Aurora MySQL 릴리스 정보

Amazon Aurora

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Aurora: Aurora MySQL 릴리스 정보

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon의 상표 및 트레이드 드레스는 Amazon 외 제품 또는 서비스와 함께, Amazon 브랜드 이미지를 떨어뜨리거나 고객에게 혼동을 일으킬 수 있는 방식으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon과 제휴 관계이거나 관련이 있거나 후원 관계와 관계없이 해당 소유자의 자산입니다.

Table of Contents

Aurora MySQL 릴리스 정보 1
Aurora MySQL 릴리스 일정 2
Aurora MySQL 메이저 버전 릴리스 일정 2
Release calendar for Aurora MySQL minor versions 3
Aurora MySQL 버전 3 5
Aurora MySQL 업데이트: 2026-02-17(버전 3.12.0, MySQL 8.0.44와 호환) 6
개선 사항 7
MySQL Community Edition 버그 수정 통합 10
Aurora MySQL 업데이트: 2025-12-16(버전 3.11.1, MySQL 8.0.43과 호환) 10
개선 사항 11
Aurora MySQL 업데이트: 2025-11-13(버전 3.11.0, MySQL 8.0.43과 호환) 11
개선 사항 12
MySQL Community Edition 버그 수정 통합 15
Aurora MySQL 업데이트: 2026-01-02(버전 3.10.3, MySQL 8.0.42와 호환) 15
개선 사항 16
Aurora MySQL 업데이트: 2025-11-20(버전 3.10.2, MySQL 8.0.42와 호환) 16
개선 사항 17
Aurora MySQL 업데이트: 2025년 9월 30일(버전 3.10.1, MySQL 8.0.42와 호환) 18
개선 사항 19
MySQL Community Edition 버그 수정 통합 20
Aurora MySQL 업데이트: 2025년 7월 31일(버전 3.10.0, MySQL 8.0.42와 호환) 20
새로운 특성 21
개선 사항 22
MySQL Community Edition 버그 수정 통합 24
Aurora MySQL 업데이트: 2025년 5월 14일(버전 3.09.0, MySQL 8.0.40과 호환) 24
릴리스 하이라이트 25
개선 사항 25
MySQL Community Edition 버그 수정 통합 28
Aurora MySQL 업데이트: 2025년 4월 7일(버전 3.08.2, MySQL 8.0.39와 호환) 29
개선 사항 29
MySQL Community Edition 버그 수정 통합 30
Aurora MySQL 업데이트: 2025년 2월 17일(버전 3.08.1, MySQL 8.0.39와 호환) 30
개선 사항 29
MySQL Community Edition 버그 수정 통합 30

Aurora MySQL 업데이트: 2024년 11월 18일 (버전 3.08.0, MySQL 8.0.39와 호환) 32
새로운 특성 33
개선 사항 34
MySQL Community Edition 버그 수정 통합 39
Aurora MySQL 업데이트: 2024-07-23(버전 3.07.1)(사용되지 않음) 39
개선 사항 40
MySQL Community Edition 버그 수정 통합 40
Aurora MySQL 업데이트: 2024-06-04(버전 3.07.0)(사용되지 않음) 40
개선 사항 41
MySQL Community Edition 버그 수정 통합 44
Aurora MySQL 업데이트: 2024-06-26(버전 3.06.1)(사용되지 않음) 44
개선 사항 45
MySQL Community Edition 버그 수정 통합 47
Aurora MySQL 업데이트: 2024-03-07(버전 3.06.0)(사용되지 않음) 48
새로운 기능 48
개선 사항 45
MySQL Community Edition 버그 수정 통합 47
Aurora MySQL 업데이트: 2024-01-31(버전 3.05.2)(사용되지 않음) 52
개선 사항 53
MySQL Community Edition 버그 수정 통합 54
Aurora MySQL 업데이트: 2023-11-21(버전 3.05.1)(사용되지 않음) 55
개선 사항 55
MySQL Community Edition 버그 수정 통합 55
Aurora MySQL 업데이트: 2023-10-30(버전 3.05.0.1)(사용되지 않음) 56
개선 사항 56
Aurora MySQL 업데이트: 2023-10-25(버전 3.05.0)(사용되지 않음) 57
개선 사항 57
MySQL Community Edition 버그 수정 통합 61
Aurora MySQL 업데이트: 2026-01-02(버전 3.04.6, MySQL 8.0.28과 호환) 61
개선 사항 62
MySQL Community Edition 버그 수정 통합 62
Aurora MySQL 업데이트: 2025년 5월 5일(버전 3.04.4, MySQL 8.0.28과 호환) 62
개선 사항 63
MySQL Community Edition 버그 수정 통합 65
Aurora MySQL 업데이트: 2024년 6월 26일(버전 3.04.3, MySQL 8.0.28과 호환) 65
개선 사항 66

MySQL Community Edition 버그 수정 통합 68
Aurora MySQL 업데이트: 2024년 3월 15일(버전 3.04.2, MySQL 8.0.28과 호환) 68
개선 사항 69
MySQL Community Edition 버그 수정 통합 71
Aurora MySQL 업데이트: 2023년 11월 13일(버전 3.04.1, MySQL 8.0.28과 호환) 72
개선 사항 73
MySQL Community Edition 버그 수정 통합 74
Aurora MySQL 업데이트: 2023년 7월 31일(버전 3.04.0, MySQL 8.0.28과 호환) 74
개선 사항 76
MySQL Community Edition 버그 수정 통합 81
Aurora MySQL 업데이트: 2023년 12월 8일(버전 3.03.3)(사용되지 않음) 81
개선 사항 82
MySQL Community Edition 버그 수정 통합 83
Aurora MySQL 업데이트: 2023년 8월 29일(버전 3.03.2)(사용되지 않음) 83
개선 사항 84
MySQL Community Edition 버그 수정 통합 86
Aurora MySQL 업데이트: 2023년 5월 11일(버전 3.03.1)(사용되지 않음) 86
개선 사항 87
MySQL Community Edition 버그 수정 통합 88
Aurora MySQL 업데이트: 2023년 3월 1일(버전 3.03.0)(사용되지 않음) 88
개선 사항 89
MySQL Community Edition 버그 수정 통합 91
Aurora MySQL 업데이트: 2023년 4월 17일(버전 3.02.3)(사용되지 않음) 92
개선 사항 93
Aurora MySQL 업데이트: 2022년 11월 18일(버전 3.02.2)(사용되지 않음) 94
개선 사항 94
MySQL Community Edition 버그 수정 통합 91
Aurora MySQL 업데이트: 2022년 9월 7일(버전 3.02.1)(사용되지 않음) 97
개선 사항 98
Aurora MySQL 업데이트: 2022년 4월 20일(버전 3.02.0)(사용되지 않음) 99
개선 사항 99
MySQL 커뮤니티 에디션 버그 픽스 통합 101
Aurora MySQL 업데이트: 2022년 4월 15일(버전 3.01.1)(사용되지 않음) 101
개선 사항 102
MySQL 커뮤니티 에디션 버그 픽스 통합 104
Aurora MySQL 업데이트: 2021년 11월 18일(버전 3.01.0)(사용되지 않음) 104

개선 사항 105
Aurora MySQL 버전 2 106
Aurora MySQL 업데이트: 2025년 4월 9일(버전 2.12.5, MySQL 5.7.44와 호환) - RDS 추가 지원 버전 108
개선 사항 109
Aurora MySQL 업데이트: 2024년 10월 23일(버전 2.12.4, MySQL 5.7.44와 호환) - RDS 추가 지원 버전 110
개선 사항 109
MySQL Community Edition 버그 수정 통합 111
Aurora MySQL 버전 2에서 지원되지 않는 기능 111
MySQL 5.7 호환성 111
Aurora MySQL 업데이트: 2024년 07월 9일(버전 2.12.3, MySQL 5.7.44와 호환) - RDS 추가 지원 버전 112
개선 사항 112
MySQL Community Edition 버그 수정 통합 115
Aurora MySQL 버전 2에서 지원되지 않는 기능 115
MySQL 5.7 호환성 115
Aurora MySQL 업데이트: 2024년 3월 19일(버전 2.12.2, MySQL 5.7.44와 호환) - RDS 추가 지원 버전 116
개선 사항 116
MySQL Community Edition 버그 수정 통합 117
Aurora MySQL 버전 2에서 지원되지 않는 기능 117
MySQL 5.7 호환성 118
Aurora MySQL 업데이트: 2023년 12월 28일(버전 2.12.1, MySQL 5.7.40과 호환) - RDS 추가 지원 버전 118
개선 사항 119
MySQL Community Edition 버그 수정 통합 121
Aurora MySQL 버전 2에서 지원되지 않는 기능 122
MySQL 5.7 호환성 122
Aurora MySQL 업데이트: 2023년 10월 25일(버전 2.12.0.1, MySQL 5.7.40과 호환) - RDS 추가 지원 버전(베타) 122
개선 사항 123
Aurora MySQL 업데이트: 2023년 7월 25일(버전 2.12.0, MySQL 5.7.40과 호환) - RDS 추가 지원 버전 123
개선 사항 124
MySQL Community Edition 버그 수정 통합 127

Aurora MySQL 버전 2에서 지원되지 않는 기능 127
MySQL 5.7 호환성 127
Aurora MySQL 업데이트: 2024년 7월 19일(버전 2.11.6, MySQL 5.7.12와 호환) - RDS 추가 지원 버전 128
개선 사항 129
MySQL Community Edition 버그 수정 통합 130
Aurora MySQL 버전 2에서 지원되지 않는 기능 130
MySQL 5.7 호환성 131
Aurora MySQL 업데이트: 2024년 3월 26일(버전 2.11.5, MySQL 5.7.12와 호환) - RDS 추가 지원 버전(기본) 131
개선 사항 132
MySQL Community Edition 버그 수정 통합 134
Aurora MySQL 버전 2에서 지원되지 않는 기능 134
MySQL 5.7 호환성 134
Aurora MySQL 업데이트: 2023년 10월 17일(버전 2.11.4, MySQL 5.7.12와 호환) - RDS 추가 지원 버전 134
개선 사항 136
MySQL Community Edition 버그 수정 통합 137
Aurora MySQL 버전 2에서 지원되지 않는 기능 138
MySQL 5.7 호환성 138
Aurora MySQL 업데이트: 2023년 6월 9일(버전 2.11.3, MySQL 5.7.12와 호환) - RDS 추가 지원 버전 138
개선 사항 139
Aurora MySQL 버전 2에서 지원되지 않는 기능 140
MySQL 5.7 호환성 141
Aurora MySQL 업데이트: 2023년 3월 24일(버전 2.11.2, MySQL 5.7.12와 호환) - RDS 추가 지원 버전 141
개선 사항 142
Aurora MySQL 버전 2에서 지원되지 않는 기능 143
MySQL 5.7 호환성 143
Aurora MySQL 업데이트: 2023년 2월 14일(버전 2.11.1, MySQL 5.7.12와 호환) - RDS 추가 지원 버전 144
개선 사항 145
Aurora MySQL 버전 1과의 비교 146
MySQL 5.7 호환성 146

Aurora MySQL 업데이트: 2022년 10월 25일(버전 2.11.0, MySQL 5.7.12와 호환) - RDS 추가 지원 버전 147
개선 사항 147
MySQL Community Edition 버그 수정 통합 151
Aurora MySQL 버전 1과의 비교 152
MySQL 5.7 호환성 153
Aurora MySQL 업데이트: 2022년 11월 1일(버전 2.10.3)(사용되지 않음) 153
개선 사항 154
MySQL Community Edition 버그 수정 통합 155
Aurora MySQL 버전 1과의 비교 156
MySQL 5.7 호환성 156
Aurora MySQL 업데이트: 2022년 1월 26일(버전 2.10.2)(사용되지 않음) 157
개선 사항 157
MySQL Community Edition 버그 수정 통합 91
Aurora MySQL 버전 1과의 비교 160
MySQL 5.7 호환성 161
Aurora MySQL 업데이트: 2021년 10월 21일(버전 2.10.1)(사용되지 않음) 161
개선 사항 162
MySQL 커뮤니티 에디션 버그 픽스 통합 163
Aurora MySQL 버전 1과의 비교 163
MySQL 5.7 호환성 164
Aurora MySQL 업데이트: 2021년 5월 25일(버전 2.10.0)(사용되지 않음) 164
개선 사항 165
MySQL 커뮤니티 에디션 버그 픽스 통합 168
Aurora MySQL 버전 1과의 비교 171
MySQL 5.7 호환성 171
Aurora MySQL 업데이트: 2021년 11월 12일(버전 2.09.3)(사용되지 않음) 172
개선 사항 172
MySQL 커뮤니티 에디션 버그 픽스 통합 175
Aurora MySQL 버전 1과의 비교 175
MySQL 5.7 호환성 176
Aurora MySQL 업데이트: 2021년 2월 26일(버전 2.09.2)(사용되지 않음) 176
개선 사항 177
Aurora MySQL 버전 1과의 비교 178
MySQL 5.7 호환성 179
Aurora MySQL 업데이트: 2020년 12월 11일(버전 2.09.1)(사용되지 않음) 179

개선 사항 180
MySQL 커뮤니티 에디션 버그 픽스 통합 181
Aurora MySQL 버전 1과의 비교 181
MySQL 5.7 호환성 182
Aurora MySQL 업데이트: 2020년 9월 17일(버전 2.09.0)(사용되지 않음) 182
개선 사항 183
MySQL 커뮤니티 에디션 버그 픽스 통합 188
Aurora MySQL 버전 1과의 비교 189
MySQL 5.7 호환성 189
Aurora MySQL 업데이트: 2022년 1월 6일(버전 2.08.4)(사용되지 않음) 190
개선 사항 191
Aurora MySQL 버전 1과의 비교 191
MySQL 5.7 호환성 192
Aurora MySQL 업데이트: 2020년 11월 12일(버전 2.08.3)(사용되지 않음) 192
개선 사항 193
MySQL 커뮤니티 에디션 버그 픽스 통합 194
Aurora MySQL 버전 1과의 비교 194
MySQL 5.7 호환성 194
Aurora MySQL 업데이트: 2020년 8월 28일(버전 2.08.2)(사용되지 않음) 195
개선 사항 196
Aurora MySQL 버전 1과의 비교 196
MySQL 5.7 호환성 197
Aurora MySQL 업데이트: 2020년 6월 18일(버전 2.08.1)(사용되지 않음) 197
개선 사항 198
Aurora MySQL 버전 1과의 비교 198
MySQL 5.7 호환성 199
Aurora MySQL 업데이트: 2020년 6월 2일(버전 2.08.0)(사용되지 않음) 199
개선 사항 200
MySQL 커뮤니티 에디션 버그 픽스 통합 203
Aurora MySQL 버전 1과의 비교 203
MySQL 5.7 호환성 204
Aurora MySQL 업데이트: 2023년 8월 15일(버전 2.07.10)(사용되지 않음) 204
개선 사항 205
Aurora MySQL 버전 2에서 지원되지 않는 기능 206
MySQL 5.7 호환성 206
Aurora MySQL 업데이트: 2023년 5월 4일(버전 2.07.9)(사용되지 않음) 207

개선 사항 207
Aurora MySQL 버전 2에서 지원되지 않는 기능 208
MySQL 5.7 호환성 208
Aurora MySQL 업데이트: 2022년 6월 16일(버전 2.07.8)(사용되지 않음) 209
개선 사항 210
MySQL 커뮤니티 에디션 버그 픽스 통합 210
Aurora MySQL 버전 1과의 비교 210
MySQL 5.7 호환성 211
Aurora MySQL 업데이트: 2021년 11월 24일(버전 2.07.7)(사용되지 않음) 211
개선 사항 212
Aurora MySQL 버전 1과의 비교 213
MySQL 5.7 호환성 214
Aurora MySQL 업데이트: 2021년 9월 2일(버전 2.07.6)(사용되지 않음) 214
MySQL 커뮤니티 에디션 버그 픽스 통합 215
Aurora MySQL 버전 1과의 비교 215
MySQL 5.7 호환성 215
Aurora MySQL 업데이트: 2021년 7월 6일(버전 2.07.5)(사용되지 않음) 216
개선 사항 217
Aurora MySQL 버전 1과의 비교 217
MySQL 5.7 호환성 217
Aurora MySQL 업데이트: 2021년 3월 4일(버전 2.07.4)(사용되지 않음) 218
개선 사항 219
MySQL 커뮤니티 에디션 버그 픽스 통합 220
Aurora MySQL 버전 1과의 비교 220
MySQL 5.7 호환성 220
Aurora MySQL 업데이트: 2020년 11월 10일(버전 2.07.3)(사용되지 않음) 221
개선 사항 222
MySQL 커뮤니티 에디션 버그 픽스 통합 223
Aurora MySQL 버전 1과의 비교 224
MySQL 5.7 호환성 224
Aurora MySQL 업데이트: 2020년 4월 17일(버전 2.07.2)(사용되지 않음) 225
개선 사항 226
MySQL 커뮤니티 에디션 버그 픽스 통합 226
Aurora MySQL 버전 1과의 비교 227
MySQL 5.7 호환성 227
Aurora MySQL 업데이트 2019년 12월 23일(버전 2.07.1)(사용되지 않음) 228

개선 사항 229
Aurora MySQL 버전 1과의 비교 229
MySQL 5.7 호환성 229
Aurora MySQL 업데이트: 2019년 11월 25일(버전 2.07.0)(사용되지 않음) 230
개선 사항 231
MySQL 커뮤니티 에디션 버그 픽스 통합 232
Aurora MySQL 버전 1과의 비교 232
MySQL 5.7 호환성 233
Aurora MySQL 업데이트: 2019년 11월 22일(버전 2.06.0)(사용되지 않음) 233
개선 사항 234
Aurora MySQL 버전 1과의 비교 237
MySQL 5.7 호환성 237
Aurora MySQL 업데이트: 2019년 11월 11일(버전 2.05.0)(사용되지 않음) 238
개선 사항 239
MySQL 버그 수정 통합 239
Aurora MySQL 버전 1과의 비교 239
MySQL 5.7 호환성 240
Aurora MySQL 업데이트: 2020년 8월 14일(버전 2.04.9)(사용되지 않음) 241
개선 사항 241
MySQL 버그 수정 통합 244
Aurora MySQL 버전 1과의 비교 245
MySQL 5.7 호환성 245
Aurora MySQL 업데이트: 2019년 11월 20일(버전 2.04.8)(사용되지 않음) 246
개선 사항 247
Aurora MySQL 버전 1과의 비교 247
MySQL 5.7 호환성 248
Aurora MySQL 업데이트: 2019년 11월 14일(버전 2.04.7)(사용되지 않음) 248
개선 사항 249
Aurora MySQL 버전 1과의 비교 250
MySQL 5.7 호환성 250
Aurora MySQL 업데이트: 2019년 9월 19일(버전 2.04.6)(사용되지 않음) 251
개선 사항 252
MySQL 버그 수정 통합 252
Aurora MySQL 버전 1과의 비교 252
MySQL 5.7 호환성 253
Aurora MySQL 업데이트: 2019년 7월 8일(버전 2.04.5)(사용되지 않음) 253

개선 사항 254
Aurora MySQL 버전 1과의 비교 255
MySQL 5.7 호환성 255
Aurora MySQL 업데이트: 2019년 5월 29일(버전 2.04.4)(사용되지 않음) 256
개선 사항 257
Aurora MySQL 버전 1과의 비교 257
MySQL 5.7 호환성 258
Aurora MySQL 업데이트: 2019년 5월 9일(버전 2.04.3)(사용되지 않음) 258
개선 사항 259
Aurora MySQL 버전 1과의 비교 259
MySQL 5.7 호환성 260
Aurora MySQL 업데이트: 2019년 5월 2일(버전 2.04.2)(사용되지 않음) 260
개선 사항 261
MySQL 버그 수정 통합 262
Aurora MySQL 버전 1과의 비교 262
MySQL 5.7 호환성 262
Aurora MySQL 업데이트: 2019년 3월 25일(버전 2.04.1)(사용되지 않음) 263
개선 사항 264
Aurora MySQL 버전 1과의 비교 264
MySQL 5.7 호환성 265
Aurora MySQL 업데이트: 2019년 3월 25일(버전 2.04.0)(사용되지 않음) 265
개선 사항 266
MySQL 버그 수정 통합 266
Aurora MySQL 버전 1과의 비교 266
MySQL 5.7 호환성 267
Aurora MySQL 업데이트: 2019년 2월 7일(버전 2.03.4)(사용되지 않음) 268
개선 사항 268
Aurora MySQL 버전 1과의 비교 269
MySQL 5.7 호환성 269
Aurora MySQL 업데이트: 2019년 1월 18일(버전 2.03.3)(사용되지 않음) 270
개선 사항 270
MySQL 버그 수정 통합 271
Aurora MySQL 버전 1과의 비교 272
MySQL 5.7 호환성 272
Aurora MySQL 업데이트: 2019년 1월 9일(버전 2.03.2)(사용되지 않음) 273
개선 사항 274

Aurora MySQL 버전 1과의 비교 274
MySQL 5.7 호환성 274
Aurora MySQL 업데이트: 2018년 10월 24일(버전 2.03.1)(사용되지 않음) 275
개선 사항 276
Aurora MySQL 버전 1과의 비교 276
MySQL 5.7 호환성 276
Aurora MySQL 업데이트: 2018년 10월 11일(버전 2.03)(사용되지 않음) 277
개선 사항 278
MySQL 커뮤니티 에디션 버그 픽스 통합 278
Aurora MySQL 버전 1과의 비교 278
MySQL 5.7 호환성 279
Aurora MySQL 업데이트: 2018년 10월 8일(버전 2.02.5)(사용되지 않음) 279
개선 사항 280
Aurora MySQL 버전 1과의 비교 280
MySQL 5.7 호환성 281
Aurora MySQL 업데이트: 2018년 9월 21일(버전 2.02.4)(사용되지 않음) 281
개선 사항 282
MySQL 커뮤니티 에디션 버그 픽스 통합 282
Aurora MySQL 버전 1과의 비교 283
MySQL 5.7 호환성 284
Aurora MySQL 업데이트: 2018년 8월 23일(버전 2.02.3)(사용되지 않음) 284
Aurora MySQL 버전 1과의 비교 285
MySQL 5.7 호환성 286
Aurora MySQL 2.x와 Aurora MySQL 1.x의 CLI 차이점 286
개선 사항 287
Aurora MySQL 업데이트: 2018년 6월 4일(버전 2.02.2)(사용되지 않음) 287
개선 사항 288
Aurora MySQL 5.6과의 비교 288
MySQL 5.7 호환성 289
Aurora MySQL 2.x와 Aurora MySQL 1.x의 CLI 차이점 286
개선 사항 288
Aurora MySQL 업데이트: 2018년 5월 3일(버전 2.02)(사용되지 않음) 290
Aurora MySQL 5.6과의 비교 290
MySQL 5.7 호환성 291
Aurora MySQL 2.x와 Aurora MySQL 1.x의 CLI 차이점 286
개선 사항 292

MySQL 버그 수정 통합 293
Aurora MySQL 업데이트: 2018년 3월 13일(버전 2.01.1)(사용되지 않음) 293
Aurora MySQL 5.6과의 비교 293
MySQL 5.7 호환성 294
Aurora MySQL 2.x와 Aurora MySQL 1.x의 CLI 차이점 294
개선 사항 295
Aurora MySQL 업데이트: 2018년 2월 6일(버전 2.01)(사용되지 않음) 295
Aurora MySQL 5.6과의 비교 295
MySQL 5.7 호환성 296
Aurora MySQL 2.x와 Aurora MySQL 1.x의 CLI 차이점 286
Aurora MySQL 버전 1(사용되지 않음) 298
Aurora MySQL 업데이트: 2021년 9월 30일(버전 1.23.4)(사용되지 않음) 299
개선 사항 300
Aurora MySQL 업데이트: 2021년 6월 28일(버전 1.23.3)(사용되지 않음) 300
개선 사항 301
Aurora MySQL 업데이트: 2021년 3월 18일(버전 1.23.2)(사용되지 않음) 301
개선 사항 302
MySQL 커뮤니티 에디션 버그 픽스 통합 303
Aurora MySQL 업데이트: 2020년 11월 24일(버전 1.23.1)(사용되지 않음) 303
개선 사항 304
Aurora MySQL 업데이트: 2020년 9월 2일(버전 1.23.0)(사용되지 않음) 305
개선 사항 306
MySQL 커뮤니티 에디션 버그 픽스 통합 309
Aurora MySQL 업데이트: 2021년 6월 3일(버전 1.22.5)(사용되지 않음) 310
개선 사항 311
Aurora MySQL 업데이트: 2021년 3월 4일(버전 1.22.4)(사용되지 않음) 311
개선 사항 312
Aurora MySQL 업데이트: 2020년 11월 9일(버전 1.22.3)(사용되지 않음) 312
개선 사항 313
MySQL 커뮤니티 에디션 버그 픽스 통합 314
Aurora MySQL 업데이트: 2020년 3월 5일(버전 1.22.2)(사용되지 않음) 315
개선 사항 316
Aurora MySQL 업데이트 2019년 12월 23일(버전 1.22.1)(사용되지 않음) 316
개선 사항 317
Aurora MySQL 업데이트: 2019년 11월 25일(버전 1.22.0)(사용되지 않음) 317
개선 사항 318

MySQL 커뮤니티 에디션 버그 픽스 통합 321
Aurora MySQL 업데이트: 2019년 11월 25일(버전 1.21.0)(사용되지 않음) 322
개선 사항 322
MySQL 커뮤니티 에디션 버그 픽스 통합 324
Aurora MySQL 업데이트: 2020년 3월 5일(버전 1.20.1)(사용되지 않음) 324
개선 사항 325
Aurora MySQL 업데이트: 2019년 11월 11일(버전 1.20.0)(사용되지 않음) 325
개선 사항 326
MySQL 커뮤니티 에디션 버그 픽스 통합 327
Aurora MySQL 업데이트: 2020년 3월 5일(버전 1.19.6)(사용되지 않음) 327
개선 사항 328
Aurora MySQL 업데이트: 2019년 9월 19일(버전 1.19.5)(사용되지 않음) 328
개선 사항 329
MySQL 커뮤니티 에디션 버그 픽스 통합 330
Aurora MySQL 업데이트: 2019년 6월 5일(버전 1.19.2)(사용되지 않음) 330
개선 사항 331
Aurora MySQL 업데이트: 2019년 5월 9일(버전 1.19.1)(사용되지 않음) 331
개선 사항 332
Aurora MySQL 업데이트: 2019년 2월 7일(버전 1.19.0)(사용되지 않음) 332
특성 333
개선 사항 333
MySQL 커뮤니티 에디션 버그 픽스 통합 334
Aurora MySQL 업데이트: 2018년 9월 20일(버전 1.18.0)(사용되지 않음) 334
특성 335
Aurora MySQL 업데이트: 2020년 3월 5일(버전 1.17.9)(사용되지 않음) 336
개선 사항 336
Aurora MySQL 업데이트: 2019년 1월 17일(버전 1.17.8)(사용되지 않음) 337
개선 사항 337
MySQL 커뮤니티 에디션 버그 픽스 통합 337
Aurora MySQL 업데이트: 2018년 10월 8일(버전 1.17.7)(사용되지 않음) 337
개선 사항 338
MySQL 커뮤니티 에디션 버그 픽스 통합 338
Aurora MySQL 업데이트: 2018년 9월 6일(버전 1.17.6)(사용되지 않음) 339
개선 사항 339
MySQL 커뮤니티 에디션 버그 픽스 통합 340
Aurora MySQL 업데이트: 2018년 8월 14일(버전 1.17.5)(사용되지 않음) 340

개선 사항 340
Aurora MySQL 업데이트: 2018년 8월 7일(버전 1.17.4)(사용되지 않음) 341
개선 사항 341
Aurora MySQL 업데이트: 2018년 6월 5일(버전 1.17.3)(사용되지 않음) 342
개선 사항 342
Aurora MySQL 업데이트: 2018년 4월 27일(버전 1.17.2)(사용되지 않음) 343
개선 사항 343
Aurora MySQL 업데이트: 2018년 3월 23일(버전 1.17.1)(사용되지 않음) 344
개선 사항 344
Aurora MySQL 업데이트: 2018년 3월 13일(버전 1.17)(사용되지 않음) 344
제로 가동 중지 패치 적용 345
새로운 기능 345
개선 사항 345
MySQL 버그 수정 통합 346
Aurora MySQL 업데이트: 2017년 12월 11일(버전 1.16)(사용되지 않음) 346
제로 가동 중지 패치 적용 346
새로운 기능 347
개선 사항 347
MySQL 버그 수정 통합 347
Aurora MySQL 업데이트: 2017년 11월 20일(버전 1.15.1)(사용되지 않음) 347
제로 가동 중지 패치 적용 348
개선 사항 348
MySQL 버그 수정 통합 348
Aurora MySQL 업데이트: 2017년 10월 24일(버전 1.15)(사용되지 않음) 349
제로 가동 중지 패치 적용 349
새로운 기능 349
개선 사항 350
MySQL 버그 수정 통합 348
Aurora MySQL 업데이트: 2018년 3월 13일(버전 1.14.4)(사용되지 않음) 351
제로 가동 중지 패치 적용 352
새로운 기능 352
개선 사항 352
MySQL 버그 수정 통합 352
Aurora MySQL 업데이트: 2017년 9월 22일(버전 1.14.1)(사용되지 않음) 352
개선 사항 353
Aurora MySQL 업데이트: 2017년 8월 7일(버전 1.14)(사용되지 않음) 353

제로 가동 중지 패치 적용 353
개선 사항 354
MySQL 버그 수정 통합 355
Aurora MySQL 업데이트: 2017년 5월 15일(버전 1.13)(사용되지 않음) 355
제로 가동 중지 패치 적용 355
새로운 기능 356
개선 사항 356
MySQL 버그 수정 통합 357
Aurora MySQL 업데이트: 2017년 4월 5일(버전 1.12)(사용되지 않음) 357
새로운 기능 357
개선 사항 358
MySQL 버그 수정 통합 358
Aurora MySQL 업데이트: 2017년 2월 23일(버전 1.11)(사용되지 않음) 359
새로운 기능 359
개선 사항 360
MySQL 버그 수정 통합 361
Aurora MySQL 업데이트: 2017년 1월 12일(버전 1.10.1)(사용되지 않음) 362
새로운 기능 362
개선 사항 362
Aurora MySQL 업데이트: 2016년 12월 14일(버전 1.10)(사용되지 않음) 362
새로운 기능 362
개선 사항 364
MySQL 버그 수정 통합 364
Aurora MySQL 업데이트: 2016년 11월 10일(버전 1.9.0, 1.9.1)(사용되지 않음) 364
새로운 기능 365
개선 사항 365
Aurora MySQL 업데이트: 2016년 10월 26일(버전 1.8.1)(사용되지 않음) 366
개선 사항 366
MySQL 버그 수정 통합 366
Aurora MySQL 업데이트: 2016년 10월 18일(버전 1.8)(사용되지 않음) 366
새로운 기능 366
개선 사항 367
MySQL 버그 수정 통합 368
Aurora MySQL 업데이트: 2016년 9월 20일(버전 1.7.1)(사용되지 않음) 368
개선 사항 368
Aurora MySQL 업데이트: 2016년 8월 30일(버전 1.7.0)(사용되지 않음) 368

새로운 기능 369
개선 사항 369
MySQL 버그 수정 통합 369
Aurora MySQL 업데이트: 2016년 6월 1일(버전 1.6.5)(사용되지 않음) 370
새로운 기능 370
개선 사항 370
MySQL 버그 수정 통합 371
Aurora MySQL 업데이트: 2016년 4월 6일(버전 1.6)(사용되지 않음) 371
새로운 기능 371
개선 사항 372
MySQL 버그 수정 통합 373
Aurora MySQL 업데이트: 2016년 1월 11일(버전 1.5)(사용되지 않음) 373
개선 사항 373
MySQL 버그 수정 통합 374
Aurora MySQL 업데이트: 2015년 12월 3일(버전 1.4)(사용되지 않음) 374
새로운 기능 374
개선 사항 375
MySQL 버그 수정 통합 375
Aurora MySQL 업데이트: 2015년 10월 16일(버전 1.2, 1.3)(사용되지 않음) 376
수정 사항 376
개선 사항 376
MySQL 버그 수정 통합 377
Aurora MySQL 업데이트: 2015년 8월 24일(버전 1.1)(사용되지 않음) 379
Aurora MySQL 업데이트를 통해 수정한 MySQL 버그 381
Aurora MySQL 3.x 업데이트를 통해 수정한 MySQL 버그 381
Aurora MySQL 2.x 업데이트를 통해 수정한 MySQL 버그 392
Aurora MySQL 1.x 업데이트를 통해 수정한 MySQL 버그 411
Aurora MySQL에서 수정된 보안 취약성 430
문서 이력 438
..... cdlv

Amazon Aurora MySQL 호환 버전 릴리스 정보

Amazon Aurora MySQL 호환 버전 릴리스는 정기적으로 업데이트됩니다. 업데이트는 시스템 유지 관리 기간 중에 Aurora MySQL DB 클러스터에 적용됩니다. 업데이트가 적용되는 시기는 업데이트의 유형뿐 아니라 DB 클러스터에 대한 유지 관리 기간 설정 및 AWS 리전에 따라 다릅니다.

