメモリを消費する。(バグ #17780517)
- 全文クエリで、num_token が max_proximity_item より少ないことを確認しようとして失敗すると、アサーションが発生する。(バグ #18233051)
- INFORMATION_SCHEMA TABLES および COLUMNS テーブルの特定のクエリが、多量の空の InnoDB がある場合に過剰にメモリを使用する。(バグ #18592390)
- トランザクションをコミットするときに、特に master_info_repository=TABLE でサーバーを実行している場合により多くのリソースを使用していたスレッドそのものを確認する方法ではなく、フラグを使用してスレッドが作成されたかどうかを確認するようになりました。(バグ #18684222)
- マスターが DML を実行している間にワーカーのクライアントスレッドが FLUSH TABLES WITH READ LOCK を実行する場合、同じクライアントで `SHOW SLAVE STATUS` を実行するとブロックされ、デッドロックが発生する。(バグ #19843808)
- GROUP_CONCAT() で順序付けを行うと、サーバーが終了する場合がある。(バグ #19880368)

Aurora MySQL データベースエンジンの更新: 2015-08-24 (バージョン 1.1) (廃止)

バージョン: 1.1

この更新には以下の改良点が含まれています。

- MySQL データベースでレプリケーションする場合のレプリケーションの安定性の改善 (バイナリログレプリケーション)。MySQL を使用した Aurora MySQL レプリケーションの詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora でのレプリケーション」を参照してください。

- レプリケーションワーカーである Aurora MySQL DB クラスターの累積されたリレーログのサイズに対する 1 ギガバイト (GB) の制限。これにより、Aurora DB クラスターのファイル管理が改善されます。
- 先読み、再帰的な外部キーの関係および Aurora レプリケーション部分の安定性の向上。
- MySQL バグ修正の統合。
  - 名前の先頭が数字の InnoDB データベースでは、全文検索 (FTS) パーサーエラーが発生します。(バグ #17607956)
  - InnoDB 全文検索は、名前の尖塔が数字のデータベースでは失敗します。(バグ #17161372)
  - Windows 上の InnoDB データベースの場合、全文検索 (FTS) オブジェクト ID が期待される 16 進形式ではありません。(バグ #16559254)
  - MySQL 5.6 に導入されたコード回帰は、DROP TABLE と ALTER TABLE のパフォーマンスにマイナスの影響を与えました。このため、MySQL Server 5.5.x から 5.6.x の間でパフォーマンスが低下する可能性があります。(バグ #16864741)
- ロギングを単純にしてログファイルのサイズおよび必要なストレージ容量を削減します。

Aurora MySQL データベースエンジンの更新で修正された MySQL のバグ

以下のセクションでは、Aurora MySQL データベースエンジンの更新で修正された MySQL のバグについて説明します。

トピック

- Aurora MySQL 3.x データベースエンジンの更新で修正された MySQL のバグ
- Aurora MySQL 2.x データベースエンジンの更新で修正された MySQL のバグ
- Aurora MySQL 1.x データベースエンジンの更新で修正された MySQL のバグ

Aurora MySQL 3.x データベースエンジンの更新で修正された MySQL のバグ

MySQL 8.0 互換バージョンの Aurora には、対応する MySQL 互換バージョンで適用された MySQL のバグ修正がすべて反映されています。次の表は、Aurora MySQL データベースエンジンの更新で修正された追加の MySQL のバグと、どの更新で修正されたかを示しています。

データベースエンジンの更新	MySQL 互換バージョン	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンアップデート 2024-03-07 (バージョン 3.06.0、My	8.0.34	3.06.0	• キャッシュライン値が誤って計算され、Graviton インスタンスでデータベースを再起動する際に障害が発生する問題を修正しました。(コミュニティバグフィックス #35479763) • ストアドルーチン内のサブクエリの一部のインスタンスが常に正しく処理されない問題を修正しました。(コミュニティバグフィックス #35377192)

データベースエンジンの更新	MySQL 互換バージョン	バージョン	修正された MySQL のバグ
SQL 8.0.34 に対応)			• バックグラウンドの TLS 証明書のローテーションが原因で CPU 使用率が高くなることがある問題を修正しました。(コミュニティのバグ修正 #34284186) • Aurora MySQL バージョン 3.05 より前の MySQL INSTANT システムスキーマのテーブルへの列の追加を InnoDB が許可していた問題を修正しました。これにより、Aurora MySQL バージョン 3.05.0 にアップグレードした後に、サーバーが予期せず終了する (データベースインスタンスが再起動する) 可能性がある問題を修正しました。(コミュニティのバグ修正 #35625510)
Aurora MySQL データベースエンジンアップデート 2024-01-31 (バージョン 3.05.2、MySQL 8.0.32 に対応)	8.0.32	3.05.2	• 複数の、XORまたは条件を含む SELECT AND ステートメントをサブクエリとして持つストアドルーチンを繰り返し実行するとOR、仮想メモリが過剰に消費され、最終的には仮想メモリが枯渇する可能性がありました。(コミュニティバグフィックス #33852530)

データベースエンジンの更新	MySQL 互換バージョン	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンアップデート 2023-11-21 (バージョン 3.05.1、MySQL 8.0.32 に対応)	8.0.32	3.05.1	• システムスキーマ内の MySQL テーブルで Aurora MySQL バージョン 3.01 から Aurora MySQL バージョン 3.04 までの間に `INSTANT ADD` 列が追加され、Aurora MySQL がバージョン 3.05.0 にアップグレードされた後に、これらのテーブルの DML によってサーバーが予期せず終了する場合の InnoDB の問題を修正しました。(コミュニティのバグ修正 #35625510)
Aurora MySQL データベースエンジンの更新 2023-10-25(バージョン 3.05.0、MySQL 8.0.32 互換)	8.0.32	3.05.0	• バックグラウンドの TLS 証明書のローテーションが原因で CPU 使用率が高くなる問題を修正しました (コミュニティのバグ修正 #34284186)。

データベースエンジンの更新	MySQL 互換バージョン	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンアップデート 2024-03-15 (バージョン 3.04.2、MySQL 8.0.28 に対応)	8.0.28	3.04.2	• キャッシュライン値が正しく計算されず、Graviton ベースのインスタンスでデータベースを再起動する際に障害が発生する問題を修正しました。(コミュニティバグフィックス #35479763) • 複数の、、XORまたは条件を含む SELECT OR ステートメントをサブクエリとして持つストアドルーチンを繰り返し実行するとAND、仮想メモリが過剰に消費され、最終的には仮想メモリが枯渇する可能性がありました。(コミュニティバグフィックス #33852530)

データベースエンジンの更新	MySQL 互換バージョン	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンアップデート 2023-11-13 (バージョン 3.04.1、MySQL 8.0.28 に対応) デフォルト	8.0.28	3.04.1	• バックグラウンドの TLS 証明書のローテーションが原因で CPU 使用率が高くなる問題を修正しました (コミュニティのバグ修正 #34284186)。

データベースエンジンの更新	MySQL 互換バージョン	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンアップデート 2023-07-31 (バージョン 3.04.0、MySQL 8.0.28 に対応)	8.0.28	3.04.0	• intrinsic の一時テーブルページを含むバッファブロックがページトラバーサル中に再配置され、アサーションエラーが発生する問題を修正しました (バグ# 33715694)。 • InnoDB: オンライン DDL out-of-bounds 操作がメモリにアクセスしないようにします (バグ番号 34750489、バグ番号 108925) • 複数の共通テーブル式 (CTE) がネスト構造になっている複雑な SQL ステートメントの処理中に、誤ったクエリ結果が生成されることがある問題を修正しました (バグ #34572040、バグ #34634469、バグ #33856374)。

データベースエンジンの更新	MySQL 互換バージョン	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンの更新 2023-12-08 (バージョン 3.03.3、MySQL 8.0.26 互換)	8.0.26	3.03.3	• バックグラウンドの TLS 証明書のローテーションが原因で CPU 使用率が高くなる問題を修正しました (コミュニティのバグ修正 #34284186)。
Aurora MySQL データベースエンジンの更新 2023-08-29 (バージョン 3.03.2、MySQL 8.0.26 互換)	8.0.26	3.03.2	• 複数の共通テーブル式 (CTE) がネスト構造になっている複雑な SQL ステートメントの処理中に、誤ったクエリ結果が生成されることがある問題を修正しました (バグ #34572040、バグ #34634469、バグ #33856374) • InnoDB: 同じテーブルの統計を初期化解除および初期化しようとするスレッド間で競合状態になり、アサーション障害が発生していました (バグ #33135425) • InnoDB: オンライン DDL out-of-bounds 操作がメモリにアクセスするのを防ぐ (バグ #34750489, バグ #108925)

データベースエンジンの更新	MySQL 互換バージョン	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンの更新 2023-05-11 (バージョン 3.03.1、MySQL 8.0.26 互換)	8.0.26	3.03.1	• intrinsic の一時テーブルページを含むバッファブロックがページトラバーサル中に再配置され、アサーションエラーが発生する問題を修正しました (バグ #33715694)。

データベースエンジンの更新	MySQL 互換バージョン	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンの更新 2023-03-01 (バージョン 3.03.0、MySQL 8.0.26 互換) このバージョンへのアップグレードはサポートされていません。	8.0.26	3.03.0	• JSON や TEXT など、一部の列タイプのソート時に、ソートバッファのサイズがソートの最大行の 15 倍未満の場合、バッファが枯渇してしまう場合がある問題を修正しました。現在は、ソートバッファのサイズは最大のソートキーの 15 倍あれば十分です。(バグ #103325、バグ #105532、バグ #32738705、バグ #33501541) • InnoDB がテーブルパーティションの一部の有効な名前を正しく処理しない場合がある問題を修正しました。(バグ #32208630) • 特定の条件下で、OR 条件を使用したクエリを実行した場合に nullability プロパティの計算が不正確になり、誤った結果が返される可能性がある問題を修正しました。(バグ #34060289) • 次の 2 つの条件が当てはまる場合に、特定の条件下で誤った結果が返される可能性がある問題を修正しました。 • 派生テーブルが外側のクエリブロックにマージされている。

データベースエンジンの更新	MySQL 互換バージョン	バージョン	修正された MySQL のバグ
			• クエリに左結合と IN サブクエリが含まれている。 (バグ #34060289) • 整数列の最大値を超えた場合に、不正な AUTO_INCREMENT 値が生成されていた問題を修正しました。このエラーは、列の最大値が考慮されていないことが原因でした。この場合、以前の有効な AUTO_INCREMENT 値が返され、重複キーエラーが発生すべきでした。(バグ #87926、バグ #26906787) • パフォーマンススキーマの DROP 権限を取り消すことができなかった問題を修正しました。(バグ #33578113) • IF ステートメントで EXISTS を使用したストアドプロシージャが、処理対象の 1 つ以上のテーブルが実行の合間に削除され、再作成された場合に、最初の呼び出し以降の呼び出しで正しく実行されない問題を修正しました。(バグ #32855634) • サブクエリのビューと外部クエリブロックを参照するクエリが、予期しない再起動を引き起こす可能性がある問題を修正しました。(バグ #32324234)

データベースエンジンの更新	MySQL 互換バージョン	バージョン	修正された MySQL のバグ

Aurora MySQL データベースエンジンの更新 2022-11-18 (バージョン 3.02.2、MySQL 8.0.23 互換) 標準サポートは 2024 年 1 月 15 日に終了。	8.0.23	3.02.2	• 特定の条件下で、OR 条件を使用したクエリを実行した場合に nullability プロパティの計算が不正確になり、誤った結果が返される可能性がある問題を修正しました。(バグ #34060289)
			• 次の 2 つの条件が当てはまる場合に、特定の条件下で誤った結果が返される可能性がある問題を修正しました。
			  • 派生テーブルが外側のクエリブロックにマージされている。
			  • クエリに左結合と IN サブクエリが含まれている。(バグ #34060289)
			• パフォーマンススキーマの DROP 権限を取り消すことができなかった問題を修正しました。(バグ #33578113)
			• IF ステートメントで EXISTS を使用したストアドプロシージャが、処理対象の 1 つ以上のテーブルが実行の合間に削除され、再作成された場合に、最初の呼び出し以降の呼び出しで正しく実行されない問題を修正しました。(MySQL バグ #32855634)
			• 整数列の最大値を超えた場合に、不正な AUTO_INCREMENT