Amazon Aurora MySQL 릴리스는 며칠 이내에 모든 AWS 리전에 공개됩니다. 일부 리전에서는 아직 다른 리전에서 사용할 수 없는 엔진 버전이 일시적으로 표시될 수 있습니다. 릴리스 정보는 모든 리전에서 새 버전을 사용할 수 있게 되면 게시됩니다.

업데이트는 DB 클러스터의 모든 인스턴스에 동시에 적용됩니다. 업데이트하려면 DB 클러스터의 모든 인스턴스에서 데이터베이스를 다시 시작해야 합니다. 따라서 20~30초 동안 가동 중지됩니다. 가동 중지 후에 DB 클러스터를 다시 사용할 수 있습니다. 에서 유지 관리 기간 설정을 확인하거나 변경할 수 있습니다..AWS Management Console

주제

- Amazon Aurora MySQL 릴리스 일정
- Amazon Aurora MySQL 버전 3에 대한 데이터베이스 엔진 업데이트
- Amazon Aurora MySQL 버전 2에 대한 데이터베이스 엔진 업데이트
- Amazon Aurora MySQL 버전 1에 대한 데이터베이스 엔진 업데이트(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그
- Aurora MySQL에서 수정된 보안 취약성

Amazon Aurora MySQL 릴리스 일정

이 페이지의 릴리스 일정은 메이저 및 마이너 버전 업그레이드를 계획하는 데 도움이 될 수 있습니다. Amazon Aurora 업그레이드, 버전 관리 및 수명 주기에 대한 자세한 내용은 Amazon Aurora 버전을 참조하세요.

주제

- Aurora MySQL 메이저 버전 릴리스 일정
- Release calendar for Aurora MySQL minor versions

Aurora MySQL 메이저 버전 릴리스 일정

Aurora MySQL 메이저 버전은 해당 커뮤니티 버전에 대한 커뮤니티 사용 주기가 끝날 때까지 표준 지원에 따라 사용할 수 있습니다. Aurora 표준 지원이 종료된 이후에도 유료로 메이저 버전을 계속 실행할 수 있습니다. 자세한 내용은 Amazon RDS 추가 지원 사용 및 Amazon Aurora 요금을 참조하세요.

Aurora MySQL은 현재 다음과 같은 메이저 버전을 지원합니다.

> **Note**
>
> AWS CLI 또는 RDS API를 사용하여 메이저 엔진 버전의 지원 날짜에 대한 정보를 볼 수도 있습니다. 자세한 내용은 Amazon RDS 추가 지원에서 엔진 버전에 대한 지원 날짜 보기를 참조하세요.

커뮤니티 메이저 버전	Aurora 메이저 버전	커뮤니티 수명 종료 날짜	Aurora 표준 지원 종료일	RDS 추가 지원 요금 부과 시작일(1년)	RDS 추가 지원 요금 부과 시작일(3년)	RDS 추가 지원 종료일	RDS 확장 지원이 가능한 마이너 버전
MySQL 8.0	Aurora MySQL 버전 3	2026년 4월	2028년 4월 30일	2028년 5월 1일	해당 사항 없음	2029년 7월 31일	추후 결정

커뮤니티 메이저 버전	Aurora 메이저 버전	커뮤니티 수명 종료 날짜	Aurora 표준 지원 종료일	RDS 추가 지원 요금 부과 시작일(1년)	RDS 추가 지원 요금 부과 시작일(3년)	RDS 추가 지원 종료일	RDS 확장 지원이 가능한 마이너 버전
MySQL 5.7	Aurora MySQL 버전 2	2023년 10월	2024년 10월 31일	2024년 12월 1일	해당 사항 없음	2027년 2월 28일	Aurora MySQL 2.11 및 2.12
MySQL 5.6(사용되지 않음)	Aurora MySQL 버전 1(사용되지 않음)	2021년 2월 5일	2023년 2월 28일	해당 사항 없음	해당 사항 없음	해당 사항 없음	해당 사항 없음

> **Note**
>
> Aurora MySQL 버전 2에 대한 Amazon RDS 추가 지원은 2024년 11월 1일에 시작되지만, 2024년 12월 1일까지는 요금이 청구되지 않습니다. 2024년 11월 1일부터 11월 30일 사이에 해당하는 모든 Aurora MySQL 버전 2 DB 클러스터는 Amazon RDS 추가 지원 범위에 포함됩니다.

Release calendar for Aurora MySQL minor versions

Aurora MySQL은 현재 다음과 같은 마이너 버전을 지원합니다.

일반적으로 Aurora 마이너 버전은 분기별로 출시됩니다. 릴리스 일정은 추가 기능 또는 수정 사항에 따라 다를 수 있습니다.

마이너 버전은 해당 메이저 버전보다 먼저 표준 지원이 종료될 수 있습니다. 예를 들어 버전 3.07은 2025년 8월에 표준 지원 종료일에 도달하는 반면 메이저 버전 3은 2028년 4월 30일에 표준 지원이 종료됩니다. Amazon RDS는 이 날짜 사이에 릴리스된 추가 3.* 마이너 버전을 지원합니다.

Amazon RDS 추가 지원 요금은 메이저 버전이 추가 지원을 받을 수 있는 이후 특정 마이너 버전에만 적용됩니다. 추가 지원이 가능한 메이저 버전에 대한 자세한 내용은 Aurora MySQL 메이저 버전 릴리스 일정 섹션을 참조하세요.

Aurora MySQL 버전	Aurora MySQL 릴리스 날짜	Aurora MySQL 표준 지원 종료일
3.12(Community MySQL 8.0.44와 호환)	2026년 2월 17일	2027년 2월 17일
3.11(Community MySQL 8.0.43과 호환)	2025년 11월 13일	2026년 11월 13일
3.10(Community MySQL 8.0.42와 호환)(LTS)	2025년 7월 31일	2028년 4월 30일
3.09(Community MySQL 8.0.40과 호환)	2025년 5월 14일	2026년 5월 31일
3.08(Community MySQL 8.0.39와 호환)	2024년 11월 18일	2026년 5월 31일
3.04(Community MySQL 8.0.28와 호환)(LTS)	2023년 7월 31일	2026년 10월 31일
2.12[1] (Community MySQL 5.7.40 또는 5.7.44와 호환[2])	2023년 7월 25일	2024년 10월 31일
2.11[1] (Community MySQL 5.7.12와 호환)	2022년 10월 25일	2024년 10월 31일

LTS – Aurora MySQL 장기 지원(LTS) 버전 자세한 내용은 Amazon Aurora MySQL 장기 지원(LTS) 및 베타 릴리스를 참조하세요.

[1]이 마이너 버전은 메이저 버전이 Amazon RDS 추가 지원에 있을 때 계속 사용할 수 있습니다.

[2] 2.12.1까지의 Aurora MySQL 2.12 버전은 MySQL 버전 5.7.40과 호환되고 버전 2.12.2 이상은 MySQL 버전 5.7.44과 호환됩니다.

Amazon Aurora MySQL 버전 3에 대한 데이터베이스 엔진 업데이트

다음은 Amazon Aurora MySQL 버전 3 데이터베이스 엔진 업데이트입니다.

주제

- Aurora MySQL 데이터베이스 엔진 업데이트 2026-02-17(버전 3.12.0, MySQL 8.0.44와 호환)
- Aurora MySQL 데이터베이스 엔진 업데이트 2025-12-16(버전 3.11.1, MySQL 8.0.43과 호환)
- Aurora MySQL 데이터베이스 엔진 업데이트 2025-11-13(버전 3.11.0, MySQL 8.0.43과 호환)
- Aurora MySQL 데이터베이스 엔진 업데이트 2026-01-02(버전 3.10.3, MySQL 8.0.42와 호환)
- Aurora MySQL 데이터베이스 엔진 업데이트 2025-11-20(버전 3.10.2, MySQL 8.0.42와 호환)
- Aurora MySQL 데이터베이스 엔진 업데이트 2025년 9월 30일(버전 3.10.1, MySQL 8.0.42와 호환)
- Aurora MySQL 데이터베이스 엔진 업데이트 2025년 7월 31일(버전 3.10.0, MySQL 8.0.42와 호환)
- Aurora MySQL 데이터베이스 엔진 업데이트 2025년 5월 14일(버전 3.09.0, MySQL 8.0.40과 호환)
- Aurora MySQL 데이터베이스 엔진 업데이트 2025년 4월 7일(버전 3.08.2, MySQL 8.0.39와 호환)
- Aurora MySQL 데이터베이스 엔진 업데이트 2025년 2월 17일(버전 3.08.1, MySQL 8.0.39와 호환)
- Aurora MySQL 데이터베이스 엔진 업데이트 2024년 11월 18일(버전 3.08.0, MySQL 8.0.39와 호환)
- Aurora MySQL 데이터베이스 엔진 업데이트 2024-07-23(버전 3.07.1)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2024-06-04(버전 3.07.0)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2024-06-26(버전 3.06.1)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2024-03-07(버전 3.06.0)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2024-01-31(버전 3.05.2)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2023-11-21(버전 3.05.1)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2023-10-30(버전 3.05.0.1)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2023-10-25(버전 3.05.0)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2026-01-02(버전 3.04.6, MySQL 8.0.28과 호환)
- Aurora MySQL 데이터베이스 엔진 업데이트 2025년 5월 5일(버전 3.04.4, MySQL 8.0.28과 호환)
- Aurora MySQL 데이터베이스 엔진 업데이트 2024년 6월 26일(버전 3.04.3, MySQL 8.0.28과 호환)
- Aurora MySQL 데이터베이스 엔진 업데이트 2024년 3월 15일(버전 3.04.2, MySQL 8.0.28과 호환)
- Aurora MySQL 데이터베이스 엔진 업데이트 2023년 11월 13일(버전 3.04.1, MySQL 8.0.28과 호환)

- Aurora MySQL 데이터베이스 엔진 업데이트 2023-07-31(버전 3.04.0) , MySQL 8.0.28과 호환)
- Aurora MySQL 데이터베이스 엔진 업데이트 2023년 12월 8일(버전 3.03.3)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2023년 8월 29일(버전 3.03.2)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2023년 5월 11일(버전 3.03.1)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2023년 3월 1일(버전 3.03.0)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2023년 4월 17일(버전 3.02.3)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2022년 11월 18일(버전 3.02.2)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2022년 9월 7일(버전 3.02.1)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2022년 4월 20일(버전 3.02.0)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2022년 4월 15일(버전 3.01.1)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2021년 11월 18일(버전 3.01.0)(사용되지 않음)

Aurora MySQL 데이터베이스 엔진 업데이트 2026-02-17(버전 3.12.0, MySQL 8.0.44와 호환)

버전: 3.12.0

Aurora MySQL 3.12.0을 정식 버전으로 사용할 수 있습니다. Aurora MySQL 3.12 버전은 MySQL 8.0.44와 호환됩니다. 발생한 커뮤니티 변경 사항에 대한 자세한 내용은 MySQL 8.0 Release Notes를 참조하세요.

Aurora MySQL 버전 3의 새로운 기능에 대한 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환을 참조하세요. Aurora MySQL 버전 3과 Aurora MySQL 버전 2의 차이점에 대한 자세한 내용은 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요. Aurora MySQL 버전 3과 MySQL 8.0 Community Edition의 비교 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3과 MySQL 8.0 Community Edition 비교를 참조하세요.

현재 지원되는 Aurora MySQL 버전 2 클러스터에서 Aurora MySQL 버전 MySQL 3.12.0 클러스터로 Amazon RDS 블루/그린 배포를 사용하여 제로 가동 중지 패치(ZDP)를 활용하여 현재 위치 업그레이드를 수행하거나 스냅샷을 복원하거나 관리형 블루/그린 업그레이드를 시작할 수 있습니다. https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/blue-green-deployments-overview.html

Aurora MySQL 버전 3으로의 업그레이드 계획에 대한 자세한 내용은 Aurora MySQL 클러스터에 대한 메이저 버전 업그레이드 계획을 참조하세요. Aurora MySQL 업그레이드에 대한 일반 정보는 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터 업그레이드를 참조하세요.

문제 해결 정보는 Amazon Aurora 사용 설명서의 Aurora MySQL 현재 위치 업그레이드 문제 해결을 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼 및 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 DB 클러스터 유지 관리를 참조하세요.

개선 사항

보안 수정

- 일부 SQL 문이 감사 로그에 로깅되지 않을 수 있는 문제를 수정했습니다.
- Aurora MySQL 리더 및 라이터 인스턴스가 업데이트된 암호를 일관되게 검증하도록 캐싱_sha2_password 플러그인을 수정했습니다.

중간 CVEs

- CVE-2025-53040
- CVE-2025-53042
- CVE-2025-53044
- CVE-2025-53045
- CVE-2025-53053
- CVE-2025-53054
- CVE-2025-53062
- CVE-2025-53069

가용성 향상:

- 반일관적 읽기 중에 잠금 테이블이 가득 차면 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 다중 스레드 복제와 `aurora_in_memory_relaylog` 함께를 사용할 때 binlog 복제본 인스턴스가 다시 시작될 수 있는 binlog 복구 문제를 수정했습니다.
- 전역 쓰기 전달 또는 로컬 쓰기 전달이 비활성화될 때 라이터 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 너무 큰 thread_stack 구성으로 인해 재시작 또는 업그레이드 중에 Aurora MySQL 서버 부팅이 방해될 수 있는 문제를 수정했습니다. 이제 Aurora MySQL 서버는 시스템 메모리를 초과할 때 thread_stack을 엔진 기본값(1MB)으로 자동 재설정하여 시작 실패를 방지합니다.

- 실행 중인 병렬 쿼리EXPLAIN FOR CONNECTION에서 실행한 KILL 후 실행할 때 엔진이 다시 시작될 수 있는 문제를 수정했습니다.
- 임의 쿼리에서 병렬 쿼리를 강제로 실행할 수 있고이 쿼리가 강제로 종료되면 엔진이 다시 시작되는 문제를 수정했습니다.
- MySQL Community Edition에서 지원되는 대로 CONNECTION_ADMIN 또는 SUPER 권한이 있는 사용자가 max_connections 제한을 초과하는 추가 연결을 수행하지 못하는 문제를 수정했습니다.
- 릴레이 로그 복구 중에 많은 수의 릴레이 로그 파일을 처리할 때 이진 로그(binlog) 복제본이 다시 시작될 수 있는 문제를 수정했습니다.
- 리더 DB 인스턴스에서 병렬 쿼리를 실행할 때 라이터 DB 인스턴스가 다시 시작되는 문제를 수정했습니다.
- 드문 경우지만 대규모 트랜잭션을 커밋하는 동안 오류가 발생할 때 이진 로깅을 비활성화할 수 있는 문제를 수정했습니다.
- 라이터 DB 인스턴스에서 드문 트랜잭션 커밋 주문이 발생하는 경우 읽기 전용 복제본이 다시 시작될 수 있는 문제를 수정했습니다.
- 향상된 binlog가 활성화된 인스턴스에서 실행 중에 예약된 이벤트가 중단될 때 데이터베이스가 다시 시작될 수 있는 문제를 수정했습니다.
- 다중 TiB 스토리지 크기를 사용하는 데이터베이스 인스턴스가 InnoDB 버퍼풀 검증 실패로 인해 재시작 중에 가동 중지 시간이 증가할 수 있는 문제를 해결했습니다.

일반적인 개선 사항:

- 실패한 연결 시도가 예기치 않게 performance_schema 테이블에 행을 추가하여 과도한 메모리 소비를 유발하는 MySQL 8.0.44에 도입된 커뮤니티 변경 사항을 제외했습니다. Ref 커뮤니티 버그 #119766
- 사용자 지정 데이터 정렬 설정이 있는 인스턴스에서 binlog 복제를 구성할 때 Aurora 특정 복제 저장 프로시저에 영향을 미치는 문제를 수정했습니다.
- 데이터베이스 엔진과 스토리지 계층 간에 전송 중 암호화를 설정하는 동안 CPU 오버헤드를 줄이는 문제를 수정했습니다.
- 시스템 변수 innodb_flush_log_at_trx_commit이 0으로 설정된 경우 쓰기 IOPS 성능이 개선되었습니다.
- aurora_oom_response가 임계 시간(몇 분) 후 메모리 압력을 해결하지 못하면 aurora_oom_response 작업(구성된 경우 인쇄 제외)을 자동으로 비활성화합니다.
- Aurora Serverless 스케일 다운 작업 중에 또는 버퍼 풀에서 페이지 제거 중에 버퍼 풀에서 잘못된 페이지 읽기가 발생할 수 있는 경합 조건을 해결했습니다. 커뮤니티 버그 #116305를 참조하세요.

- 라이터에서 권한 관련 명령을 실행할 때 리더가 다시 시작될 수 있는 out-of-memory(OOM) 문제를 수정했습니다.
- 에 다음 권한이 추가되었습니다`rds_superuser_role`FLUSH_OPTIMIZER_COSTS, FLUSH_STATUS, FLUSH_TABLES, FLUSH_USER_RESOURCES. 에 대한 자세한 내용은 Amazon Master User Accounts with Aurora 설명서를 참조`rds_superuser_role`하세요. 이러한 동적 권한에 대한 자세한 내용은 MySQL 설명서를 참조하세요.
- `Aurora_pq_request_in_progress` 카운터가 정확하게 업데이트되지 않은 `EXPLAIN ANALYZE` 문을 실행하는 동안 병렬 쿼리 요청을 부정확하게 추적하는 문제를 수정했습니다.
- ZDP(제로 가동 중지 패치)/ZDR(제로 가동 중지 재시작) 중에 보존된 연결이 잘못 처리되어 클라이언트가 쿼리가 완료될 때까지 무기한 대기하는 문제를 수정했습니다.
- 업데이트 중에 공간 인덱스를 통해 행을 읽을 수 없게 되는 문제를 해결합니다.
- 제로 가동 중지 재시작(ZDR) 또는 제로 가동 중지 패치 적용(ZDP) 작업 중에 중단된 옵티마이저 힌트가 포함된 쿼리가 잘못 처리될 수 있는 문제를 수정했습니다.
- innodb_flush_log_at_trx_commit이 0으로 설정된 경우 커밋 지연 시간이 측정되지 않는 문제를 해결합니다.
- 테이블 수가 많을 때 논리적 데이터 사전 복구 중에 메모리 사용량을 줄이기 위한 최적화를 도입했습니다.
- 향상된 Binlog가 활성화되었거나 이전에 활성화된 클러스터에서 `SHOW BINARY LOGS` 명령이 실행되는 데 시간이 더 오래 걸리는 문제를 수정했습니다. 이 문제로 인해 여러 `SHOW BINARY LOGS` 명령이 동시에 실행되는 경우 커밋 지연 시간이 증가할 수도 있습니다.
- 뷰 정의자의 기본 역할과 관련된 메모리 문제를 해결합니다.
- "쓰기 전달" 기능을 비활성화하는 프로세스를 완료하지 못할 수 있는 문제를 해결했습니다.
- 쓰기 전달을 사용하는 리더 인스턴스가 타임스탬프 값을 포함하고 time_zone 데이터베이스 파라미터가 "UTC"로 설정된 DML 문을 실행할 때 라이터 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 드문 경우지만 라이터의 복제본 및 DDL 작업에 대한 동시 읽기 쿼리로 인해 복제본에서 'Table does not exist' 오류와 함께 Aurora 읽기 전용 복제본 또는 테이블 정의 불일치가 간헐적으로 발생하지 않는 문제를 수정했습니다.
- 병렬 쿼리와 함께 해시 조인 알고리즘을 사용하여 LEFT- 또는 RIGHT-JOIN 작업과 관련된 쿼리를 실행할 때 불완전한 결과 집합이 발생할 수 있는 문제를 수정했습니다.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 8.0.44까지의 모든 커뮤니티 버그 수정이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 3.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그 단원을 참조하십시오.

- 양식 쿼리로 인해에서 어설션 실패가 `SELECT 1 FROM t WHERE CAST(a AS UNSIGNED INTEGER) = 1 AND a = (SELECT 1 FROM t)` 발생하는 문제를 수정했습니다`item_func.cc`. (커뮤니티 버그 수정 #36128964)
- FLUSH STATUS, COM_CHANGE_USER 및 SHOW PROCESS LIST가 동시에 실행될 때 교착 상태를 해결하는 문제를 수정했습니다. (버그#35218030)

Aurora MySQL 데이터베이스 엔진 업데이트 2025-12-16(버전 3.11.1, MySQL 8.0.43과 호환)

버전: 3.11.1

Aurora MySQL 3.11.1이 정식 버전입니다. Aurora MySQL 3.11 버전은 MySQL 8.0.43과 호환됩니다. 발생한 커뮤니티 변경 사항에 대한 자세한 내용은 MySQL 8.0 Release Notes를 참조하세요.

Aurora MySQL 버전 3의 새로운 기능에 대한 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환을 참조하세요. Aurora MySQL 버전 3과 Aurora MySQL 버전 2의 차이점에 대한 자세한 내용은 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요. Aurora MySQL 버전 3과 MySQL 8.0 Community Edition의 비교 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3과 MySQL 8.0 Community Edition 비교를 참조하세요.

현재 지원되는 Aurora MySQL 버전 2 클러스터에서 Aurora MySQL 버전 MySQL 3.11.1 클러스터로 Amazon RDS 블루/그린 배포를 사용하여 제로 가동 중지 패치(ZDP)를 활용하여 현재 위치 업그레이드를 수행하거나 스냅샷을 복원하거나 관리형 블루/그린 업그레이드를 시작할 수 있습니다. https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/blue-green-deployments-overview.html

Aurora MySQL 버전 3으로의 업그레이드 계획에 대한 자세한 내용은 Aurora MySQL 클러스터에 대한 메이저 버전 업그레이드 계획을 참조하세요. Aurora MySQL 업그레이드에 대한 일반적인 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터 업그레이드를 참조하세요.

문제 해결 정보는 Amazon Aurora 사용 설명서의 Aurora MySQL 현재 위치 업그레이드 문제 해결을 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼 및를 통해를 지원 사용할 수 있습니다지원. 자세한 내용은 Amazon Aurora 사용 설명서의 DB 클러스터 유지 관리를 참조하세요.

개선 사항

일반 개선 사항

- 데이터베이스 클러스터 생성 또는 복원을 완료하는 데 시간이 오래 걸릴 수 있는 문제를 수정했습니다.
- 데이터베이스 인스턴스와 스토리지 계층 간에 전송 중 암호화를 설정하는 동안 CPU 사용량 오버헤드를 줄였습니다.
- 일부 SQL 문이 감사 로그에 로깅되지 않을 수 있는 문제를 수정했습니다.

Aurora MySQL 데이터베이스 엔진 업데이트 2025-11-13(버전 3.11.0, MySQL 8.0.43과 호환)

버전: 3.11.0

Aurora MySQL 3.11.0을 정식 버전으로 사용할 수 있습니다. Aurora MySQL 3.11 버전은 MySQL 8.0.43과 호환됩니다. 발생한 커뮤니티 변경 사항에 대한 자세한 내용은 MySQL 8.0 Release Notes를 참조하세요.

Aurora MySQL 버전 3의 새로운 기능에 대한 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환을 참조하세요. Aurora MySQL 버전 3과 Aurora MySQL 버전 2의 차이점에 대한 자세한 내용은 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요. Aurora MySQL 버전 3과 MySQL 8.0 Community Edition의 비교 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3과 MySQL 8.0 Community Edition 비교를 참조하세요.

현재 지원되는 Aurora MySQL 버전 2 클러스터에서 Aurora MySQL 버전 MySQL 3.11.0 클러스터로 Amazon RDS 블루/그린 배포를 사용하여 제로 가동 중지 패치(ZDP)를 활용하여 현재 위치 업그레이드를 수행하거나 스냅샷을 복원하거나 관리형 블루/그린 업그레이드를 시작할 수 있습니다.

Aurora MySQL 버전 3으로의 업그레이드 계획에 대한 자세한 내용은 Aurora MySQL 클러스터에 대한 메이저 버전 업그레이드 계획을 참조하세요. Aurora MySQL 업그레이드에 대한 일반적인 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터 업그레이드를 참조하세요.

문제 해결 정보는 Amazon Aurora 사용 설명서의 Aurora MySQL 현재 위치 업그레이드 문제 해결을 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼 및를 통해를지원사용할 수 있습니다지원. 자세한 내용은 Amazon Aurora 사용 설명서의 DB 클러스터 유지 관리를 참조하세요.

개선 사항

보안 문제 및 CVEs 수정

중간 CVEs

- CVE-2025-50077
- CVE-2025-50078
- CVE-2025-50079
- CVE-2025-50080
- CVE-2025-50082
- CVE-2025-50083
- CVE-2025-50084
- CVE-2025-50085
- CVE-2025-50086
- CVE-2025-50087
- CVE-2025-50088
- CVE-2025-50091
- CVE-2025-50092
- CVE-2025-50093
- CVE-2025-50094
- CVE-2025-50096
- CVE-2025-50097
- CVE-2025-50099
- CVE-2025-50101
- CVE-2025-50102
- CVE-2025-53023

낮은 CVEs

- CVE-2025-50098
- CVE-2025-50100

- CVE-2025-50104

가용성 개선

- 중단된 트랜잭션이 발생할 경우 데이터베이스가 다시 시작되지 않도록 할 수 aurora_enhanced_binlog 있는의 논리적 복구 문제를 수정했습니다.
- 전역 쓰기 전달 또는 로컬 쓰기 전달이 비활성화될 때 라이터 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 둘 이상의 복제본이 요청을 전달하는 동안 쓰기 전달이 비활성화될 때 라이터가 응답하지 않을 수 있는 문제를 수정했습니다.
- 실행 중인 병렬 쿼리EXPLAIN FOR CONNECTION <query-id>에서 실행한 KILL <query-id> 후 실행할 때 엔진이 다시 시작될 수 있는 문제를 수정했습니다.
- 데이터베이스 재시작 후 out-of-memory(OOM) 회피가 구성된 aurora_oom_response DB 파라미터 값을 유지하지 않는 문제를 수정했습니다.
- MySQL Community Edition에서 지원되는 대로 CONNECTION_ADMIN 또는 SUPER 권한이 있는 사용자가 max_connections 제한을 초과하는 추가 연결을 수행하지 못하는 문제를 수정했습니다.
- 중요한 메모리 페이지가 스왑되지 않도록 하여 DB 인스턴스가 다시 시작되는 Aurora Serverless V2 조정 문제를 수정했습니다.
- innodb_purge_threads 파라미터가 기본값과 다른 값으로 수동으로 구성된 경우 Aurora MySQL Serverless v2 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다. 이제 Aurora Serverless v2 인스턴스에 대해 innodb_purge_threads 파라미터가 자동으로 관리되며 수정할 수 없습니다.
- 라이터가 많은 수의 DDL 작업을 실행할 때 리더 인스턴스의 재시작이 실패할 수 있는 문제를 해결했습니다.
- 전역 쓰기 전달 또는 로컬 쓰기 전달을 사용하는 동안 리더 인스턴스가 다시 시작될 경우 라이터 인스턴스가 응답하지 않을 수 있는 문제를 수정했습니다.
- 활성 트랜잭션과의 SSL/TLS 연결을 보존할 때 제로 가동 중지 패치 적용(ZDP)/제로 가동 중지 재시작(ZDR) 중에 엔진이 다시 시작될 수 있는 문제를 수정했습니다.
- 드문 경우지만 쿼리 처리 스레드와 호환되지 않는 순서로 라이터 인스턴스의 변경 사항이 리더 인스턴스로 전송될 수 있는 문제를 수정했습니다. 이로 인해 쿼리 처리 스레드와 복제 적용 스레드 간에 교착 상태가 발생하여 리더 인스턴스가 다시 시작될 수 있습니다.
- 모니터링 에이전트의 잘못된 상태 평가로 인해 불필요한 데이터베이스 서버 재시작이 발생하지 않도록 문제를 수정했습니다.

일반 개선 사항

- 장기 실행 트랜잭션을 실행하는 동안 데이터베이스에 대한 쓰기가 중지되어 데이터베이스가 다시 시작되거나 메이저 버전 업그레이드가 실패하는 문제를 수정했습니다.
- 라이터를 조인할 때 복제본이 잘못 다시 시작되는 문제를 수정했습니다.
- Aurora 스토리지 볼륨이 큰 데이터베이스 인스턴스에 최적화된 데이터베이스 성능.
- 범위 경계를 계산할 때 상수 표현식이 인식되지 않는 범위 최적화 프로그램의 문제를 수정했습니다. (커뮤니티 버그 #112737)
- null 값이 있는 JSON 열이 포함된 테이블을 내보낼 때 DB 스냅샷을 Amazon S3로 내보내는 속도가 느려지는 문제를 수정했습니다.
- 이진 로깅을 사용할 때 트랜잭션 롤백 후 임시 이진 로그 파일이 제대로 정리되지 않는 문제를 수정했습니다. 이 수정은 스토리지 소비가 임시 파일을 보관하는 것을 방지하며 경우에 따라 바이너리 로그 파일의 이상을 방지할 수도 있습니다.
- 시스템 변수가 로 설정된 경우 쓰기 IOPS 성능이 개선`innodb_flush_log_at_trx_commit`되었습니다`0`.
- 잘못된 오류 코드로 경고를 복원하는 동안 제로 가동 중지 패치 적용(ZDP)/제로 가동 중지 재시작(ZDR)으로 인해 DB 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 가 임계 시간(몇 분) 후 메모리 압력을 해결하지 `aurora_oom_response` 못하면 `aurora_oom_response` 작업(구성된 경우 인쇄 제외)을 자동으로 비활성화합니다.
- 부분 JSON 업데이트가 있는 열이 병렬 내보내기에 실패하여 내부 폴백이 훨씬 느린 Amazon RDS 내보내기로 이어지는 문제를 수정했습니다.
- Blob 필드가 있는 테이블에서 병렬 쿼리 작업을 수행할 때 메모리 관리 문제가 발생할 수 있는 문제를 수정했습니다.
- 향상된 Binlog가 활성화되고 ON 또는 로 `gtid_mode` 설정된 상태에서 `Previous_gtids` binlog 이벤트가 특정 GTIDs를 제외할 수 있는 문제를 수정했습니다`ON_PERMISSIVE`.
- 쓰기 전달이 활성화된 경우 리더 인스턴스에서 예기치 않은 "내부 쓰기 전달 오류"가 발생하는 문제를 수정했습니다.
- Aurora Serverless 스케일 다운 작업 중에 또는 버퍼 풀에서 페이지 제거 중에 버퍼 풀에서 잘못된 페이지 읽기가 발생할 수 있는 경합 조건을 해결했습니다. 커뮤니티 버그 #116305를 참조하세요.
- 생성된 표현식이 있는 보조 인덱스가 포함된 테이블을 내보낼 때 DB 스냅샷을 Amazon S3로 내보내는 속도가 느려지는 문제를 수정했습니다.
- 여러 연결이 예기치 않게 종료될 때 `AbortedClients` 지표가 부정확해지는 문제를 수정했습니다.
- 사용자가AWS Lambda요청 제한 시간(기본값: 10초)을 구성할 수 `aurora_lambda_request_timeout` 있도록 새 전역 변수를 추가했습니다. Aurora MySQL DB 클

러스터에서 Lambda 함수를 호출하는 방법에 대한 자세한 내용은 Aurora MySQL DB 클러스터에서 Lambda 함수 호출을 참조하세요.

- 제로 가동 중지 시간 재시작으로 데이터베이스 인스턴스가 다시 시작된 후 로컬 쓰기 전달 작동이 중지되는 문제를 수정했습니다.
- 가 낮은 값으로 설정된 경우 데이터베이스 인스턴스 재시작 작업이 실패할 수 있는 문제를 수정max_user_connections했습니다.
- 대용량(>64TB)에 대한 부트스트랩 프로세스를 최적화하여 병렬 내보내기 성능을 개선하여 전체 내보내기 작업 시간을 단축했습니다.
- 읽기 쿼리와 ALTER TABLE 병렬로 수행할 때 라이터 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 14TB보다 큰 테이블이 있는 경우 DB 클러스터 내보내기가 예상보다 훨씬 오래 걸릴 수 있는 문제를 수정했습니다.
- Aurora_pq_request_in_progress 카운터가 정확하게 업데이트되지 않은 EXPLAIN ANALYZE 문을 실행하는 동안 병렬 쿼리 요청을 부정확하게 추적하는 문제를 수정했습니다.
- 흔하지 않은 조건에서 데이터베이스 볼륨이 최대 허용 크기에 가까울 때 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 제로 가동 중지 재시작 및 제로 가동 중지 패치 적용 중에 연결을 복원할 때 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 8.0.43까지의 모든 커뮤니티 버그 수정이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 3.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그 단원을 참조하십시오.

- 양식의 쿼리로 인해에서 어설션 실패가 SELECT 1 FROM t WHERE CAST(a AS UNSIGNED INTEGER) = 1 AND a = (SELECT 1 FROM t) 발생하는 문제를 수정했습니다item_func.cc. (커뮤니티 버그 수정 #36128964)
- 양식 쿼리는에서 어설션으로 SELECT 1 FROM t WHERE CAST(a AS UNSIGNED INTEGER) = 1 AND a = (SELECT 1 FROM t) 이어졌습니다item_func.cc. (버그 #36128964)

커뮤니티 변경 사항에 대한 자세한 내용은 MySQL 8.0.43 릴리스 정보를 참조하세요.

Aurora MySQL 데이터베이스 엔진 업데이트 2026-01-02(버전 3.10.3, MySQL 8.0.42와 호환)

버전: 3.10.3

Aurora MySQL 3.10.3이 정식 버전입니다. Aurora MySQL 3.10 버전은 MySQL 8.0.42와 호환됩니다. 발생한 커뮤니티 변경 사항에 대한 자세한 내용은 MySQL 8.0 Release Notes를 참조하세요.

Aurora MySQL 버전 3의 새로운 기능에 대한 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환을 참조하세요. Aurora MySQL 버전 3과 Aurora MySQL 버전 2의 차이점에 대한 자세한 내용은 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요. Aurora MySQL 버전 3과 MySQL 8.0 Community Edition의 비교 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3과 MySQL 8.0 Community Edition 비교를 참조하세요.

현재 지원되는 Aurora MySQL 버전 2 클러스터에서 Aurora MySQL 버전 MySQL 3.10.3 클러스터로 Amazon RDS 블루/그린 배포를 사용하여 제로 가동 중지 패치(ZDP)를 활용하여 현재 위치 업그레이드를 수행하거나 스냅샷을 복원하거나 관리형 블루/그린 업그레이드를 시작할 수 있습니다. https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/blue-green-deployments-overview.html

Aurora MySQL 버전 3으로의 업그레이드 계획에 대한 자세한 내용은 Aurora MySQL 클러스터에 대한 메이저 버전 업그레이드 계획을 참조하세요. Aurora MySQL 업그레이드에 대한 일반적인 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터 업그레이드를 참조하세요.

문제 해결 정보는 Amazon Aurora 사용 설명서의 Aurora MySQL 현재 위치 업그레이드 문제 해결을 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼 및를 통해를 지원 사용할 수 있습니다지원. 자세한 내용은 Amazon Aurora 사용 설명서의 DB 클러스터 유지 관리를 참조하세요.

개선 사항

일반적인 개선 사항:

- 데이터베이스 인스턴스와 스토리지 계층 간에 전송 중 암호화를 설정하는 동안 CPU 사용량 오버헤드를 줄였습니다.
- 일부 SQL 문이 감사 로그에 로깅되지 않을 수 있는 문제를 수정했습니다.

Aurora MySQL 데이터베이스 엔진 업데이트 2025-11-20(버전 3.10.2, MySQL 8.0.42와 호환)

버전: 3.10.2

Aurora MySQL 3.10.2가 정식 버전입니다. Aurora MySQL 3.10 버전은 MySQL 8.0.42와 호환됩니다. 발생한 커뮤니티 변경 사항에 대한 자세한 내용은 MySQL 8.0 Release Notes를 참조하세요.

Aurora MySQL 버전 3의 새로운 기능에 대한 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환을 참조하세요. Aurora MySQL 버전 3과 Aurora MySQL 버전 2의 차이점에 대한 자세한 내용은 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요. Aurora MySQL 버전 3과 MySQL 8.0 Community Edition의 비교 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3과 MySQL 8.0 Community Edition 비교를 참조하세요.

현재 지원되는 Aurora MySQL 버전 2 클러스터에서 Aurora MySQL 버전 MySQL 3.10.2 클러스터로 Amazon RDS 블루/그린 배포를 사용하여 제로 가동 중지 패치(ZDP)를 활용하여 현재 위치 업그레이드를 수행하거나 스냅샷을 복원하거나 관리형 블루/그린 업그레이드를 시작할 수 있습니다. https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/blue-green-deployments-overview.html

Aurora MySQL 버전 3으로의 업그레이드 계획에 대한 자세한 내용은 Aurora MySQL 클러스터에 대한 메이저 버전 업그레이드 계획을 참조하세요. Aurora MySQL 업그레이드에 대한 일반적인 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터 업그레이드를 참조하세요.

문제 해결 정보는 Amazon Aurora 사용 설명서의 Aurora MySQL 현재 위치 업그레이드 문제 해결을 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼 및를 통해를지원사용할 수 있습니다지원. 자세한 내용은 Amazon Aurora 사용 설명서의 DB 클러스터 유지 관리를 참조하세요.

개선 사항

가용성 향상:

- 활성 트랜잭션과 SSL/TLS 연결을 유지할 때 제로 가동 중지 패치 적용(ZDP)/제로 가동 중지 재시작(ZDR) 중에 엔진이 다시 시작될 수 있는 문제를 수정했습니다.
- 실행 중인 병렬 쿼리`EXPLAIN FOR CONNECTION <query-id>`에서 실행한 `KILL <query-id>` 후 실행할 때 엔진이 다시 시작될 수 있는 문제를 수정했습니다.
- 2개 이상의 복제본이 요청을 전달하는 동안 쓰기 전달이 비활성화될 때 라이터가 응답하지 않을 수 있는 문제를 수정했습니다.
- 전역 쓰기 전달 또는 로컬 쓰기 전달이 비활성화될 때 라이터 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.

일반적인 개선 사항:

- `Aurora_pq_request_in_progress` 카운터가 정확하게 업데이트되지 않은 `EXPLAIN ANALYZE` 문을 실행하는 동안 병렬 쿼리 요청을 부정확하게 추적하는 문제를 수정했습니다.
- 가 임계 시간(몇 분) 후 메모리 압력을 해결하지 `aurora_oom_response` 못하면 `aurora_oom_response` 작업(구성된 경우 인쇄 제외)을 자동으로 비활성화합니다.
- 시스템 변수가 로 설정된 경우 쓰기 IOPS 성능이 개선`innodb_flush_log_at_trx_commit`되었습니다`0`.
- 라이터를 조인할 때 복제본이 잘못 다시 시작되는 문제를 수정했습니다.
- 장기 실행 트랜잭션을 실행하는 동안 데이터베이스에 대한 쓰기가 중지되어 데이터베이스가 다시 시작되거나 메이저 버전 업그레이드가 실패하는 문제를 수정했습니다.

Aurora MySQL 데이터베이스 엔진 업데이트 2025년 9월 30일(버전 3.10.1, MySQL 8.0.42와 호환)

버전: 3.10.1

Aurora MySQL 3.10.1이 정식 버전입니다. Aurora MySQL 3.10 버전은 MySQL 8.0.42와 호환됩니다. 발생한 커뮤니티 변경 사항에 대한 자세한 내용은 MySQL 8.0 Release Notes를 참조하세요.

Aurora MySQL 버전 3의 새로운 기능에 대한 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환을 참조하세요. Aurora MySQL 버전 3과 Aurora MySQL 버전 2의 차이점에 대한 자세한 내용은 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요. Aurora MySQL 버전 3과 MySQL 8.0 Community Edition의 비교 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3과 MySQL 8.0 Community Edition 비교를 참조하세요.

제로 가동 중지 패치(ZDP)를 활용하는 현재 위치 업그레이드를 수행하고, Amazon RDS 블루/그린 배포를 사용하여 현재 지원되는 Aurora MySQL 버전 2 클러스터에서 Aurora MySQL 버전 3.10.1 클러스터로 스냅샷을 복원하거나 관리형 블루/그린 업그레이드를 시작할 수 있습니다.

Aurora MySQL 버전 3으로의 업그레이드 계획에 대한 자세한 내용은 Aurora MySQL 클러스터에 대한 메이저 버전 업그레이드 계획을 참조하세요. Aurora MySQL 업그레이드에 대한 일반 정보는 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터 업그레이드를 참조하세요.

문제 해결 정보는 Amazon Aurora 사용 설명서의 Aurora MySQL 현재 위치 업그레이드 문제 해결을 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 DB 클러스터 유지 관리를 참조하세요.

개선 사항

가용성 향상:

- 리더 인스턴스로 전송된 페이지 래칭 순서 관련 문제를 수정했습니다. 라이터가 업데이트하는 테이블에 대해 SELECT 쿼리를 실행할 때 교착 상태로 인해 리더 인스턴스가 다시 시작될 수 있습니다.
- 라이터가 많은 수의 DDL 작업을 실행할 때 리더 인스턴스의 재시작이 실패할 수 있는 문제를 해결했습니다.
- innodb_purge_threads 파라미터를 기본값과 다른 값으로 수동으로 구성할 때 Aurora MySQL Serverless v2 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다. 이제 innodb_purge_threads 파라미터가 Aurora Serverless v2 인스턴스에 대해 자동으로 관리되며 수정할 수 없습니다.
- 중요한 메모리 페이지가 스왑되지 않도록 하여 DB 인스턴스가 다시 시작되는 Aurora Serverless V2 조정 문제를 수정했습니다.
- 데이터베이스 재시작 후 Aurora 메모리 부족(OOM) 회피로 구성된 aurora_oom_response DB 파라미터 값이 유지되지 않는 문제를 수정했습니다.
- 다중 AZ 클러스터에서 장애 조치 후 가용성 문제를 일으킬 수 있는 경쟁 조건을 수정했습니다.
- aurora_enhanced_binlog를 활성화하면 트랜잭션 중단 시 데이터베이스가 다시 시작되지 않는 논리적 복구 문제를 수정했습니다.
- 병렬 쿼리(PQ) 워크로드 중에 데이터베이스 인스턴스가 응답하지 않을 수 있는 문제를 해결했습니다.
- MySQL Community Edition에서 지원되는 대로 CONNECTION_ADMIN 또는 SUPER 권한이 있는 사용자가 max_connections 제한을 초과하는 추가 연결을 수행하지 못하는 문제를 수정했습니다.

일반적인 개선 사항

- 볼륨이 특정 임곗값보다 큰 후 특정 유형의 쿼리가 실수로 허용되어 볼륨이 최대 크기로 증가할 때 연속 데이터베이스 서버가 다시 시작될 수 있는 문제를 수정했습니다.
- 전역 쓰기 전달 또는 로컬 쓰기 전달을 사용하는 동안 리더 인스턴스가 다시 시작될 경우 라이터 인스턴스가 응답하지 않을 수 있는 문제를 수정했습니다.
- 읽기 쿼리와 함께 ALTER TABLE을 수행할 때 라이터 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 병렬 내보내기 작업 중에 대용량(> 64TB) 부트스트래핑으로 가용성을 개선하는 문제를 수정했습니다.
- max_user_connections가 낮은 값으로 설정된 경우 데이터베이스 인스턴스 재시작 작업이 실패할 수 있는 문제를 수정했습니다.

- 여러 연결이 예기치 않게 종료될 때 AbortedClients 지표가 부정확해지는 문제를 수정했습니다.
- Aurora Serverless 스케일 다운 작업 중에 또는 버퍼 풀에서 페이지 제거 중에 버퍼 풀에서 잘못된 페이지 읽기가 발생할 수 있는 경합 조건을 해결했습니다. 커뮤니티 버그 #116305를 참조하세요.
- 쓰기 전달이 활성화된 경우 리더 인스턴스에서 예기치 않은 "내부 쓰기 전달 오류"가 발생하는 문제를 수정했습니다.
- 향상된 Binlog가 활성화되고 gtid_mode가 ON 또는 ON_PERMISSIVE로 설정된 특정 GTID에서 제외할 수 있는 문제를 수정했습니다.
- Blob 필드가 있는 테이블에서 병렬 쿼리 작업을 수행할 때 메모리 관리 문제가 발생할 수 있는 문제를 수정했습니다.
- 부분 JSON 업데이트가 있는 열이 병렬 내보내기에 실패하여 내부 폴백이 훨씬 느린 RDS 내보내기로 이어지는 문제를 수정했습니다.
- 동시 쓰기 전달 워크로드가 많을 때 예기치 않은 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 잘못된 오류 코드로 경고를 복원하는 동안 제로 가동 중지 패치(ZDP)/제로 가동 중지 재시작(ZDR)으로 인해 DB 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 이진 로깅을 사용할 때 트랜잭션 롤백 후 임시 이진 로그 파일이 제대로 정리되지 않는 문제를 수정했습니다. 이 수정은 불필요한 스토리지 소비가 임시 파일을 보관하는 것을 방지하며, 경우에 따라 이진 로그 파일의 이상을 방지할 수도 있습니다.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 MySQL 8.0.42까지의 모든 커뮤니티 버그 수정 사항이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 3.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그 섹션을 참조하세요.

- SELECT 1 FROM t WHERE CAST(a AS UNSIGNED INTEGER) = 1 AND a = (SELECT 1 FROM t) 형식의 쿼리는 item_func.cc에서 어설션으로 이어졌습니다. (버그 #36128964)

Aurora MySQL 데이터베이스 엔진 업데이트 2025년 7월 31일(버전 3.10.0, MySQL 8.0.42와 호환)

버전: 3.10.0

Aurora MySQL 3.10.0이 정식 버전입니다. Aurora MySQL 3.10 버전은 MySQL 8.0.42와 호환됩니다. 발생한 커뮤니티 변경 사항에 대한 자세한 내용은 MySQL 8.0 Release Notes를 참조하세요.

> **Note**
>
> 이 버전은 장기 지원(LTS) 릴리스로 지정됩니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL LTS(장기 지원) 릴리스를 참조하세요.
> LTS 버전의 경우 `AutoMinorVersionUpgrade` 파라미터를 `true`로 설정(또는 AWS Management Console에서 자동 마이너 버전 업그레이드 활성화)하지 않는 것이 좋습니다. 이렇게 하면 DB 클러스터가 LTS 버전이 아닐 수 있는 마이너 버전 자동 업그레이드 캠페인의 다음 대상 버전으로 업그레이드될 수 있습니다.

Aurora MySQL 버전 3의 새로운 기능에 대한 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환을 참조하세요. Aurora MySQL 버전 3과 Aurora MySQL 버전 2의 차이점에 대한 자세한 내용은 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요. Aurora MySQL 버전 3과 MySQL 8.0 Community Edition의 비교 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3과 MySQL 8.0 Community Edition 비교를 참조하세요.

제로 가동 중지 패치(ZDP)를 활용하는 현재 위치 업그레이드를 수행하고, Amazon RDS 블루/그린 배포를 사용하여 현재 지원되는 Aurora MySQL 버전 2 클러스터에서 Aurora MySQL 버전 3.10.0 클러스터로 스냅샷을 복원하거나 관리형 블루/그린 업그레이드를 시작할 수 있습니다.

Aurora MySQL 버전 3으로의 업그레이드 계획에 대한 자세한 내용은 Aurora MySQL 클러스터에 대한 메이저 버전 업그레이드 계획을 참조하세요. Aurora MySQL 업그레이드에 대한 일반 정보는 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터 업그레이드를 참조하세요.

문제 해결 정보는 Amazon Aurora 사용 설명서의 Aurora MySQL 현재 위치 업그레이드 문제 해결을 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 DB 클러스터 유지 관리를 참조하세요.

새로운 특성

- Aurora MySQL 버전 3.10은 바이너리 로그 복제본에 대한 인 메모리 릴레이 로그 캐시 지원을 확장합니다. 버전 3.05에 처음 도입된 이 기능은 이진 로그 복제 처리량을 최대 40%까지 개선할 수 있습니다. 인 메모리 릴레이 로그 캐시는 기본적으로 단일 스레드 이진 로그 복제, GTID 자동 위치 지정이 활성화된 다중 스레드 복제에 대해 활성화되며 버전 3.10부터 `replica_preserve_commit_order = ON`(GTID가 없는 경우에도)을 사용한 다중 스레드 복제

에도 활성화됩니다. 3.10의 새 파라미터, `aurora_in_memory_relaylog`를 사용하여 이 기능을 제어할 수 있습니다. 자세한 내용은 Aurora MySQL에서 이진 로그 최적화를 참조하세요.

- Amazon Aurora는 최대 스토리지 용량을 128TiB에서 256TiB로 두 배로 늘려 단일 데이터베이스 클러스터에서 더 큰 워크로드를 지원합니다. Aurora MySQL에서 증가된 스토리지 제한을 사용하려면 클러스터를 버전 3.10(MySQL 8.0.42와 호환) 이상으로 업그레이드합니다. 업그레이드 후 Aurora 스토리지는 클러스터 볼륨의 데이터 양에 따라 최대 256TiB까지 자동으로 확장됩니다.

개선 사항

보안 수정

중간 CVE:

- CVE-2025-21501
- CVE-2025-21500
- CVE-2025-21543
- CVE-2025-21540
- CVE-2025-21491
- CVE-2025-21490
- CVE-2025-21559
- CVE-2025-21555
- CVE-2025-21497
- CVE-2025-21519
- CVE-2025-21529
- CVE-2025-21505
- CVE-2025-21531
- CVE-2025-21523
- CVE-2025-21503
- CVE-2025-21522
- CVE-2025-21518
- CVE-2025-21577
- CVE-2025-30682
- CVE-2025-30687

- CVE-2025-30688
- CVE-2025-21574
- CVE-2025-21575
- CVE-2025-30693
- CVE-2025-30695
- CVE-2025-30715
- CVE-2025-21584
- CVE-2025-21580
- CVE-2025-21581
- CVE-2025-21585
- CVE-2025-30689
- CVE-2025-21579
- CVE-2025-30696
- CVE-2025-30705
- CVE-2025-30683
- CVE-2025-30684
- CVE-2025-30685
- CVE-2025-30699
- CVE-2025-30704
- CVE-2025-30721

낮은 CVE:

- CVE-2025-21520
- CVE-2025-21546
- CVE-2025-30703
- CVE-2025-30681

가용성 향상:

- 복구 진행 상황을 잘못 평가하여 불필요한 데이터베이스 서버 재시작이 발생하는 문제를 수정했습니다.

일반적인 개선 사항

- I/O 최적화가 활성화된 경우 커밋 지연 시간이 단축됩니다.
- 쓰기 전달이 활성화되고 파라미터 `aurora_replica_read_consistency`가 수정될 때 리더 인스턴스가 오류 로그를 생성하지 않는 문제를 수정했습니다.
- 복제 스레드와 버퍼 캐시에 없는 테이블에 액세스하는 쿼리 간의 상호 작용으로 인해 리더 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 제로 가동 중지 재시작으로 데이터베이스 인스턴스가 다시 시작된 후 로컬 쓰기 전달 작동이 중지되는 문제를 수정했습니다.
- 세트 옵션이 포함된 문에 대해 전달된 쿼리가 잘못 작동할 수 있는 쓰기 전달 문제를 수정했습니다.
- 실행 취소 테이블스페이스에 메타데이터를 삽입하면 예기치 않은 데이터베이스 재시작이 트리거되는 안정성 문제가 수정되었습니다.
- 14TB보다 큰 테이블이 있는 경우 DB 클러스터 내보내기가 예상보다 훨씬 오래 걸릴 수 있는 문제를 수정했습니다.
- `Innodb_buffer_pool_pages_misc` 상태 변수를 잘못 보고할 수 있는 문제를 수정했습니다.
- 제로 가동 중지 패치(ZDP) 또는 제로 가동 중지 재시작(ZDR) 중 `LAST_INSERT_ID` 보존에 대한 지원이 추가되었습니다.
- 이 제한을 초과하면 클러스터를 사용할 수 없게 될 수 있으므로 Aurora MySQL은 가상 인덱스 ID에 8비트 값을 사용하여 MySQL 실행 취소 형식 문제를 방지합니다. 이 제한에 가까워지면 시스템은 이제 MySQL 오류 로그에 경고 메시지를 기록합니다. 한도에 도달하면 새 인덱스를 추가하려고 할 때 오류가 반환됩니다. 가상 인덱스 모범 사례에 대한 자세한 내용은 Aurora MySQL 설명서의 가상 인덱스 ID 오버플로 오류를 참조하세요.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 MySQL 8.0.42까지의 모든 커뮤니티 버그 수정 사항이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 3.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그 섹션을 참조하세요.

- 다중 스레드가 메타데이터 성능 스키마 개체를 등록 및 등록 취소하거나 메타데이터 잠금을 획득 및 해제하려고 동시에 시도하면 서버 종료가 발생할 수 있습니다. (버그 #26502135)

Aurora MySQL 데이터베이스 엔진 업데이트 2025년 5월 14일(버전 3.09.0, MySQL 8.0.40과 호환)

버전: 3.09.0

Aurora MySQL 3.09.0이 정식 버전입니다. Aurora MySQL 3.09 버전은 MySQL 8.0.40와 호환됩니다. 발생한 커뮤니티 변경 사항에 대한 자세한 내용은 MySQL 8.0 Release Notes를 참조하세요.

Aurora MySQL 버전 3의 새로운 기능에 대한 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환을 참조하세요. Aurora MySQL 버전 3과 Aurora MySQL 버전 2의 차이점에 대한 자세한 내용은 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요. Aurora MySQL 버전 3과 MySQL 8.0 Community Edition의 비교 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3과 MySQL 8.0 Community Edition 비교를 참조하세요.

제로 가동 중지 패치를 활용하는 현재 위치 업그레이드를 수행하고, Amazon RDS 블루/그린 배포를 사용하여 현재 지원되는 Aurora MySQL 버전 2 클러스터에서 Aurora MySQL 버전 3.09.0 클러스터로 스냅샷을 복원하거나 관리형 블루/그린 업그레이드를 시작할 수 있습니다.

Aurora MySQL 버전 3으로의 업그레이드 계획에 대한 자세한 내용은 Aurora MySQL 클러스터에 대한 메이저 버전 업그레이드 계획을 참조하세요. Aurora MySQL 업그레이드에 대한 일반 정보는 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터 업그레이드를 참조하세요.

문제 해결 정보는 Amazon Aurora 사용 설명서의 Aurora MySQL 현재 위치 업그레이드 문제 해결을 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 DB 클러스터 유지 관리를 참조하세요.

릴리스 하이라이트

- 보조 리더 인스턴스가 예상치 못한 이벤트(하드웨어 장애, 네트워크 중단) 중에 시작을 완료하고 읽기 요청을 처리할 수 있도록 Aurora MySQL 글로벌 데이터베이스를 개선했습니다. 이전에는 이러한 이벤트 중에 보조 리더 인스턴스를 다시 시작할 수 없었습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 글로벌 데이터베이스 보조 클러스터에 대한 리전 간 복원력을 참조하세요.
- Aurora MySQL Global Database 리전 간 전환 중 라이터 가동 중지 시간을 일반적으로 1분 미만으로 줄여 계획된 리전 전환 중 가동 중지 시간을 최소화했습니다.

개선 사항

보안 수정

중요 CVE:

- CVE-2024-11053
- CVE-2024-37371

중간 CVE:

- CVE-2024-7264
- CVE-2024-21193
- CVE-2024-21194
- CVE-2024-21196
- CVE-2024-21197
- CVE-2024-21198
- CVE-2024-21199
- CVE-2024-21201
- CVE-2024-21203
- CVE-2024-21207
- CVE-2024-21212
- CVE-2024-21213
- CVE-2024-21218
- CVE-2024-21219
- CVE-2024-21230
- CVE-2024-21236
- CVE-2024-21238
- CVE-2024-21239
- CVE-2024-21241
- CVE-2025-21494
- CVE-2025-21504
- CVE-2025-21525
- CVE-2025-21534
- CVE-2025-21536

가용성 향상:

- `innodb_flush_log_at_trx_commit`이 라이터에서 `0`으로 설정되고 리더에서 0이 아닌 경우 리더에서 라이터 인스턴스로 전달된 다중 문 쿼리가 중단되어 잠재적인 쓰기 전달 실패를 방지하는 문제를 수정했습니다.
- 향상된 Binlog를 활성화하면 트랜잭션을 BLACKHOLE 엔진에 커밋하거나 `XA PREPARE` 문을 실행하여 `SHOW BINARY LOGS`를 동시에 실행할 때 데이터베이스가 재부팅될 수 있는 교착 문제를 수정하여 잠재적인 쓰기 중단 및 인스턴스 가용성 문제를 방지했습니다.
- 이전 요청이 완전히 완료되기 전에 새 요청이 수락되지 않도록 하여 Aurora 라이터 인스턴스가 다시 시작될 수 있는 쓰기 전달의 경합 조건을 수정하여 쓰기 전달 작업의 안정성을 개선했습니다.
- 복제본에서 네트워크 중단으로 인해 라이터와의 연결이 올바르게 재설정되지 않아 복제가 중단되고 잠재적 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- Aurora MySQL 메모리 부족(OOM) 응답은 이제 `aurora_oom_response` DB 파라미터를 통해 활성화할 때 시스템 메모리 상태(LOW/RESERVED)를 기반으로 메모리 사용량을 점진적으로 줄이는 단계별 버퍼 풀 크기 조정을 구현하여 메모리 압력 상황에서 더 나은 메모리 관리를 제공합니다.
- Binlog 파일 크기에 관계없이 일정한 시간이 소요되도록 복구 프로세스를 최적화하여 데이터베이스 재시작 중 Binlog 파일 복구 시간을 개선했습니다. 이전에는 경우에 따라 복구 시간이 마지막 Binlog 파일의 크기에 비례합니다.
- `performance_schema.data_lock_waits` 쿼리 중에 동시 InnoDB 테이블 잘라내기 작업을 실행할 때 예기치 않은 MySQL 서버가 다시 시작될 수 있는 문제를 수정했습니다.
- 스토리지가 부족한 상태에서 대용량 binlog 이벤트를 커밋할 때 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 워크로드가 많은 시나리오에서 메모리 부족(OOM) 방지 중에 트리거된 버퍼 풀 크기 조정 작업이 응답하지 않아 데이터베이스가 다시 시작될 수 있는 문제를 수정했습니다.
- 트리거를 생성할 때 데이터베이스 재시작 루프가 발생할 수 있는 문제를 해결했습니다. 새 Binlog 또는 Relaylog 파일이 추가되거나 이러한 파일이 교체될 때도 문제가 발생할 수 있습니다.
- 다중 문 또는 암시적 커밋 쿼리와 함께 쓰기 전달을 사용할 때 Aurora 리더 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.

일반적인 개선 사항

- `ALTER TABLE ... REBUILD / OPTIMIZE TABLE` 작업에서 스레드 간에 버퍼 크기를 나누는 대신 DDL 스레드당 `innodb_ddl_buffer_size` 바이트를 할당하여 과도한 메모리를 소비할 수 있는 문제를 수정하여 DDL 작업 중에 잠재적인 메모리 과다 사용률을 방지했습니다.

- 메모리가 4GiB를 초과하는 모든 DB 인스턴스 클래스에서 `aurora_oom_response`의 기본값을 print에서 print,decline,kill_connect로 변경했습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL 메모리 부족 문제를 참조하세요.
- 다음 권한이 `rds_superuser_role`, `FLUSH_OPTIMIZER_COSTS`, `FLUSH_STATUS`, `FLUSH_TABLES`, `FLUSH_USER_RESOURCES`에 추가되었습니다. `rds_superuser_role`에 대한 자세한 내용은 Amazon Aurora를 사용하는 Amazon 관리자 사용자 계정을 참조하세요. 이러한 동적 권한에 대한 자세한 내용은 MySQL 설명서를 참조하세요.
- 이 Aurora MySQL 버전부터는 빠른 삽입 최적화가 더 이상 활성화되지 않습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL 성능 개선 사항을 참조하세요.
- `max_user_connections` 임곗값을 잘못 위반하여 일부 사용자에게 연결 오류가 발생하는 문제를 수정했습니다. 이는 연결이 거의 즉시 생성되고 종료되는 경우와 같은 일부 엣지 사례에서 발생합니다.
- CPU 사용률이 높아 응답하지 않는 데이터베이스 서버 인스턴스가 발생하는 감사 로깅 문제를 수정했습니다.
- XA 트랜잭션 사용 시 메모리 관리 문제를 수정하여 향상된 Binlog가 활성화된 경우 인스턴스가 다시 시작되지 않도록 했습니다.
- 데이터베이스 서버 재시작 후 버퍼풀 인덱스 통계가 잘못 업데이트되어 옵티마이저가 잘못된 비용을 추정할 때 쿼리 성능이 저하되는 문제를 수정했습니다.
- 멈춘 작업자 스레드로 인해 고객이 로컬 쓰기 전달 기능을 끄지 못하는 문제를 수정했습니다.
- 향상된 Binlog가 활성화되었거나 이전에 활성화된 클러스터에서 `SHOW BINARY LOGS` 명령이 실행되는 데 시간이 더 오래 걸리는 문제를 수정했습니다. 이 문제로 인해 여러 `SHOW BINARY LOGS` 명령이 동시에 실행되는 경우 커밋 지연 시간이 증가할 수도 있습니다.

업그레이드 및 마이그레이션:

- 삭제된 사용자에게 속한 연결을 유지하려고 시도하는 동안 제로 가동 중지 패치(ZDP)가 실패할 수 있는 문제를 수정했습니다. `DROP USER` 명령과 활성 연결에 미치는 영향에 대한 자세한 내용은 MySQL 설명서에서 확인할 수 있습니다.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 MySQL 8.0.40까지의 모든 커뮤니티 버그 수정 사항이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 3.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그 섹션을 참조하세요.

- 대규모 트랜잭션이 수신 및 적용되고 있고 복제 채널 중지 요청이 `STOP REPLICA`를 사용하여 이루어진 동안 MySQL은 제대로 처리하지 못했으며 이후에 채널 명령을 처리하지 않았습니다. 또한 서버 종료 프로세스가 정상적으로 완료되지 않아 MySQL 프로세스를 종료하거나 호스트 시스템을 다시 시작해야 했습니다. (버그 #115966, 버그 #37008345)

Aurora MySQL 데이터베이스 엔진 업데이트 2025년 4월 7일(버전 3.08.2, MySQL 8.0.39와 호환)

버전: 3.08.2

Aurora MySQL 3.08.2이 정식 버전입니다. Aurora MySQL 3.08 버전은 MySQL 8.0.39와 호환됩니다. 발생한 커뮤니티 변경 사항에 대한 자세한 내용은 MySQL 8.0 Release Notes를 참조하세요.

Aurora MySQL 버전 3의 새로운 기능에 대한 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환을 참조하세요. Aurora MySQL 버전 3과 Aurora MySQL 버전 2의 차이점에 대한 자세한 내용은 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요. Aurora MySQL 버전 3과 MySQL 8.0 Community Edition의 비교 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3과 MySQL 8.0 Community Edition 비교를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는2.11.*, 2.12.*, 3.04.*, 3.05.*, 3.06.*, 3.07.* 및 3.08.*입니다.

Amazon RDS 블루/그린 배포를 사용하여 현재 지원되는 Aurora MySQL 버전 2 클러스터에서 Aurora MySQL 버전 3.08.2 클러스터로 인플레이스 업그레이드를 수행하거나, 스냅샷을 복원하거나, 관리형 블루/그린 업그레이드를 시작할 수 있습니다.

Aurora MySQL 버전 3으로의 업그레이드 계획에 대한 자세한 내용은 Aurora MySQL 클러스터에 대한 메이저 버전 업그레이드 계획을 참조하세요. Aurora MySQL 업그레이드에 대한 일반 정보는 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터 업그레이드를 참조하세요.

문제 해결 정보는 Amazon Aurora 사용 설명서의 Aurora MySQL 현재 위치 업그레이드 문제 해결을 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

보안 수정:

- CVE-2024-11053

가용성 향상:

- 네트워크 중단으로 인해 라이터와의 연결이 올바르게 재설정되지 않는 복제본 문제를 수정했습니다.
- 라이터 인스턴스에서 `innodb_flush_log_at_trx_commit`이 `0`으로 설정된 경우 전달된 다중 문 쿼리가 실패할 수 있는 문제를 수정했습니다.
- 쓰기 전달이 활성화된 경우 Aurora 라이터 DB 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 다중 문 또는 암시적 커밋 쿼리와 함께 쓰기 전달을 사용할 때 Aurora 리더 DB 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.

일반적인 개선 사항

- 이 Aurora MySQL 버전부터는 빠른 삽입 최적화가 더 이상 활성화되지 않습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL 성능 개선 사항을 참조하세요.
- `max_user_connections` 임곗값을 잘못 위반하여 일부 사용자에게 연결 오류가 발생하는 문제를 수정했습니다. 이는 연결이 거의 즉시 생성되고 종료되는 경우와 같은 일부 엣지 사례에서 발생합니다.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 아래 사항 외에도 8.0.39까지의 모든 커뮤니티 버그 수정 사항이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 3.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그를 참조하세요.

- 대규모 트랜잭션이 수신 및 적용되고 있고 복제 채널 중지 요청이 `STOP REPLICA`를 사용하여 이루어진 동안 MySQL은 제대로 처리하지 못했으며 이후에 채널 명령을 처리하지 않았습니다. 또한 서버 종료 프로세스가 정상적으로 완료되지 않아 MySQL 프로세스를 종료하거나 호스트 시스템을 다시 시작해야 했습니다. (커뮤니티 버그 수정 #115966 및 #37008345)

Aurora MySQL 데이터베이스 엔진 업데이트 2025년 2월 17일(버전 3.08.1, MySQL 8.0.39와 호환)

버전: 3.08.1

Aurora MySQL 3.08.1이 정식 버전입니다. Aurora MySQL 3.08 버전은 MySQL 8.0.39와 호환됩니다. 발생한 커뮤니티 변경 사항에 대한 자세한 내용은 MySQL 8.0 Release Notes를 참조하세요.