データベースエンジンの更新	MySQL 互換バージョン	バージョン	修正された MySQL のバグ
			値が生成されていました。このエラーは、列の最大値が考慮されていないことが原因でした。この場合、以前の有効な AUTO_INCREMENT 値が返され、重複キーエラーが発生すべきでした。(バグ #87926、バグ #26906787) • 特定のテーブル ID を持つユーザー作成のテーブルを含む Aurora MySQL バージョン 1 (MySQL 5.6 互換) データベースクラスターをアップグレードする際に、エラーが発生する可能性がある問題を修正しました。Aurora MySQL バージョン 2 (MySQL 5.7 互換) から Aurora MySQL バージョン 3 (MySQL 8.0 互換) にアップグレードする際に、これらのテーブル ID を割り当てると、データディクショナリテーブル ID が競合する可能性があります。(バグ #33919635)

データベースエンジンの更新	MySQL 互換バージョン	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンの更新 2022-04-20 (バージョン 3.02.0、MySQL 8.0.23 互換) 標準サポートは 2024 年 1 月 15 日に終了。このバージョンへのアップグレードはサポートされていません。	8.0.23	3.02.0	予期しないサーバーの動作を引き起こす可能性のある、ストアドプロシージャ内のカーソルに使用される一時テーブルの不適切な処理を修正しました。(バグ #32416811)

データベースエンジンの更新	MySQL 互換バージョン	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンの更新 2022-04-15 (バージョン 3.01.1、MySQL 8.0.23 互換) 標準サポートは 2024 年 1 月 15 日に終了。このバージョンへのアップグレードはサポートされていません。	8.0.23	3.01.1	予期しないサーバーの動作を引き起こす可能性のある、ストアドプロシージャ内のカーソルに使用される一時テーブルの不適切な処理を修正しました。(バグ #32416811)

Aurora MySQL 2.x データベースエンジンの更新で修正された MySQL のバグ

MySQL 5.7 と互換性のあるバージョンの Aurora には、MySQL 5.7.40 によるすべての MySQL バグ修正が含まれています。次の表は、Aurora MySQL データベースエンジンの更新で修正された追加の MySQL のバグと、どの更新で修正されたかを示しています。

データベースエンジンの更新	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンの更新 2023-12-28 (バージョン 2.12.1、MySQL 5.7.40 互換)	2.12.1	• `SHOW PROCESSLIST` ステートメントと同時に実行すると、既存および新規のリモート接続が停止する問題を修正しました (コミュニティバグ #34857411) • レプリケーション: 一部のバイナリログイベントが必ずしも正しく処理されていませんでした (バグ #34617506) • 全文検索 (FTS) パーサープラグインによる単一文字トークンの処理を修正しました (バグ #35432973)
Aurora MySQL データベースエンジンの更新 2023-07-25 (バージョン 2.12.0、MySQL 5.7.40 互換)	2.12.0	• バックグラウンドの TLS 証明書のローテーションが原因で CPU 使用率が高くなる問題を修正しました。(コミュニティのバグ修正 #34284186)
Aurora MySQL データベースエンジンの更新 2023-10-17 (バージョン 2.11.4、MySQL 5.7.12 互換)	2.11.4	• レプリケーション: 一部のバイナリログイベントが必ずしも正しく処理されていませんでした。(バグ #34617506) • バックグラウンドの TLS 証明書のローテーションが原因で CPU 使用率が高くなる問題を修正しました。(コミュニティのバグ修正 #34284186) • プリペアドステートメントで、一部のタイプのサブクエリでサーバーが終了する場合がありました。(バグ #33100586)
Aurora MySQL データベースエンジンの更新	2.11.0	• パフォーマンススキーマのステートメントイベントテーブル (events_statements_current など) から文字セット情報を読み取るコードが、その文字セット情報への同時書き込みを阻止

データベースエンジンの更新	バージョン	修正された MySQL のバグ
2022-10-25 (バージョン 2.11.0、MySQL 5.7.12 互換) このバージョンは新規作成には使用不可。		できなかった問題を修正しました。その結果、SQL クエリテキストの文字セットが無効になり、サーバーが終了する可能性がありました。今回の修正により、無効な文字セットがあると SQL_TEXT 列が切り捨てられ、サーバーの終了が阻止されるようになりました。(バグ #23540008) • InnoDB: コミュニティバグ #25189192、バグ #84038 の修正のバックポート。テーブルを別のスキーマに移動する RENAME TABLE オペレーションの後、InnoDB が INNODB_SYS_DATAFILES データディクショナリテーブルを更新できなかった問題を修正しました。その結果、テーブルスペースデータファイルが見つからないというエラーが再起動時に発生していました。 • InnoDB: 新しいインデックスの追加時に、サーバーが内部定義の外部キーインデックスを削除し、仮想生成列に定義されたセカンダリインデックスを外部キーインデックスとして使用しようとすると、サーバーが終了する問題を修正しました。InnoDB は、外部キー制約が仮想生成列で定義されたセカンダリインデックスを参照することを許可するようになりました。(バグ #23533396) • 2 つのセッションが INSERT ... ON DUPLICATE KEY UPDATE オペレーションを同時に実行している場合にデッドロック状態になる問題を修正しました。タプルの部分的ロールバック中に、別のセッションがタプルを更新する可能性があります。このバグの修正に伴い、バグ #11758237、バグ #17604730、バグ #20040791 の修正が取り消されます。(バグ #25966845) • automatic_sp_privileges が有効になっていても、EXECUTE 権限と ALTER ROUTINE 権限がルーチン作成者に正しく付与されない問題を修正しました。(バグ #27407480) • コミュニティバグ #24671968 の修正のバックポート: WHERE 句に相関サブクエリが含まれていて、テーブルのセカンダリインデックスが選択リストの列とサブクエリの列に定義されており、GROUP BY または DISTINCT でクエリに

データベースエンジンの更新	バージョン	修正された MySQL のバグ
		ルースインデックススキャンを適用できる場合に、クエリが誤った結果を生成する可能性がある問題を修正しました。 • 外部キーを持つ複数のテーブルに対して複数テーブルの DELETE ステートメントを発行すると、レプリケーションが中断される問題を修正しました。(バグ #80821) • slave_skip_errors が有効になっていても、特殊なケースで特定のスレーブエラーが無視されない問題を修正しました。行ベースのレプリケーションを実行しているサーバーでテーブルを開いたりロックしたりできない場合や、フィールド変換に失敗した場合、エラーは重大と見なされ、slave_skip_errors の状態は無視されます。今回の修正により、slave_skip_errors が有効な場合は、トランザクションの適用中に報告されたすべてのエラーが正しく処理されるようになりました。(バグ #70640、バグ #17653275) • SET PASSWORD ステートメントが MySQL 5.6 マスターから MySQL 5.7 スレーブにレプリケートされる場合や、log_built in_as_identified_by_password システム変数が ON に設定された MySQL 5.7 マスターから MySQL 5.7 スレーブにレプリケートされる場合に、パスワードハッシュ自体もスレーブに格納される前にハッシュされる問題を修正しました。この問題は修正され、レプリケートされたパスワードハッシュは、スレーブに当初渡されたままの形で保存されます。(バグ #24687073) • 多数のレベルの JSON 配列、オブジェクト、またはその両方でラップされた大きなサブドキュメントで構成される JSON 値のシリアル化が、完了するまで長時間かかることがあった問題を修正しました。(バグ #23031146) • 構文エラーなどの理由で解析できないステートメントは、スロークエリログに書き込まれなくなりました。(バグ #33732907)

データベースエンジンの更新	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンの更新 2022-11-01 (バージョン 2.10.3) (廃止)	2.10.3	• パフォーマンススキーマのステートメントイベントテーブル (events_statements_current など) から文字セット情報を読み取るコードが、その文字セット情報への同時書き込みを阻止できなかった問題を修正しました。その結果、SQL クエリテキストの文字セットが無効になり、サーバーが終了する可能性がありました。今回の修正により、無効な文字セットがあると SQL_TEXT 列が切り捨てられ、サーバーの終了が阻止されるようになりました。(バグ #23540008) • プライマリキーが 1024 バイト超の一時テーブルを UPDATE が必要とし、そのテーブルが InnoDB を使用して作成された場合に、サーバーが終了することがある問題を修正しました。(バグ #25153670)

データベースエンジンの更新	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンの更新 2022-01-26 (バージョン 2.10.2) (廃止)	2.10.2	• テーブルの統計に関連するコードのエラーが、dict0stats.cc (http://dict0stats.cc/) ソースファイルでアサーションを引き起こすという、InnoDB の問題を修正しました。(バグ #24585978) • インデックスがオンラインで構築される際に、仮想列上のセカンダリインデックスが破損していました。UPDATE (https://dev.mysql.com/doc/refman/5.7/en/update.html) ステートメントについて、この問題を次のように修正します。インデックスレコードの仮想列にある値が NULL に設定されている場合、この値はクラスターインデックスレコードから生成します。(バグ #30556595) • ASSERTION "!OTHER_LOCK" IN LOCK_REC_ADD_TO_QUEUE (バグ #29195848) • HANDLE_FATAL_SIGNAL (SIG=11) IN __STRCHR_SSE2 (バグ #28653104) • ロック待機中にクエリが中断すると、InnoDB でエラーが発生する可能性がある問題を修正しました。(バグ #28068293) • トランザクション分離レベルが REPEATABLE READ に設定されている場合、インターリーブトランザクションがレプリカアプライヤをデッドロックすることがありました。(バグ #25040331) • ロック待機タイムアウトが原因でバイナリログレプリカが低速になる問題を修正しました。 (バグ #27189701)
Aurora MySQL データベースエンジンの更新 2021-10-21 (バージョン 2.10.1) (廃止)	2.10.1	CURRENT_TIMESTAMP は、トリガーでゼロを生成します。(バグ #25209512)

データベースエンジンの更新	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンの更新 2021-05-25 (バージョン 2.10.0) (廃止)	2.10.0	• トランザクション分離レベルが REPEATABLE READ に設定されている場合、インターリーブトランザクションがレプリカアプライヤをデッドロックすることがありました。(バグ #25040331) • ストアドプロシージャに、あるビューを参照するステートメントであって、代わりに別のビューを参照したものが含まれていた場合、プロシージャを複数回正常に呼び出すことができませんでした。(バグ #87858、バグ #26864199) • 多くの OR 条件を持つクエリでは、オプティマイザのメモリ効率がより高くなり、システム可変 range_optimizer_max_mem_size によって課されるメモリ制限を超える可能性が低くなりました。さらに、その可変のデフォルト値は 1,536,000 から 8,388,608 に引き上げられました。(バグ #79450、バグ #22283790) • レプリケーション: リレーログから次のイベントを読み込むためにレプリカの SQL スレッドによって呼び出される `next_event()` 関数では、(例えば、クローズされたリレーログを原因として) SQL スレッドが実行されてエラーになったときに取得した `relaylog.log_lock` を解放しなかったため、他のすべてのスレッドが、終了するためにリレーログのロックを取得するまで待機する状態を生じさせました。この修正により、SQL スレッドがその状況下で関数を離れる前にロックが解放されます。(バグ #21697821) • 仮想列を使用した `ALTER TABLE` のメモリ破損を修正しました。(バグ #24961167、バグ #24960450) • レプリケーション: マルチスレッドレプリカは、そのサイズより大きいトランザクションを処理する必要があった場合、slave_pending_jobs_size_max を使用してより小さなキューサイズで設定できませんでした。slave_pending_jobs_size_max より大きいパケットは、slave_max_allowed_packet で設定された制限よりもパケットが小さい場合でも、エラー `ER_MTS_EVENT_BIGGER_PENDING_JOBS_SIZE_MAX`