Aurora MySQL 버전 3의 새로운 기능에 대한 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환을 참조하세요. Aurora MySQL 버전 3과 Aurora MySQL 버전 2의 차이점에 대한 자세한 내용은 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요. Aurora MySQL 버전 3과 MySQL 8.0 Community Edition의 비교 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3과 MySQL 8.0 Community Edition 비교를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는2.11.*, 2.12.*, 3.04.*, 3.05.*, 3.06.*, 3.07.* 및 3.08.*입니다.

Amazon RDS 블루/그린 배포를 사용하여 현재 지원되는 Aurora MySQL 버전 2 클러스터에서 Aurora MySQL 버전 3.08.1 클러스터로 인플레이스 업그레이드를 수행하거나, 스냅샷을 복원하거나, 관리형 블루/그린 업그레이드를 시작할 수 있습니다.

Aurora MySQL 버전 3으로의 업그레이드 계획에 대한 자세한 내용은 Aurora MySQL 클러스터에 대한 메이저 버전 업그레이드 계획을 참조하세요. Aurora MySQL 업그레이드에 대한 일반 정보는 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터 업그레이드를 참조하세요.

문제 해결 정보는 Amazon Aurora 사용 설명서의 Aurora MySQL 현재 위치 업그레이드 문제 해결을 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

가용성 향상:

- 데이터베이스 재시작, 패치 또는 마이너 버전 업그레이드 및 장애 조치 중에 DB 인스턴스 재시작 시간을 줄이는 개선이 이루어졌습니다.

업그레이드 및 마이그레이션:

- 삭제된 사용자의 역할 및 권한 부여를 복원하려고 시도하는 동안 제로 가동 중지 패치(ZDP)가 실패할 수 있는 문제를 수정했습니다. `DROP USER` 작업에 대한 자세한 내용은 MySQL 설명서의 DROP USER Statement를 참조하세요.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 아래 사항 외에도 8.0.39까지의 모든 커뮤니티 버그 수정 사항이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 3.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그를 참조하세요.

Aurora MySQL 데이터베이스 엔진 업데이트 2024년 11월 18일(버전 3.08.0, MySQL 8.0.39와 호환)

버전: 3.08.0

Aurora MySQL 3.08.0이 정식 버전입니다. Aurora MySQL 3.08 버전은 MySQL 8.0.39와 호환됩니다. 발생한 커뮤니티 변경 사항에 대한 자세한 내용은 MySQL 8.0 Release Notes를 참조하세요.

Aurora MySQL 버전 3의 새로운 기능에 대한 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환을 참조하세요. Aurora MySQL 버전 3과 Aurora MySQL 버전 2의 차이점에 대한 자세한 내용은 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요. Aurora MySQL 버전 3과 MySQL 8.0 Community Edition의 비교 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3과 MySQL 8.0 Community Edition 비교를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는2.11.*, 2.12.*, 3.04.*, 3.05.*, 3.06.*, 3.07.* 및 3.08.*입니다.

Amazon RDS 블루/그린 배포를 사용하여 현재 지원되는 Aurora MySQL 버전 2 클러스터에서 Aurora MySQL 버전 3.08.0 클러스터로 인플레이스 업그레이드를 수행하거나, 스냅샷을 복원하거나, 관리형 블루/그린 업그레이드를 시작할 수 있습니다.

Aurora MySQL 버전 3으로의 업그레이드 계획에 대한 자세한 내용은 Aurora MySQL 클러스터에 대한 메이저 버전 업그레이드 계획을 참조하세요. Aurora MySQL 업그레이드에 대한 일반 정보는 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터 업그레이드를 참조하세요.

문제 해결 정보는 Amazon Aurora 사용 설명서의 Aurora MySQL 현재 위치 업그레이드 문제 해결을 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

새로운 특성

- 사용자가 InnoDB 제거 프로세스를 모니터링할 수 있도록 세 가지 새로운 Amazon CloudWatch 지표가 추가되었습니다.
  - `PurgeBoundary`
  - `PurgeFinishedPoint`
  - `TruncateFinishedPoint`

  자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora에 대한 Amazon CloudWatch 지표를 참조하세요.
- 사용자가 제거 경계를 유지할 수 있는 장기 실행 트랜잭션을 식별할 수 있도록 새 CloudWatch 지표 `TransactionAgeMaximum`를 추가했습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora에 대한 Amazon CloudWatch 지표를 참조하세요.
- 메모리 부족(OOM) 방지를 위한 세 가지 새로운 CloudWatch 지표가 추가되었습니다.
  - `AuroraMillisecondsSpentInOomRecovery`
  - `AuroraNumOomRecoverySuccessful`
  - `AuroraNumOomRecoveryTriggered`

  자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora에 대한 Amazon CloudWatch 지표를 참조하세요.
- 메모리 부족(OOM) 회피에 대한 세 가지 CloudWatch 지표를 합계 실행에서 증분 카운터로 변경했습니다.
  - `AuroraMemoryNumDeclinedSqlTotal`
  - `AuroraMemoryNumKillConnTotal`
  - `AuroraMemoryNumKillQueryTotal`

  자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora에 대한 Amazon CloudWatch 지표를 참조하세요.
- 내부 임시 테이블에서 사용하는 메모리 양을 표시하기 위해 `aurora_temptable_ram_allocation` 및 `aurora_temptable_max_ram_allocation`이라는 두 가지 전역 상태 변수를 추가했습니다. 이러한 전역 상태 변수는 관찰성을 높이고 내부 임시 테이블 메모리 사용과 관련된 문제를 진단하는 데 도움이 됩니다.

  자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 전역 상태 변수를 참조하세요.

- 서버가 메모리에서 트레이스를 제거하기 전에 오류 로그에 옵티마이저 트레이스를 인쇄하는 새 시스템 변수 `aurora_optimizer_trace_print_before_purge`를 도입했습니다. 시스템 변수 optimizer_trace_offset 및 optimizer_trace_limit에 의해 설정된 임곗값을 기반으로 제거를 트리거할 수 있습니다.
- 다음 DB 인스턴스 클래스에 대한 지원을 추가했습니다.
  - db.r7i
  - db.r8g

  자세한 내용은 Amazon Aurora 사용 설명서의 DB 인스턴스 클래스에 지원되는 DB 엔진을 참조하세요.

개선 사항

보안 문제 및 CVE 해결:

- 바이너리 로그(binlog) 복제를 위한 새 사용자, `rdsrepladmin_priv_checks_user`를 도입했습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 바이너리 로그 복제를 위한 권한 있는 사용자 확인을 참조하세요.
- Aurora MySQL 저장 프로시저에 대한 입력 파라미터가 잘못 처리될 수 있는 문제를 수정했습니다.

이 릴리스에는 MySQL 8.0.39까지의 모든 커뮤니티 CVE 수정 사항이 포함되어 있습니다. 다음 CVE 수정 사항이 포함되어 있습니다.

- CVE-2023-44487
- CVE-2024-0853
- CVE-2024-20996
- CVE-2024-21000
- CVE-2024-21013
- CVE-2024-21125
- CVE-2024-21127
- CVE-2024-21129
- CVE-2024-21130
- CVE-2024-21134
- CVE-2024-21135

- CVE-2024-21137
- CVE-2024-21142
- CVE-2024-21157
- CVE-2024-21159
- CVE-2024-21160
- CVE-2024-21162
- CVE-2024-21163
- CVE-2024-21165
- CVE-2024-21166
- CVE-2024-21173
- CVE-2024-21177
- CVE-2024-21179
- CVE-2024-21185
- CVE-2024-25062
- CVE-2024-37371
- CVE-2025-21492

가용성 향상:

- 이미 열을 즉시 삭제한 테이블에서 데이터 조작 언어(DML) 명령을 실행한 후 서버가 예기치 않게 다시 시작되어 삭제된 열과 삭제되지 않은 열의 수가 1017보다 큰 문제를 수정했습니다.
- 데이터베이스 로그 파일이 올바르게 교체되지 않아 DB 인스턴스의 로컬 스토리지 공간 사용량이 증가할 수 있는 문제를 수정했습니다.
- aurora_oom_response가 활성화된 경우 메모리 부족 조건에서 DB 인스턴스의 연결 종료가 중지되어 메모리 부족 재부팅이 발생하여 피할 수 있는 문제를 수정했습니다.
- 로그 애플리케이션에 사용되는 메모리를 비울 때 리더 DB 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 전체 텍스트 검색(FTS) 인덱스에 대한 내부 지표를 계산할 때 데이터베이스가 다시 시작될 수 있는 문제를 수정했습니다.
- 인덱스 조건 푸시다운과 함께 하위 쿼리에서 `LIMIT` 절을 사용할 때 때때로 잘못된 쿼리 결과를 생성할 수 있는 커뮤니티 문제를 수정했습니다.

- 메모리 액세스 위반으로 인해 스레드가 더 이상 소유하지 않은 뮤텍스 객체가 해제되는 문제를 수정했습니다.
- 릴레이 로그 복구 중에 많은 수의 릴레이 로그 파일을 처리할 때 이진 로그(binlog) 복제본이 다시 시작될 수 있는 문제를 수정했습니다.
- 쓰기 전달이 활성화된 경우 Aurora 리더 DB 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 여러 `UNION` 절이 포함된 쿼리가 대용량 메모리를 할당하여 DB 인스턴스를 다시 시작하는 문제를 수정했습니다.
- 리더 DB 인스턴스에서 병렬 쿼리를 실행할 때 라이터 DB 인스턴스가 다시 시작되는 문제를 수정했습니다.
- `FLUSH RELAY LOGS` 명령을 실행할 때 복제본의 I/O 스레드에 교착 상태가 발생하여 복제본에서 binlog 복제가 중지되는 문제를 수정했습니다.
- 향상된 binlog가 활성화된 DB 클러스터에서 대규모 GTID 세트를 처리할 때 DB 인스턴스가 다시 시작되는 문제를 수정했습니다.
- 인 메모리 릴레이 로그 캐시가 활성화된 경우 binlog 복제본이 다시 시작될 수 있는 문제를 수정했습니다. 단일 스레드 바이너리 로그 복제 또는 GTID 자동 위치 지정이 활성화된 다중 스레드 복제를 사용하는 경우 Aurora MySQL 관리형 binlog 복제본에서 인 메모리 릴레이 로그 캐시가 활성화됩니다.
- 데이터베이스 엔진 시작 중에 데이터 제어 언어(DCL) 문을 적용할 때 binlog 복제본 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 드문 경우지만 라이터 DB 인스턴스로 업데이트되는 테이블에서 `SELECT` 쿼리를 실행할 때 교착 상태로 인해 리더 DB 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- Aurora Global Database 리더 인스턴스가 활성 쓰기 전달 세션으로 다시 시작될 수 있는 문제를 수정했습니다.
- 라이터 DB 인스턴스에서 드문 트랜잭션 커밋 주문이 발생하는 경우 Aurora 읽기 전용 복제본이 다시 시작될 수 있는 문제를 수정했습니다.
- 향상된 binlog가 활성화된 DB 인스턴스에서 실행되는 동안 예약된 이벤트가 취소될 때 데이터베이스가 다시 시작될 수 있는 문제를 수정했습니다.

일반적인 개선 사항

- 제로 가동 중지 패치(ZDP) 또는 제로 가동 중지 재시작(ZDR) 중에 클라이언트 연결이 멈출 수 있는 문제를 수정했습니다.
- 드문 경우지만 열린 테이블 오류를 처리하는 동안 발생할 수 있는 메모리 관리 문제로 인해 데이터베이스 인스턴스가 다시 시작되는 문제를 수정했습니다.

- AUTO_INCREMENT 프라이머리 키 열과 고유 키 열이 있는 테이블에서 동시 INSERT 문을 수행할 때와 INSERT 문에 서로 다른 행에 고유한 키 위반이 있을 때 SQL 문에 예기치 않은 프라이머리 키 위반 오류 또는 경고가 발생할 수 있는 문제를 수정했습니다.
- 성능 스키마 계측 memory/sql/sp_head::main_mem_root는 이제 제어된 계측입니다. 따라서 저장된 프로그램의 구문 분석 및 표현에 할당된 메모리가 이제 연결당 메모리 제한에 포함됩니다.
- 업데이트 중에 공간 인덱스를 통해 행을 읽을 수 없게 되는 문제를 수정했습니다.
- 사용자가 로컬 쓰기 전달을 비활성화하지 못하는 문제를 수정했습니다.
- 쓰기 전달을 사용하는 동안 SELECT COUNT 쿼리가 잘못된 결과를 반환할 수 있는 문제를 수정했습니다.
- ZDP가 쿼리의 힌트로 설정된 세션 변수를 잘못 복원할 때 잘못된 쿼리 결과가 발생할 수 있는 문제를 수정했습니다.
- 업그레이드 시나리오에서 innodb_max_undo_log_size 임곗값보다 큰 실행 취소 테이블스페이스의 자동 잘림 문제를 수정했습니다.
- innodb_flush_log_at_trx_commit가 0으로 설정된 경우 커밋 지연 시간과 커밋 처리량이 측정되지 않는 문제를 수정했습니다.
- Aurora 복제본 지연이 잘못 보고될 수 있는 문제를 수정했습니다.
- Aurora Global Database를 사용할 때 threads_running 변수에 잘못된 값을 제공한 문제를 수정했습니다.
- 외래 키가 있는 테이블에 복제 변경 사항을 적용할 때 병렬 보조 인덱스 최적화가 활성화된 Aurora MySQL binlog 복제본이 다시 시작되는 문제를 수정했습니다.
- 스케일 업 또는 스케일 다운 이벤트 중에 잠금 해시 테이블의 장기 크기 조정으로 인한 재시작 문제를 수정했습니다.
- 향상된 binlog가 이전에 활성화되었을 때 이진 로깅을 비활성화한 후 DB 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 테이블 수가 많을 때 논리적 데이터 사전 복구 중에 메모리 사용량을 줄이기 위한 최적화를 도입했습니다.
- 바이너리 로그 복제가 이미 구성된 경우 binlog 복제본에서 mysql.rds_set_external_source 저장 프로시저를 실행하는 동안 사용자에게 ERROR 1377 (HY000): Fatal error during log purge 오류가 발생할 수 있는 문제를 수정했습니다.
- ZDR 이후 사용자 역할 권한의 지속성을 방해하는 결함을 수정했습니다.
- 뷰 정의자의 기본 역할과 관련된 메모리 문제를 수정했습니다.
- SHOW VOLUME STATUS가 실행될 때 DB 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.

- 스케일 업 또는 스케일 다운 이벤트 중 버퍼 풀 크기 조정 연장으로 인한 재시작 문제를 수정했습니다.
- 스케일 다운 이벤트 중에 버퍼 크기 조정으로 인해 해제된 페이지에 액세스하는 논리적 미리 읽기(LRA)로 인한 재시작 문제를 수정했습니다.
- 쿼리가 구체화를 전략으로 사용하는 세미조인 변환과 `LEFT OUTER JOIN`을 함께 사용할 때 잘못된 결과를 반환하는 `SELECT` 쿼리를 해결하는 문제를 수정했습니다.
- 쓰기 전달을 비활성화하는 프로세스를 완료하지 못할 수 있는 문제를 해결했습니다.
- `ActiveTransactions` 및 `BlockedTransactions` CloudWatch 지표가 예상보다 낮은 값을 보고하는 문제를 수정했습니다.
- 복제본이 상위 테이블과 하위 테이블 모두에서 명시적으로 삭제하는 멀티테이블 `DELETE` 문을 처리할 때 binlog 복제가 중단되는 문제를 수정했습니다.
- enum 유형을 반환하는 사용자 정의 함수로 트리거를 처리할 때 DB 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- `row`로 설정된 `binlog_format`을 사용하여 복제할 때 `DMLLatency` CloudWatch 지표에 이진 로그 복제본 인스턴스의 잘못된 값이 표시되는 문제를 수정했습니다.
- 느린 `SELECT` 쿼리가 선행되지 않는 한 MySQL Event Scheduler에서 실행하는 느린 `INSERT`, `DELETE` 및 `UPDATE` 쿼리가 느린 쿼리 로그에 기록되지 않는 문제를 수정했습니다.
- 드문 경우지만 Aurora 읽기 전용 복제본을 간헐적으로 사용할 수 없거나 테이블 정의 불일치가 발생하여 복제본에서 `Table does not exist` 오류가 발생하는 문제를 수정했습니다. 이는 라이터 DB 인스턴스의 복제본 및 데이터 정의 언어(DDL) 작업에 대한 동시 읽기 쿼리 때문입니다.
- 향상된 binlog가 활성화되거나 이전에 활성화된 DB 클러스터에서 `SHOW BINARY LOGS` 명령을 실행하는 데 시간이 더 오래 걸리는 문제를 수정했습니다. 이 문제로 인해 여러 `SHOW BINARY LOGS` 명령이 동시에 실행되는 경우 커밋 지연 시간이 증가할 수도 있습니다.

업그레이드 및 마이그레이션:

- 데이터베이스 객체(예: 테이블, 트리거 및 루틴)가 많은 DB 클러스터의 메이저 버전 업그레이드 성능을 Aurora MySQL 버전 2에서 버전 3으로 개선했습니다.

  더 큰 DB 인스턴스 클래스의 경우 데이터베이스 업그레이드 프로세스는 여러 스레드를 사용하여 기존의 MySQL 객체 메타데이터를 새로운 원자성 MySQL 8.0 데이터 사전으로 병렬로 업그레이드합니다.
- 대상 Aurora MySQL DB 엔진 버전이 3.04.0 이상일 때 업그레이드 또는 마이그레이션이 실패하는 문제를 수정했습니다. 이는 `lower_case_table_names` DB 클러스터 파라미터가 `1`로 설정되어 있고 MySQL 데이터베이스 데이터 정렬이 소문자 테이블 이름과 호환되지 않는 경우에 발생합니다.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 아래 사항 외에도 8.0.39까지의 모든 커뮤니티 버그 수정 사항이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 3.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그를 참조하세요.

- `JOIN` 및 `UNION` 작업이 모두 있는 특정 쿼리의 결과 세트에서 `NULL` 값이 잘못 생략되는 문제를 수정했습니다. (커뮤니티 버그 수정 #114301)

Aurora MySQL 데이터베이스 엔진 업데이트 2024-07-23(버전 3.07.1)(사용되지 않음)

버전: 3.07.1

Aurora MySQL 3.07.1이 정식 버전입니다. Aurora MySQL 3.07 버전은 MySQL 8.0.36과 호환됩니다. 발생한 커뮤니티 변경 사항에 대한 자세한 내용은 MySQL 8.0 Release Notes를 참조하세요.

Aurora MySQL 버전 3의 새로운 기능에 대한 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환을 참조하세요. Aurora MySQL 버전 3과 Aurora MySQL 버전 2의 차이점에 대한 자세한 내용은 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요. Aurora MySQL 버전 3과 MySQL 8.0 커뮤니티 에디션의 비교 내용은 Amazon Aurora 사용 설명서에서 Aurora MySQL 버전 3과 MySQL 8.0 커뮤니티 에디션 비교를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 2.11.*, 2.12.*, 3.03.*, 3.04.*, 3.05.*, 3.06.* 및 3.07.*입니다.

현재 지원되는 Aurora MySQL 버전 2 클러스터에서 Aurora MySQL 버전 MySQL 3.07.1 클러스터로 Amazon RDS 블루/그린 배포를 사용하여 현재 위치 업그레이드를 수행하거나 스냅샷을 복원하거나 관리형 블루/그린 업그레이드를 시작할 수 있습니다.

Aurora MySQL 버전 3으로의 업그레이드 계획에 대한 자세한 내용은 Aurora MySQL 클러스터에 대한 메이저 버전 업그레이드 계획을 참조하세요. Aurora MySQL 업그레이드에 대한 일반 정보는 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터 업그레이드를 참조하세요.

문제 해결 정보는 Amazon Aurora 사용 설명서의 Aurora MySQL 현재 위치 업그레이드 문제 해결을 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

보안 문제 및 CVE 해결:

- 바이너리 로그(binlog) 복제를 위한 새 사용자, `rdsrepladmin_priv_checks_user`를 도입했습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 바이너리 로그 복제를 위한 권한 있는 사용자 확인을 참조하세요.

이 릴리스에는 MySQL 8.0.36까지의 모든 커뮤니티 CVE 수정 사항이 포함되어 있습니다.

가용성 향상:

- 로그 애플리케이션에 사용되는 메모리를 비울 때 리더 DB 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 전체 텍스트 검색(FTS) 인덱스에 대한 내부 지표를 계산할 때 데이터베이스가 다시 시작될 수 있는 문제를 수정했습니다.
- 대용량 트랜잭션을 커밋하는 동안 오류가 발생할 때 이진 로깅을 비활성화할 수 있는 문제를 수정했습니다.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 MySQL 8.0.36까지의 모든 커뮤니티 버그 수정 사항이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 3.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그 단원을 참조하십시오.

Aurora MySQL 데이터베이스 엔진 업데이트 2024-06-04(버전 3.07.0)(사용되지 않음)

버전: 3.07.0

Aurora MySQL 3.07.0이 정식 버전입니다. Aurora MySQL 3.07 버전은 MySQL 8.0.36과 호환됩니다. 발생한 커뮤니티 변경 사항에 대한 자세한 내용은 MySQL 8.0 Release Notes를 참조하세요.

Aurora MySQL 버전 3의 새로운 기능에 대한 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환을 참조하세요. Aurora MySQL 버전 3과 Aurora MySQL 버전 2의 차이점에 대한 자세한 내용은 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요. Aurora MySQL 버전 3과 MySQL 8.0 커뮤니티 에디션의 비교 내용은 Amazon Aurora 사용 설명서에서 Aurora MySQL 버전 3과 MySQL 8.0 커뮤니티 에디션 비교를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.03.*, 3.04.*, 3.05.*, 3.06.* 및 3.07.*입니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

보안 문제 및 CVE 해결:

- 완전 소유 AWS 구현인 FIPS 검증 암호화에 대한 지원을 활성화했습니다. 자세한 내용은 AWS 보안 블로그에서 AWS-LC 에서 이제 FIPS 140-3 인증을 받았습니다를 참조하세요.

이 릴리스에는 MySQL 8.0.36까지의 모든 커뮤니티 CVE 수정 사항이 포함되어 있습니다. 다음 CVE 수정 사항이 포함되어 있습니다.

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

가용성 향상:

- 라이터 DB 인스턴스에서 변경되거나 삭제되는 테이블을 읽을 때 리더 DB 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 전달된 쿼리를 실행하는 동안 쓰기 전달 세션이 닫힐 때 Aurora MySQL 라이터 DB 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 바이너리 로그가 활성화된 인스턴스에서 대규모 GTID 세트를 처리할 때 DB 인스턴스가 다시 시작되는 문제를 수정했습니다.
- InnoDB 분할 테이블에서 `INSERT` 쿼리를 처리할 때 인스턴스의 여유 메모리가 점진적으로 감소할 수 있는 문제를 해결했습니다.
- 드문 경우지만 리더 DB 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.

- SHOW STATUS 및 PURGE BINARY LOGS 문을 동시에 실행할 때 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다. PURGE BINARY LOGS는 사용자가 구성한 binlog 보존 기간을 준수하기 위해 실행되는 관리형 문입니다.
- 비가상 열이 MODIFY COLUMN 또는 CHANGE COLUMN 문으로 재정렬된 테이블에서 데이터 조작 언어(DML) 문을 실행한 후 서버가 예기치 않게 닫힐 수 있는 문제를 수정했습니다.
- 데이터베이스 인스턴스를 다시 시작하는 동안 추가 재시작이 발생할 수 있는 문제를 수정했습니다.
- 전달된 암시적 커밋 문에 오류가 발생할 때 쓰기 전달을 사용하는 리더 DB 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 드문 경우지만 외래 키 제약 조건이 있는 테이블에서 SELECT 쿼리를 수행할 때 리더 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 다중 TB Aurora 클러스터 볼륨을 사용하는 DB 인스턴스가 InnoDB 버퍼 풀 검증 실패로 인해 재시작 중에 가동 중지 시간이 증가할 수 있는 문제를 수정했습니다.
- 가상 열이 외래 키 제약 조건의 열 또는 참조된 테이블의 멤버로 포함되는 테이블에 계단식 UPDATE 또는 DELETE 외래 키 제약 조건이 정의될 때 데이터베이스가 다시 시작될 수 있는 문제를 수정했습니다.
- AUTO_INCREMENT 열과 관련된 많은 삽입 작업을 실행하는 동안 다시 시작이 발생하는 경우 시작 시 데이터베이스 복구가 중단될 수 있는 문제를 해결했습니다.
- 확장 중에 데이터베이스가 다시 시작될 수 있는 Aurora Serverless v2의 문제를 수정했습니다.

일반적인 개선 사항:

- 병렬 쿼리를 사용하는 프라이머리 키 범위 스캔 쿼리의 하위 세트에 대한 I/O 사용량을 줄이고 성능을 개선했습니다.
- Aurora MySQL 버전 3.06.0에 Amazon Bedrock 통합에 대한 지원이 추가되었습니다. 이 과정에서 새로운 예약 키워드(accept, aws_bedrock_invoke_model, aws_sagemaker_invoke_endpoint, content_type 및 timeout_ms)가 추가되었습니다. Aurora MySQL 버전 3.07.0에서 이러한 키워드는 예약되지 않은 키워드로 변경되었으며 이는 따옴표 없이 식별자로 허용됩니다. MySQL에서 예약 및 예약되지 않은 키워드를 처리하는 방법에 대한 자세한 내용은 MySQL 설명서의 키워드 및 예약어를 참조하세요.
- Amazon Bedrock을 아직 사용할 수 없는의 Aurora MySQL DB 클러스터에서 AWS 리전 Amazon Bedrock 서비스를 호출할 때 클라이언트에 오류 메시지를 명확하게 반환하지 않는 문제를 수정했습니다.
- Aurora 병렬 쿼리를 사용하여 BLOB 열을 쿼리할 때 과도한 메모리 소비를 일으킬 수 있는 문제를 해결했습니다.

- 세션 수준에서 설정할 `connection_memory_limit` 및 `connection_memory_chunk_size` 파라미터에 대한 지원이 MySQL Community Edition과 동일하게 동작하도록 추가되었습니다. `connection_memory_limit`는 단일 사용자 연결에서 사용할 수 있는 최대 메모리 양을 설정하는 데 사용됩니다. `connection_memory_chunk_size` 파라미터를 사용하여 글로벌 메모리 사용량 카운터 업데이트에 대한 청킹 크기를 설정할 수 있습니다.
- 사용자가 쿼리를 중단하거나 `performance_schema` 쿼리에 대한 세션 제한 시간을 설정할 수 없는 문제를 수정했습니다.
- 복제 인스턴스가 호스트 교체 중일 때 사용자 지정 SSL 인증서(mysql.rds_import_binlog_ssl_material)를 사용하도록 구성된 바이너리 로그(binlog) 복제가 실패할 수 있는 문제를 수정했습니다.
- 모든 테이블에서 전체 텍스트 검색 시스템의 메모리 사용량을 추적하는 `Aurora_fts_cache_memory_used` 전역 상태 변수를 추가했습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 전역 상태 변수를 참조하세요.
- 향상된 Binlog 및 제로 ETL 통합이 활성화된 상태에서 Amazon Aurora MySQL DB 클러스터가 바이너리 로그 복제본으로 구성된 경우 제로 ETL 대상으로 구성된 Amazon Redshift 클러스터에서 IntegrationLag가 일시적으로 증가할 수 있는 문제를 수정했습니다.
- 다운로드 또는 교체 시 로그 파일에 액세스할 수 없고 경우에 따라 CPU 사용량이 증가할 수 있는 감사 로그 파일 관리 관련 문제를 해결했습니다.
- 스냅샷 복원, 특정 시점 복구 수행, 데이터베이스에 많은 수의 테이블이 있는 DB 클러스터 복제의 완료 시간을 줄이기 위해 `AUTO_INCREMENT` 키 복구를 최적화했습니다.
- 성능 스키마 대기 이벤트 요약 테이블에 wait/io/redo_log_flush 이벤트가 표시되지 않는 문제를 수정했습니다.
- 스냅샷 복원, 역추적 또는 데이터베이스 복제 작업 후 내림차순 인덱스를 사용하는 `AUTO_INCREMENT` 열에 대해 중복 키 오류가 발생할 수 있는 문제를 해결했습니다.
- 쓰기 전달을 사용하는 리더 DB 인스턴스가 타임스탬프 값을 포함하고 `time_zone` 데이터베이스 파라미터가 UTC로 설정된 데이터 조작 언어(DML) 문을 실행할 때 라이터 DB 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 테이블에 전체 텍스트 검색(FTS) 인덱스가 하나 이상 있고 Aurora 라이터 DB 인스턴스에서 `TRUNCATE` 문이 실행 중인 경우 테이블이 존재하지 않습니다 오류와 함께 Aurora 리더 인스턴스의 `SELECT` 쿼리가 실패할 수 있는 문제가 해결되었습니다.
- 드문 경우지만 제로 가동 중지 시간 패치(ZDP)가 실패하는 문제를 수정했습니다.
- 병렬 쿼리와 함께 해시 조인 알고리즘을 사용하여 `LEFT JOIN` 또는 `RIGHT JOIN` 작업과 관련된 쿼리를 실행할 때 불완전한 결과 세트를 일으킬 수 있는 문제를 수정했습니다.

업그레이드 및 마이그레이션:

- 테이블 스키마에 사용자 정의 FTS_DOC_ID 열이 있는 경우 Aurora MySQL 버전 2에서 Aurora MySQL 버전 3으로 업그레이드 실패를 일으킬 수 있는 문제를 수정했습니다.
- InnoDB 테이블스페이스를 처리하는 동안 동기화 오류로 인해 Aurora MySQL 버전 2에서 Aurora MySQL 버전 3으로의 업그레이드가 실패할 수 있는 문제를 해결했습니다.
- Aurora MySQL 버전 2의 InnoDB 시스템 테이블에 이미 삭제된 테이블스페이스에 대한 분리된 항목이 존재하여 Aurora MySQL 버전 3으로의 메이저 버전 업그레이드가 실패할 수 있는 문제를 수정했습니다.
- Amazon RDS 블루/그린 배포 전환 후 SERVER_ID 값이 업데이트되지 않는 문제를 수정했습니다. 이로 인해 Amazon Web Services(AWS) JDBC 드라이버와 같은 스마트 드라이버가 블루/그린 전환 후 DB 클러스터 토폴로지를 검색할 수 없는 문제가 발생했습니다. 이 수정을 통해 Aurora MySQL 버전 3.07 이상에서 실행되는 RDS 블루/그린 배포의 일부로 이름이 변경된 Aurora DB 클러스터는 전환의 일부로 SERVER_ID 값이 업데이트됩니다. 이전 버전의 경우 블루 및 그린 클러스터의 DB 인스턴스를 재부팅하여 SERVER_ID 값을 업데이트할 수 있습니다.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 아래 사항 외에도 8.0.36까지의 모든 커뮤니티 버그 수정 사항이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 3.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그를 참조하세요.

- 캐시 라인 값을 잘못 계산하여 Graviton 기반 인스턴스에서 데이터베이스를 다시 시작하는 동안 오류가 발생하는 문제를 수정했습니다. (커뮤니티 버그 수정 #35479763)
- 저장된 루틴 내의 일부 하위 쿼리 인스턴스가 올바르게 처리되지 않는 문제를 수정했습니다. (커뮤니티 버그 수정 #35377192)
- 백그라운드 TLS 인증서 교체로 인해 CPU 사용이 높아질 수 있는 문제를 해결했습니다(커뮤니티 버그 수정 #34284186).
- InnoDB가 3.05 이하의 Aurora MySQL 버전에서 MySQL 시스템 스키마의 테이블에 INSTANT 열을 추가하도록 허용하여 Aurora MySQL 버전 3.05.0으로 업그레이드한 후 서버가 예기치 않게 종료(데이터베이스 인스턴스 다시 시작)될 수 있는 문제를 수정했습니다. (커뮤니티 버그 수정 #35625510)

Aurora MySQL 데이터베이스 엔진 업데이트 2024-06-26(버전 3.06.1)(사용되지 않음)

버전: 3.06.1

Aurora MySQL 3.06.1이 정식 버전입니다. Aurora MySQL 3.06 버전은 MySQL 8.0.34와 호환됩니다. 발생한 커뮤니티 변경 사항에 대한 자세한 내용은 MySQL 8.0 Release Notes를 참조하세요.

Aurora MySQL 버전 3의 새로운 기능에 대한 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환을 참조하세요. Aurora MySQL 버전 3과 Aurora MySQL 버전 2의 차이점에 대한 자세한 내용은 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요. Aurora MySQL 버전 3과 MySQL 8.0 커뮤니티 에디션의 비교 내용은 Amazon Aurora 사용 설명서에서 Aurora MySQL 버전 3과 MySQL 8.0 커뮤니티 에디션 비교를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.03.*, 3.04.*, 3.05.*, 3.06.* 및 3.07.*입니다.

현재 지원되는 Aurora MySQL 버전 2 클러스터에서 Aurora MySQL 버전 MySQL 3.06.1 클러스터로 Amazon RDS 블루/그린 배포를 사용하여 현재 위치 업그레이드를 수행하거나 스냅샷을 복원하거나 관리형 블루/그린 업그레이드를 시작할 수 있습니다.

Aurora MySQL 버전 3으로의 업그레이드 계획에 대한 자세한 내용은 Aurora MySQL DB 클러스터에 대한 메이저 버전 업그레이드 계획을 참조하세요. Aurora MySQL 업그레이드에 대한 일반 정보는 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터 업그레이드를 참조하세요.

문제 해결 정보는 Amazon Aurora 사용 설명서의 Aurora MySQL 현재 위치 업그레이드 문제 해결을 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

보안 문제 및 CVE 해결:

이 릴리스에는 MySQL 8.0.34까지의 모든 커뮤니티 CVE 수정 사항이 포함되어 있습니다. 다음 CVE 수정 사항이 포함되어 있습니다.

- CVE-2023-44487
- CVE-2024-0853

가용성 향상:

- 병렬 쿼리를 실행할 때 Aurora MySQL DB 인스턴스가 다시 시작되는 문제를 수정했습니다.
- 라이터 DB 인스턴스에서 변경되거나 삭제되는 테이블을 읽을 때 리더 DB 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 메모리 액세스 위반으로 인해 스레드가 더 이상 소유하지 않은 뮤텍스 객체가 해제되는 문제를 수정했습니다.
- 전달된 쿼리를 실행하는 동안 쓰기 전달 세션이 닫힐 때 Aurora MySQL 라이터 DB 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 바이너리 로그가 활성화된 인스턴스에서 대규모 GTID 세트를 처리할 때 DB 인스턴스가 다시 시작되는 문제를 수정했습니다.
- 드문 경우지만 외래 키 제약 조건이 있는 테이블에서 `SELECT` 쿼리를 수행할 때 리더 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 데이터베이스 복구 중에 InnoDB 데이터 사전을 복구하려고 할 때 DB 인스턴스가 다시 시작되는 문제를 수정했습니다.
- 확장 중에 데이터베이스가 다시 시작될 수 있는 Aurora Serverless v2의 문제를 수정했습니다.

일반적인 개선 사항:

- 해제된 후 메모리를 사용할 수 있는 지표 게시 코드 문제를 수정했습니다.
- 존재하지 않는 실행 취소 테이블스페이스 객체로 인해 DB 엔진이 반복적으로 다시 시작되는 문제를 수정했습니다.
- 업그레이드 시나리오에서 innodb_max_undo_log_size 임곗값보다 큰 실행 취소 테이블스페이스의 자동 잘림 문제를 수정했습니다.
- Aurora Global Database를 사용할 때 `threads_running` 상태 변수에 잘못된 값을 제공한 문제를 수정했습니다.
- 외래 키가 있는 테이블에 복제 변경 사항을 적용할 때 병렬 보조 인덱스 최적화가 활성화된 Aurora MySQL 이진 로그(binlog) 읽기 전용 복제본이 다시 시작되는 문제를 수정했습니다.
- Aurora MySQL 버전 3.06.0에 Amazon Bedrock 통합에 대한 지원이 추가되었습니다. 이 과정에서 새로운 예약 키워드(`accept`, `aws_bedrock_invoke_model`, `aws_sagemaker_invoke_endpoint`, `content_type` 및 `timeout_ms`)가 추가되었습니다. Aurora MySQL 버전 3.06.1에서 이러한 키워드는 예약되지 않은 키워드로 변경되었으며 이는 따옴표 없이 식별자로 허용됩니다. MySQL에서 예약 및 예약되지 않은 키워드를 처리하는 방법에 대한 자세한 내용은 MySQL 설명서의 키워드 및 예약어를 참조하세요.

- Amazon Bedrock을 아직 사용할 수 없는의 Aurora MySQL DB 클러스터에서 AWS 리전 Amazon Bedrock 서비스를 호출할 때 클라이언트에 오류 메시지를 명확하게 반환하지 않는 문제를 수정했습니다.
- 병렬 읽기를 사용할 때 `rw_lock`의 잠금 소유자 정보가 부정확하여 DB 인스턴스가 다시 시작되는 문제를 수정했습니다.
- `SHOW VOLUME STATUS`가 실행될 때 DB 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- `SELECT ... INTO OUTFILE ...` 쿼리를 실행할 때 시간이 지남에 따라 사용 가능한 메모리가 감소하는 메모리 관리 문제를 수정했습니다.
- 세션 수준에서 설정할 `connection_memory_limit` 및 `connection_memory_chunk_size` 파라미터에 대한 지원이 MySQL Community Edition의 해당 기능과 유사하게 동작하도록 추가되었습니다. `connection_memory_limit` 파라미터는 단일 사용자 연결에서 사용할 수 있는 최대 메모리 양을 설정합니다. `connection_memory_chunk_size` 파라미터는 글로벌 메모리 사용량 카운터에 대한 업데이트를 위한 청킹 크기를 설정합니다.
- DB 인스턴스의 로컬 스토리지가 전체 용량에 도달하면 DB 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- db.t4g.medium 및 db.t4g.large DB 인스턴스에 대해 Performance Insights 자동 관리가 켜져 있을 때 성능 스키마가 활성화되지 않는 문제를 수정했습니다.
- 쓰기 전달을 사용하는 리더 DB 인스턴스가 타임스탬프 값을 포함하고 `time_zone` 데이터베이스 파라미터가 UTC로 설정된 데이터 조작 언어(DML) 문을 실행할 때 라이터 DB 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 고객이 구성한 `wait_timeout` 또는 `interactive_timeout`의 최소 값에 도달하면 DB 인스턴스가 클라이언트 연결을 닫지 못하게 하는 제로 가동 중지 시간 패치(ZDP) 중 문제를 수정했습니다.

업그레이드 및 마이그레이션:

- 대상 Aurora MySQL DB 엔진 버전이 3.04.0 이상일 때 업그레이드 또는 마이그레이션이 실패하는 문제를 수정했습니다. 이는 `lower_case_table_names` DB 클러스터 파라미터가 1로 설정되어 있고 MySQL 데이터베이스 데이터 정렬이 소문자 테이블 이름과 호환되지 않는 경우에 발생합니다.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 MySQL 8.0.34까지의 모든 커뮤니티 버그 수정 사항이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 3.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그를 참조하세요.

Aurora MySQL 데이터베이스 엔진 업데이트 2024-03-07(버전 3.06.0)(사용되지 않음)

버전: 3.06.0

Aurora MySQL 3.06.0이 정식 버전입니다. Aurora MySQL 3.06 버전은 MySQL 8.0.34와 호환됩니다. 발생한 커뮤니티 변경 사항에 대한 자세한 내용은 MySQL 8.0 Release Notes를 참조하세요.

Aurora MySQL 버전 3의 새로운 기능에 대한 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환을 참조하세요. Aurora MySQL 버전 3과 Aurora MySQL 버전 2의 차이점에 대한 자세한 내용은 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요. Aurora MySQL 버전 3과 MySQL 8.0 커뮤니티 에디션의 비교 내용은 Amazon Aurora 사용 설명서에서 Aurora MySQL 버전 3과 MySQL 8.0 커뮤니티 에디션 비교를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.03.*, 3.04.*, 3.05.* 및 3.06.* 입니다.

현재 지원되는 Aurora MySQL 버전 2 클러스터에서 Aurora MySQL 버전 MySQL 3.06.0 클러스터로 Amazon RDS 블루/그린 배포를 사용하여 현재 위치 업그레이드를 수행하거나 스냅샷을 복원하거나 관리형 블루/그린 업그레이드를 시작할 수 있습니다.

Aurora MySQL 버전 3으로의 업그레이드 계획에 대한 자세한 내용은 Aurora MySQL DB 클러스터에 대한 메이저 버전 업그레이드 계획을 참조하세요. Aurora MySQL 업그레이드에 대한 일반 정보는 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터 업그레이드를 참조하세요.

문제 해결 정보는 Amazon Aurora 사용 설명서의 Aurora MySQL 현재 위치 업그레이드 문제 해결을 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

새로운 기능

- Aurora MySQL 버전 3.06.0은 Amazon Bedrock 통합을 지원하고 새로운 예약 키워드 `accept`, `aws_bedrock_invoke_model`, `aws_sagemaker_invoke_endpoint`, `content_type` 및 `timeout_ms`를 도입합니다. 버전 3.06.0으로 업그레이드하기 전에 새 예약 키워드의 사용에 대한 객체 정의를 확인합니다. 새 예약 키워드와의 충돌을 완화하려면 객체 정의에 사용된 예약 키워드를

인용합니다. Amazon Bedrock 통합 및 예약된 키워드 처리에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Bedrock이란 무엇인가요?를 참조하세요. 자세한 내용은 MySQL 설명서의 키워드 및 예약어, INFORMATION_SCHEMA KEYWORDS 테이블 및 스키마 객체 이름을 참조하세요.

- 보조 인덱스가 두 개 이상인 대규모 테이블의 트랜잭션을 복제할 때 이진 로그 복제본의 성능이 개선되었습니다. 이 기능은 binlog 복제본에서 보조 인덱스 변경 사항을 병렬로 적용하는 스레드 풀을 도입합니다. 이 기능은 보조 인덱스 변경 사항을 적용하는 데 사용할 수 있는 총 병렬 스레드 수를 제어하는 `aurora_binlog_replication_sec_index_parallel_workers` DB 클러스터 파라미터에 의해 제어됩니다. 자세한 내용은 Amazon Aurora 사용 설명서의 바이너리 로그 복제 최적화를 참조하세요.
- Aurora MySQL 클러스터의 데이터베이스 인스턴스에서 글로벌 시스템 변수 `read_only`의 값을 변경할 수 있는 새로운 저장 프로시저 `mysql.rds_set_read_only`가 추가되었습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 복제를 참조하세요.
- `SOURCE_SSL`에 값을 지정하여 이진 로그 복제본에서 암호화를 설정할 수 있는 새로운 저장 프로시저 `mysql.rds_set_binlog_source_ssl`이 추가되었습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 복제를 참조하세요.
- Amazon Aurora Machine Learning은 Aurora MySQL 데이터베이스와 AWS 기계 학습(ML) 서비스 간의 최적화된 통합입니다. 이제 Amazon Bedrock이 지원되므로 SQL을 사용하여 Aurora MySQL DB 클러스터에서 직접 Amazon Bedrock의 기계 학습 모델을 호출할 수 있습니다. Aurora MySQL DB 클러스터에서 Amazon Bedrock을 사용하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL에서 Amazon Aurora 기계 학습 사용을 참조하세요.
- Aurora MySQL 버전 3.06에서는 테이블스페이스 자르기 자동 실행 취소에 대한 지원이 추가됩니다. 이 최적화를 통해 실행 취소 로그가 제거된 후 실행 취소 테이블스페이스에서 미사용 공간을 회수할 수 있습니다.

개선 사항

보안 문제 및 CVE 해결:

이 릴리스에는 다음 CVE 수정 사항이 포함되어 있습니다.

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

가용성 향상:

- 라이터 DB 인스턴스에 워크로드가 많을 때 읽기 전용 복제본 DB 인스턴스를 성공적으로 시작할 수 없는 문제를 수정했습니다.
- Aurora 스토리지와의 통신 결함으로 인해 Aurora MySQL 라이터 DB 인스턴스가 장애 조치될 수 있는 문제를 수정했습니다. 결함은 Aurora 스토리지 인스턴스의 소프트웨어 업데이트 후 DB 인스턴스와 기본 스토리지 간의 통신이 분석된 결과로 발생합니다.
- InnoDB 분할 테이블에서 `INSERT` 쿼리를 처리할 때 인스턴스의 여유 메모리가 점진적으로 감소할 수 있는 문제를 해결했습니다.
- 쿼리를 실행하는 동안 해시 조인을 사용할 때 사용 가능한 메모리가 감소하여 Aurora MySQL DB 인스턴스가 다시 시작되거나 장애 조치될 수 있는 문제를 수정했습니다.
- SHOW STATUS 및 PURGE BINARY LOGS 문을 동시에 실행할 때 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다. `PURGE BINARY LOGS`는 사용자가 구성한 binlog 보존 기간을 준수하기 위해 실행되는 관리형 문입니다.
- 비가상 열이 `MODIFY COLUMN` 또는 `CHANGE COLUMN` 문으로 재정렬된 테이블에서 데이터 조작 언어(DML) 문을 실행한 후 서버가 예기치 않게 닫힐 수 있는 문제를 수정했습니다.
- 데이터베이스 인스턴스를 다시 시작하는 동안 추가 재시작이 발생할 수 있는 문제를 수정했습니다.
- 가상 열이 외래 키 제약 조건의 열 또는 참조된 테이블의 멤버로 포함되는 테이블에 계단식 `UPDATE` 또는 `DELETE` 외래 키 제약 조건이 정의될 때 데이터베이스가 다시 시작될 수 있는 문제를 수정했습니다.
- Aurora MySQL 2.10에서는 읽기 가용성으로 Aurora DB 클러스터 재부팅에 대한 지원이 추가되었습니다. 이 기능을 사용하면 라이터 DB 인스턴스가 재부팅되는 동안 리더 DB 인스턴스를 온라인 상태로 유지할 수 있습니다. 이제 Aurora MySQL 글로벌 데이터베이스의 보조 AWS 리전 에서 이 기능이 지원되므로 기본 클러스터에서 라이터 인스턴스를 다시 시작하는 동안에도 읽기 요청을 계속 처리할 수 있습니다. 이전에는 라이터 인스턴스가 다시 시작될 때 Aurora MySQL 클러스터의 모든 리더 인스턴스도 다시 시작되었습니다. 이번 릴리스에서는 보조 클러스터 리더 인스턴스가 라이터 인스턴스 재시작 중에도 계속해서 읽기 요청을 처리하여 클러스터 내 읽기 가용성을 향상시킵니다. 자세한 내용은 읽기 가용성을 포함하여 Aurora 클러스터 재부팅을 참조하세요.
- `AUTO_INCREMENT` 열과 관련된 많은 삽입 작업을 실행하는 동안 다시 시작이 발생하는 경우 시작 시 데이터베이스 복구가 중단될 수 있는 문제를 해결했습니다.

일반적인 개선 사항:

- Aurora 클러스터 볼륨에서 데이터를 읽는 동안 일시적인 네트워크 문제로 인해 병렬 쿼리가 실패할 수 있는 문제를 해결했습니다.

- 사용자가 쿼리를 중단하거나 `performance_schema` 쿼리에 대한 세션 제한 시간을 설정할 수 없는 문제를 수정했습니다.
- 복제 인스턴스가 호스트 교체 중일 때 사용자 지정 SSL 인증서 (mysql.rds_import_binlog_ssl_material)를 사용하도록 구성된 바이너리 로그(binlog) 복제가 실패할 수 있는 문제를 수정했습니다.
- 메모리가 4GiB 이하인 작은 DB 인스턴스는 이제 DB 인스턴스에 메모리가 부족할 때 메모리 사용량이 가장 많은 연결을 닫습니다. 버퍼 풀을 조정하여 크기를 줄일 수도 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL 메모리 부족 문제를 참조하세요.
- 메모리가 4GiB를 초과하는 모든 DB 인스턴스 클래스에서 `aurora_oom_response`에 대한 기본 응답을 비어 있음에서 `print`로 변경했습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL 메모리 부족 문제를 참조하세요.
- 다운로드 또는 교체 시 로그 파일에 액세스할 수 없고 경우에 따라 CPU 사용량이 증가할 수 있는 감사 로그 파일 관리 관련 문제를 해결했습니다.
- 스냅샷 복원, 특정 시점 복구 수행, 데이터베이스에 많은 수의 테이블이 있는 DB 클러스터 복제의 완료 시간을 줄이기 위해 `AUTO_INCREMENT` 키 복구를 최적화했습니다.
- 성능 스키마 대기 이벤트 요약 테이블에 wait/io/redo_log_flush 이벤트가 표시되지 않는 문제를 수정했습니다.
- 잠금 관리자의 메모리 사용량을 추적하기 위해 `Aurora_lockmgr_memory_used` 및 `Aurora_lockmgr_buffer_pool_memory_used` 지표를 추가했습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 전역 상태 변수를 참조하세요.
- Aurora MySQL 2.11.* 이하 버전에서 업그레이드한 후 소규모 읽기 전용 복제본 인스턴스에서 복제 지연이 증가할 수 있는 문제를 해결했습니다.
- 스냅샷 복원, 역추적 또는 데이터베이스 복제 작업 후 내림차순 인덱스를 사용하는 `AUTO_INCREMENT` 열에 대해 중복 키 오류가 발생할 수 있는 문제를 해결했습니다.
- 테이블에 전체 텍스트 검색(FTS) 인덱스가 하나 이상 있고 Aurora 라이터 DB 인스턴스에서 `TRUNCATE` 문이 실행 중인 경우 테이블이 존재하지 않습니다 오류와 함께 Aurora 리더 인스턴스의 `SELECT` 쿼리가 실패할 수 있는 문제가 해결되었습니다.
- 병렬 쿼리와 함께 해시 조인 알고리즘을 사용하여 `LEFT JOIN` 또는 `RIGHT JOIN` 작업과 관련된 쿼리를 실행할 때 불완전한 결과 세트를 일으킬 수 있는 문제를 수정했습니다.

업그레이드 및 마이그레이션:

- 테이블 스키마에 사용자 정의 `FTS_DOC_ID` 열이 있는 경우 메이저 버전 업그레이드가 실패할 수 있는 문제를 수정했습니다.

- InnoDB 테이블스페이스를 처리하는 동안 동기화 오류로 인해 Aurora MySQL 버전 2에서 Aurora MySQL 버전 3으로의 업그레이드가 실패할 수 있는 문제를 해결했습니다.
- Aurora MySQL 버전 2의 InnoDB 시스템 테이블에 이미 삭제된 테이블스페이스에 대한 분리된 항목이 존재하여 Aurora MySQL 버전 3으로의 메이저 버전 업그레이드가 실패할 수 있는 문제를 수정했습니다.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 아래 사항 외에도 8.0.34까지의 모든 커뮤니티 버그 수정 사항이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 3.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그를 참조하세요.

- 캐시 라인 값을 잘못 계산하여 Graviton 기반 인스턴스에서 데이터베이스를 다시 시작하는 동안 오류가 발생하는 문제를 수정했습니다. (커뮤니티 버그 수정 #35479763)
- 저장된 루틴 내의 일부 하위 쿼리 인스턴스가 항상 올바르게 처리되지 않는 문제를 수정했습니다. (커뮤니티 버그 수정 #35377192)
- 백그라운드 TLS 인증서 교체로 인해 CPU 사용이 높아질 수 있는 문제를 해결했습니다(커뮤니티 버그 수정 #34284186).
- InnoDB가 3.05 이하의 Aurora MySQL 버전에서 MySQL 시스템 스키마의 테이블에 `INSTANT` 열을 추가하도록 허용하여 Aurora MySQL 버전 3.05.0으로 업그레이드한 후 서버가 예기치 않게 종료(데이터베이스 인스턴스 다시 시작)될 수 있는 문제를 수정했습니다. (커뮤니티 버그 수정 #35625510)

Aurora MySQL 데이터베이스 엔진 업데이트 2024-01-31(버전 3.05.2)(사용되지 않음)

버전: 3.05.2

Aurora MySQL 3.05.2가 정식 버전입니다. Aurora MySQL 3.05 버전은 MySQL 8.0.32와 호환됩니다. 발생한 커뮤니티 변경 사항에 대한 자세한 내용은 MySQL 8.0 Release Notes를 참조하세요.

Aurora MySQL 버전 3의 새로운 기능에 대한 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환을 참조하세요. Aurora MySQL 버전 3과 Aurora MySQL 버전 2의 차이점에 대한 자세한 내용은 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요. Aurora MySQL 버전 3과 MySQL 8.0 커뮤니티 에디션의 비교 내용은 Amazon Aurora 사용 설명서에서 Aurora MySQL 버전 3과 MySQL 8.0 커뮤니티 에디션 비교를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.03.*, 3.04.* 및 3.05.*입니다.

현재 지원되는 Aurora MySQL 버전 2 클러스터에서 Aurora MySQL 버전 MySQL 3.05.2 클러스터로 Amazon RDS 블루/그린 배포를 사용하여 현재 위치 업그레이드를 수행하거나 스냅샷을 복원하거나 관리형 블루/그린 업그레이드를 시작할 수 있습니다.

Aurora MySQL 버전 3으로의 업그레이드 계획에 대한 자세한 내용은 Aurora MySQL 버전 3에 대한 업그레이드 계획을 참조하세요. Aurora MySQL 업그레이드에 대한 일반 정보는 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터 업그레이드를 참조하세요.

문제 해결 정보는 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3의 업그레이드 문제 해결을 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

보안 문제 및 CVE 해결:

이 릴리스에는 다음 CVE 수정 사항이 포함되어 있습니다.

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

가용성 향상:

- InnoDB 분할 테이블에서 `INSERT` 쿼리를 처리하면 인스턴스의 여유 메모리가 점진적으로 감소할 수 있는 문제를 수정했습니다.
- SHOW STATUS 및 PURGE BINARY LOGS 문을 동시에 실행할 때 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다. `PURGE BINARY LOGS`는 사용자가 구성한 binlog 보존 기간을 준수하기 위해 실행되는 관리형 문입니다.
- 비가상 열이 `MODIFY COLUMN` 또는 `CHANGE COLUMN` 문으로 재정렬된 테이블에서 데이터 조작 언어(DML) 문을 실행한 후 서버가 예기치 않게 닫힐 수 있는 문제를 수정했습니다.
- 데이터베이스 인스턴스를 다시 시작하는 동안 추가 재시작이 발생할 수 있는 문제를 수정했습니다.

일반적인 개선 사항:

- 사용자가 쿼리를 중단하거나 `performance_schema` 쿼리에 대한 세션 제한 시간을 설정할 수 없는 문제를 수정했습니다.
- 복제 인스턴스가 호스트 교체 중일 때 사용자 지정 SSL 인증서 (mysql.rds_import_binlog_ssl_material)를 사용하는 이진 로그(binlog) 복제 설정이 실패할 수 있는 문제를 수정했습니다.
- 다운로드 또는 교체 시 로그 파일에 액세스할 수 없고 경우에 따라 CPU 사용량이 증가할 수 있는 감사 로그 파일 관리 관련 문제를 해결했습니다.
- Aurora MySQL 3.05.2 이하 버전에서는 사용자가 Aurora MySQL 리더 DB 인스턴스에서 `SHOW ENGINE INNODB STATUS`의 출력을 검색할 수 없습니다. 이는 `innodb_read_only`가 활성화된 경우 기본 InnoDB 동작 때문입니다.

  Aurora MySQL 버전 3.05.2 이상에서는 `SHOW ENGINE INNODB STATUS`가 리더 인스턴스에서 실행되면 출력이 MySQL 오류 로그에 기록되므로 문제 해결이 더 쉬워집니다.

  MySQL 오류 로그 작업에 대한 자세한 내용은 Aurora MySQL 오류 로그를 참조하세요. `SHOW ENGINE INNODB STATUS`에 대한 자세한 내용은 MySQL 설명서의 SHOW ENGINE 문을 참조하세요.

업그레이드 및 마이그레이션:

- 테이블 스키마에 사용자 정의 `FTS_DOC_ID` 열이 있는 경우 Aurora MySQL 버전 2에서 Aurora MySQL 버전 3으로 업그레이드 실패를 일으킬 수 있는 문제를 수정했습니다.
- InnoDB 테이블스페이스를 처리하는 동안 동기화 오류로 인해 Aurora MySQL 버전 2에서 Aurora MySQL 버전 3으로의 업그레이드가 실패할 수 있는 문제를 해결했습니다.
- Aurora MySQL 버전 2의 InnoDB 시스템 테이블에 이미 삭제된 테이블스페이스에 대한 분리된 항목이 존재하여 Aurora MySQL 버전 3으로의 메이저 버전 업그레이드가 실패할 수 있는 문제를 수정했습니다.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 아래 사항 외에도 8.0.32까지의 모든 커뮤니티 버그 수정 사항이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 3.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그를 참조하세요.

- `records_in_range`가 `INSERT` 작업에 대해 과도한 수의 디스크 읽기를 수행하여 성능이 점진적으로 저하되는 문제를 수정했습니다. (커뮤니티 버그 수정 #34976138)

Aurora MySQL 데이터베이스 엔진 업데이트 2023-11-21(버전 3.05.1)(사용되지 않음)

버전: 3.05.1

Aurora MySQL 3.05.1이 정식 버전Release Notes입니다. Aurora MySQL 3.05 버전은 MySQL 8.0.32와 호환됩니다. 자세한 내용은 MySQL 8.0 Release Notes를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 2.07.*, 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.*, 3.04.* 및 3.05.*입니다.

기존 Aurora MySQL 3.* 데이터베이스 클러스터를 Aurora MySQL 3.05.1로 업그레이드할 수 있습니다. 현재 지원되는 모든 Aurora MySQL 릴리스에서 Aurora MySQL 3.05.1로 스냅샷을 복원할 수도 있습니다.

Aurora MySQL Global Database를 버전 3.05.*로 업그레이드하려는 경우 기본 및 보조 DB 클러스터를 패치 수준을 포함하여 정확히 동일한 버전으로 업그레이드해야 합니다. Aurora 글로벌 데이터베이스의 마이너 버전 업그레이드에 대한 자세한 내용은 마이너 버전 업그레이드를 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

아래 나열된 보안 문제 및 CVE 해결:

이 릴리스에는 MySQL 8.0.32까지의 모든 커뮤니티 CVE 수정 사항이 포함되어 있습니다.

- CVE-2023-38545

MySQL Community Edition 버그 수정 통합

이 릴리스에는 아래 사항 외에도 8.0.32까지의 모든 커뮤니티 버그 수정이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 3.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그를 참조하세요.

- InnoDB에서 시스템 스키마의 MySQL 테이블에 Aurora MySQL 버전 3.01부터 Aurora MySQL 버전 3.04까지 `INSTANT ADD` 열이 추가된 경우 Aurora MySQL이 버전 3.05.0으로 업그레이드된 후 이러한 테이블의 DML로 인해 서버가 예기치 않게 종료되는 문제를 해결했습니다. (커뮤니티 버그 수정 #35625510)

Aurora MySQL 데이터베이스 엔진 업데이트 2023-10-30(버전 3.05.0.1)(사용되지 않음)

버전: 3.05.0.1

Aurora MySQL 3.05.0.1은 일반적으로 미국 동부(버지니아 북부), 미국 동부(오하이오), 미국 서부(캘리포니아 북부), 미국 서부(오레곤), AWS GovCloud(미국 동부), AWS GovCloud(미국 서부) 리전에서 사용할 수 있습니다. 이는 보안 수정 사항만 포함된 초기 릴리스입니다. 다음 패치 릴리스인 3.05.1에서는 이러한 수정 사항이 모든 리전에 더 광범위하게 배포될 예정입니다. Aurora MySQL 3.05 버전은 MySQL 8.0.32와 호환됩니다. 발생한 커뮤니티 변경 사항에 대한 자세한 내용은 MySQL 8.0 Release Notes를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 2.07.*, 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.*, 3.04.* 및 3.05.*입니다.

기존 Aurora MySQL 3.* 데이터베이스 클러스터를 Aurora MySQL 3.05.0.1로 업그레이드할 수 있습니다. 또한 현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 3.05.0.1로 스냅샷을 복원할 수 있습니다.

Aurora MySQL Global Database를 버전 3.05.*로 업그레이드하려는 경우 기본 및 보조 DB 클러스터를 패치 수준을 포함하여 정확히 동일한 버전으로 업그레이드해야 합니다. Aurora 글로벌 데이터베이스의 마이너 버전 업그레이드에 대한 자세한 내용은 마이너 버전 업그레이드를 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

아래 나열된 보안 문제 및 CVE 해결:

이 릴리스에는 MySQL 8.0.32까지의 모든 커뮤니티 CVE 수정 사항이 포함되어 있습니다.

- CVE-2023-38545

Aurora MySQL 데이터베이스 엔진 업데이트 2023-10-25(버전 3.05.0)(사용되지 않음)

버전: 3.05.0

Aurora MySQL 3.05.0이 정식 출시되었습니다. Aurora MySQL 3.05 버전은 MySQL 8.0.32와 호환됩니다. 발생한 커뮤니티 변경 사항에 대한 자세한 내용은 MySQL 8.0 Release Notes를 참조하세요.

Aurora MySQL 버전 3의 새로운 기능에 대한 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환을 참조하세요. Aurora MySQL 버전 3과 Aurora MySQL 버전 2의 차이점에 대한 자세한 내용은 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요. Aurora MySQL 버전 3과 MySQL 8.0 Community Edition의 비교 내용은 Aurora MySQL 버전 3과 MySQL 8.0 Community Edition 비교를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.03.*, 3.04.* 및 3.05.*입니다.

Amazon RDS 블루/그린 배포를 사용하여 현재 지원되는 Aurora MySQL 버전 2 클러스터에서 Aurora MySQL 버전 3.05.0 클러스터로 인플레이스 업그레이드를 수행하거나, 스냅샷을 복원하거나, 관리형 블루/그린 업그레이드를 시작할 수 있습니다.

Aurora MySQL 버전 3으로의 업그레이드 계획에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3에 대한 업그레이드 계획을 참조하세요. Aurora MySQL 업그레이드에 대한 일반 정보는 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터 업그레이드를 참조하세요.

문제 해결 정보는 Aurora MySQL 버전 3의 업그레이드 문제 해결을 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

새로운 기능:

- Aurora MySQL 데이터베이스 클러스터의 데이터를 KMS 키(SSE-KMS)로 암호화된 Amazon S3 버킷의 텍스트 파일로 저장하기 위한 지원을 추가했습니다. 자세한 내용은 Amazon Aurora MySQL DB 클러스터에서 Amazon S3 버킷의 텍스트 파일로 데이터 저장을 참조하세요.
- 엔진에 사용되는 표준 시간대(TZ) 정보의 현재 버전을 나타내는 새로운 글로벌 상태 변수 `aurora_tmz_version`을 도입했습니다. 값은 IANA 표준 시간대 데이터베이스 버전을 따르며

'YYYYsuffix' 형식(예: 2022a 및 2023c)으로 지정됩니다. 자세한 내용은 Aurora MySQL 글로벌 상태 변수를 참조하세요.

아래 나열된 보안 문제 및 CVE 해결:

관리된 환경에서 처리를 미세 조정하기 위한 수정 사항 및 기타 향상된 기능 추가 CVE 수정은 다음과 같습니다.

- CVE-2022-37434

가용성 향상:

- 병렬 쿼리를 사용하는 Aurora MySQL 데이터베이스 인스턴스에서 많은 수의 동시 병렬 쿼리를 실행할 때 데이터베이스가 다시 시작될 수 있는 문제를 해결했습니다.
- 감사 로깅 스레드로 인해 발생하는 잠금 경합에 따라 결국 CPU 사용률이 높아지고 클라이언트 애플리케이션 제한 시간이 초과되는 문제를 해결했습니다.
- binlog 소스에 `gtid_mode`가 `ON` 또는 `ON_PERMISSIVE`로 설정된 경우 향상된 binlog가 활성화된 바이너리 로그(binlog) 복제본 클러스터에서 실행된 GTID 세트가 잘못 복구될 수 있는 문제를 해결했습니다. 이 문제로 인해 복구 중에 복제본 클러스터의 라이터 인스턴스가 추가로 다시 시작되거나 실행된 GTID 세트를 쿼리할 때 잘못된 결과가 생성될 수 있습니다.
- 향상된 바이너리 로그가 활성화된 경우 사용 가능한 메모리 감소로 인해 Aurora MySQL 데이터베이스 인스턴스가 다시 시작되거나 장애 조치가 발생할 수 있는 메모리 관리 문제를 해결했습니다.
- 삭제된 테이블에 속한 데이터베이스 페이지를 읽으려고 할 때 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 라이터 인스턴스에서 데이터베이스 볼륨이 160GB의 배수로 증가하면 리더 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 바이너리 로그 복구 프로세스가 실행되는 동안 향상된 바이너리 로그 기능이 활성화된 Aurora MySQL 데이터베이스 인스턴스가 시작될 경우 시작 도중 멈출 수 있는 문제를 해결했습니다.
- 대규모 롤백 세그먼트가 초기화되는 동안 Aurora MySQL 데이터베이스 인스턴스가 시작될 경우 여러 번 다시 시작될 수 있는 문제를 해결했습니다.
- 제로 가동 중지 패치 적용 중에 인스턴스가 다시 시작되어 데이터베이스 연결이 예기치 않게 종료되는 문제를 해결했습니다.
- SHOW STATUS 및 PURGE BINARY LOGS 문을 동시에 실행할 때 교착 상태로 인해 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다. purge binary logs는 사용자가 구성한 binlog 보존 기간을 준수하기 위해 실행되는 관리형 문입니다.