データベースエンジンの更新	バージョン	修正された MySQL のバグ
		で拒否されました。今回の修正により、slave_pending_jobs_size_max がハード制限ではなくソフト制限になります。パケットのサイズが slave_pending_jobs_size_max を超えるが、slave_max_allowed_packet より小さい場合、すべてのレプリカワーカーが空のキューを持つまでトランザクションが保持されてから処理されます。後続のすべてのトランザクションは、大規模なトランザクションが完了するまで保持されます。したがって、レプリカワーカーのキューサイズは制限されますが、時折発生するより大きなトランザクションは引き続き許可されます。(バグ #21280753、バグ #77406) • レプリケーション: マルチスレッドレプリカを使用する場合、適用元エラーで、Performance Schemaのレプリケーションテーブルで外部化されたデータと一致しないワーカー ID データが表示されました。(バグ #25231367) • レプリケーション:-Gtid-mode=ON、-LOG-bin=off で -slave-skip-errors を使用して実行されている GTID ベースのレプリケーションレプリカで、無視すべきエラーが発生したときに正しく更新されず、との同期が失われる原因となっていました。`Exec_Master_Log_Pos` `Exec_Master_Log_Pos` `Read_master_log_pos` `GTID_NEXT` が指定されていない場合、レプリカは、単一のステートメントトランザクションからロールバックするときに GTID 状態を更新しません。トランザクションが終了しても、GTID 状態は別途表示されるので、`Exec_Master_Log_Pos` は更新されません。この修正により、`GTID_NEXT` が指定された場合にのみトランザクションがロールバックされるときに GTID 状態の更新の制約がなくなります。(バグ #22268777) • レプリケーション: バイナリログ記録が無効になっているときに、部分的に失敗したステートメントが、自動生成または指定された GTID を正しく消費しませんでした。この修正により、部分的に失敗した DROP TABLE、部分的に失敗した DROP USER、または部分的に失敗した DROP VIEW がそれぞれ関連する GTID を消費することと、バイナリログ記録が

データベースエンジンの更新	バージョン	修正された MySQL のバグ
		無効である場合は、それを @@GLOBAL.GTID_EXECUTED および mysql.gtid_executed テーブルに保存することが保証されます。(バグ #21686749) • レプリケーション: MySQL 5.7 を実行しているレプリカは、MySQL 5.5 の一部ではない server_uuid の取得エラーにより、MySQL 5.5 ソースに接続できませんでした。これは、server_uuid の取得方法が変更されたことが原因となって生じました。(バグ #22748612) • バイナリログレプリケーション: この修正の前に、GTID トランザクションのスキップメカニズムが XA トランザクションで正常に動作していませんでした。サーバーは、ある GTID トランザクションが過去に既に実行している場合、その GTID トランザクションを (サイレントに) スキップするメカニズムを備えています。(BUG#25041920) • 渡されたトランザクション ID が正しくないせいで失敗した XA ROLLBACK ステートメントが、正しいトランザクション ID でバイナリログに記録され、レプリケーションレプリカによって処理される可能性があります。バイナリログ記録が実行される前にエラー状況をチェックし、失敗した XA ROLLBACK ステートメントはログに記録されなくなりました。(バグ #26618925) • レプリケーション:ソースログファイル名とソースログ位置を指定しない CHANGE MASTER TO ステートメントを使用してレプリカを設定した場合、START SLAVE が発行される前にシャットダウンし、オプション -relay-log-recovery set を使用して再起動すると、レプリケーションは開始されませんでした。これは、リレーログの復旧が試行される前にレシーバースレッドがスタートされておらず、ソースログファイル名とソースログの位置を提供するログローテーションイベントがリレーログで使用できないために発生しました。この場合、レプリカはリレーログの復旧をスキップし、警告をログに記録し、レプリケーションをスタートします。(バグ #28996606、バグ #93397)

データベースエンジンの更新	バージョン	修正された MySQL のバグ
		• レプリケーション: 行ベースのレプリケーションでは、utf8mb3 列を持つテーブルから、列が utf8mb4 文字セットで定義されている同じ定義のテーブルにレプリケートするときに、フィールド文字数を誤って表示するメッセージが返されました。(バグ #25135304、バグ #83918) • レプリケーション: GTID が使用中のレプリケーションレプリカで RESET SLAVE ステートメントが発行されると、既存のリレーログファイルは消去されましたが、チャネル用の受信した GTID のセットがクリアされる前に、代わりの新しいリレーログファイルが生成されました。したがって、以前の GTID セットは PREVIOUS_GTIDS イベントとして新しいリレーログファイルに書き込まれ、両方のサーバーの gtid_executed セットが空であっても、レプリカがソースよりも多くの GTID を持つことを示す致命的なエラーがレプリケーションで発生しました。ここで、RESET SLAVE が発行されると、新しいリレーログファイルが生成される前に受信した GTID のセットがクリアされるため、この状況は発生しません。(バグ #27411175) • レプリケーション: レプリケーションのために GTID が使用されているときに、分析エラー (ER_PARSE_ERROR) を引き起こしたステートメントを含むトランザクションを、同じ GTID で空のトランザクションまたは代替トランザクションの挿入のための推奨される方法によって、手動でスキップできませんでした。このアクションにより、レプリカは既に使用されている GTID を識別し、GTID を共有した不要なトランザクションをスキップします。ただし、分析エラーの場合、GTID をスキップする必要があるかどうかを確認する前にステートメントが分析されたため、いかなる場合でもトランザクションをスキップする意図があったにもかかわらず、分析エラーによりレプリケーション適用元スレッドが停止しました。この修正により、GTID が既に使用されていたため、関連するトランザクションをスキップする必要がある場合に、レプリケーションの適用元スレッドが分析エラーを無視するようにな

データベースエンジンの更新	バージョン	修正された MySQL のバグ

		りました。この動作の変更は、mysqlbinlog によって生成されるバイナリログ出力で構成されるワークロードの場合には適用されないことに注意してください。そのような状況では、スキップされたトランザクションの直後に実行される、分析エラーがあるトランザクションも、エラーを発生させるべきときに警告なしでスキップされるリスクがあります。(バグ #27638268)
		• レプリケーション: GTID に対する SQL スレッドが部分的なトランザクションをスキップできるようにします。(バグ #25800025)
		• レプリケーション: 負のタイムアウトパラメータまたは小数のタイムアウトパラメータが WAIT_UNTIL_SQL_THREAD_AFTER_GTIDS() に提供されると、サーバーは予期しない動作をしました。この修正によりもたらされる結果は次のとおりです。
		  • 小数のタイムアウト値はそのまま読み込まれ、丸められることはありません。
		  • サーバーが厳密な SQL モードの場合、負のタイムアウト値はエラーで拒否されます。サーバーが厳密な SQL モードでない場合、この値は、関数が待機せずに、即時に NULL を返し、その後に警告を発行するようにします。(バグ #24976304、バグ #83537)
		• レプリケーション: WAIT_FOR_EXECUTED_GTID_SET() 関数が小数部 (1.5 など) を含むタイムアウト値とともに使用された場合、キャストロジックのエラーは、タイムアウトが最も近い 1 秒に切り捨てられ、1 秒未満の値 (0.1 など) はゼロに切り捨てられたことを意味していました。キャストロジックが修正され、タイムアウト値が当初指定されたとおりに、丸められることなく、適用されるようになりました。Dirkjan Bussink の貢献に感謝の意を表します。(バグ #29324564、バグ #94247)

データベースエンジンの更新	バージョン	修正された MySQL のバグ
		• GTID を有効にすると、複数ステートメントトランザクション内の切断された XA トランザクションで XA COMMIT がアサーションを発生させました。(バグ #22173903) • レプリケーション: gtid_next 値が手動で設定されたときに、不明なトランザクション識別子に対して XA ROLLBACK ステートメントが発行された場合、デバッグ構築でアサーションが発生しました。XA ROLLBACK ステートメントがエラーで失敗した場合、サーバーは GTID 状態の更新を試行しなくなりました。(バグ #27928837、バグ #90640) • 複数の CASE 関数が ORDER BY 句で使用されている場合に、ソート順が正しくなくなる問題を修正しました。(バグ #22810883) • 順序付けを使用する一部のクエリが、初期化されていない列に最適化中にアクセスし、サーバーを終了させる可能性があります。(バグ #27389294)
Aurora MySQL データベースエンジンの更新 2021-11-12 (バージョン 2.09.3) (廃止)	2.09.3	• アサーション !M_PREBUILT->TRX->CHECK_FORIGNS。(バグ #23533396) • レプリケーション:* WAIT_FOR_EXECUTED_GTID_SET () 関数のロックの不具合により、特定の状況でサーバーがハングする可能性があります。この問題はすでに修正されました。(バグ #29550513)

データベースエンジンの更新	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンの更新 2020-12-11 (バージョン 2.09.1) (廃止)	2.09.1	• レプリケーション: トランザクション分離レベルが REPEATABLE READ に設定されている場合、インターリーブトランザクションがスレーブアプライヤをデッドロックすることがありました。(バグ #25040331) • デフォルト値の CURRENT_TIMESTAMP が書き込まれた TIMESTAMP または DATETIME 列を持つテーブルで、そのテーブルに BEFORE INSERT トリガーがある場合は、これらの列が 0000-00-00 00:00:00 に初期化されることがあります。(バグ #25209512、バグ #84077) • VALUES リストが結合を含むサブクエリを使用して 2 行目以降の値を生成した INSERT ステートメントの場合、必要な権限の解決に失敗するとサーバーが終了する可能性があります。(バグ #23762382)
Aurora MySQL データベースエンジンの更新 2020-11-12 (バージョン 2.08.3) (廃止)	2.08.3	• バグ #23762382 - INSERT VALUES QUERY WITH JOIN IN A SELECT CAUSES INCORRECT BEHAVIOR。 • バグ #25209512 - CURRENT_TIMESTAMP PRODUCES ZEROS IN TRIGGER。

データベースエンジンの更新	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンの更新 2020-06-02 (バージョン 2.08.0) (廃止)	2.08.0	• バグ #25289359: フルテキストキャッシュサイズがフルテキストキャッシュサイズの制限を超えた場合、データの同期時に行われるフルテキストキャッシュロックが解放されませんでした。 • バグ #29138644: MySQL サーバーの実行中にシステム時刻を手動で変更すると、ページクリーナーのスレッドの遅延が発生しました。 • バグ #25222337: 仮想インデックスの仮想列フィールド名が NULL の場合、外部キー制約の影響を受ける仮想列への入力中に発生するフィールド名の比較中にサーバー終了が発生しました。 • バグ #25053286: ビューにアクセスしたクエリを含むストアドプロシージャを実行すると、セッションが終了するまで解放されなかったメモリが割り当てられました。 • バグ #25586773: 特定の SELECT ステートメントの内容からテーブルを作成したステートメントを含むストアドプロシージャを実行すると、メモリリークが発生する可能性があります。 • バグ #28834208: ログアプリケーション中、OPTIMIZE TABLE オペレーション後に、InnoDB が仮想列のインデックスの更新をチェックする前に仮想列を挿入しませんでした。 • バグ #26666274: 32 ビット符号なし整数オーバーフローが原因で、Performance Schemaバッファコンテナに無限ループが発生しました。
Aurora MySQL データベースエンジンの更新 2022-06-16 (バージョン 2.07.8) (廃止)	2.07.8	プライマリキーが 1024 バイト超の一時テーブルを UPDATE が必要とし、そのテーブルが InnoDB を使用して作成された場合、サーバーが終了する可能性があります。(バグ #25153670)

データベースエンジンの更新	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンの更新 2021-09-02 (バージョン 2.07.6) (廃止)	2.07.6	• INSERTING 64K SIZE RECORDS TAKE TOO MUCH TIME. (64K サイズのレコードの挿入に時間がかかりすぎています) (バグ #23031146)
Aurora MySQL データベースエンジンの更新 2021-03-04 (バージョン 2.07.4) (廃止)	2.07.4	• ' ' (スペース)、'%'、',' を含むトークンを処理するときにフルテキスト ngram パーサーで発生する問題を修正しました。ngram パーサーを使用する場合は、FTS インデックスを再構築する必要があります。(バグ #25873310) • ネストされた SQL ビューでクエリを実行しているときにエンジンを再起動することがある問題を修正しました。(バグ #27214153、バグ #26864199)

データベースエンジンの更新	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンの更新 2020-11-10 (バージョン 2.07.3) (廃止)	2.07.3	• InnoDB: マスターの XA 準備段階に正常に実行された同時 XA トランザクションは、スレーブで再生されたときに競合し、アプライヤスレッドでロック待機タイムアウトが発生しました。競合は、トランザクションがスレーブ上で連続して再生された際のギャップロック範囲が異なることによって発生します。このタイプの競合を防ぐために、READ COMMITTED の分離レベルの XA トランザクションによる ギャップロックは、XA トランザクションが準備段階に達したときに解放される (継承されなくなる) ようになりました。(バグ #27189701、バグ #25866046) • InnoDB: READ COMMITTED の分離レベルの使用中、外部キーの検証で不要なギャップロックが実行されました。(バグ #25082593) • レプリケーション: XA トランザクションを使用中、レプリケーションスレーブ上の applier (SQL) スレッドでロック待機タイムアウトまたはデッドロックが発生した場合、自動再試行は機能しません。これは、SQL スレッドがロールバックを行う間、XA トランザクションをロールバックしないことが原因でした。トランザクションが再試行された際の初期のイベントは XA START であり、XA トランザクションがすでに進行中だったために無効となり、XAER_RMFAIL エラーが発生したということです。(バグ #24764800) • レプリケーション: トランザクション分離レベルが REPEATABLE READ に設定されている場合、インターリーブトランザクションがスレーブアプライヤをデッドロックすることがありました。(バグ #25040331) • レプリケーション: すべての既存のリレーログファイル (Relay_Log_Space) の合計サイズに対する SHOW SLAVE STATUS ステートメントによって返される値は、リレーログファイルによって使用される実際のディスク容量よりもはるかに大きくなる可能性があります。I/O スレッドが値を更新する間に可変をロックしなかったため、SQL スレッドはリレーログファイルを自動的に削除し、I/O スレッドが値の

データベースエンジンの更新	バージョン	修正された MySQL のバグ
		更新を完了する前に低減された値を書き込むことができました。その後 I/O スレッドは元のサイズ計算を書き込み、SQL スレッドの更新を無視して削除したファイルの容量を再追加しました。同時更新を防ぎ正確に計算するため、更新中の Relay_Log_Space の値はロックされます。(バグ #26997096、バグ #87832) • VALUES リストが結合を含むサブクエリを使用して 2 行目以降の値を生成した INSERT ステートメントの場合、必要な権限の解決に失敗するとサーバーが終了する可能性があります。(バグ #23762382) • デフォルト値の CURRENT_TIMESTAMP が書き込まれた TIMESTAMP または DATETIME 列を持つテーブルで、そのテーブルに `BEFORE INSERT` トリガーがある場合は、これらの列が `0000-00-00 00:00:00` に初期化されることがあります。(バグ #25209512、バグ #84077) • メタデータのPerformance Schemaオブジェクトの登録と登録解除を複数のスレッドが同時に試行すると、サーバーが終了する可能性があります。(バグ #26502135) • 特定の SELECT ステートメントの内容からテーブルを作成したステートメントを含むストアドプロシージャを実行すると、メモリリークが発生する可能性があります。(バグ #25586773) • ビューにアクセスしたクエリを含むストアドプロシージャを実行すると、セッションが終了するまで解放されなかったメモリが割り当てられる可能性があります。(バグ #25053286) • サブクエリのマテリアル化の特定のケースで、サーバーが終了する可能性があります。これらのクエリは、マテリアル化が無効であることを示すエラーを生成するようになりました。(バグ #26402045) • 結合バッファリングが使用されている場合、多くの左結合を持つクエリが低速になります (ブロックネストされたループ

データベースエンジンの更新	バージョン	修正された MySQL のバグ
		アルゴリズムを使用する場合など)。(バグ #18898433、バグ #72854) • オプティマイザは、2 番目の列に対して LIKE の句との内部結合を実行する際、複合インデックスの 2 番目の列をスキップしました。(バグ #28086754)
Aurora MySQL データベースエンジンの更新 2020-04-17 (バージョン 2.07.2) (廃止)	2.07.2	• バグ #23104498: メモリ使用量の合計を報告する際のPerformance Schemaの問題を修正しました。(https://github.com/mysql/mysql-server/commit/20b6840df5452f47313c6f9a6ca075bfbc00a96b) • バグ #22551677: オフラインにしようとするとデータベースエンジンがクラッシュする可能性があったPerformance Schemaの問題を修正しました。(https://github.com/mysql/mysql-server/commit/05e2386eccd32b6b444b900c9f8a87a1d8d531e9) • バグ #23550835、バグ #23298025、バグ #81464: 内部バッファの容量を超えたことが原因で、データベースエンジンがクラッシュしていたPerformance Schemaの問題を修正しました。(https://github.com/mysql/mysql-server/commit/b4287f93857bf2f99b18fd06f555bbe5b12debfc、https://github.com/mysql/mysql-server/commit/b4287f93857bf2f99b18fd06f555bbe5b12debfc)

データベースエンジンの更新	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンの更新 2019-11-25 (バージョン 2.07.0) (廃止)	2.07.0	• バグ #26251621: INCORRECT BEHAVIOR WITH TRIGGER AND GCOL • バグ #22574695: ASSERTION `!TABLE || (!TABLE-> READ_SET || BITMAP_IS_SET(TABLE->READ_SET, FIEL • バグ #25966845: INSERT ON DUPLICATE KEY GENERATE A DEADLOCK • バグ #23070734: CONCURRENT TRUNCATE TABLES CAUSE STALL • バグ #26191879: FOREIGN KEY CASCADES USE EXCESSIVE MEMORY • バグ #20989615: INNODB AUTO_INCREMENT PRODUCES SAME VALUE TWICE
Aurora MySQL データベースエンジンの更新 2019-11-11 (バージョン 2.05.0) (廃止)	2.05.0	• バグ #23054591: PURGE BINARY LOGS TO がバイナリログファイル全体を読み取り、処理が停止する MySql

データベースエンジンの更新	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンの更新 2020-08-14 (バージョン 2.04.9) (廃止)	2.04.9	• バグ #23070734、バグ #80060: Concurrent TRUNCATE TABLEs cause stalls • バグ #23103937: PS_TRUNCATE_ALL_TABLES() が SUPER_READ_ONLY モードで機能しない • バグ #22551677: サーバーをオフラインにすると、Performance Schema内の競合状態により、サーバーが終了する可能性があります。 • バグ #27082268: 無効な FTS の同期。 • バグ #12589870: クエリキャッシュが有効なときにマルチクエリステートメントで再起動が発生する問題を修正しました。 • バグ #26402045: サブクエリのマテリアル化の特定のケースで、サーバーが終了する可能性があります。これらのクエリは、マテリアル化が無効であることを示すエラーを生成するようになりました。 • バグ #18898433: 結合バッファリングが使用されている場合 (ブロックネストされたループアルゴリズムを使用する場合など)、多くの左結合を持つクエリが低速になります。 • バグ #25222337: 仮想インデックスの仮想列フィールド名が NULL の場合、外部キー制約の影響を受ける仮想列への入力中に発生するフィールド名の比較中にサーバー終了が発生しました。(https://github.com/mysql/mysql-server/commit/273d5c9d7072c63b6c47dbef6963d7dc491d5131) • バグ #25053286: ビューにアクセスしたクエリを含むストアドプロシージャを実行すると、セッションが終了するまで解放されなかったメモリが割り当てられました。(https://github.com/mysql/mysql-server/commit/d7b37d4d141a95f577916448650c429f0d6e193d) • バグ #25586773: 特定の SELECT (https://dev.mysql.com/doc/refman/5.7/en/select.html) ステートメントの内容からテーブルを作成したステートメントを含むストアドプロシージャを実行すると、メモリリークが発生する可能性があり

データベースエンジンの更新	バージョン	修正された MySQL のバグ
		ます。(https://github.com/mysql/mysql-server/commit/88301e5adab65f6750f66af284be410c4369d0c1) • バグ #26666274: Performance Schemaバッファコンテナ内の無限ループ • バグ #23550835、バグ #23298025、バグ #81464: 内部バッファがいっぱいになったときに SELECT Performance Schemaテーブルによってサーバーが終了する可能性があります。
Aurora MySQL データベースエンジンの更新 2019-09-19 (バージョン 2.04.6) (廃止)	2.04.6	• バグ #23054591: PURGE BINARY LOGS TO がバイナリログファイル全体を読み取っているため、処理が停止する MySql
Aurora MySQL データベースエンジンの更新 2019-05-02 (バージョン 2.04.2) (廃止)	2.04.2	Bug #24829050 - INDEX_MERGE_INTERSECTION OPTIMIZATION CAUSES WRONG QUERY RESULTS
Aurora MySQL データベースエンジンの更新 2018-10-11 (バージョン 2.03) (廃止)	2.03	• パーティショニングされたテーブルの逆スキャンが、ICP を行う - ORDER BY DESC (バグ #24929748)。 • JSON_OBJECT は無効な JSON コードを作成する (バグ #26867509)。 • 大規模な JSON データを挿入すると、膨大な時間がかかる (バグ #22843444)。 • パーティショニングされたテーブルが 5.6 より 5.7 でより多くのメモリーを使用する (バグ #25080442)。

データベースエンジンの更新	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンの更新 2018-09-21 (バージョン 2.02.4) (廃止)	2.02.4	• 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(!VIC TIM_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

データベースエンジンの更新	バージョン	修正された MySQL のバグ
		• 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
Aurora MySQL データベースエンジンの更新 2018-05-03 (バージョン 2.02) (廃止)	2.02.0	左結合は外部側で不正な結果を返します (バグ #22833364)。

Aurora MySQL 1.x データベースエンジンの更新で修正された MySQL のバグ

MySQL 5.6 と互換性のあるバージョンの Aurora には、MySQL 5.6.10 によるすべての MySQL バグ修正が含まれています。次の表は、Aurora MySQL データベースエンジンの更新で修正された追加の MySQL のバグと、どの更新で修正されたかを示しています。

データベースエンジンの更新	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンの更新 2021-03-18 (バージョン 1.23.2) (廃止)	1.23.2	• レプリケーション: SHOW BINLOG EVENTS 文の実行中に、パラレルトランザクションがブロックされました。この修正により、SHOW BINLOG EVENTS プロセスはファイルの終了位置を計算する間だけロックを取得するため、パラレルトランザクションが長期間ブロックされることはありません。(バグ #76618、バグ #20928790)
Aurora MySQL データベースエンジンの更新	1.23.0	• ALTER TABLE ADD COLUMN ALGORITHM=QUICK のバイナリログイベントは、コミュニティ版と互換性を持つように ALGORITHM=DEFAULT として書き換えられます。

データベースエンジンの更新	バージョン	修正された MySQL のバグ
2020-09-02 (バージョン 1.23.0) (廃止)		• バグ #22350047: IF CLIENT KILLED AFTER ROLLBACK TO SAVEPOINT PREVIOUS STMTS COMMITTED • バグ #29915479: RUNNING COM_REGISTER_SLAVE WITHOUT COM_BINLOG_DUMP CAN RESULTS IN SERVER EXIT • バグ #30441969: BUG #29723340: MYSQL SERVER CRASH AFTER SQL QUERY WITH DATA ?AST • バグ #30628268: OUT OF MEMORY CRASH • バグ #27081349: UNEXPECTED BEHAVIOUR WHEN DELETE WITH SPATIAL FUNCTION • バグ #27230859: UNEXPECTED BEHAVIOUR WHILE HANDLING INVALID POLYGON" • バグ #27081349: UNEXPECTED BEHAVIOUR WHEN DELETE WITH SPATIAL" • バグ #26935001: ALTER TABLE AUTO_INCREMENT TRIES TO READ INDEX FROM DISCARDED TABLESPACE • バグ #29770705: SERVER CRASHED WHILE EXECUTING SELECT WITH SPECIFIC WHERE CLAUSE • バグ #27659490: SELECT USING DYNAMIC RANGE AND INDEX MERGE USE TOO MUCH MEMORY(OOM) • バグ #24786290: REPLICATION BREAKS AFTER BUG #74145 HAPPENS IN MASTER • バグ #27703912: EXCESSIVE MEMORY USAGE WITH MANY PREPARE • バグ #20527363: TRUNCATE TEMPORARY TABLE CRASH: !DICT_TF2_FLAG_IS_SET(TABLE, DICT_TF2_TEMPORARY) • バグ #23103937: PS_TRUNCATE_ALL_TABLES() が SUPER_READ_ONLY モードで機能しない

データベースエンジンの更新	バージョン	修正された MySQL のバグ
		• バグ #25053286: USE VIEW WITH CONDITION IN PROCEDURE CAUSES INCORRECT BEHAVIOR (fixed in 5.6.36) • バグ #25586773: INCORRECT BEHAVIOR FOR CREATE TABLE SELECT IN A LOOP IN SP (fixed in 5.6.39) • バグ #27407480: AUTOMATIC_SP_PRIVILEGES REQUIRES NEED THE INSERT PRIVILEGES FOR MYSQL.USER TABLE • バグ #26997096: `relay_log_space` 値は同期された方法で更新されないため、その値はリレーログで使用される実際のディスク領域よりもはるかに大きくなることがあります。 • バグ#15831300 SLAVE_TYPE_CONVERSIONS=ALL_NON_LOSSY NOT WORKING AS EXPECTED • SSL バグのバックポート: バグ #17087862、バグ #20551271 • バグ #16894092: PERFORMANCE REGRESSION IN 5.6.6+ FOR INSERT INTO ..。SELECT ..。FROM (5.6.15 で修正)。 • `SLAVE_TYPE_CONVERSIONS` に関連するバグ修正を移植します。