- 데이터베이스가 내부 시스템 테이블에서 트리거를 생성하거나 삭제하는 동안 라이터 인스턴스가 다시 시작되면 데이터베이스 클러스터를 사용할 수 없게 되는 문제를 해결했습니다.
- Aurora 복제본이 있는 클러스터에서 향상된 binlog 기능을 사용할 때 긴 세마포어 대기 시간 때문에 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 집계 함수를 참조하는 쿼리를 실행하는 동안 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 드문 경우지만 Aurora Serverless v2가 규모 조정 중에 테이블 캐시 업데이트를 잘못 시도하면 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 중간 임시 테이블을 구체화하는 동안 공통 테이블 표현식(CTE)에 지원되지 않는 인덱스 스캔 액세스 방법이 고려되어 데이터베이스 다시 시작 또는 잘못된 쿼리 결과를 비롯한 원치 않는 동작이 발생할 수 있는 문제를 해결했습니다. 이 문제는 TempTable 스토리지 엔진을 사용하여 테이블에서 지원되지 않는 인덱스 스캔 액세스 방법의 사용을 방지함으로써 해결되었습니다.

일반적인 개선 사항:

- Aurora MySQL 3.04.0에서 실행되는 Aurora Serverless v2 데이터베이스 클러스터에서 향상된 binlog가 활성화된 경우 데이터베이스를 사용할 수 없게 되는 문제를 해결했습니다.
- 향상된 binlog 기능이 활성화된 경우 Aurora 스토리지에 쓰기 전에 사용되지 않은 스토리지 메타데이터를 제거했습니다. 이렇게 하면 네트워크를 통해 전송되는 바이트 증가에 따른 쓰기 지연 시간 증가로 인해 데이터베이스 다시 시작 또는 장애 조치가 발생할 수 있는 특정 상황이 방지됩니다.
- `malloc_stats` 및 `malloc_stats_totals` 테이블이 `performance_schema`에 추가됨에 따라 내부 메모리 할당자인 Jemalloc의 동작을 제어하는 다음과 같은 세 가지 고급 시스템 변수를 추가했습니다.
  - `aurora_jemalloc_background_thread`.
  - `aurora_jemalloc_dirty_decay_ms`.
  - `aurora_jemalloc_tcache_enabled`.
- 업그레이드 또는 마이그레이션 시 Aurora 전용 성능 스키마 테이블이 생성되지 않는 문제를 해결했습니다.
- 새로운 시스템 변수 `aurora_use_vector_instructions`를 추가했습니다. 이 파라미터가 활성화되면 Aurora MySQL은 최적화된 벡터 처리 명령을 사용하여 I/O가 많은 워크로드의 성능을 개선합니다. 이 설정은 Aurora MySQL 3.05 이상에서는 기본적으로 `ON`으로 지정됩니다. 자세한 내용은 Aurora MySQL 구성 파라미터를 참조하세요.
- 향상된 binlog가 활성화된 경우 CloudWatch의 `NumBinaryLogFiles` 지표에 잘못된 결과가 표시될 수 있는 문제를 해결했습니다.

- Amazon SageMaker에 대한 Aurora MySQL 기계 학습 작업의 요청 제한 시간이 3초에서 30초로 늘어났습니다. 이는 더 큰 배치 크기를 사용할 때 Aurora MySQL 기계 학습 기능에서 Amazon SageMaker에 대한 요청 재시도 또는 실패 횟수가 증가하는 문제를 해결하는 데 도움이 됩니다.
- performance_schema 데이터베이스에 `malloc_stats` 및 `malloc_stats_totals` 테이블에 대한 지원을 추가했습니다.
- `LOAD DATA FROM S3` 명령의 `FROM` 키워드를 선택 사항으로 업데이트했습니다. 자세한 내용은 Amazon S3 버킷의 텍스트 파일에서 Amazon Aurora MySQL DB 클러스터로 데이터 로드를 참조하세요.
- `INSTANT` 알고리즘을 `ALTER COLUMN` 작업에 사용할 수 있는지 여부를 제어하는 파라미터 `innodb_aurora_instant_alter_column_allowed`에 대한 지원을 추가했습니다. 자세한 내용은 클러스터 수준 파라미터를 참조하세요.
- 쓰기 전달이 활성화된 경우 데이터베이스에 새 클라이언트 연결이 설정되지 않는 문제를 해결했습니다.
- 데이터베이스 인스턴스를 다시 시작할 때까지 `table_open_cache` 데이터베이스 파라미터의 수정 사항이 적용되지 않는 문제를 해결했습니다.
- 스냅샷 복원, 역추적 또는 데이터베이스 복제 작업 후 내림차순 인덱스를 사용하는 `AUTO_INCREMENT` 열에 대해 중복 키 오류가 발생할 수 있는 문제를 해결했습니다.
- `GROUP BY` 절과 `aurora_parallel_query` 파라미터가 `ON`으로 설정된 상태에서 `SELECT` 쿼리를 실행할 때 부정확한 결과가 반환될 수 있는 인덱스 스캔 관련 문제를 해결했습니다.
- `INFORMATION_SCHEMA INNODB_TABLESPACES` 테이블에 대해 쿼리를 실행할 때 사용 가능한 메모리가 고갈될 수 있는 문제를 해결했습니다.
- ERROR 1146으로 인해 리더 인스턴스가 테이블을 열 수 없는 문제를 해결했습니다. 이 문제는 라이터 인스턴스에서 `INPLACE` 알고리즘을 사용하는 동안 특정 유형의 온라인 데이터 정의 언어(DDL)를 실행할 때 발생합니다.
- 내부 모니터링 프로세스에서 중복 규모 조정 요청을 잘못 제출하는 경우 Aurora Serverless v2 규모 조정 중에 인스턴스가 다시 시작되지 않도록 문제를 해결했습니다.
- 연결된 바이너리 로그(binlog) 소비자가 중복된 binlog 복제 서버 ID를 사용할 때 데이터베이스가 다시 시작될 수 있는 문제를 해결했습니다.
- Aurora MySQL 관리형 바이너리 로그 복제본에 대한 인메모리 릴레이 로그 캐시를 도입했습니다. 이러한 개선을 통해 바이너리 로그 복제 처리량을 최대 40% 늘릴 수 있습니다. 이 개선 사항은 단일 스레드 바이너리 로그 복제를 사용하거나 GTID 자동 위치 지정이 활성화된 다중 스레드 복제를 사용할 때 자동으로 활성화됩니다.

업그레이드 및 마이그레이션:

- 단일 데이터베이스에 매우 많은 수의 테이블이 있는 경우 MySQL 5.7에서 MySQL 8.0으로 업그레이드할 때 서버가 과도한 메모리를 소비했습니다. 테이블 업그레이드 가능 여부를 확인하는 과정에서 모든 데이터 딕셔너리 `Table` 객체를 미리 가져와서 각각 처리하고 이름을 가져온 다음 목록에 대해 버전 호환성 확인을 수행한 것으로 나타났습니다. 이 경우에는 모든 객체를 미리 가져오는 것이 필요 없었고 메모리 소비량도 크게 증가시켰습니다. 이 문제를 해결하기 위해 이제 이러한 경우 한 번에 하나의 `Table` 객체를 가져와서 필요한 확인을 수행하고 이름을 가져온 다음 해당 객체를 해제하고 다음 객체에 대한 작업을 진행합니다. (버그 #34526001)
- 데이터베이스 인스턴스에서 사용 가능한 모든 vCPU를 사용하여 테이블스페이스 확인을 병렬로 실행함으로써 Aurora MySQL 버전 2에서 버전 3으로의 메이저 버전 업그레이드 성능을 개선했습니다.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 아래 사항 외에도 8.0.32까지의 모든 커뮤니티 버그 수정이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 3.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그를 참조하세요.

- 백그라운드 TLS 인증서 교체로 인해 CPU 사용률이 높아질 수 있는 문제를 해결했습니다. (커뮤니티 버그 수정 #34284186)

Aurora MySQL 데이터베이스 엔진 업데이트 2026-01-02(버전 3.04.6, MySQL 8.0.28과 호환)

버전: 3.04.6

Aurora MySQL 3.04.6이 정식 버전입니다. Aurora MySQL 3.04 버전은 MySQL 8.0.28과 호환됩니다. 발생한 커뮤니티 변경 사항에 대한 자세한 내용은 MySQL 8.0 Release Notes를 참조하세요.

Aurora MySQL 버전 3의 새로운 기능에 대한 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환을 참조하세요. Aurora MySQL 버전 3과 Aurora MySQL 버전 2의 차이점에 대한 자세한 내용은 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요. Aurora MySQL 버전 3과 MySQL 8.0 Community Edition의 비교 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3과 MySQL 8.0 Community Edition 비교를 참조하세요.

현재 지원되는 Aurora MySQL 버전 2 클러스터에서 Aurora MySQL 버전 MySQL 3.04.6 클러스터로 Amazon RDS 블루/그린 배포를 사용하여 zero-downtime-patch를 활용하거나 스냅샷을 복원하거나 관리형 블루/그린 업그레이드를 시작하는 현재 위치 업그레이드를 수행할 수 있습니다. https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/blue-green-deployments-overview.html

Aurora MySQL 버전 3으로의 업그레이드 계획에 대한 자세한 내용은 Aurora MySQL 클러스터에 대한 메이저 버전 업그레이드 계획을 참조하세요. Aurora MySQL 업그레이드에 대한 일반 정보는 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터 업그레이드를 참조하세요.

문제 해결 정보는 Amazon Aurora 사용 설명서의 Aurora MySQL 현재 위치 업그레이드 문제 해결을 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

가용성 개선

- 실행 중인 병렬 쿼리`EXPLAIN FOR CONNECTION <query-id>`에서 실행한 `KILL <query-id>` 후 실행할 때 엔진이 다시 시작될 수 있는 문제를 수정했습니다.
- Global Write Forwarding 또는 Local Write Forwarding을 사용할 때 쓰기 전달이 비활성화되거나 리더 인스턴스가 다시 시작될 경우 라이터 인스턴스를 사용할 수 없게 되는 문제를 수정했습니다.

일반 개선 사항

- 쓰기 전달이 활성화되고 파라미터 "aurora_replica_read_consistency"가 수정될 때 리더 인스턴스가 오류 로그를 생성하지 않는 문제를 수정했습니다.
- 일부 SQL 문이 감사 로그에 로깅되지 않을 수 있는 문제를 수정했습니다.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 MySQL 8.0.28까지의 모든 커뮤니티 버그 수정 사항이 포함되어 있습니다. 자세한 정보는 Aurora MySQL 3.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그을 참조하십시오.

Aurora MySQL 데이터베이스 엔진 업데이트 2025년 5월 5일(버전 3.04.4, MySQL 8.0.28과 호환)

버전: 3.04.4

Aurora MySQL 3.04.4가 정식 버전입니다. Aurora MySQL 3.04 버전은 MySQL 8.0.28와 호환됩니다. 발생한 커뮤니티 변경 사항에 대한 자세한 내용은 MySQL 8.0 Release Notes를 참조하세요.

Aurora MySQL 버전 3의 새로운 기능에 대한 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환을 참조하세요. Aurora MySQL 버전 3과 Aurora MySQL 버전 2의 차이점에 대한 자세한 내용은 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요. Aurora MySQL 버전 3과 MySQL 8.0 Community Edition의 비교 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3과 MySQL 8.0 Community Edition 비교를 참조하세요.

> **Note**
>
> 이 버전은 장기 지원(LTS) 릴리스로 지정됩니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL LTS(장기 지원) 릴리스를 참조하세요.
> LTS 버전의 경우 `AutoMinorVersionUpgrade` 파라미터를 `true`로 설정(또는 AWS Management Console에서 자동 마이너 버전 업그레이드 활성화)하지 않는 것이 좋습니다. 이렇게 하면 DB 클러스터가 LTS 버전이 아닐 수 있는 마이너 버전 자동 업그레이드 캠페인의 다음 대상 버전으로 업그레이드될 수 있습니다.

제로 가동 중지 패치를 활용하는 현재 위치 업그레이드를 수행하고, Amazon RDS 블루/그린 배포를 사용하여 현재 지원되는 Aurora MySQL 버전 2 클러스터에서 Aurora MySQL 버전 3.04.4 클러스터로 스냅샷을 복원하거나 관리형 블루/그린 업그레이드를 시작할 수 있습니다.

Aurora MySQL 버전 3으로의 업그레이드 계획에 대한 자세한 내용은 Aurora MySQL 클러스터에 대한 메이저 버전 업그레이드 계획을 참조하세요. Aurora MySQL 업그레이드에 대한 일반 정보는 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터 업그레이드를 참조하세요.

문제 해결 정보는 Amazon Aurora 사용 설명서의 Aurora MySQL 현재 위치 업그레이드 문제 해결을 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

보안 수정

중요 CVE:

- CVE-2024-11053
- CVE-2024-37371

가용성 향상:

- 네트워크 중단으로 인해 라이터와의 연결이 올바르게 재설정되지 않는 복제본 문제를 수정했습니다.
- 릴레이 로그 복구 중에 많은 수의 릴레이 로그 파일을 처리할 때 이진 로그(binlog) 복제본이 다시 시작될 수 있는 문제를 수정했습니다.
- 병렬 쿼리 기능을 사용하여 쿼리를 실행할 때 데이터베이스 리더 인스턴스가 다시 시작되는 문제를 수정했습니다.
- 드문 경우지만 대규모 트랜잭션을 커밋하는 동안 오류가 발생할 때 이진 로깅을 비활성화할 수 있는 문제를 수정했습니다.
- 라이터 DB 인스턴스에서 드문 트랜잭션 커밋 주문이 발생하는 경우 Aurora 읽기 전용 복제본이 다시 시작될 수 있는 문제를 수정했습니다.
- 향상된 Binlog가 활성화된 인스턴스에서 실행 중에 예약된 이벤트가 중단될 때 데이터베이스가 다시 시작될 수 있는 문제를 수정했습니다.
- 다중 TB Aurora 클러스터 볼륨을 사용하는 데이터베이스 인스턴스가 InnoDB 버퍼풀 검증 실패로 인해 재시작 중에 가동 중지 시간이 증가할 수 있는 문제를 해결했습니다.

일반적인 개선 사항

- 다음 권한이 `rds_superuser_role`, `FLUSH_OPTIMIZER_COSTS`, `FLUSH_STATUS`, `FLUSH_TABLES` 및 `FLUSH_USER_RESOURCES`에 추가되었습니다. `rds_superuser_role`에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora를 사용하는 Amazon 관리자 사용자 계정을 참조하세요. 이러한 동적 권한에 대한 자세한 내용은 MySQL 설명서를 참조하세요.
- 제로 가동 중지 패치(ZDP)/제로 가동 중지 재시작(ZDR) 중에 보존된 연결이 잘못 처리되어 클라이언트가 쿼리가 완료될 때까지 무기한 대기하는 문제를 수정했습니다.
- 업데이트 중에 공간 인덱스를 통해 행을 읽을 수 없게 되는 문제를 수정했습니다.
- 제로 가동 중지 재시작 또는 제로 가동 중지 패치 적용 작업 중에 중단된 옵티마이저 힌트가 포함된 쿼리가 잘못 처리될 수 있는 문제를 해결했습니다.
- `innodb_flush_log_at_trx_commit`가 `0`으로 설정된 경우 커밋 지연 시간이 측정되지 않는 문제를 수정했습니다.
- 테이블 수가 많을 때 논리적 데이터 사전 복구 중에 메모리 사용량을 줄이기 위한 최적화를 도입했습니다.

- 향상된 Binlog가 활성화되었거나 이전에 활성화된 클러스터에서 `SHOW BINARY LOGS` 명령이 실행되는 데 시간이 더 오래 걸리는 문제를 수정했습니다. 이 문제로 인해 여러 `SHOW BINARY LOGS` 명령이 동시에 실행되는 경우 커밋 지연 시간이 증가할 수도 있습니다.
- 뷰 정의자의 기본 역할과 관련된 메모리 문제를 수정했습니다.
- 쓰기 전달 기능을 비활성화하는 프로세스를 완료하지 못할 수 있는 문제를 해결했습니다.
- 쓰기 전달이 활성화된 경우 데이터베이스에 새 클라이언트 연결이 설정되지 않는 문제를 해결했습니다.
- 쓰기 전달을 사용하는 리더 인스턴스가 `timestamp` 값을 포함하고 `time_zone` 데이터베이스 파라미터가 "UTC"로 설정된 DML 문을 실행할 때 라이터 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 복제본의 동시 읽기 쿼리와 라이터의 DDL 작업으로 인해 복제본에 `'Table does not exist'` 오류가 발생하여 Aurora 읽기 전용 복제본 또는 테이블 정의 불일치가 간헐적으로 발생하지 않는 문제를 수정했습니다.
- 병렬 쿼리와 함께 해시 조인 알고리즘을 사용하여 `LEFT-` 또는 `RIGHT-JOIN` 작업과 관련된 쿼리를 실행할 때 불완전한 결과 세트가 발생할 수 있는 문제를 수정했습니다.
- 이 Aurora MySQL 버전부터는 빠른 삽입 최적화가 더 이상 활성화되지 않습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL 성능 개선 사항을 참조하세요.

업그레이드 및 마이그레이션:

- AMS2에서 AMS3으로 업그레이드하는 동안 불필요하게 생성된 기본 역할을 제거했습니다.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 MySQL 8.0.28까지의 모든 커뮤니티 버그 수정 사항이 포함되어 있습니다. 자세한 내용은 섹션을 참조하세요Aurora MySQL 3.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그

- `FLUSH STATUS`, `COM_CHANGE_USER` 및`SHOW PROCESS LIST`가 동시에 실행될 때 교착 상태를 해결하는 문제를 수정했습니다. (버그#35218030)

Aurora MySQL 데이터베이스 엔진 업데이트 2024년 6월 26일(버전 3.04.3, MySQL 8.0.28과 호환)

버전: 3.04.3

Aurora MySQL 3.04.3이 정식 버전입니다. Aurora MySQL 3.04 버전은 MySQL 8.0.28과 호환됩니다. 발생한 커뮤니티 변경 사항에 대한 자세한 내용은 MySQL 8.0 Release Notes를 참조하세요.

Aurora MySQL 버전 3의 새로운 기능에 대한 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환을 참조하세요. Aurora MySQL 버전 3과 Aurora MySQL 버전 2의 차이점에 대한 자세한 내용은 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요. Aurora MySQL 버전 3과 MySQL 8.0 Community Edition의 비교 내용은 Aurora MySQL 버전 3과 MySQL 8.0 Community Edition 비교를 참조하세요.

> **Note**
>
> 이 버전은 장기 지원(LTS) 릴리스로 지정됩니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL LTS(장기 지원) 릴리스를 참조하세요.
> LTS 버전의 경우 `AutoMinorVersionUpgrade` 파라미터를 `true` 로 설정(또는에서 마이너 버전 자동 업그레이드 활성화 AWS Management Console)하지 않는 것이 좋습니다. 이렇게 하면 DB 클러스터가 LTS 버전이 아닐 수 있는 마이너 버전 자동 업그레이드 캠페인의 다음 대상 버전으로 업그레이드될 수 있습니다.

현재 지원되는 Aurora MySQL 릴리스는 2.07.9, 2.7.10, 2.11.*, 2.12.*, 3.03.*, 3.04.*, 3.05.*, 3.06.* 및 3.07.*입니다.

현재 사용 가능한 Aurora MySQL 버전 2 클러스터에서 Aurora MySQL 버전 MySQL 3.04.3 클러스터로 Amazon RDS 블루/그린 배포를 사용하여 현재 위치 업그레이드를 수행하거나 스냅샷을 복원하거나 관리형 블루/그린 업그레이드를 시작할 수 있습니다.

Aurora MySQL 버전 3으로의 업그레이드 계획에 대한 자세한 내용은 Aurora MySQL DB 클러스터에 대한 메이저 버전 업그레이드 계획을 참조하세요. Aurora MySQL 업그레이드에 대한 일반 정보는 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터 업그레이드를 참조하세요.

문제 해결 정보는 Amazon Aurora 사용 설명서의 Aurora MySQL 현재 위치 업그레이드 문제 해결을 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

보안 문제 및 CVE 해결:

이 릴리스에는 MySQL 8.0.28까지의 모든 커뮤니티 CVE 수정 사항이 포함되어 있습니다. 다음 CVE 수정 사항이 포함되어 있습니다.

- CVE-2024-0853

가용성 향상:

- 병렬 쿼리를 실행할 때 Aurora MySQL DB 인스턴스가 다시 시작되는 문제를 수정했습니다.
- 라이터 DB 인스턴스에서 변경되거나 삭제되는 테이블을 읽을 때 리더 DB 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 메모리 액세스 위반으로 인해 스레드가 더 이상 소유하지 않은 뮤텍스 객체가 해제되는 문제를 수정했습니다.
- 전달된 쿼리를 실행하는 동안 쓰기 전달 세션이 닫힐 때 Aurora MySQL 라이터 DB 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 바이너리 로그가 활성화된 인스턴스에서 대규모 GTID 세트를 처리할 때 DB 인스턴스가 다시 시작되는 문제를 수정했습니다.
- InnoDB 분할 테이블에서 `INSERT` 쿼리를 처리할 때 DB 인스턴스의 여유 메모리가 점진적으로 감소할 수 있는 문제를 해결했습니다.
- 드문 경우지만 외래 키 제약 조건이 있는 테이블에서 `SELECT` 쿼리를 수행할 때 리더 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 데이터베이스 복구 중에 InnoDB 데이터 사전 복구에 오랜 시간이 걸릴 때 데이터베이스가 다시 시작될 수 있는 문제를 수정했습니다.
- 가상 열이 외래 키 제약 조건의 열 또는 참조된 테이블의 멤버로 포함되는 테이블에 계단식 `UPDATE` 또는 `DELETE` 외래 키 제약 조건이 정의될 때 데이터베이스가 다시 시작될 수 있는 문제를 수정했습니다.
- 스케일 업 도중에 데이터베이스가 다시 시작될 수 있는 Aurora Serverless v2의 문제를 수정했습니다.

일반적인 개선 사항:

- Aurora Global Database를 사용할 때 `threads_running` 상태 변수에 잘못된 값을 제공한 문제를 수정했습니다.
- 병렬 읽기를 사용할 때 `rw_lock`의 잠금 소유자 정보가 부정확하여 DB 인스턴스가 다시 시작되는 문제를 수정했습니다.

- `SELECT ... INTO OUTFILE ...` 쿼리를 실행할 때 시간이 지남에 따라 사용 가능한 메모리가 감소하는 메모리 관리 문제를 수정했습니다.
- DB 인스턴스의 로컬 스토리지가 전체 용량에 도달하면 DB 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- db.t4g.medium 및 db.t4g.large DB 인스턴스에 대해 Performance Insights 자동 관리가 켜져 있을 때 성능 스키마가 활성화되지 않는 문제를 수정했습니다.
- 고객이 구성한 `wait_timeout` 또는 `interactive_timeout`에 도달할 때 DB 인스턴스가 클라이언트 연결을 닫지 못하게 하는 제로 가동 중지 시간 패치(ZDP) 중 문제를 수정했습니다.
- 테이블에 전체 텍스트 검색(FTS) 인덱스가 하나 이상 있고 Aurora 라이터 DB 인스턴스에서 `TRUNCATE` 문이 실행 중인 경우 테이블이 존재하지 않습니다 오류와 함께 Aurora 리더 인스턴스의 `SELECT` 쿼리가 실패할 수 있는 문제가 해결되었습니다.

업그레이드 및 마이그레이션:

- 대상 Aurora MySQL DB 엔진 버전이 3.04.0 이상일 때 업그레이드 또는 마이그레이션이 실패하는 문제를 수정했습니다. 이는 `lower_case_table_names` DB 클러스터 파라미터가 1로 설정되어 있고 MySQL 데이터베이스 데이터 정렬이 소문자 테이블 이름과 호환되지 않는 경우에 발생합니다.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 MySQL 8.0.28까지의 모든 커뮤니티 버그 수정 사항이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 3.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그 단원을 참조하십시오.

Aurora MySQL 데이터베이스 엔진 업데이트 2024년 3월 15일(버전 3.04.2, MySQL 8.0.28과 호환)

버전: 3.04.2

Aurora MySQL 3.04.2를 정식 버전으로 사용할 수 있습니다. Aurora MySQL 3.04 버전은 MySQL 8.0.28과 호환됩니다. 발생한 커뮤니티 변경 사항에 대한 자세한 내용은 MySQL 8.0 Release Notes를 참조하세요.

Aurora MySQL 버전 3의 새로운 기능에 대한 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환을 참조하세요. Aurora MySQL 버전 3과 Aurora MySQL 버전 2의 차이점에 대한 자세한 내용은 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요. Aurora MySQL 버전 3

과 MySQL 8.0 Community Edition의 비교 내용은 Aurora MySQL 버전 3과 MySQL 8.0 Community Edition 비교를 참조하세요.

> **Note**
>
> 이 버전은 장기 지원(LTS) 릴리스로 지정됩니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL LTS(장기 지원) 릴리스를 참조하세요.
> LTS 버전의 경우 `AutoMinorVersionUpgrade` 파라미터를 `true` 로 설정(또는에서 마이너 버전 자동 업그레이드 활성화 AWS Management Console)하지 않는 것이 좋습니다. 이렇게 하면 DB 클러스터가 LTS 버전이 아닐 수 있는 마이너 버전 자동 업그레이드 캠페인의 다음 대상 버전으로 업그레이드될 수 있습니다.

현재 지원되는 Aurora MySQL 릴리스는 2.07.9, 2.7.10, 2.11.*, 2.12.*, 3.03.*, 3.04.*, 3.05.* 및 3.06.*입니다.

현재 사용 가능한 Aurora MySQL 버전 2 클러스터에서 Aurora MySQL 버전 MySQL 3.04.2 클러스터로 Amazon RDS 블루/그린 배포를 사용하여 현재 위치 업그레이드를 수행하거나 스냅샷을 복원하거나 관리형 블루/그린 업그레이드를 시작할 수 있습니다.

Aurora MySQL 버전 3으로의 업그레이드 계획에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3에 대한 업그레이드 계획을 참조하세요. Aurora MySQL 업그레이드에 대한 일반 정보는 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터 업그레이드를 참조하세요.

문제 해결 정보는 Aurora MySQL 버전 3의 업그레이드 문제 해결을 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

보안 문제 및 CVE 해결:

이 릴리스에는 다음 CVE 수정 사항이 포함되어 있습니다.

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

- CVE-2023-38546
- CVE-2023-39975

가용성 향상:

- 라이터 DB 인스턴스에 워크로드가 많을 때 읽기 전용 복제본 DB 인스턴스를 성공적으로 시작할 수 없는 문제를 수정했습니다.
- Aurora 스토리지와 통신하는 구성 요소의 결함으로 인해 Aurora MySQL 라이터 DB 인스턴스가 장애 조치될 수 있는 문제를 수정했습니다. 결함은 소프트웨어 업데이트 후 DB 인스턴스와 기본 스토리지 간의 통신이 분석된 결과로 발생합니다.
- SHOW STATUS 및 PURGE BINARY LOGS 문을 동시에 실행할 때 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다. `PURGE BINARY LOGS`는 사용자가 구성한 binlog 보존 기간을 준수하기 위해 실행되는 관리형 문입니다.
- 데이터베이스 인스턴스를 다시 시작하는 동안 추가 재시작이 발생할 수 있는 문제를 수정했습니다.
- 감사 로깅 스레드로 인해 발생하는 잠금 경합에 따라 CPU 사용률이 높아지고 클라이언트 애플리케이션 제한 시간이 초과될 수 있는 문제를 해결했습니다.
- 대규모 롤백 세그먼트가 초기화되는 동안 Aurora MySQL 데이터베이스 인스턴스가 시작될 경우 여러 번 다시 시작될 수 있는 문제를 해결했습니다.
- 집계 함수를 참조하는 쿼리를 실행하는 동안 DB 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.

일반적인 개선 사항:

- Aurora DB 클러스터 볼륨에서 데이터를 읽는 동안 일시적인 네트워크 문제로 인해 병렬 쿼리가 실패할 수 있는 문제를 해결했습니다.
- 사용자가 쿼리를 중단하거나 `performance_schema` 쿼리에 대한 세션 제한 시간을 설정할 수 없는 문제를 수정했습니다.
- 복제 인스턴스가 호스트 교체 중일 때 사용자 지정 SSL 인증서 (mysql.rds_import_binlog_ssl_material)를 사용하도록 구성된 바이너리 로그(binlog) 복제가 실패할 수 있는 문제를 수정했습니다.
- 다운로드 또는 교체 시 로그 파일에 액세스할 수 없고 경우에 따라 CPU 사용량이 증가할 수 있는 감사 로그 파일 관리 관련 문제를 해결했습니다.
- 스냅샷 복원, 특정 시점 복구 수행, 데이터베이스에 많은 수의 테이블이 있는 DB 클러스터 복제의 완료 시간을 줄이기 위해 `AUTO_INCREMENT` 키 복구를 최적화했습니다.

- 일부 `performance_schema` 테이블을 참조하는 SQL 문이 Community MySQL에서 Aurora MySQL 버전 3.04.0 및 3.04.1로 마이그레이션한 후 이러한 테이블이 누락되어 오류를 반환할 수 있는 문제를 수정했습니다.
- Aurora MySQL 2.11.* 이하 버전에서 업그레이드한 후 소규모 읽기 전용 복제본 인스턴스에서 복제 지연이 증가할 수 있는 문제를 해결했습니다.
- 스냅샷 복원, 역추적 또는 데이터베이스 복제 작업 후 내림차순 인덱스를 사용하는 `AUTO_INCREMENT` 열에 대해 중복 키 오류가 발생할 수 있는 문제를 해결했습니다.
- DB 인스턴스를 다시 시작할 때까지 `table_open_cache` 데이터베이스 파라미터의 수정 사항이 적용되지 않는 문제를 해결했습니다.
- ERROR 1146으로 인해 리더 DB 인스턴스가 테이블을 열 수 없는 문제를 해결했습니다. 이 문제는 라이터 DB 인스턴스에서 `INPLACE` 알고리즘을 사용하는 동안 특정 유형의 온라인 데이터 정의 언어(DDL) 문을 실행할 때 발생합니다.
- 내부 모니터링 프로세스에서 중복 규모 조정 요청을 잘못 제출하는 경우 Aurora Serverless v2 규모 조정 중에 인스턴스가 다시 시작되지 않도록 문제를 해결했습니다.
- 연결된 바이너리 로그(binlog) 소비자가 중복된 binlog 복제 서버 ID를 사용할 때 데이터베이스가 다시 시작될 수 있는 문제를 해결했습니다.

업그레이드 및 마이그레이션:

- Aurora MySQL 버전 2의 InnoDB 시스템 테이블에 이미 삭제된 테이블스페이스에 대한 분리된 항목이 존재하여 Aurora MySQL 버전 3으로의 메이저 버전 업그레이드가 실패할 수 있는 문제를 수정했습니다.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 아래 사항 외에도 8.0.28까지의 모든 커뮤니티 버그 수정 사항이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 3.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그를 참조하세요.

- 캐시 라인 값을 잘못 계산하여 Graviton 기반 인스턴스에서 데이터베이스를 다시 시작하는 동안 오류가 발생하는 문제를 수정했습니다. (커뮤니티 버그 수정 #35479763)
- 여러 `AND`, `OR` 또는 `XOR` 조건이 포함된 SELECT 문을 하위 쿼리로 사용하여 저장된 루틴을 반복적으로 실행하면 과도한 소비와 최종 가상 메모리 소진이 발생할 수 있습니다. (커뮤니티 버그 수정 #33852530)

Aurora MySQL 데이터베이스 엔진 업데이트 2023년 11월 13일(버전 3.04.1, MySQL 8.0.28과 호환)

버전: 3.04.1

Aurora MySQL 3.04.1이 정식 버전입니다. Aurora MySQL 3.04 버전은 MySQL 8.0.28과 호환됩니다. 발생한 커뮤니티 변경 사항에 대한 자세한 내용은 MySQL 8.0 Release Notes를 참조하세요.

> **Note**
>
> 이 버전은 장기 지원(LTS) 릴리스로 지정됩니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL LTS(장기 지원) 릴리스를 참조하세요.
> LTS 버전의 경우 `AutoMinorVersionUpgrade` 파라미터를 `true` 로 설정(또는에서 마이너 버전 자동 업그레이드 활성화 AWS Management Console)하지 않는 것이 좋습니다. 이렇게 하면 DB 클러스터가 LTS 버전이 아닐 수 있는 마이너 버전 자동 업그레이드 캠페인의 다음 대상 버전으로 업그레이드될 수 있습니다.

Aurora MySQL 버전 3의 새로운 기능에 대한 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환을 참조하세요. Aurora MySQL 버전 3과 Aurora MySQL 버전 2의 차이점에 대한 자세한 내용은 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요. Aurora MySQL 버전 3과 MySQL 8.0 Community Edition의 비교 내용은 Aurora MySQL 버전 3과 MySQL 8.0 Community Edition 비교를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 2.07.9, 2.7.10, 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.*, 3.04.* 및 3.05.*입니다.

현재 사용 가능한 Aurora MySQL 버전 2 클러스터에서 Aurora MySQL 버전 MySQL 3.04.1 클러스터로 Amazon RDS 블루/그린 배포를 사용하여 현재 위치 업그레이드를 수행하거나 스냅샷을 복원하거나 관리형 블루/그린 업그레이드를 시작할 수 있습니다.