データベースエンジンの更新	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンの更新 2020-11-09 (バージョン 1.22.3) (廃止)	1.22.3	• バグ #26654685: 外部キーチェック中に破損したインデックス ID が検出され、アサーションが発生しました • バグ #15831300: デフォルトでは、整数をマスター上の小さいタイプからスレーブ上のより大きなタイプ (例えば、マスターの SMALLINT 列からスレーブの BIGINT 列へ) に昇格すると、昇格された値は署名付きのように処理されます。このような場合、サーバーシステム可変 slave_type_converss に指定された値のセットで、ALL_SIGNED および ALL_UNSIGNED の一方または両方を使用してこの動作を変更またはオーバーライドすることができます。詳細については、「行ベースのレプリケーション: 属性の昇格と降格」、および可変についての説明を参照してください。 • バグ #17449901: foreign_key_checks=0 で、InnoDB は外部キー制約で必要なインデックスを削除することを許可し、テーブルを不整合に配置したため、テーブルのロード時に発生する外部キーチェックに失敗しました。InnoDB で、foreign_key_checks=0 であっても、外部キーの制約により必要なインデックスが削除できなくなりました。外部キーの制約は、外部キーのインデックスを削除するよりも前に削除する必要があります。 • バグ #20768847: ALTER TABLE ... 外部キーの依存関係を持つテーブルでの DROP INDEX のオペレーションによってアサーションが発生しました。

データベースエンジンの更新	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンの更新 2019-11-25 (バージョン 1.22.0) (廃止)	1.22.0	• バグ #16346241 - SERVER CRASH IN ITEM_PARAM::QUERY_VAL_STR • バグ #17733850 - NAME_CONST() CRASH IN ITEM_NAME_CONST::ITEM_NAME_CONST() • バグ #20989615 - INNODB AUTO_INCREMENT PRODUCES SAME VALUE TWICE • バグ #20181776 - ACCESS CONTROL DOESN'T MATCH MOST SPECIFIC HOST WHEN IT CONTAINS WILDCARD • バグ #27326796 - MYSQL CRASH WITH INNODB ASSERTION FAILURE IN FILE PARS0PARS.CC • バグ #20590013 - IF YOU HAVE A FULLTEXT INDEX AND DROP IT YOU CAN NO LONGER PERFORM ONLINE DDL
Aurora MySQL データベースエンジンの更新 2019-11-25 (バージョン 1.21.0) (廃止)	1.21.0	• バグ #19929406: HANDLE_FATAL_SIGNAL (SIG=11) IN __MEMMOVE_SSSE3_BACK FROM STRING::COPY • バグ #17059925: UNION ステートメントに、行検証の値が正しく計算されませんでした。これは、Performance Schema ステートメントテーブル (events_statements_current など) の `ROWS_EXAMINED` 列に対して大きすぎる値として現れていました。 • バグ #11827369: `SELECT ... FROM DUAL` ネストされたサブクエリのいくつかのクエリでアサーションが発生しました。 • バグ #16311231: `IN` 句の XOR オペレーションを含む `WHERE` 句でクエリがサブクエリを含む場合に、正しくない結果が返されました。

データベースエンジンの更新	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンの更新 2019-11-11 (バージョン 1.20.0) (廃止)	1.20.0	• バグ #19929406: HANDLE_FATAL_SIGNAL (SIG=11) IN __MEMMOVE_SSSE3_BACK FROM STRING::COPY • バグ #17059925: UNION ステートメントに、行検証の値が正しく計算されませんでした。これは、Performance Schema ステートメントテーブル (events_statements_current など) の ROWS_EXAMINED 列に対して大きすぎる値として現れていました。 • バグ #11827369: SELECT ... FROM DUAL ネストされたサブクエリのいくつかのクエリでアサーションが発生しました。 • バグ #16311231: IN 句の XOR オペレーションを含む WHERE 句でクエリがサブクエリを含む場合に、正しくない結果が返されました。
Aurora MySQL データベースエンジンの更新 2019-09-19 (バージョン 1.19.5) (廃止)	1.19.5	• CVE-2018-2696 • CVE-2015-4737 • バグ #19929406: HANDLE_FATAL_SIGNAL (SIG=11) IN __MEMMOVE_SSSE3_BACK FROM STRING::COPY • バグ #17059925: UNION ステートメントに、行検証の値が正しく計算されませんでした。Performance Schema ステートメントテーブル (events_statements_current など) の ROWS_EXAMINED 列の値が大きすぎることに関するマニフェストでした。 • バグ #11827369: SELECT ... FROM DUAL ネストされたサブクエリのいくつかのクエリでアサーションが発生しました。 • バグ #16311231: IN 句の XOR オペレーションを含む WHERE 句でクエリがサブクエリを含む場合に、正しくない結果が返されました。

データベースエンジンの更新	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンの更新 2019-02-07 (バージョン 1.19.0) (廃止)	1.19.0	• BUG #32917: DETECT ORPHAN TEMP-POOL FILES, AND HANDLE GRACEFULLY • BUG #63144 CREATE TABLE IF NOT EXISTS METADATA LOCK IS TOO RESTRICTIVE
Aurora MySQL データベースエンジンの更新 2019-01-17 (バージョン 1.17.8) (廃止)	1.17.8	• バグ #13418638: CREATE TABLE IF NOT EXISTS METADATA LOCK IS TOO RESTRICTIVE
Aurora MySQL データベースエンジンの更新 2018-10-08 (バージョン 1.17.7) (廃止)	1.17.7	• 外部キー列のインデックスを削除すると、テーブルが見つからなくなります。(バグ #16208542) • add_derived_key() のメモリリーク。(バグ #76349) • パーティショニングされたテーブルの場合、インデックスマージが使用されたかどうかによって、クエリが異なる結果を返すことがありました。(バグ #16862316) • HASH でパーティショニングされたテーブルに対して実行すると、index_merge 最適化 (インデックスマージの最適化を参照) を使用するクエリが無効な結果を返す場合があります。(バグ #17588348)
Aurora MySQL データベースエンジンの更新 2018-09-06 (バージョン 1.17.6) (廃止)	1.17.6	• BINARY 列のデフォルト値の名称変更または変更した ALTER TABLE ステートメントの場合、変更はテーブルコピーを使用して実行されており、適切ではありません。(バグ #67141、バグ #14735373、バグ #69580、バグ #17024290) • 暗黙的にグループ化された通常のテーブルと派生したテーブルの間の外部結合によって、サーバーが終了する可能性があります。(バグ #16177639)

データベースエンジンの更新	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンの更新 2018-03-13 (バージョン 1.17) (廃止)	1.17.0	• レプリケーションフィルタが使用されていると LAST_INSERT_ID が不正にレプリケートされる (バグ #69861) • クエリは、INDEX_MERGE 設定の有無に応じて異なる結果を返す (バグ #16862316) • ストアドルーチン、非効率なクエリプランのクエリ処理の再実行 (バグ #16346367) • InnoDB FTS : FTS_CACHE_APPEND_DELETED_DOC_IDS のアサート (バグ #18079671) • ALTER TABLE CHANGE COLUMN の RBT_EMPTY (INDEX_CACHE->WORDS) のアサート (バグ #17536995) • 保存ポイントが関わる場合に InnoDB 全文検索でレコードが見つからない (バグ #70333、バグ #17458835)
Aurora MySQL データベースエンジンの更新 2017-11-20 (バージョン 1.15.1) (廃止)	1.15.1	• 元に戻す — MySQL インスタンスで停止される “doing SYNC index” (バグ #73816) • 元に戻す — ALTER TABLE CHANGE COLUMN の RBT_EMPTY(INDEX_CACHE->WORDS) のアサート (バグ #17536995) • 元に戻す — 保存ポイントが関わる場合に InnoDB 全文検索でレコードが見つからない (バグ #70333)

データベースエンジンの更新	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンの更新 2017-10-24 (バージョン 1.15) (廃止)	1.15.0	• CREATE USER はプラグインおよびパスワードハッシュを受け入れるが、パスワードハッシュを無視する (バグ #78033) • パーティションエンジンは、読み取りビットのセットにフィールドを追加し、パーティションインデックスからソートされたエントリを返せるようにします。これにより、結合バッファでは不要なフィールドまで読み取ろうとします。すべてのパーティションフィールドを read_set に追加するのではなく、read_set の設定済みのプレフィックスフィールドでのみソートするように修正しました。DBUG_ASSERT を追加し、key_cmp を行う場合、少なくとも初期のフィールドが必ず読み取られるようにしました (バグ #16367691) • MySQL インスタンスで停止される “doing SYNC index” (バグ #73816) • ALTER TABLE CHANGE COLUMN の RBT_EMPTY (INDEX_CACHE->WORDS) のアサート (バグ #17536995) • 保存ポイントが関わる場合に InnoDB 全文検索でレコードが見つからない (バグ #70333)
Aurora MySQL データベースエンジンの更新: 2018-03-13 (バージョン 1.14.4) (廃止)	1.14.4	• 無視できるイベントが動作せず、テストされない (バグ #74683) • NEW->OLD ASSERT FAILURE 'GTID_MODE > 0' (バグ #20436436)
Aurora MySQL データベースエンジンの更新: 2017-08-07 (バージョン 1.14) (廃止)	1.14.0	派生テーブル (FROM 句のサブクエリ) と結合された全文検索では、サーバーが終了しました。ここで、全文操作が派生テーブルに依存する場合、サーバーは、マテリアライズされたテーブルで全文検索を実行できないことを示すエラーを生成します。(バグ #68751、バグ #16539903)

データベースエンジンの更新	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンの更新: 2017-05-15 (バージョン 1.13) (廃止)	1.13.0	• 空のときに削除されたテーブルをリロードすると、AUTO INCREMENT 値がリセットされました。(バグ #21454472、バグ #77743) • purge_node_t 構造の不一致により、ロールバック時にインデックスレコードが見つかりませんでした。この不一致により、「秒インデックスエントリの更新でエラーが発生しました」、「レコードをパージできませんでした」、「削除用にマークされていない秒インデックスエントリをパージしようとしました」などの警告やエラーメッセージが表示されました。(バグ #19138298、バグ #70214、バグ #21126772、バグ #21065746) • qsort オペレーションのスタックサイズの計算が正しくないと、スタックのオーバーフローが発生します。(バグ #73979) • ロールバック時にインデックスにレコードが見つかりませんでした。(バグ #70214、バグ #72419) • CURRENT_TIMESTAMP 更新時に ALTER TABLE の列 TIMESTAMP を追加すると、ZERO-datas が挿入されます (バグ #17392)

データベースエンジンの更新	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンの更新: 2017-04-05 (バージョン 1.12) (廃止)	1.12.0	• 空のときに削除されたテーブルをリロードすると、AUTO INCREMENT 値がリセットされました。(バグ #21454472、バグ #77743) • purge_node_t 構造の不一致により、ロールバック時にインデックスレコードが見つかりませんでした。この不一致により、「秒インデックスエントリの更新でエラーが発生しました」、「レコードをパージできませんでした」、「削除用にマークされていない秒インデックスエントリをパージしようとしました」などの警告やエラーメッセージが表示されました。(バグ #19138298、バグ #70214、バグ #21126772、バグ #21065746) • qsort オペレーションのスタックサイズの計算が正しくないと、スタックのオーバーフローが発生します。(バグ #73979) • ロールバック時にインデックスにレコードが見つかりませんでした。(バグ #70214、バグ #72419) • CURRENT_TIMESTAMP 更新時に ALTER TABLE の列 TIMESTAMP を追加すると、ZERO-datas が挿入されます (バグ #17392)
Aurora MySQL データベースエンジンの更新: 2017-02-23 (バージョン 1.11) (廃止)	1.11.0	• 別の DROP オペレーションと同時に ALTER テーブルの DROP 外部キーを実行すると、テーブルがなくなる。(バグ #16095573) • ORDER BY を使用した一部の INFORMATION_SCHEMA クエリで、以前と同じように filesort 最適化が使用されない。(バグ #16423536) • FOUND_ROWS () によって返されるテーブルの行数が正しくない。(バグ #68458) • 開いているテンポラリテーブルが多すぎると、エラーが発生する代わりに、サーバーに障害が発生する。(バグ #18948649)

データベースエンジンの更新	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンの更新: 2016-12-14 (バージョン 1.10) (廃止)	1.10.0	• 派生テーブルの UNION は、「1=0/false」句のある不正な結果を返します。(バグ #69471) • サーバーはストアドプロシージャの 2 回目の実行の際、ITEM_FUNC_GROUP_CONCAT::FIX_FIELDS でクラッシュします。(バグ #20755389) • キャッシュのサイズが合計サイズの 10% を超えたらすぐに、別のスレッドにキャッシュ同期タスクをオフロードすることで、FTS キャッシュのディスクへの同期中に MySQL のクエリが長時間停止することを回避できます。(バグ #22516559、#73816)
Aurora MySQL データベースエンジンの更新: 2016-10-26 (バージョン 1.8.1) (廃止)	1.8.1	• この問題により、OpenSSL はディフィー・ヘルマンのキー長パラメーターを変更しました。 LogJam (バグ #18367167)
Aurora MySQL データベースエンジンの更新: 2016-10-18 (バージョン 1.8) (廃止)	1.8.0	• 複数のインデックスがある列ですべてのインデックスを削除する場合、外部キーの制約に基づいてインデックスが必要なときに、InnoDB は DROP INDEX オペレーションをブロックできませんでした。(バグ #16896810) • 外部キーの制約に伴うクラッシュの解決策を追加しました。(バグ #16413976) • ストアドプロシージャでカーソルを取得し、同時にテーブルを分析またはフラッシュするときに発生するクラッシュを修正しました。(バグ #18158639) • テーブルを変更して AUTO_INCREMENT の値を自動インクリメント列の最大値より小さくしたときに発生する自動インクリメントバグを修正しました。(バグ #16310273)

データベースエンジンの更新	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンの更新: 2016-08-30 (バージョン 1.7.0) (廃止)	1.7.0	• LOCK_grant ロックのパーティショニングにより安定性を向上しました。(ポート WL #8355) • ストアドプロシージャの SELECT でカーソルを開くとセグメンテーション違反が発生します。(ポートのバグ #16499751) • MySQL のいくつかの特別な使用法で、誤った結果となります。(バグ #11751794) • GET_SEL_ARG_FOR_KEYPART でのクラッシュ - バグ #11751794 のパッチにより発生します。(バグ #16208709) • GROUP BY を使った単純なクエリで誤った結果となります。(バグ #17909656) • 範囲の述語を使った半結合 (semi-join) クエリで、余分な行が返されます。(バグ #16221623) • IN サブクエリの後に ORDER BY 句を追加すると、重複列が返されます。(バグ #16308085) • MyISAM で GROUP BY によるルーススキャンのクエリで EXPLAIN を使うとクラッシュします。(バグ #16222245) • 引用符付きの INT 述語を使ったルースインデックススキャンでランダムなデータが返されます。(バグ #16394084) • オプティマイザがルースインデックススキャンを使用した場合、サーバーがテンポラリテーブルを作成しようとするときに終了する場合があります。(バグ #16436567) • COUNT(DISTINCT) は NULL 値をカウントするべきではありませんが、オプティマイザがルースインデックススキャンを使用したときにカウントされます。(バグ #17222452) • クエリが MIN()/MAX() の両方と集計関数 (DISTINCT) (例えば SUM(DISTINCT)) を含む場合に、ルースインデックススキャンを使って実行されると、MIN()/MAX() の結果の値が正しく設定されません。(バグ #17217128)

データベースエンジンの更新	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンの更新: 2016-06-01 (バージョン 1.6.5) (廃止)	1.6.5	• SLAVE CAN'T CONTINUE REPLICATION AFTER MASTER'S CRASH RECOVERY (Port Bug #17632285)
Aurora MySQL データベースエンジンの更新: 2016-04-06 (バージョン 1.6) (廃止)	1.6.0	• アサーション `!M_ORDERED_REC_BUFFER' 失敗のバグ #18694052 修正を 5.6 にバックポート (ポートのバグ #18305270) • MEMCPY()、HA_PARTITION::POSITION のセグメンテーション違反 (ポートのバグ # 18383840) • PARTITIONING,INDEX_MERGE AND NO PK の結果が正しくない (ポートのバグ # 18167648) • EXPORT: ASSERTION IN HA_PARTITION::EXTRA の FLUSH TABLES (ポートのバグ # 16943907) • 仮想 HA_ROWS HANDLER::MULTI_RANGE_READ_INFO_CONST におけるサーバークラッシュ (ポートのバグ # 16164031) • 範囲オプティマイザが SEL_ARG::RB_INSERT() でクラッシュする (ポートのバグ # 16241773)
Aurora MySQL データベースエンジンの更新: 2016-01-11 (バージョン 1.5) (廃止)	1.5.0	• MySQL の全文検索が、データベース名が数字で始まるテーブルに影響する件を一部修正しました。(ポートのバグ #17607956)

データベースエンジンの更新	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンの更新: 2015-12-03 (バージョン 1.4) (廃止)	1.4	• FTSPARSE() のセグメンテーション違反。(バグ #16446108) • InnoDB データディクショナリが列名を変更しながら更新されない。(バグ #19465984) • 別のデータベースにテーブル名を変更した後に FTS がクラッシュする。(バグ #16834860) • 削除されたテーブルでトリガーの準備が失敗すると エラー 1054 が発生する。(バグ #18596756) • メタデータを変更するとトリガーの実行で問題が発生する場合がある。(バグ #18684393) • 長い UTF8 VARCHAR フィールドに対してマテリアル化が選択されない。(バグ #17566396) • ORDER BY で制限を X にした場合に実行プランのパフォーマンスが悪い (バグ #16697792)。 • バグ #11765744 を 5.1、5.5 および 5.6 にバックポート。(バグ #17083851) • SQL/SQL_SHOW.CC が SIG6 になるミューテックスの問題。ソースが頻繁に FILL_VARIABLES になる。(バグ #20788853) • バグ #18008907 を 5.5 以上のバージョンにバックポート。(バグ #18903155) • MySQL 5.7 にスタックのオーバーフローエラーの修正を適応。(バグ #19678930)

データベースエンジンの更新	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンの更新: 2015-10-16 (バージョン 1.2、1.3) (廃止)	1.2、1.3	• innodb 内でクエリを強制するとアサーションとともにクラッシュすることがある。(バグ #1608883) • イベントスケジューラ、イベント実行、または新規接続で新規スレッドの作成に失敗すると、エラーログにメッセージが書き込まれない。(バグ #16865959) • 1 つの接続がデフォルトのデータベースを変更すると同時に別の接続が SHOW PROCESSLIST を実行した場合、2 番目の接続が 1 番目の接続のデフォルトのデータベースメモリを表示しようとすると無効なメモリにアクセスする。(バグ #11765252) • 設計によるバイナリログの消去が使用中またはアクティブなバイナリログファイルを削除せず、これが発生したときに通知しない。(バグ #13727933) • 一部のステートメントで、オプティマイザが不必要なサブクエリ句を削除するとメモリリークが発生することがある。(バグ #15875919) • シャットダウン時に、サーバーが初期化されていないミューテックスをロックする場合がある。(バグ #16016493) • 複数の列に名前をつける GROUP_CONCAT() および ORDER BY 句を使用した準備済みステートメントが、サーバーを終了させる場合がある。(バグ #16075310) • Performance Schemaの計測がレプリカワーカースレッドで見つからない。(バグ #16083949) • `STOP SLAVE` は、1 つ以上のステータス可変 `Slave_retried_transactions` 、`Slave_heartbeat_period` 、`Slave_received_heartbeats` 、`Slave_last_heartbeat` 、または `Slave_running` の値を取得する SHOW STATUS などのステートメントと同時に発行されると、デッドロックが発生する可能性がある。(バグ #16088188)

データベースエンジンの更新	バージョン	修正された MySQL のバグ
		• 検索する用語が引用句で囲まれた語句の場合に、ブールモードを使用した全文クエリの結果がゼロになることがある。(バグ #16206253) • サブクエリの結合の ON 句でサブクエリを使用して準備済みステートメントを実行する際に、オプティマイザが冗長サブクエリ句を削除しようとすると、アサーションが立ち上がる。(バグ #16318585) • GROUP_CONCAT が不安定、ITEM_SUM::CLEAN_UP_AFTER_REMOVAL でクラッシュが発生する。(バグ #16347450) • デフォルトの InnoDB 全文検索 (FTS) ストップワードリストを、INFORMATION_SCHEMA.INNODB_FT_DEFAULT_STOPWORD と同じ構造を持つ InnoDB テーブルを作成することで置き換えようとすると、エラーになる。(バグ #16373868) • ワーカーのクライアントスレッドが FLUSH TABLES WITH READ LOCK を実行した後、続いてマスターで更新があると、`SHOW SLAVE STATUS` を実行したときにワーカーがハングアップする。(バグ #16387720) • 全文検索で「abc-def」などの区切り検索文字列を分析する場合、InnoDB で MyISAM と同じ単語区切り記号を使用するようになりました。(バグ #16419661) • FTS_AST_TERM_SET_WILDCARD がクラッシュする。(バグ #16429306) • FTS RQG テストの FTS_AST_VISIT() のセグメンテーション違反。(バグ #16435855) • デバッグを構築するために、オプティマイザがサブクエリを指す Item_ref を削除すると、サーバーが終了する。(バグ #16509874) • InnoDB テーブルの全文検索が、リテラル句を + または - 演算子と組み合わせた検索に失敗する。(バグ #16516193) • `START SLAVE--master-info-repository=TABLE relay-log-info-repository=TABLE` オプションを指定し

データベースエンジンの更新	バージョン	修正された MySQL のバグ
		、autocommit を 0 に設定してサーバーを起動すると、失敗しました。`--skip-slave-start` (バグ #16533802) • 非常に大きい InnoDB 全文検索 (FTS) 結果でメモリの消費が大きすぎる。(バグ #16625973) • デバッグの構築で、検索文字列に直接バイナリを使用すると、バイナリに NULL バイトや意味のない文字が含まれている場合があるため、OPT_CHECK_ORDER_BY でアサーションが発生する。(バグ #16766016) • 一部のステートメントで、オプティマイザが不必要なサブクエリ句を削除するとメモリリークが発生することがある。(バグ #16807641) • ワーカーへの新規接続で `STOP SLAVE` を発行した後に元の接続を使用して `SHOW SLAVE STATUS` を発行した場合、FLUSH TABLES WITH READ LOCK を発行した後にデッドロックが発生する可能性がある。(バグ #16856735) • 無効な区切り記号を使用した GROUP_CONCAT でサーバーが終了する場合がある。(バグ #16870783) • サーバーが SHOW STATUS LIKE 'pattern' ステートメントの LOCK_active_mi および active_mi-> rli-> data_lock ミューテックスを過度にロックする。パターンがミューテックス (`Slave_heartbeat_period` 、`Slave_last_heartbeat` 、`Slave_received_heartbeats` 、`Slave_retried_transactions` 、`Slave_running`) を使用するステータス可変と一致しなかった場合でも、同じ結果になる。(バグ #16904035) • ブールモード修飾子を使用した全文検索がアサーションを伴う失敗になる。(バグ #16927092) • + ブール演算子を使用した検索で InnoDB テーブルの全文検索が失敗する。(バグ #17280122) • 4 ウェイデッドロック: ゾンビ、バイナリログの破棄、プロセスリストの表示、バイナリログの表示。(バグ #17283409)

データベースエンジンの更新	バージョン	修正された MySQL のバグ
		• コミットロックを待っている SQL スレッドが強制終了され再びスタートされると、トランザクションがワーカーにスキップされる。(バグ #17450876) • 「unended」トークンが原因で InnoDB 全文検索失敗が発生する。文字列および文字列長を、文字列比較に渡す必要があります。(バグ #17659310) • 多数のパーティショニングされた InnoDB テーブルを MySQL 5.6 または 5.7 で使用すると、MySQL サーバーの以前のリリースで同じテーブルを使用したときよりも多くのメモリを消費する。(バグ #17780517) • 全文クエリで、num_token が max_proximity_item より少ないことを確認しようとして失敗すると、アサーションが発生する。(バグ #18233051) • INFORMATION_SCHEMA TABLES および COLUMNS テーブルの特定のクエリが、多量の空の InnoDB がある場合に過剰にメモリを使用する。(バグ #18592390) • トランザクションをコミットするときに、特に master_info_repository=TABLE でサーバーを実行している場合により多くのリソースを使用していたスレッドそのものを確認する方法ではなく、フラグを使用してスレッドが作成されたかどうかを確認するようになりました。(バグ #18684222) • マスターが DML を実行している間にワーカーのクライアントスレッドが FLUSH TABLES WITH READ LOCK を実行する場合、同じクライアントで `SHOW SLAVE STATUS` を実行するとブロックされ、デッドロックが発生する。(バグ #19843808) • GROUP_CONCAT() で順序付けを行うと、サーバーが終了する場合がある。(バグ #19880368)