Aurora MySQL 버전 3으로의 업그레이드 계획에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3에 대한 업그레이드 계획을 참조하세요. Aurora MySQL 업그레이드에 대한 일반 정보는 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터 업그레이드를 참조하세요.

문제 해결 정보는 Aurora MySQL 버전 3의 업그레이드 문제 해결을 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

가용성 향상:

- 병렬 쿼리를 사용하는 Aurora MySQL 데이터베이스 인스턴스에서 많은 수의 동시 병렬 쿼리를 실행할 때 데이터베이스가 다시 시작될 수 있는 문제를 해결했습니다.
- binlog 소스에 `gtid_mode`가 `ON` 또는 `ON_PERMISSIVE`로 설정된 경우 향상된 binlog가 활성화된 바이너리 로그(binlog) 복제본 클러스터에서 실행된 GTID 세트가 잘못 복구될 수 있는 문제를 해결했습니다. 이 문제로 인해 복구 중에 복제본 클러스터의 라이터 인스턴스가 추가로 다시 시작되거나 실행된 GTID 세트를 쿼리할 때 잘못된 결과가 생성될 수 있습니다.
- 향상된 바이너리 로그가 활성화된 경우 사용 가능한 메모리 감소로 인해 Aurora MySQL 데이터베이스 인스턴스가 다시 시작되거나 장애 조치가 발생할 수 있는 메모리 관리 문제를 해결했습니다.
- 라이터 인스턴스에서 데이터베이스 볼륨이 160GB의 배수로 증가하면 리더 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 바이너리 로그 복구 프로세스가 실행되는 동안 향상된 바이너리 로그 기능이 활성화된 Aurora MySQL 데이터베이스 인스턴스가 시작될 경우 시작 도중 멈출 수 있는 문제를 해결했습니다.
- `SHOW STATUS` 및 `PURGE BINARY LOGS` 문을 동시에 실행할 때 교착 상태로 인해 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다. purge binary logs는 사용자가 구성한 binlog 보존 기간을 준수하기 위해 실행되는 관리형 문입니다.
- 데이터베이스가 내부 시스템 테이블에서 트리거를 생성하거나 삭제하는 동안 라이터 인스턴스가 다시 시작되면 데이터베이스 클러스터를 사용할 수 없게 되는 문제를 해결했습니다.
- Aurora 복제본이 있는 클러스터에서 향상된 binlog 기능을 사용할 때 긴 세마포어 대기 시간 때문에 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.

일반적인 개선 사항:

- Aurora MySQL 3.04.0에서 실행되는 Aurora Serverless v2 데이터베이스 클러스터에서 향상된 binlog가 활성화된 경우 데이터베이스를 사용할 수 없게 되는 문제를 해결했습니다.
- 향상된 binlog 기능이 활성화된 경우 Aurora Storage에 쓰기 전에 사용되지 않은 스토리지 메타데이터를 제거했습니다. 이렇게 하면 네트워크를 통해 전송되는 바이트 증가에 따른 쓰기 지연 시간 증가로 인해 데이터베이스 다시 시작 또는 장애 조치가 발생할 수 있는 특정 상황이 방지됩니다.

- 업그레이드 또는 마이그레이션 시 Aurora 전용 성능 스키마 테이블이 생성되지 않는 문제를 해결했습니다.
- 향상된 binlog가 활성화된 경우 CloudWatch의 `NumBinaryLogFiles` 지표에 잘못된 결과가 표시될 수 있는 문제를 해결했습니다.

업그레이드 및 마이그레이션:

- 단일 데이터베이스에 매우 많은 수의 테이블이 있는 경우 MySQL 5.7에서 MySQL 8.0으로 업그레이드할 때 서버가 과도한 메모리를 소비했습니다. 테이블 업그레이드 가능 여부를 확인하는 과정에서 모든 데이터 딕셔너리 `Table` 객체를 미리 가져와서 각각 처리하고 이름을 가져온 다음, 목록에 대해 `CHECK TABLE ... FOR UPGRADE`를 수행한 것으로 나타났습니다. 이 경우에는 모든 객체를 미리 가져오는 것이 필요 없었고 메모리 소비량도 크게 증가시켰습니다. 이 문제를 해결하기 위해 이제 이러한 경우 한 번에 하나의 `Table` 객체를 가져와서 필요한 확인을 수행하고 이름을 가져온 다음 해당 객체를 해제하고 다음 객체에 대한 작업을 진행합니다. (버그 #34526001)

MySQL Community Edition 버그 수정 통합

이 릴리스에는 아래 사항 외에도 8.0.28까지의 모든 커뮤니티 버그 수정이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 3.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그를 참조하세요.

- 백그라운드 TLS 인증서 교체로 인해 CPU 사용률이 높아질 수 있는 문제를 해결했습니다(커뮤니티 버그 수정 #34284186).

Aurora MySQL 데이터베이스 엔진 업데이트 2023-07-31(버전 3.04.0) , MySQL 8.0.28과 호환)

버전: 3.04.0

Aurora MySQL 3.04.0이 정식 출시되었습니다. Aurora MySQL 3.04 버전은 MySQL 8.0.28과 호환되고, Aurora MySQL 3.03 버전은 MySQL 8.0.26과 호환되며, Aurora MySQL 3.02 버전은 MySQL 8.0.23과 호환됩니다. 8.0.23에서 8.0.28까지 이루어진 커뮤니티 변경 사항에 대한 자세한 내용은 MySQL 8.0 Release Notes를 참조하세요.

> **Note**
>
> 이 버전은 장기 지원(LTS) 릴리스로 지정됩니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL LTS(장기 지원) 릴리스를 참조하세요.
> LTS 버전의 경우 `AutoMinorVersionUpgrade` 파라미터를 `true` 로 설정(또는에서 마이너 버전 자동 업그레이드 활성화 AWS Management Console)하지 않는 것이 좋습니다. 이렇게 하면 DB 클러스터가 LTS 버전이 아닐 수 있는 마이너 버전 자동 업그레이드 캠페인의 다음 대상 버전으로 업그레이드될 수 있습니다.

Aurora MySQL 버전 3의 새로운 기능에 대한 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환을 참조하세요. Aurora MySQL 버전 3과 Aurora MySQL 버전 2의 차이점에 대한 자세한 내용은 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요. Aurora MySQL 버전 3과 MySQL 8.0 Community Edition의 비교 내용은 Aurora MySQL 버전 3과 MySQL 8.0 Community Edition 비교를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 2.07.9, 2.11.1, 2.11.2, 3.01.*, 3.02.*, 3.03.* 및 3.04.0입니다.

Amazon RDS 블루/그린 배포를 사용하여 현재 지원되는 Aurora MySQL 버전 2 클러스터에서 Aurora MySQL 버전 3.04.0 클러스터로 인플레이스 업그레이드를 수행하거나, 스냅샷을 복원하거나, 관리형 블루/그린 업그레이드를 시작할 수 있습니다.

Aurora MySQL 버전 3으로의 업그레이드 계획에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3에 대한 업그레이드 계획을 참조하세요. Aurora MySQL 업그레이드에 대한 일반 정보는 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터 업그레이드를 참조하세요.

문제 해결 정보는 Aurora MySQL 버전 3의 업그레이드 문제 해결을 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> Aurora MySQL 향상된 바이너리 로그(binlog)는 현재 Aurora MySQL 버전 3.04.0의 Aurora Serverless v2 데이터베이스 인스턴스에서는 지원되지 않습니다. 이 기능을 활성화하면 데이터베이스를 사용하지 못하게 될 수 있습니다. Aurora MySQL 버전 3.04.0에서 향상된 바이너리 로그를 사용해야 하는 경우 비서버리스 데이터베이스 인스턴스 클래스를 사용하거나

Serverless v2 데이터베이스 인스턴스의 최소 및 최대 ACU를 동일한 값으로 설정하는 것이 좋습니다.
Aurora MySQL의 향상된 바이너리 로깅에 대한 자세한 내용은 Aurora 사용 설명서에서 확인할 수 있습니다.

개선 사항

새로운 기능:

- InnoDB 전체 텍스트 인덱스를 사용하여 자연어 모드에서 구문을 검색하는 쿼리의 성능을 개선했습니다. MySQL에서의 전체 텍스트 검색에 대한 자세한 내용은 Full-Text Search Functions를 참조하세요.
- Amazon Aurora MySQL이 로컬(클러스터 내) 쓰기 전달을 지원합니다. 이제 리더 DB 인스턴스의 쓰기 작업을 Aurora MySQL DB 클러스터 내의 라이터 DB 인스턴스로 전달할 수 있습니다. 자세한 내용은 Aurora MySQL DB 클러스터에서 로컬 쓰기 전달 사용을 참조하세요.
- `autocommit`이 비활성화된 세션에서 Amazon Aurora 글로벌 데이터베이스에서 쓰기 전달 사용 기능에 대한 `aurora_replica_read_consistency` 파라미터의 값을 변경하는 기능을 추가했습니다. 자세한 내용은 쓰기 전달에 대한 구성 파라미터를 참조하세요.
- Aurora MySQL 3.04부터 글로벌 데이터베이스 쓰기 전달 기능의 경우 이제 데이터베이스 클러스터 및 데이터베이스 인스턴스 파라미터 그룹을 통해 `aurora_replica_read_consistency` 파라미터 값을 설정할 수 있습니다. Aurora MySQL 버전 3.04 이전에는 이 파라미터 값을 세션 수준에서만 구성할 수 있었습니다.

보안 문제 및 CVE 해결:

- SSL/TLS 공급자를 OpenSSL에서 AWS-LC로 변경했습니다. 이로 인해 다음을 포함하되 이에 국한되지 않는 다양한 변경 사항이 발생합니다.
  - 이제 Aurora MySQL 버전 3.04.0에서 상위 버전으로 업그레이드할 때 제로 가동 중지 다시 시작 및 제로 가동 중지 패치 적용을 통해 SSL을 사용한 데이터베이스 연결을 복원할 수 있습니다.
  - TLS_AES_128_GCM_SHA256, TLS_AES_256_GCM_SHA384 및 TLS_CHACHA20_POLY1305_SHA256 SSL 암호에 대한 지원을 포함하여 TLSv1.3을 지원합니다.
  - 보안 수준이 낮은 DHE-RSA-* 암호에 대한 지원이 제거되었습니다.

  자세한 내용은 Aurora MySQL DB 클러스터에서 TLS 사용을 참조하세요.

- 저장 프로시저 및 함수와 같은 모든 저장 루틴의 정의 및 속성에 액세스할 수 있는 동적 권한 `SHOW_ROUTINE`을 `rds_superuser_role`에 추가했습니다. 자세한 내용은 SHOW_ROUTINE을 참조하세요.
- 감사 로그 파일 교체 중에 감사 로그에서 이벤트가 누락될 수 있는 문제를 해결했습니다.
- TLS 1.2 버전과의 호환성을 유지하면서 안전하고 성능이 우수한 전송 계층 보안(TLS) 1.3 프로토콜에 대한 지원을 활성화했습니다.
- TLS 버전 TLSv1 및 TLSv1.1은 커뮤니티 MySQL 8.0.26에서 사용 중단되었으며 이에 따라 Aurora MySQL 3.03에서 더 이상 사용되지 않습니다. 이러한 프로토콜은 이제 커뮤니티 MySQL 8.0.28과 Aurora MySQL 3.04에서 제거되었습니다. 기본적으로 TLS 1.2 이상을 통해 통신할 수 없는 모든 보안 클라이언트는 거부됩니다. TLS를 사용하여 데이터베이스 인스턴스에 연결하는 방법에 대한 자세한 내용은 Amazon Aurora MySQL을 사용한 보안을 참조하세요.

이 릴리스에는 다음 CVE 수정 사항이 포함되어 있습니다.

- 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

가용성 향상:

- 긴 트랜잭션 복구 중에 데이터베이스가 다시 시작될 수 있는 문제를 해결했습니다.
- 데이터베이스 활동 스트림 이벤트 암호화 내에서 데이터베이스 다시 시작을 일으킬 수 있는 문제를 해결했습니다.
- 시작 또는 Aurora Serverless v2 스케일 인 중에 InnoDB 버퍼 풀이 초기화될 때 메모리 부족 오류로 인한 메모리 관리 문제를 해결했습니다. 이 문제로 인해 데이터베이스 인스턴스가 다시 시작되거나 처리량 감소 또는 지연 시간 증가를 비롯한 성능 저하가 발생할 수 있습니다.

- Aurora MySQL 병렬 쿼리 실행 계획을 사용하는 쿼리를 실행하는 동안 Aurora MySQL 리더 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 특정 상황에서 범위 추정 중에 Aurora 리더 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 자동 증가 열과 관련된 많은 삽입 작업을 실행하는 동안 다시 시작이 발생하는 경우 시작 시 데이터베이스 복구가 중단될 수 있는 문제를 해결했습니다.
- 서버 변수 `server_audit_events`가 `ALL` 또는 `QUERY`로 설정된 경우 Aurora MySQL 오류 로그에 정보 메시지가 과도하게 로깅되는 Aurora 고급 감사 관련 문제를 해결했습니다. 이 문제로 인해 데이터베이스 인스턴스가 다시 시작될 수 있습니다.
- 병렬 쿼리가 활성화된 경우 `INSERT` 문을 롤백하는 동안 데이터베이스가 다시 시작될 수 있는 문제를 해결했습니다.
- `EXTRA` 정보 열에 출력 `all select tables were optimized away`를 반환한 쿼리에서 `EXPLAIN ANALYZE` 프로파일링 도구를 실행할 때 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다. 자세한 내용은 MySQL 설명서의 EXPLAIN Output Format을 참조하세요.
- 전달된 암시적 커밋 문에 오류가 발생할 경우 전역 쓰기 전달을 사용하는 Aurora 글로벌 데이터베이스 보조 리전 리더 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- Aurora 글로벌 데이터베이스 보조 리전에서 글로벌 쓰기 전달을 사용하여 `SELECT FOR UPDATE` 쿼리를 실행할 때 Aurora 글로벌 데이터베이스 기본 리전의 라이터 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.

일반적인 개선 사항:

- `GTID_PURGED` 시스템 변수를 설정할 수 있도록 새 저장 프로시저 `mysql.rds_gtid_purged`를 추가했습니다. 자세한 내용은 mysql.rds_gtid_purged를 참조하세요.
- 바이너리 로그 복제를 중지할 위치를 구성할 수 있도록 두 개의 새로운 저장 프로시저 `mysql.rds_start_replication_until` 및 `mysql.rds_start_replication_until_gtid`를 추가했습니다. Aurora MySQL에서 바이너리 로그 복제의 중지 위치를 구성하는 방법에 대한 자세한 내용은 mysql.rds_start_replication_until을 참조하세요.
- 자동 커밋 모드가 비활성화된 세션에서 직접적으로 호출될 때 Aurora MySQL 복제 제어 저장 프로시저가 `sql_log_bin` 변수를 수정하지 못하는 문제를 해결했습니다.
- 데이터 제어 언어(DCL) 문 `GRANT/REVOKE` 및 `CREATE/DROP/ALTER/RENAME USER`에 대한 논리적 복제 지원을 추가했습니다.
- InnoDB 통계가 오래되어 쿼리 실행 계획이 최적화되지 않음으로 인해 쿼리 실행 시간이 길어질 수 있는 문제를 해결했습니다.

- `information_schema.aurora_global_db_instance_status` 및 `information_schema.aurora_global_db_status`라는 두 개의 새로운 시스템 뷰를 추가했습니다. 이러한 뷰를 사용하여 Aurora MySQL 글로벌 데이터베이스 클러스터에 있는 기본 및 보조 리소스의 상태와 토폴로지를 표시할 수 있습니다. 두 시스템 뷰에 대한 세부 정보는 Aurora MySQL 전용 information_schema 테이블에서 확인할 수 있습니다.
- 이스케이프된 와일드카드 문자를 사용하여 `SET ROLE` 문을 실행한 후 데이터베이스 이름에 와일드카드 문자가 포함된 데이터베이스에 액세스할 수 없는 문제를 해결했습니다.
- 감사 로그 교체를 처리하는 동안 보고된 이벤트가 감사 로그에 기록되지 않는 문제를 해결했습니다.
- `TRIGGER` 실행을 통해 내부 임시 테이블을 생성하면 라이터 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 새로운 시스템 변수 `innodb_aurora_max_partitions_for_range`를 추가했습니다. 영구 통계를 사용할 수 없는 경우 이 파라미터를 사용하여 파티셔닝된 테이블의 행 수 추정 실행 시간을 개선할 수 있습니다. 자세한 내용은 Aurora MySQL 구성 파라미터 문서에서 확인할 수 있습니다.
- 파티셔닝된 테이블을 생성할 때 `ROW_FORMAT`을 `COMPRESSED`로 잘못 설정할 수 있는 문제를 해결했습니다. Aurora MySQL이 압축된 테이블을 지원하지 않는다는 경고 메시지와 함께 테이블이 암시적으로 `COMPACT` 형식으로 변환됩니다.
- `replica_parallel_type` 변수가 `LOGICAL_CLOCK`으로 설정되고 `replica_preserve_commit_order` 변수가 `ON`으로 설정된 경우 멀티스레드 바이너리 로그 복제가 중지될 수 있는 문제를 해결했습니다. 이 문제는 500MB보다 큰 트랜잭션이 소스에서 실행될 때 발생할 수 있습니다.
- 글로벌 데이터베이스 쓰기 전달 기능이 활성화되면 보조 리전의 리더 인스턴스에 대한 `performance_schema` 구성 변경 사항이 의도치 않게 기본 리전의 라이터 인스턴스로 전달될 수 있는 문제를 해결했습니다.
- Aurora 스토리지 파일 시스템에서 데이터 페이지를 읽은 후 서버 상태 변수 `innodb_buffer_pool_reads`가 업데이트되지 않을 수 있는 문제를 해결했습니다.
- Aurora I/O 최적화 클러스터 구성을 선택하는 경우 Aurora MySQL 병렬 쿼리가 지원되지 않습니다. 자세한 내용은 Amazon Aurora MySQL 병렬 쿼리의 제한 사항을 참조하세요.
- 병렬 쿼리가 활성화되었을 때 쿼리 계획 최적화 프로그램이 기본 또는 보조 인덱스를 활용하는 특정 `SELECT` 쿼리에 대해 비효율적인 실행 계획을 선택하는 문제를 해결했습니다.
- 표준 시간대 정의를 IANA 2023c 버전으로 업그레이드했습니다.
- 릴레이 로그 파일에 쓸 때 경합을 줄일 수 있도록 binlog 복제본에 파일 관리 성능 최적화 기능을 도입했습니다.

- `information_schema.aurora_global_db_status` 테이블의 `RPO_LAG_IN_MILLISECONDS` 열과 `AuroraGlobalDBRPOLag` CloudWatch 지표가 사용자 워크로드에 관계없이 항상 0으로 표시되는 문제를 해결했습니다.
- 새 파라미터 `aurora_tmptable_enable_per_table_limit`를 도입했습니다. 이 파라미터가 활성화된 경우 `tmp_table_size` 변수가 TempTable 스토리지 엔진에서 생성된 개별 인메모리 내부 임시 테이블의 최대 크기를 정의합니다. 자세한 내용은 내부(암묵적) 임시 테이블에 대한 스토리지 엔진을 참조하세요.
- 글로벌 데이터베이스 쓰기 전달 기능이 활성화된 경우 추가 연결이 생성되는 문제를 해결했습니다. 이 문제는 리더 인스턴스의 읽기 전용 트랜잭션이 암시적 커밋을 라이터에 잘못 전달하는 경우 발생합니다.
- 글로벌 데이터베이스 쓰기 전달 기능을 사용하는 연결의 경우 기본 리전의 라이터에서 `performance_schema.threads` 테이블의 `PROCESSLIST_USER` 및 `PROCESSLIST_HOST` 필드가 채워지지 않는 문제를 해결했습니다. 이 테이블과 성능 스키마에 대한 자세한 내용은 MySQL Reference Manual의 The threads Table과 Amazon Aurora 사용 설명서의 성능 스키마 개요에서 확인할 수 있습니다.
- 글로벌 데이터베이스 쓰기 전달 기능을 사용할 때 `CommitLatency` CloudWatch 지표가 보조 리전의 리더 인스턴스에 대해 잘못된 값을 표시하는 문제를 해결했습니다. 보조 데이터베이스 클러스터에서 전달된 DML 문 지연 시간을 모니터링하려면 `ForwardingReplicaDMLLatency` 및 `ForwardingWriterDMLLatency` 지표를 사용하는 것이 좋습니다. 기본 리전 라이터 인스턴스의 `CommitLatency` 지표를 사용하여 커밋 지연 시간을 관찰할 수도 있습니다. 자세한 내용은 Aurora 사용 설명서의 쓰기 전달을 위한 Amazon CloudWatch 지표에서 확인할 수 있습니다.
- `replica_parallel_workers` 변수 값을 0보다 크게 설정하여 다중 스레드 바이너리 로그 복제를 구성할 때 바이너리 로그 복제를 관리 및 구성하는 데 사용되는 Aurora MySQL 복제 제어 저장 프로시저가 오류를 잘못 보고하는 문제를 해결했습니다.
- 여러 세션이 메모리에 없는 페이지에 액세스하려고 할 때 CPU 소비가 높아질 수 있는 문제를 해결했습니다.

업그레이드 및 마이그레이션:

- Aurora MySQL 버전 3.01, 3.02 또는 3.03에서 Aurora MySQL 버전 3.04 이상으로 마이너 버전 업그레이드를 수행하려면 엔진 버전을 수정하여 Aurora MySQL 업그레이드를 참조하세요.
- Aurora MySQL 2에서 Aurora MySQL 3으로 업그레이드할 때 `mysql.general_log_backup`, `mysql.general_log`, `mysql.slow_log_backup` 및 `mysql.slow_log` 테이블에 대해 보고된 스키마 불일치 오류로 인해 업그레이드 사전 확인이 실패할 수 있는 문제를 해결했습니다. 업그레이드 문제 해결에 대한 자세한 내용은 Aurora MySQL 버전 3의 업그레이드 문제 해결을 참조하세요.

- 트리거 정의에 따옴표로 묶이지 않은 예약된 키워드가 포함된 경우 Aurora MySQL 3으로 업그레이드할 때 메이저 버전 업그레이드가 실패할 수 있는 문제를 해결했습니다.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 아래 사항 외에도 8.0.28까지의 모든 커뮤니티 버그 수정이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 3.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그를 참조하세요.

- 고유 임시 테이블 페이지를 포함하는 버퍼 블록이 페이지 순회 중에 재배치되어 어설션 오류가 발생하는 문제를 해결했습니다. (버그 #33715694)
- InnoDB: 온라인 DDL 작업이 범위를 벗어난 메모리에 액세스하는 것을 방지합니다. (버그 #34750489, 버그 #108925)
- 여러 개의 중첩된 공통 테이블 표현식(CTE)으로 구성된 복잡한 SQL 문을 처리할 때 때때로 잘못된 쿼리 결과가 생성될 수 있는 문제를 해결했습니다. (버그 #34572040, 버그 #34634469, 버그 #33856374)

Aurora MySQL 데이터베이스 엔진 업데이트 2023년 12월 8일(버전 3.03.3)(사용되지 않음)

버전: 3.03.3

Aurora MySQL 3.03.3이 정식 출시되었습니다. Aurora MySQL 3.03 버전은 MySQL 8.0.26과 호환됩니다. 8.0.23에서 8.0.28까지 이루어진 커뮤니티 변경 사항에 대한 자세한 내용은 MySQL 8.0 Release Notes를 참조하세요.

Aurora MySQL 버전 3의 새로운 기능에 대한 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환을 참조하세요. Aurora MySQL 버전 3과 Aurora MySQL 버전 2의 차이점에 대한 자세한 내용은 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요. Aurora MySQL 버전 3과 MySQL 8.0 Community Edition의 비교 내용은 Aurora MySQL 버전 3과 MySQL 8.0 Community Edition 비교를 참조하세요.

현재 사용 가능한 Aurora MySQL 릴리스는 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.*, 3.04.*, 3.05.*입니다.

Amazon RDS 블루/그린 배포를 사용하여 현재 사용 가능한 Aurora MySQL 버전 2 클러스터에서 Aurora MySQL 버전 3.03.3 클러스터로 인플레이스 업그레이드를 수행하거나, 스냅샷을 복원하거나, 관리형 블루/그린 업그레이드를 시작할 수 있습니다.

Aurora MySQL 버전 3으로의 업그레이드 계획에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3에 대한 업그레이드 계획을 참조하세요. Aurora MySQL 업그레이드에 대한 일반 정보는 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터 업그레이드를 참조하세요.

문제 해결 정보는 Aurora MySQL 버전 3의 업그레이드 문제 해결을 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

아래 나열된 보안 문제 및 CVE 해결:

관리된 환경에서 처리를 미세 조정하기 위한 수정 사항 및 기타 향상된 기능 추가 CVE 수정은 다음과 같습니다.

- CVE-2023-38545

가용성 향상:

- 병렬 쿼리를 사용하는 Aurora MySQL 데이터베이스 인스턴스에서 많은 수의 동시 병렬 쿼리를 실행할 때 데이터베이스가 다시 시작될 수 있는 문제를 해결했습니다.
- binlog 소스에 `gtid_mode`가 `ON` 또는 `ON_PERMISSIVE`로 설정된 경우 향상된 binlog가 활성화된 바이너리 로그(binlog) 복제본 클러스터에서 실행된 GTID 세트가 잘못 복구될 수 있는 문제를 해결했습니다. 이 문제로 인해 복구 중에 복제본 클러스터의 라이터 인스턴스가 추가로 다시 시작되거나 실행된 GTID 세트를 쿼리할 때 잘못된 결과가 생성될 수 있습니다.
- 향상된 바이너리 로그가 활성화된 경우 사용 가능한 메모리 감소로 인해 Aurora MySQL 데이터베이스 인스턴스가 다시 시작되거나 장애 조치가 발생할 수 있는 메모리 관리 문제를 해결했습니다.
- 라이터 인스턴스에서 데이터베이스 볼륨이 160GB의 배수로 증가하면 리더 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 바이너리 로그 복구 프로세스가 실행되는 동안 향상된 바이너리 로그 기능이 활성화된 Aurora MySQL 데이터베이스 인스턴스가 시작될 경우 시작 도중 멈출 수 있는 문제를 해결했습니다.
- 제로 가동 중지 패치 적용 중에 인스턴스가 다시 시작되어 데이터베이스 연결이 예기치 않게 종료되는 문제를 해결했습니다.

- `SHOW STATUS` 및 `PURGE BINARY LOGS` 문을 동시에 실행할 때 교착 상태로 인해 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다. purge binary logs는 사용자가 구성한 binlog 보존 기간을 준수하기 위해 실행되는 관리형 문입니다.
- Aurora 복제본이 있는 클러스터에서 향상된 binlog 기능을 사용할 때 긴 세마포어 대기 시간 때문에 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.

일반적인 개선 사항

- 향상된 binlog 기능이 활성화된 경우 Aurora 스토리지에 쓰기 전에 사용되지 않은 스토리지 메타데이터를 제거했습니다. 이렇게 하면 네트워크를 통해 전송되는 바이트 증가에 따른 쓰기 지연 시간 증가로 인해 데이터베이스 다시 시작 또는 장애 조치가 발생할 수 있는 특정 상황이 방지됩니다.
- 향상된 binlog가 활성화된 경우 CloudWatch의 `NumBinaryLogFiles` 지표에 잘못된 결과가 표시될 수 있는 문제를 해결했습니다.
- 데이터베이스 인스턴스를 다시 시작할 때까지 `table_open_cache` 데이터베이스 파라미터의 수정 사항이 적용되지 않는 문제를 해결했습니다.
- 연결된 바이너리 로그(binlog) 소비자가 중복된 binlog 복제 서버 ID를 사용할 때 데이터베이스가 다시 시작될 수 있는 문제를 해결했습니다.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 아래 사항 외에도 8.0.26까지의 모든 커뮤니티 버그 수정이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 3.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그를 참조하세요.

- 백그라운드 TLS 인증서 교체로 인해 CPU 사용률이 높아질 수 있는 문제를 해결했습니다(커뮤니티 버그 수정 #34284186).

Aurora MySQL 데이터베이스 엔진 업데이트 2023년 8월 29일(버전 3.03.2)(사용되지 않음)

버전: 3.03.2

Aurora MySQL 3.03.2가 정식 출시되었습니다. Aurora MySQL 3.04 버전은 MySQL 8.0.28과 호환되고, Aurora MySQL 3.03 버전은 MySQL 8.0.26과 호환되며, Aurora MySQL 3.02 버전은 MySQL 8.0.23과 호환됩니다. 8.0.23에서 8.0.28까지 이루어진 커뮤니티 변경 사항에 대한 자세한 내용은 MySQL 8.0 Release Notes를 참조하세요.

Aurora MySQL 버전 3의 새로운 기능에 대한 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환을 참조하세요. Aurora MySQL 버전 3과 Aurora MySQL 버전 2의 차이점에 대한 자세한 내용은 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요. Aurora MySQL 버전 3과 MySQL 8.0 Community Edition의 비교 내용은 Aurora MySQL 버전 3과 MySQL 8.0 Community Edition 비교를 참조하세요.

현재 사용 가능한 Aurora MySQL 릴리스는 2.07.9, 2.07.10, 2.11.*, 3.01.*, 3.02.*, 3.03.* 및 3.04.*입니다.

Amazon RDS 블루/그린 배포를 사용하여 현재 사용 가능한 Aurora MySQL 버전 2 클러스터에서 Aurora MySQL 버전 3.03.2 클러스터로 인플레이스 업그레이드를 수행하거나, 스냅샷을 복원하거나, 관리형 블루/그린 업그레이드를 시작할 수 있습니다.

Aurora MySQL 버전 3으로의 업그레이드 계획에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3에 대한 업그레이드 계획을 참조하세요. Aurora MySQL 업그레이드에 대한 일반 정보는 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터 업그레이드를 참조하세요.

문제 해결 정보는 Aurora MySQL 버전 3의 업그레이드 문제 해결을 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

보안 문제 및 CVE 해결:

- 감사 로그 파일 교체 중에 감사 로그에서 이벤트가 누락될 수 있는 문제를 해결했습니다.

이 릴리스에는 다음 CVE 수정 사항이 포함되어 있습니다.

- CVE-2023-21963
- CVE-2023-21912
- CVE-2023-0215
- CVE-2022-43551
- CVE-2022-37434

가용성 향상:

- 긴 트랜잭션 복구 중에 데이터베이스가 다시 시작될 수 있는 문제를 해결했습니다.
- 데이터베이스가 내부 시스템 테이블에서 트리거를 생성하거나 삭제하는 동안 라이터 인스턴스가 다시 시작되면 데이터베이스 클러스터를 사용할 수 없게 되는 문제를 해결했습니다.
- 집계 함수를 참조하는 쿼리를 실행하는 동안 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 병렬 쿼리가 활성화된 경우 `INSERT` 문을 롤백하는 동안 데이터베이스가 다시 시작될 수 있는 문제를 해결했습니다.
- 빠른 입력은 Aurora MySQL 버전 3.03.2 이상에서 일반 InnoDB 테이블에 대해서만 활성화됩니다. InnoDB 임시 테이블에서는 이 최적화가 작동하지 않습니다. 빠른 삽입 최적화에 대한 자세한 내용은 Amazon Aurora MySQL 성능 향상을 참조하세요.

일반적인 개선 사항

- `ERROR 1146`으로 인해 리더 인스턴스가 테이블을 열 수 없는 문제를 해결했습니다. 이 문제는 라이터 인스턴스에서 `INPLACE` 알고리즘을 사용하는 동안 특정 유형의 온라인 데이터 정의 언어(DDL)를 실행할 때 발생합니다.
- 릴레이 로그 파일에 쓸 때 경합을 줄일 수 있도록 binlog 복제본에 파일 관리 성능 최적화 기능을 도입했습니다.
- 병렬 쿼리가 활성화되었을 때 쿼리 계획 최적화 프로그램이 기본 또는 보조 인덱스를 활용하는 특정 `SELECT` 쿼리에 대해 비효율적인 실행 계획을 선택하는 문제를 해결했습니다.
- 데이터 제어 언어(DCL) 문 `GRANT/REVOKE` 및 `CREATE/DROP/ALTER/RENAME USER`에 대한 논리적 복제 지원을 추가했습니다.
- Aurora I/O 최적화 클러스터 구성을 선택하는 경우 Amazon Aurora MySQL에 대한 병렬 쿼리가 지원되지 않습니다. 자세한 내용은 Aurora MySQL 병렬 쿼리의 제한 사항을 참조하세요.

업그레이드 및 마이그레이션:

- Aurora MySQL 버전 3.01 또는 3.02에서 Aurora MySQL 버전 3.03 이상으로 마이너 버전 업그레이드를 수행하려면 엔진 버전을 수정하여 Aurora MySQL 업그레이드를 참조하세요.
- 트리거 정의에 따옴표로 묶이지 않은 예약된 키워드가 포함된 경우 Aurora MySQL 버전 3으로 업그레이드할 때 메이저 버전 업그레이드가 실패할 수 있는 문제를 해결했습니다.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 아래 사항 외에도 8.0.26까지의 모든 커뮤니티 버그 수정이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 3.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그를 참조하세요.