データベースエンジンの更新	バージョン	修正された MySQL のバグ
Aurora MySQL データベースエンジンの更新: 2015-08-24 (バージョン 1.1) (廃止)	1.1	• 名前の先頭が数字の InnoDB データベースでは、全文検索 (FTS) パーサーエラーが発生します。(バグ #17607956) • InnoDB 全文検索は、名前の尖塔が数字のデータベースでは失敗します。(バグ #17161372) • Windows 上の InnoDB データベースの場合、全文検索 (FTS) オブジェクト ID が期待される 16 進形式ではありません。(バグ #16559254) • MySQL 5.6 に導入されたコード回帰は、DROP TABLE と ALTER TABLE のパフォーマンスにマイナスの影響を与えました。このため、MySQL Server 5.5.x から 5.6.x の間でパフォーマンスが低下する可能性があります。(バグ #16864741)

Aurora MySQL で修正されたセキュリティの脆弱性

一般的な脆弱性と露出 (CVE) は、一般的に知られているサイバーセキュリティの脆弱性に関するエントリの一覧です。各エントリには、識別番号、説明、および少なくとも 1 つの公開リファレンスが含まれています。

このページには、Aurora MySQL で修正されたセキュリティの脆弱性に関するリストがあります。Aurora のセキュリティに関する一般的な情報については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora でのセキュリティ」を参照してください。Aurora MySQL のその他のセキュリティ情報については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL でのセキュリティ」を参照してください。

既知の脆弱性から保護するために、常に最新の Aurora リリースにアップグレードすることをお勧めします。このページを使用して、特定のバージョンの Aurora MySQL に特定のセキュリティの脆弱性に対する修正があるかどうかを確認できます。クラスターにセキュリティ修正プログラムがない場合は、その修正用にアップグレードする必要がある Aurora MySQL バージョンを確認できます。

Aurora MySQL バージョン 1、2、3 で修正された CVE も、該当バージョンのリリースノートに記載されています。

- Amazon Aurora MySQL バージョン 3 のデータベースエンジンの更新
- Amazon Aurora MySQL バージョン 2 のデータベースエンジンの更新
- Amazon Aurora MySQL バージョン 1 のデータベースエンジンの更新 (廃止)

> **Note**
>
> Aurora MySQL バージョン 3 の初期リリースには、コミュニティ MySQL 8.0.23 までに修正されたすべての CVE が含まれています。今後修正される CVE については、こちらの一覧および Aurora MySQL バージョン 3 リリースノートよりお探しください。

CVE および最小固定 Aurora MySQL バージョン

- 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、3.05.1、3.05.0.1、3.04.2、3.03.3、2.12.2、2.12.1、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、2.10.2、2.07.8
- 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 リリースノートのドキュメント履歴

次の表には、Aurora MySQL リリースノートのドキュメントリリースをまとめています。

変更	説明	日付
Aurora MySQL バージョン 2.11.5、MySQL 5.7.12 に対応	Aurora MySQL バージョン 2.11.5 が公開されています。このバージョンは MySQL 5.7.12 と互換性があります。詳細については、「MySQL 5.7 互換 Aurora MySQL バージョン 2」を参照してください。	2024 年 3 月 26 日
Aurora MySQL バージョン 2.12.2、MySQL 5.7.44 に対応	Aurora MySQL バージョン 2.12.2 が公開されています。このバージョンは MySQL 5.7.44 と互換性があります。詳細については、「MySQL 5.7 互換 Aurora MySQL バージョン 2」を参照してください。	2024 年 3 月 19 日
Aurora MySQL バージョン 3.04.2、MySQL 8.0.28 に対応	Aurora MySQL バージョン 3.04.2 が利用可能です。このバージョンは MySQL 8.0.28 と互換性があります。詳細については、MySQL 8.0 と互換性のある Aurora MySQL バージョン 3 を参照してください。	2024 年 3 月 15 日
Aurora MySQL バージョン 3.06.0、MySQL 8.0.34 に対応	Aurora MySQL バージョン 3.06.0 が利用可能です。このバージョンは MySQL 8.0.34 と互換性があります。詳細に	2024 年 3 月 7 日

	ついては、MySQL 8.0 と互換性のある Aurora MySQL バージョン 3 を参照してください。	
Aurora MySQL バージョン 3.05.2、MySQL 8.0.32 に対応	Aurora MySQL バージョン 3.05.2 が利用可能です。このバージョンは MySQL 8.0.32 と互換性があります。詳細については、MySQL 8.0 と互換性のある Aurora MySQL バージョン 3 を参照してください。	2024 年 1 月 31 日
Aurora MySQL バージョン 2.12.1、MySQL 5.7.40 互換	Aurora MySQL バージョン 2.12.1 が利用可能です。このバージョンは MySQL 5.7.40 と互換性があります。詳細については、「MySQL 5.7 互換 Aurora MySQL バージョン 2」を参照してください。	2023 年 12 月 28 日
Aurora MySQL バージョン 3.03.3、MySQL 8.0.26 互換	Aurora MySQL バージョン 3.03.3 が利用可能です。このバージョンは MySQL 8.0.26 と互換性があります。詳細については、MySQL 8.0 と互換性のある Aurora MySQL バージョン 3 を参照してください。	2023 年 12 月 8 日

Aurora MySQL バージョン 3.05.1、MySQL 8.0.32 互換	Aurora MySQL バージョン 3.05.1 が利用可能です。このバージョンは MySQL 8.0.32 と互換性があります。詳細については、MySQL 8.0 と互換性のある Aurora MySQL バージョン 3 を参照してください。	2023 年 11 月 21 日
Aurora MySQL バージョン 3.04.1、MySQL 8.0.28 互換	Aurora MySQL バージョン 3.04.1 が利用可能です。このバージョンは MySQL 8.0.28 と互換性があります。詳細については、MySQL 8.0 と互換性のある Aurora MySQL バージョン 3 を参照してください。	2023 年 11 月 13 日
Aurora MySQL バージョン 3.05.0.1、MySQL 8.0.32 互換、ベータ	Aurora MySQL バージョン 3.05.0.1 が利用可能です。このバージョンは MySQL 8.0.32 と互換性があります。詳細については、MySQL 8.0 と互換性のある Aurora MySQL バージョン 3 を参照してください。	2023 年 10 月 30 日
Aurora MySQL バージョン 3.05.0、MySQL 8.0.32 互換	Aurora MySQL バージョン 3.05.0 が利用可能です。このバージョンは MySQL 8.0.32 と互換性があります。詳細については、MySQL 8.0 と互換性のある Aurora MySQL バージョン 3 を参照してください。	2023 年 10 月 25 日

Aurora MySQL バージョン 2.12.0.1、MySQL 5.7.40 互換、ベータ	Aurora MySQL バージョン 2.12.0.1 ベータが利用可能です。このバージョンは MySQL 5.7.40 と互換性があります。詳細については、「MySQL 5.7 互換 Aurora MySQL バージョン 2」を参照してください。	2023 年 10 月 25 日
Aurora MySQL バージョン 2.11.4、MySQL 5.7.12 互換	Aurora MySQL バージョン 2.11.4 が利用可能です。このバージョンは MySQL 5.7.12 と互換性があります。詳細については、「MySQL 5.7 互換 Aurora MySQL バージョン 2」を参照してください。	2023 年 10 月 17 日
Aurora MySQL バージョン 3.03.2、MySQL 8.0.26 互換	Aurora MySQL バージョン 3.03.2 が利用可能です。このバージョンは MySQL 8.0.26 と互換性があります。詳細については、MySQL 8.0 と互換性のある Aurora MySQL バージョン 3 を参照してください。	2023 年 8 月 29 日
Aurora MySQL バージョン 2.07.10、MySQL 5.7.12 互換	Aurora MySQL バージョン 2.07.10 が利用可能です。このバージョンは MySQL 5.7.12 と互換性があります。詳細については、「MySQL 5.7 互換 Aurora MySQL バージョン 2」を参照してください。	2023 年 8 月 15 日

Aurora MySQL バージョン 3.04.0、MySQL 8.0.28 互換	Aurora MySQL バージョン 3.04.0 が利用可能です。このバージョンは MySQL 8.0.28 と互換性があります。詳細については、MySQL 8.0 と互換性のある Aurora MySQL バージョン 3 を参照してください。	2023 年 7 月 31 日
Aurora MySQL バージョン 2.12.0、MySQL 5.7.40 互換	Aurora MySQL バージョン 2.12.0 が利用可能です。このバージョンは MySQL 5.7.40 と互換性があります。詳細については、「MySQL 5.7 互換 Aurora MySQL バージョン 2」を参照してください。	2023 年 7 月 25 日
Aurora MySQL バージョン 2.11.3、MySQL 5.7.12 互換	Aurora MySQL バージョン 2.11.3 が利用可能です。このバージョンは MySQL 5.7.12 と互換性があります。詳細については、「MySQL 5.7 互換 Aurora MySQL バージョン 2」を参照してください。	2023 年 6 月 9 日
Aurora MySQL バージョン 3.03.1、MySQL 8.0.26 互換	Aurora MySQL バージョン 3.03.1 が利用可能です。このバージョンは MySQL 8.0.26 と互換性があります。詳細については、MySQL 8.0 と互換性のある Aurora MySQL バージョン 3 を参照してください。	2023 年 5 月 11 日

Aurora MySQL バージョン 2.07.9、MySQL 5.7.12 互換	Aurora MySQL バージョン 2.07.9 が利用可能です。このバージョンは MySQL 5.7.12 と互換性があります。詳細については、「MySQL 5.7 互換 Aurora MySQL バージョン 2」を参照してください。	2023 年 5 月 4 日
Aurora MySQL バージョン 3.02.3、MySQL 8.0.23 互換	Aurora MySQL バージョン 3.02.3 が利用可能です。このバージョンは MySQL 8.0.23 と互換性があります。詳細については、MySQL 8.0 と互換性のある Aurora MySQL バージョン 3 を参照してください。	2023 年 4 月 17 日
Aurora MySQL バージョン 2.11.2、MySQL 5.7.12 互換	Aurora MySQL バージョン 2.11.2 が利用可能です。このバージョンは MySQL 5.7.12 と互換性があります。詳細については、「MySQL 5.7 互換 Aurora MySQL バージョン 2」を参照してください。	2023 年 3 月 24 日
Aurora MySQL バージョン 3.03.0、MySQL 8.0.26 互換	Aurora MySQL バージョン 3.03.0 が利用可能です。このバージョンは MySQL 8.0.26 と互換性があります。詳細については、MySQL 8.0 と互換性のある Aurora MySQL バージョン 3 を参照してください。	2023 年 3 月 1 日

Aurora MySQL バージョン 2.11.1、MySQL 5.7.12 互換	Aurora MySQL バージョン 2.11.1 が利用可能です。このバージョンは MySQL 5.7.12 と互換性があります。詳細については、「MySQL 5.7 互換 Aurora MySQL バージョン 2」を参照してください。	2023 年 2 月 14 日
Aurora MySQL バージョン 3.02.2、MySQL 8.0.23 互換	Aurora MySQL バージョン 3.02.2 が利用可能です。このバージョンは MySQL 8.0.23 と互換性があります。詳細については、MySQL 8.0 と互換性のある Aurora MySQL バージョン 3 を参照してください。	2022 年 11 月 18 日
Aurora MySQL バージョン 2.10.3、MySQL 5.7 互換	Aurora MySQL バージョン 2.10.3 が利用可能です。このバージョンは MySQL 5.7 と互換性があります。詳細については、「MySQL 5.7 互換 Aurora MySQL バージョン 2」を参照してください。	2022 年 11 月 1 日
Aurora MySQL バージョン 2.11.0、MySQL 5.7.12 互換	Aurora MySQL バージョン 2.11.0 が利用可能です。このバージョンは MySQL 5.7.12 と互換性があります。詳細については、「MySQL 5.7 互換 Aurora MySQL バージョン 2」を参照してください。	2022 年 10 月 25 日

Aurora MySQL バージョン 3.02.1、MySQL 8.0.23 互換	Aurora MySQL バージョン 3.02.1 が利用可能です。このバージョンは MySQL 8.0.23 と互換性があります。詳細については、MySQL 8.0 と互換性のある Aurora MySQL バージョン 3 を参照してください。	2022 年 9 月 7 日
MySQL 5.6 互換 Aurora Serverless v1 のインプレースアップグレード	MySQL 5.6 互換の Aurora Serverless v1 クラスターのインプレースアップグレードを実行して、既存のクラスターを MySQL 5.7 互換の Aurora Serverless v1 クラスターに変更できます。詳細については、「Aurora MySQL Serverless 5.7 engine updates 2020-06-18 (version 2.07.1)」を参照してください。	2022 年 6 月 16 日
MySQL 5.6 互換の Aurora Serverless v1 のインプレースアップグレード	MySQL 5.6 互換の Aurora Serverless v1 クラスターのインプレースアップグレードを実行して、既存のクラスターを MySQL 5.7 互換の Aurora Serverless v1 クラスターに変更できます。詳細については、「Aurora MySQL Serverless 5.7 engine updates 2020-06-18 (version 2.07.1)」を参照してください。	2022 年 6 月 16 日
Aurora MySQL データベースエンジンの更新 2022-06-16 (バージョン 2.07.8) が利用可能です。	Aurora MySQL バージョン 2.07.8 が利用可能です。	2022 年 6 月 16 日

Aurora MySQL バージョン 3.02.0、MySQL 8.0.23 互換	Aurora MySQL バージョン 3.02.0 が利用可能です。このバージョンは MySQL 8.0.23 と互換性があります。詳細については、MySQL 8.0 と互換性のある Aurora MySQL バージョン 3 を参照してください。	2022 年 4 月 20 日
Aurora MySQL バージョン 3.01.1、MySQL 8.0.23 互換	Aurora MySQL バージョン 3.01.1 が利用可能です。このバージョンは MySQL 8.0.23 と互換性があります。詳細については、MySQL 8.0 と互換性のある Aurora MySQL バージョン 3 を参照してください。	2022 年 4 月 15 日
初回リリース	Aurora MySQL リリースノートの初回リリース。	2022 年 3 月 22 日
Aurora MySQL バージョン 2.10.2	Aurora MySQL バージョン 2.10.2 が利用可能です。	2022 年 1 月 26 日
Aurora MySQL バージョン 2.08.4	Aurora MySQL バージョン 2.08.4 が利用可能です。	2022 年 1 月 6 日
Aurora MySQL バージョン 2.07.7	Aurora MySQL バージョン 2.07.7 が利用可能です。	2021 年 11 月 24 日
Aurora MySQL バージョン 3.01.0、MySQL 8.0.23 互換	Aurora MySQL バージョン 3.01.0 が利用可能です。このバージョンは MySQL 8.0.23 と互換性があります。詳細については、MySQL 8.0 と互換性のある Aurora MySQL バージョン 3 を参照してください。	2021 年 11 月 18 日

Aurora MySQL バージョン 2.09.3	Aurora MySQL バージョン 2.09.3 が利用可能です。	2021 年 11 月 12 日
Aurora MySQL バージョン 2.10.1	Aurora MySQL バージョン 2.10.1 が利用可能です。	2021 年 10 月 21 日
Aurora MySQL バージョン 1.23.4	Aurora MySQL バージョン 1.23.4 が利用可能です。	2021 年 9 月 30 日
Aurora MySQL バージョン 2.07.6	Aurora MySQL バージョン 2.07.6 が利用可能です。	2021 年 9 月 2 日
Aurora MySQL バージョン 2.07.5	Aurora MySQL バージョン 2.07.5 が利用可能です。	2021 年 7 月 6 日
Aurora MySQL バージョン 1.23.3	Aurora MySQL バージョン 1.23.3 が利用可能です。	2021 年 6 月 28 日
Aurora MySQL バージョン 1.22.5	Aurora MySQL バージョン 1.22.5 が利用可能です。	2021 年 6 月 3 日
Aurora MySQL バージョン 2.10.0	Aurora MySQL バージョン 2.10.0 が利用可能です。ハイライトには、ライターの再起動中のリーダーインスタンスの可用性の向上、ダウンタイムのないパッチ適用 (ZDP) の改善、ダウンタイムのない再起動 (ZDR) の改善、およびバイナリログ I/O キャッシュ最適化などが含まれています。	2021 年 5 月 25 日
Aurora MySQL バージョン 1.23.2	Aurora MySQL バージョン 1.23.2 が利用可能です。	2021 年 3 月 18 日
Aurora MySQL バージョン 2.07.4	Aurora MySQL バージョン 2.07.4 が利用可能です。	2021 年 3 月 4 日

Aurora MySQL バージョン 1.22.4	Aurora MySQL バージョン 1.22.4 が利用可能です。	2021 年 3 月 4 日
Aurora MySQL バージョン 2.09.2	Aurora MySQL バージョン 2.09.2 が利用可能です。	2021 年 2 月 26 日
Aurora MySQL バージョン 2.09.1	Aurora MySQL バージョン 2.09.1 が利用可能です。	2020 年 12 月 11 日
Aurora MySQL バージョン 1.23.1	Aurora MySQL バージョン 1.23.1 が利用可能です。	2020 年 11 月 24 日
Aurora MySQL バージョン 2.08.3	Aurora MySQL バージョン 2.08.3 が利用可能です。	2020 年 11 月 12 日
Aurora MySQL バージョン 2.07.3	Aurora MySQL バージョン 2.07.3 が利用可能です。	2020 年 11 月 10 日
Aurora MySQL バージョン 1.22.3	Aurora MySQL バージョン 1.22.3 が利用可能です。	2020 年 11 月 9 日
Aurora MySQL バージョン 2.09.0	Aurora MySQL バージョン 2.09.0 が利用可能です。	2020 年 9 月 17 日
Aurora MySQL バージョン 1.23.0	Aurora MySQL バージョン 1.23.0 が利用可能です。	2020 年 9 月 2 日
Aurora MySQL バージョン 2.08.2	Aurora MySQL バージョン 2.08.2 が利用可能です。	2020 年 8 月 28 日
Aurora MySQL バージョン 2.04.9	Aurora MySQL バージョン 2.04.9 が利用可能です。	2020 年 8 月 14 日
Aurora MySQL バージョン 2.08.1	Aurora MySQL バージョン 2.08.1 が利用可能です。	2020 年 6 月 18 日
Aurora MySQL バージョン 1.22.2 (パラレルクエリクラスターに対応)	パラレルクエリクラスターの作成時に、Aurora MySQL バージョン 1.22.2 を使用できます。	2020 年 6 月 18 日

Aurora MySQL バージョン 1.20.1 (パラレルクエリクラスターに対応)	パラレルクエリクラスターの作成時に、Aurora MySQL バージョン 1.20.1 を使用できます。	2020 年 6 月 11 日
Aurora MySQL バージョン 2.08.0	Aurora MySQL バージョン 2.08.0 が利用可能です。	2020 年 6 月 2 日
Aurora MySQL バージョン 1.19.6 (パラレルクエリクラスターに対応)	パラレルクエリクラスターの作成時に、Aurora MySQL バージョン 1.19.6 を使用できます。	2020 年 6 月 2 日
Aurora MySQL バージョン 2.07.2	Aurora MySQL バージョン 2.07.2 が利用可能です。	2020 年 4 月 17 日
Aurora MySQL バージョン 1.22.2	Aurora MySQL バージョン 1.22.2 が利用可能です。	2020 年 3 月 5 日
Aurora MySQL バージョン 1.20.1	Aurora MySQL バージョン 1.20.1 が利用可能です。	2020 年 3 月 5 日
Aurora MySQL バージョン 1.19.6	Aurora MySQL バージョン 1.19.6 が利用可能です。	2020 年 3 月 5 日
Aurora MySQL バージョン 1.17.9	Aurora MySQL バージョン 1.17.9 が利用可能です。	2020 年 3 月 5 日
Aurora MySQL バージョン 2.07.1	Aurora MySQL バージョン 2.07.1 が利用可能です。	2019 年 12 月 23 日
Aurora MySQL バージョン 1.22.1	Aurora MySQL バージョン 1.22.1 が利用可能です。	2019 年 12 月 23 日
Aurora MySQL バージョン 2.07.0	Aurora MySQL バージョン 2.07.0 が利用可能です。	2019 年 11 月 25 日
Aurora MySQL バージョン 1.22.0	Aurora MySQL バージョン 1.22.0 が利用可能です。	2019 年 11 月 25 日

Aurora MySQL バージョン 1.21.0	Aurora MySQL バージョン 1.21.0 が利用可能です。	2019 年 11 月 25 日
Aurora MySQL バージョン 2.06.0	Aurora MySQL バージョン 2.06.0 が利用可能です。	2019 年 11 月 22 日
Aurora MySQL バージョン 2.04.8	Aurora MySQL バージョン 2.04.8 が利用可能です。	2019 年 11 月 20 日
Aurora MySQL バージョン 2.04.7	Aurora MySQL バージョン 2.04.7 が利用可能です。	2019 年 11 月 14 日
Aurora MySQL バージョン 2.05.0	Aurora MySQL バージョン 2.05.0 が利用可能です。	2019 年 11 月 11 日
Aurora MySQL バージョン 1.20.0	Aurora MySQL バージョン 1.20.0 が利用可能です。	2019 年 11 月 11 日
Aurora MySQL バージョン 2.04.6	Aurora MySQL バージョン 2.04.6 が利用可能です。	2019 年 9 月 19 日
Aurora MySQL バージョン 1.19.5	Aurora MySQL バージョン 1.19.5 が利用可能です。	2019 年 9 月 19 日
Aurora MySQL バージョン 2.04.5	Aurora MySQL バージョン 2.04.5 が利用可能です。	2019 年 7 月 8 日
Aurora MySQL バージョン 1.19.2	Aurora MySQL バージョン 1.19.2 が利用可能です。	2019 年 6 月 5 日
Aurora MySQL バージョン 2.04.4	Aurora MySQL バージョン 2.04.4 が利用可能です。	2019 年 5 月 29 日
Aurora MySQL バージョン 2.04.3	Aurora MySQL バージョン 2.04.3 が利用可能です。	2019 年 5 月 9 日
Aurora MySQL バージョン 1.19.1	Aurora MySQL バージョン 1.19.1 が利用可能です。	2019 年 5 月 9 日

Aurora MySQL バージョン 2.04.2	Aurora MySQL バージョン 2.04.2 が利用可能です。	2019 年 5 月 2 日
Aurora MySQL バージョン 2.04.1	Aurora MySQL バージョン 2.04.1 が利用可能です。	2019 年 3 月 25 日
Aurora MySQL バージョン 2.04	Aurora MySQL バージョン 2.04 が利用可能です。	2019 年 3 月 25 日
Aurora MySQL バージョン 2.03.4	Aurora MySQL バージョン 2.03.4 が利用可能です。	2019 年 2 月 7 日
Aurora MySQL バージョン 1.19.0	Aurora MySQL バージョン 1.19.0 が利用可能です。	2019 年 2 月 7 日
Aurora MySQL バージョン 2.03.3	Aurora MySQL バージョン 2.03.3 が利用可能です。	2019 年 1 月 18 日
Aurora MySQL バージョン 1.17.8	Aurora MySQL バージョン 1.17.8 が利用可能です。	2019 年 1 月 17 日
Aurora MySQL バージョン 2.03.2	Aurora MySQL バージョン 2.03.2 が利用可能です。	2019 年 1 月 9 日
Aurora MySQL バージョン 2.03.1	Aurora MySQL バージョン 2.03.1 が利用可能です。	2018 年 10 月 24 日
Aurora MySQL バージョン 2.03	Aurora MySQL バージョン 2.03 が利用可能です。	2018 年 10 月 11 日
Aurora MySQL バージョン 2.02.5	Aurora MySQL バージョン 2.02.5 が利用可能です。	2018 年 10 月 8 日
Aurora MySQL バージョン 1.17.7	Aurora MySQL バージョン 1.17.7 が利用可能です。	2018 年 10 月 8 日
Aurora MySQL バージョン 2.02.4	Aurora MySQL バージョン 2.02.4 が利用可能です。	2018 年 9 月 21 日

Aurora MySQL バージョン 1.18.0	Aurora MySQL バージョン 1.18.0 が利用可能です。	2018 年 9 月 20 日
Aurora MySQL バージョン 1.17.6	Aurora MySQL バージョン 1.17.6 が利用可能です。	2018 年 9 月 6 日

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。