- 여러 개의 중첩된 공통 테이블 표현식(CTE)으로 구성된 복잡한 SQL 문을 처리할 때 때때로 잘못된 쿼리 결과가 생성될 수 있는 문제를 해결했습니다. (버그 #34572040, 버그 #34634469, 버그 #33856374)
- InnoDB: 동일한 테이블에 대한 통계를 초기화 해제하고 초기화하려는 스레드 간의 경합 상태로 인해 어설션 오류가 발생했습니다. (버그 #33135425)
- InnoDB: 온라인 DDL 작업이 범위를 벗어난 메모리에 액세스하는 것을 방지합니다. (버그 #34750489, 버그 #108925)

Aurora MySQL 데이터베이스 엔진 업데이트 2023년 5월 11일(버전 3.03.1)(사용되지 않음)

버전: 3.03.1

Aurora MySQL 3.03.1이 정식 출시되었습니다. Aurora MySQL 3.03 버전은 MySQL 8.0.26과 호환 가능하고, Aurora MySQL 3.02 버전은 MySQL 8.0.23과 호환됩니다. 8.0.23에서 8.0.26까지 이루어진 커뮤니티 변경 사항에 대한 자세한 내용은 MySQL 8.0 Release Notes를 참조하세요.

Aurora MySQL 버전 3의 새로운 기능에 대한 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환을 참조하세요. Aurora MySQL 버전 3과 Aurora MySQL 버전 2의 차이점에 대한 자세한 내용은 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요. Aurora MySQL 버전 3과 MySQL 8.0 Community Edition의 비교 내용은 Aurora MySQL 버전 3과 MySQL 8.0 Community Edition 비교를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 2.07.9, 2.11.1, 2.11.2, 3.01.*, 3.02.* 및 3.03.*입니다.

현재 지원되는 Aurora MySQL 버전 2 클러스터에서 Aurora MySQL 3.03.1로 인플레이스 업그레이드를 수행하거나 스냅샷을 복원할 수 있습니다.

Aurora MySQL 버전 3으로의 업그레이드 계획에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3에 대한 업그레이드 계획을 참조하세요. Aurora MySQL 업그레이드에 대한 일반 정보는 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터 업그레이드를 참조하세요.

문제 해결 정보는 Aurora MySQL 버전 3의 업그레이드 문제 해결을 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

새로운 기능:

- 버전 3.03.1부터 Aurora 최적화 I/O 스토리지 구성을 사용할 수 있습니다. 자세한 내용은 Amazon Aurora DB 클러스터 스토리지 구성을 참조하세요.
- 새로운 시스템 변수 `innodb_aurora_max_partitions_for_range`를 추가했습니다. 영구 통계를 사용할 수 없는 경우 이 파라미터를 사용하여 파티셔닝된 테이블의 행 수 추정 실행 시간을 개선할 수 있습니다. 자세한 내용은 Aurora MySQL 구성 파라미터 문서에서 확인할 수 있습니다.

가용성 향상:

- 트랜잭션을 커밋한 직후 연결이 종료될 때 잘못된 메모리에 잘못 액세스하여 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 서버 변수 `server_audit_events`가 ALL 또는 QUERY로 설정된 경우 Aurora MySQL 오류 로그에 정보 메시지가 과도하게 로깅되는 Aurora 고급 감사 관련 문제를 해결했습니다. 이 문제로 인해 데이터베이스 인스턴스가 다시 시작될 수 있습니다.
- 특정 상황에서 범위 추정 중에 더 이상 액세스할 수 없는 페이지를 읽으려고 할 때 Aurora 리더 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- Aurora 병렬 쿼리 실행 계획을 사용하는 쿼리를 실행하는 동안 Aurora MySQL 리더 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 여러 바이너리 로그 복제 소비자가 연결된 경우 바이너리 로그 복제를 사용하는 데이터베이스 인스턴스에서 CPU 사용률이 증가하고 연결 실패가 발생할 수 있는 문제를 해결했습니다.
- 중간 임시 테이블을 구체화하는 동안 공통 테이블 표현식(CTE)에 지원되지 않는 인덱스 스캔 액세스 방법이 고려되어 데이터베이스 다시 시작 또는 잘못된 쿼리 결과를 비롯한 원치 않는 동작이 발생할 수 있는 문제를 해결했습니다. 이 문제는 TempTable 스토리지 엔진을 사용하여 테이블에서 지원되지 않는 인덱스 스캔 액세스 방법의 사용을 방지함으로써 해결되었습니다.
- `INSERT INTO SELECT` 및 `FROM`과 같은 쿼리를 실행할 때 불일치를 일으킬 수 있는 문제로 인해 이 Aurora MySQL 버전에서는 빠른 삽입이 활성화되지 않습니다. 빠른 삽입 최적화에 대한 자세한 내용은 Amazon Aurora MySQL 성능 향상을 참조하세요.

일반적인 개선 사항

- SHOW BINARY LOGS 문의 실행 시간이 예상보다 길어질 수 있는 문제를 해결했습니다. 이로 인해 데이터베이스의 커밋 처리량이 감소할 수 있습니다.
- 즉각적인 ADD COLUMN 기능을 사용하여 추가된 열이 있는 사용자 테이블의 병렬 내보내기가 실패할 수 있는 문제를 해결했습니다.
- 감사 로그 교체를 처리하는 동안 보고된 이벤트가 감사 로그에 기록되지 않는 문제를 해결했습니다.
- INFORMATION_SCHEMA INNODB_TABLESPACES 테이블에 대해 쿼리를 실행할 때 사용 가능한 메모리가 고갈될 수 있는 문제를 해결했습니다.
- 파티셔닝된 테이블을 생성할 때 ROW_FORMAT을 COMPRESSED로 잘못 설정할 수 있는 문제를 해결했습니다. Aurora MySQL이 압축된 테이블을 지원하지 않는다는 경고 메시지와 함께 테이블이 암시적으로 COMPACT 형식으로 변환됩니다.

업그레이드 및 마이그레이션:

- Aurora MySQL 버전 3.01 또는 3.02에서 Aurora MySQL 버전 3.03 이상으로 마이너 버전 업그레이드를 수행하려면 엔진 버전을 수정하여 Aurora MySQL 업그레이드를 참조하세요.
- Aurora MySQL 2에서 Aurora MySQL 3으로 업그레이드할 때 `mysql.general_log_backup`, `mysql.general_log`, `mysql.slow_log_backup` 및 `mysql.slow_log` 테이블에 대해 보고된 스키마 불일치 오류로 인해 업그레이드 사전 확인이 실패할 수 있는 문제를 해결했습니다. 업그레이드 문제 해결에 대한 자세한 내용은 Aurora MySQL 버전 3의 업그레이드 문제 해결을 참조하세요.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 아래 사항 외에도 8.0.26까지의 모든 커뮤니티 버그 수정이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 3.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그를 참조하세요.

- 고유 임시 테이블 페이지를 포함하는 버퍼 블록이 페이지 순회 중에 재배치되어 어설션 오류가 발생하는 문제를 해결했습니다. (버그 #33715694)

Aurora MySQL 데이터베이스 엔진 업데이트 2023년 3월 1일(버전 3.03.0)(사용되지 않음)

버전: 3.03.0

Aurora MySQL 3.03.0이 정식 출시되었습니다. Aurora MySQL 3.03 버전은 MySQL 8.0.26과 호환 가능하고, Aurora MySQL 3.02 버전은 MySQL 8.0.23과 호환됩니다. 8.0.23에서 8.0.26까지 이루어진 커뮤니티 변경 사항에 대한 자세한 내용은 MySQL 8.0 Release Notes를 참조하세요.

Aurora MySQL 버전 3의 새로운 기능에 대한 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환을 참조하세요. Aurora MySQL 버전 3과 Aurora MySQL 버전 2의 차이점에 대한 자세한 내용은 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요. Aurora MySQL 버전 3과 MySQL 8.0 Community Edition의 비교 내용은 Aurora MySQL 버전 3과 MySQL 8.0 Community Edition 비교를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 2.07.*, 2.11.*, 3.01.*, 3.02.* 및 3.03.*입니다.

현재 지원되는 Aurora MySQL 버전 2 클러스터에서 Aurora MySQL 3.03.0으로 인플레이스 업그레이드를 수행하거나 스냅샷을 복원할 수 있습니다.

Aurora MySQL 버전 3으로의 업그레이드 계획에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3에 대한 업그레이드 계획을 참조하세요. Aurora MySQL 업그레이드에 대한 일반 정보는 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터 업그레이드를 참조하세요.

문제 해결 정보는 Aurora MySQL 버전 3의 업그레이드 문제 해결을 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

아래 나열된 보안 문제 및 CVE 해결:

관리된 환경에서 처리를 미세 조정하기 위한 수정 사항 및 기타 향상된 기능 아래의 추가 CVE 수정:

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

가용성 향상:

- 버퍼 풀 초기화 시간이 예상보다 오래 걸려 다시 시작 시 더 큰 DB 인스턴스 클래스에서 오류가 발생할 수 있는 문제를 해결했습니다.

- 바이너리 로깅이 활성화된 경우 데이터베이스 복구 프로세스 중에 DB 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- GRANT 및 REVOKE 같은 데이터 제어 언어(DCL) 문을 실행하거나 라이터 인스턴스에서 새 연결을 설정하는 동안 리더 인스턴스에서 연결 실패가 발생할 수 있는 문제를 해결했습니다.
- 현재 지원되지 않는 DELETE 및 UPDATE 문과 같은 데이터 조작 언어(DML) 작업에 병렬 쿼리가 잘못 사용되어 DB 인스턴스가 다시 시작되는 문제를 해결했습니다. 병렬 쿼리에서 지원되는 작업에 대한 자세한 내용은 Aurora MySQL 병렬 쿼리의 제한 사항을 참조하세요.
- 드문 경우지만 라이터 인스턴스의 대규모 업데이트 작업 또는 데이터 정의 언어(DDL) 워크로드와 Aurora 복제본의 동일한 테이블 세트에 대한 읽기 작업을 동시에 실행하는 동안 Aurora 복제본이 다시 시작될 수 있는 문제를 해결했습니다.
- Aurora Serverless v2 리더 인스턴스 스케일 다운 작업에서 리더 인스턴스가 다시 시작되고 드문 경우지만 데이터 불일치가 발생할 수 있는 문제를 해결했습니다.
- DB 인스턴스에 대한 연결이 종료될 때 잘못된 메모리 위치에 잘못 액세스하여 DB 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 드문 경우지만 10진수 열을 소수점 이하 0자리로 자르는 GROUP BY 절을 사용하여 쿼리를 처리할 때 DB 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 공간 인덱스를 사용하여 범위 쿼리를 실행할 때 레코드에 잘못 액세스하여 DB 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 내부 임시 테이블이 기본값 또는 고객이 구성한 메모리 또는 mmap 값을 초과하는 경우 Aurora MySQL 복제본 인스턴스에서 DB 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 고급 감사 로그 교체로 인해 메모리 관리 오류가 발생할 수 있는 문제를 해결했습니다.
- INSERT INTO SELECT 및 FROM과 같은 쿼리를 실행할 때 불일치를 일으킬 수 있는 문제로 인해 이 Aurora MySQL 버전에서는 빠른 삽입이 활성화되지 않습니다. 빠른 삽입 최적화에 대한 자세한 내용은 Amazon Aurora MySQL 성능 향상을 참조하세요.

일반적인 개선 사항

- GLOBAL 읽기 일관성 설정을 사용하여 글로벌 데이터베이스 쓰기 전달 세션의 읽기 쿼리 지연 시간을 개선했습니다.
- 클라이언트 세션에서 `reset_connection` 또는 `change_user` 명령을 실행한 후 `wait_timeout` 파라미터 값이 적용되지 않는 문제를 해결했습니다.
- 인스턴스의 수신 연결이 갑자기 증가할 때 DB 인스턴스에 연결하면서 애플리케이션의 지연 시간이 길어지는 문제를 해결했습니다. Aurora MySQL 데이터베이스 인스턴스의 연결 설정 지연 문제를 해결하는 데 도움이 되는 두 가지 새로운 Amazon CloudWatch 지표 `AuroraSlowHandshakeCount`

및 AuroraSlowConnectionHandleCount를 도입했습니다. 이러한 지표에 대한 자세한 내용은 Aurora CloudWatch 지표 정의 문서, Amazon Aurora에 대한 Amazon CloudWatch 지표에서 확인할 수 있습니다.

- temptable_use_mmap 파라미터는 사용이 중단되었으며 향후 MySQL 릴리스에서 지원이 제거될 예정입니다. 자세한 내용은 내부(암묵적) 임시 테이블에 대한 스토리지 엔진을 참조하세요.
- SHOW BINARY LOGS 문의 실행 시간이 예상보다 길어질 수 있는 문제를 해결했습니다. 이로 인해 데이터베이스의 커밋 처리량이 감소할 수 있습니다.

업그레이드 및 마이그레이션:

- Aurora MySQL 버전 3.01 또는 3.02에서 Aurora MySQL 버전 3.03 이상으로 마이너 버전 업그레이드를 수행하려면 엔진 버전을 수정하여 Aurora MySQL 업그레이드를 참조하세요.
- 클러스터에 많은 테이블(75만 개 초과)이 있는 경우 Aurora MySQL 버전 2에서 Aurora MySQL 버전 3으로의 메이저 버전 업그레이드가 실패할 수 있는 문제를 해결했습니다.
- mysql.innodb_table_stats 및 mysql.innodb_index_stats 테이블을 마이그레이션하는 데 예상보다 시간이 오래 걸려 Aurora MySQL 버전 2에서 Aurora MySQL 버전 3으로의 메이저 버전 업그레이드가 실패할 수 있는 문제를 해결했습니다. 이 문제는 주로 테이블이 수백만 개 있는 DB 클러스터에 영향을 미쳤습니다.
- 스키마 불일치 오류로 인해 Aurora MySQL 버전 2에서 Aurora MySQL 버전 3으로의 업그레이드가 실패할 수 있는 문제를 해결했습니다. 이러한 오류는 mysql.general_log_template 및 mysql.slow_log_template 테이블에 대한 업그레이드 사전 검사기에서 보고됩니다. 업그레이드 문제 해결에 대한 자세한 내용은 Aurora MySQL 버전 3의 업그레이드 문제 해결을 참조하세요.
- schemaInconsistencyCheck 오류로 인해 Aurora MySQL 버전 2에서 Aurora MySQL 버전 3으로의 업그레이드가 실패할 수 있는 문제를 해결했습니다. 이 오류는 upgrade-prechecks.log에 보고되는 mysql.table_migration_index_info 테이블 내 스키마 불일치로 인해 발생합니다. Aurora MySQL version 3으로의 업그레이드 문제 해결에 대한 자세한 내용은 Aurora MySQL 버전 3의 업그레이드 문제 해결을 참조하세요.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 아래 사항 외에도 8.0.26까지의 모든 커뮤니티 버그 수정이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 3.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그를 참조하세요.

- JSON 및 TEXT를 비롯한 일부 열 유형의 크기가 정렬에서 가장 큰 행의 크기보다 15배 이상 크지 않은 경우 정렬 버퍼가 소진되는 문제를 해결했습니다. 이제 정렬 버퍼의 크기가 가장 큰 정렬 키의 15배이기만 하면 됩니다. (버그 #103325, 버그 #105532, 버그 #32738705, 버그 #33501541)
- InnoDB가 경우에 따라 테이블 파티션의 일부 정식 이름을 올바르게 처리하지 않는 문제를 해결했습니다. (버그 #32208630)
- OR 조건을 사용한 쿼리를 실행할 때 null 허용 여부 속성의 부정확한 계산으로 인해 특정 상황에서 잘못된 결과가 반환될 수 있는 문제를 해결했습니다. (버그 #34060289)
- 다음 두 조건이 충족될 때 특정 상황에서 잘못된 결과가 반환될 수 있는 문제를 해결했습니다.
  - 파생 테이블이 외부 쿼리 블록에 병합됨
  - 쿼리에 왼쪽 조인과 IN 하위 쿼리가 포함됨

  (버그 #34060289)
- 최대 정수 열 값을 초과할 때 잘못된 AUTO_INCREMENT 값이 생성되는 문제를 해결했습니다. 이 오류는 최대 열 값을 고려하지 않았기 때문에 발생했습니다. 이 경우 이전의 유효한 AUTO_INCREMENT 값이 반환되어 중복 키 오류가 발생합니다. (버그 #87926, 버그 #26906787)
- 성능 스키마에 대한 DROP 권한을 취소할 수 없는 문제를 해결했습니다. (버그 #33578113)
- 실행 사이에 삭제되었다가 다시 생성된 하나 이상의 테이블에 대해 작동하며 EXISTS를 사용하는 IF 문이 포함된 저장 프로시저가 첫 번째 간접 호출 이후 후속 간접 호출에서 올바르게 실행되지 않는 문제를 해결했습니다. (버그 #32855634)
- 하위 쿼리와 외부 쿼리 블록의 뷰를 참조하는 쿼리가 예기치 않게 다시 시작되는 문제를 해결했습니다. (버그 #32324234)

Aurora MySQL 데이터베이스 엔진 업데이트 2023년 4월 17일(버전 3.02.3)(사용되지 않음)

버전: 3.02.3

Aurora MySQL 3.02.3이 정식 출시되었습니다. Aurora MySQL 3.02 버전은 MySQL 8.0.23과 호환 가능하고, Aurora MySQL 2.x 버전은 MySQL 5.7과 호환됩니다.

Aurora MySQL 버전 3의 새로운 기능과 Aurora MySQL 버전 3과 Aurora MySQL 버전 2 또는 커뮤니티 MySQL 8.0의 차이점에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 2.07.*, 2.11.1, 2.11.2, 3.01.*, 3.02.* 및 3.03.*입니다.

현재 지원되는 Aurora MySQL 버전 2 클러스터에서 Aurora MySQL 3.02.3으로 인플레이스 업그레이드를 수행하거나 스냅샷을 복원할 수 있습니다.

Aurora MySQL 버전 3으로의 업그레이드 계획에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3에 대한 업그레이드 계획을 참조하세요. 업그레이드 절차 자체는 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3으로 업그레이드를 참조하세요. Aurora MySQL 업그레이드에 대한 일반 정보는 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터 업그레이드를 참조하세요.

문제 해결 정보는 Aurora MySQL 버전 3의 업그레이드 문제 해결을 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

가용성 향상:

- 트랜잭션을 커밋한 직후 연결이 종료될 때 잘못된 메모리에 잘못 액세스하여 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- `INSERT INTO SELECT` 및 `FROM`과 같은 쿼리를 실행할 때 불일치를 일으킬 수 있는 문제로 인해 이 Aurora MySQL 버전에서는 빠른 삽입이 활성화되지 않습니다. 빠른 삽입 최적화에 대한 자세한 내용은 Amazon Aurora MySQL 성능 향상을 참조하세요.

일반적인 개선 사항

- 중간 임시 테이블을 구체화하는 동안 공통 테이블 표현식(CTE)에 지원되지 않는 인덱스 스캔 액세스 방법이 고려되어 데이터베이스 다시 시작 또는 잘못된 쿼리 결과를 비롯한 원치 않는 동작이 발생할 수 있는 문제를 해결했습니다. 이 문제는 TempTable 스토리지 엔진을 사용하여 테이블에서 지원되지 않는 인덱스 스캔 액세스 방법의 사용을 방지함으로써 해결되었습니다.
- 드문 경우지만 Aurora MySQL 라이터 인스턴스에서 대규모 업데이트 또는 데이터 정의 언어(DDL) 작업이 동시에 실행되는 테이블에 액세스할 때 Aurora MySQL 리더 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 특정 상황에서 범위 추정 중에 더 이상 액세스할 수 없는 페이지를 읽으려고 할 때 Aurora MySQL 리더 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 여러 바이너리 로그 복제 소비자가 연결된 경우 바이너리 로그 복제를 사용하는 데이터베이스 인스턴스에서 CPU 사용률이 증가하고 연결 실패가 발생할 수 있는 문제를 해결했습니다.

- Aurora 병렬 쿼리 실행 계획을 사용하는 쿼리를 실행하는 동안 Aurora MySQL 리더 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.

Aurora MySQL 데이터베이스 엔진 업데이트 2022년 11월 18일(버전 3.02.2)(사용되지 않음)

버전: 3.02.2

Aurora MySQL 3.02.2가 정식 출시되었습니다. Aurora MySQL 3.02 버전은 MySQL 8.0.23과 호환되고, Aurora MySQL 2.x 버전은 MySQL 5.7과 호환되며, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

Aurora MySQL 버전 3의 새로운 기능과 Aurora MySQL 버전 3과 Aurora MySQL 버전 2 또는 커뮤니티 MySQL 8.0의 차이점에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

현재 지원되는 Aurora MySQL 버전 2 클러스터에서 Aurora MySQL 3.02.2로 스냅샷을 복원할 수 있습니다.

Aurora MySQL 버전 3으로의 업그레이드 계획에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3에 대한 업그레이드 계획을 참조하세요. 업그레이드 절차 자체는 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3으로 업그레이드를 참조하세요. Aurora MySQL 업그레이드에 대한 일반 정보는 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터 업그레이드를 참조하세요.

문제 해결 정보는 Aurora MySQL 버전 3의 업그레이드 문제 해결을 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

Aurora MySQL 버전 3.02.2는 정식 출시되었으며 일반적으로 커뮤니티 MySQL 8.0.23과 호환됩니다.

아래 나열된 보안 문제 및 CVE 해결:

관리된 환경에서 처리를 미세 조정하기 위한 수정 사항 및 기타 향상된 기능 아래의 추가 CVE 수정:

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

- CVE-2022-21444

가용성 향상:

- 데이터베이스 인스턴스에 대한 연결이 명시적으로 또는 암시적으로 종료될 때 잘못된 메모리에 잘못 액세스하여 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 대규모 인스턴스 클래스에서 버퍼 풀 초기화가 예상보다 오래 걸려 데이터베이스 시작이 반복적으로 중단될 수 있는 문제를 해결했습니다.
- 드문 경우지만 Aurora Serverless v2가 규모 조정 중에 테이블 캐시 업데이트를 잘못 시도하면 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 드문 경우지만 10진수 열을 소수점 이하 0자리로 자르는 GROUP BY 절을 사용하여 쿼리를 처리할 때 데이터베이스가 다시 시작될 수 있는 문제를 해결했습니다.
- `INSERT INTO SELECT` 및 `FROM`과 같은 쿼리를 실행할 때 불일치를 일으킬 수 있는 문제로 인해 이 Aurora MySQL 버전에서는 빠른 삽입이 활성화되지 않습니다. 빠른 삽입 최적화에 대한 자세한 내용은 Amazon Aurora MySQL 성능 향상을 참조하세요.

일반적인 개선 사항

- mysql.host 테이블의 메타데이터 불일치로 인해 Aurora MySQL 버전 2(MySQL 5.7과 호환)에서 Aurora MySQL 버전 3(MySQL 8.0과 호환)으로의 업그레이드가 실패할 수 있는 문제를 해결했습니다.
- Aurora MySQL 버전 2(MySQL 5.7과 호환)에서 Aurora MySQL 버전 3(MySQL 8.0과 호환)으로의 업그레이드 시간을 단축하는 성능 개선 사항을 추가했습니다. 특정 업그레이드 단계를 병렬화하면 db.r6g.16xlarge 또는 db.r5.24xlarge와 같은 더 큰 인스턴스 클래스를 사용할 때 시간이 더욱 단축됩니다.
- 이전 버전의 경우 50개 오류만 표시하도록 제한되었는데 Aurora MySQL 버전 2(MySQL 5.7과 호환)에서 Aurora MySQL 버전 3(MySQL 8.0과 호환)으로 업그레이드할 때 모든 오류를 표시하도록 지원을 추가했습니다.

- 드문 경우지만 Aurora MySQL 버전 2(MySQL 5.7과 호환)에서 Aurora MySQL 버전 3(MySQL 8.0과 호환)으로 메이저 버전을 업그레이드한 후 자동 증가 카운터가 잘못될 수 있는 문제를 해결했습니다.
- 'mysql.innodb_table_stats' 및 'mysql.innodb_index_stats' 테이블을 마이그레이션하는 데 예상보다 시간이 오래 걸리기 때문에 Aurora MySQL 버전 2에서 Aurora MySQL 버전 3으로의 메이저 버전 업그레이드가 실패할 수 있는 문제를 해결했습니다. 이 문제는 주로 테이블 수가 많은(150만 개 초과) 데이터베이스 클러스터에 영향을 미쳤습니다.
- AMS 8.0 엔진 업그레이드 워크플로의 결함으로 인해 Aurora MySQL 버전 2에서 Aurora MySQL 버전 3으로의 메이저 버전 업그레이드가 실패하여 로그 레코드가 Aurora 스토리지 클러스터 볼륨에 누적되고 정상적인 쓰기 작업이 중지될 수 있는 문제를 해결했습니다. 이 문제는 주로 테이블 수가 많은(대략 75만 개 초과) 데이터베이스 클러스터에 영향을 미쳤습니다.
- MySQL 제거 스레드가 활성 상태로 잘못 유지되어 Aurora MySQL Serverless v2 유휴 인스턴스가 0.5 ACU로 축소되지 않는 문제를 해결했습니다.
- 인스턴스의 수신 연결이 갑자기 증가할 때 데이터베이스 인스턴스에 연결하면서 애플리케이션의 지연 시간이 길어지는 문제를 해결했습니다.
- Aurora MySQL 데이터베이스 인스턴스의 연결 설정 지연 문제를 해결하는 데 도움이 되는 두 가지 새로운 Amazon CloudWatch 지표를 도입했습니다. AuroraSlowHandshakeCount 및 AuroraSlowConnectionHandleCount 지표에 대한 자세한 내용은 Aurora CloudWatch 지표 정의에서 확인할 수 있습니다.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 아래 사항 외에도 8.0.23까지의 모든 커뮤니티 버그 수정이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 3.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그를 참조하세요.

- OR 조건을 사용한 쿼리를 실행할 때 null 허용 여부 속성의 부정확한 계산으로 인해 특정 상황에서 잘못된 결과가 반환될 수 있는 문제를 해결했습니다. (버그 #34060289)
- 다음 두 조건이 충족될 때 특정 상황에서 잘못된 결과가 반환될 수 있는 문제를 해결했습니다.
  - 파생 테이블이 외부 쿼리 블록에 병합되어 있습니다.
  - 쿼리에 왼쪽 조인과 IN 하위 쿼리가 포함되어 있습니다. (버그 #34060289)
- 성능 스키마에 대한 DROP 권한을 취소할 수 없는 문제를 해결했습니다. (버그 #33578113)
- 실행 사이에 삭제되었다가 다시 생성된 하나 이상의 테이블에 대해 작동하며 EXISTS를 사용하는 IF 문이 포함된 저장 프로시저가 첫 번째 간접 호출 이후 후속 간접 호출에서 올바르게 실행되지 않는 문제를 해결했습니다. (MySQL 버그 #32855634)

- 최대 정수 열 값을 초과할 때 잘못된 AUTO_INCREMENT 값이 생성되었습니다. 이 오류는 최대 열 값을 고려하지 않았기 때문에 발생했습니다. 이 경우 이전의 유효한 AUTO_INCREMENT 값이 반환되어 중복 키 오류가 발생합니다. (버그 #87926, 버그 #26906787)
- 특정 테이블 ID로 사용자가 생성한 테이블이 포함된 Aurora MySQL 버전 1(MySQL 5.6과 호환) 데이터베이스 클러스터를 업그레이드하는 동안 오류가 발생할 수 있는 문제를 해결했습니다. 이러한 테이블 ID를 할당하면 Aurora MySQL 버전 2(MySQL 5.7과 호환)에서 Aurora MySQL 버전 3(MySQL 8.0과 호환)으로 업그레이드하는 동안 데이터 딕셔너리 테이블 ID가 충돌할 수 있습니다. (버그 #33919635)

Aurora MySQL 데이터베이스 엔진 업데이트 2022년 9월 7일(버전 3.02.1)(사용되지 않음)

버전: 3.02.1

Aurora MySQL 3.02.1이 정식 출시되었습니다. Aurora MySQL 3.02 버전은 MySQL 8.0.23과 호환되고, Aurora MySQL 2.x 버전은 MySQL 5.7과 호환되며, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

Aurora MySQL 버전 3의 새로운 기능과 Aurora MySQL 버전 3과 Aurora MySQL 버전 2 또는 커뮤니티 MySQL 8.0의 차이점에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

Aurora MySQL 버전 3으로의 업그레이드 계획에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3에 대한 업그레이드 계획을 참조하세요. 업그레이드 절차 자체는 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3으로 업그레이드를 참조하세요. Aurora MySQL 업그레이드에 대한 일반 정보는 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터 업그레이드를 참조하세요.

문제 해결 정보는 Aurora MySQL 버전 3의 업그레이드 문제 해결을 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

Aurora MySQL 버전 3.02.1은 정식 출시되었으며 일반적으로 커뮤니티 MySQL 8.0.23과 호환됩니다.

아래 나열된 보안 문제 및 CVE 해결:

관리된 환경에서 처리를 미세 조정하기 위한 수정 사항 및 기타 향상된 기능 아래의 추가 CVE 수정:

- CVE-2022-0778

가용성 향상:

- 여러 MySQL 바이너리 로그(binlog) 복제본이 Aurora 라이터 노드에 연결되거나 새 연결 요청이 급증하면서 동시에 오래 실행되는 쿼리가 많을 때 연결이 실패하고 지연 시간이 길어지는 문제를 해결했습니다.
- `CONNECT` 이벤트에 대한 고급 감사가 설정되어 있을 때 데이터베이스가 다시 시작되는 문제를 해결했습니다.
- 내부 임시 테이블이 고객이 구성한 값 또는 기본값으로 설정된 메모리 및 mmap 파일의 할당된 크기를 소진할 경우 Aurora MySQL 읽기 전용 복제본 인스턴스에서 데이터베이스가 다시 시작될 수 있는 문제를 해결했습니다.
- 저장 프로시저에서 동시 DDL 작업 중에 읽기 전용 복제본이 반복적으로 다시 시작될 수 있는 문제를 해결했습니다.
- `INSERT INTO SELECT` 및 `FROM`과 같은 쿼리를 실행할 때 불일치를 일으킬 수 있는 문제로 인해 이 Aurora MySQL 버전에서는 빠른 삽입이 활성화되지 않습니다. 빠른 삽입 최적화에 대한 자세한 내용은 Amazon Aurora MySQL 성능 향상을 참조하세요.

일반적인 개선 사항

- R6i 인스턴스에 대한 지원을 추가했습니다.

추가 정보:

- Aurora MySQL 버전 3.02.1는 Aurora MySQL 버전 2(MySQL 5.7과 호환 가능)에서 직접 메이저 버전을 업그레이드하는 것을 지원하지 않습니다. 이 버전으로 메이저 버전을 업그레이드하려면 먼저 Aurora MySQL 버전 3.02.0으로 메이저 버전을 업그레이드한 다음 Aurora MySQL 버전 3.02.1로 인플레이스 마이너 버전 업그레이드를 수행하세요.

Aurora MySQL 데이터베이스 엔진 업데이트 2022년 4월 20일(버전 3.02.0)(사용되지 않음)

버전: 3.02.0

Aurora MySQL 3.02.0이 정식 출시되었습니다. Aurora MySQL 3.02 버전은 MySQL 8.0.23과 호환되고, Aurora MySQL 2.x 버전은 MySQL 5.7과 호환되며, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

Aurora MySQL 버전 3의 새로운 기능과 Aurora MySQL 버전 3과 Aurora MySQL 버전 2 또는 커뮤니티 MySQL 8.0의 차이점에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

또한 현재 지원되는 Aurora MySQL 버전 2 클러스터에서 Aurora MySQL 3.02.0으로 스냅샷을 복원할 수 있습니다.

Aurora MySQL 버전 3으로의 업그레이드 계획에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3에 대한 업그레이드 계획을 참조하세요. 업그레이드 절차 자체는 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3으로 업그레이드를 참조하세요. Aurora MySQL 업그레이드에 대한 일반 정보는 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터 업그레이드를 참조하세요.

문제 해결 정보는 Aurora MySQL 버전 3의 업그레이드 문제 해결을 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

Aurora MySQL 버전 3.02.0은 정식 출시되었으며 일반적으로 커뮤니티 MySQL 8.0.23과 호환됩니다.

아래 나열된 보안 문제 및 CVE 해결:

관리된 환경에서 처리를 미세 조정하기 위한 수정 사항 및 기타 향상된 기능 아래의 추가 CVE 수정:

- CVE-2021-22946

새로운 기능:

- Amazon Aurora Serverless v2이 정식 버전입니다. 자세한 내용은 Amazon Aurora Serverless 개요, 블로그 및 Aurora Serverless v2 사용 문서를 참조하세요. AWS Management Console에서 몇 단계만 거치면 Aurora Serverless v2 데이터베이스를 생성하여 지금 바로 시작할 수 있습니다.

가용성 향상:

- 레코드를 삭제하거나 두 개 이상의 가변 길이 열(VARCHAR, VARBINARY, BLOB 및 TEXT 유형)이 포함된 테이블을 삭제하는 동안 서버가 다시 시작 루프에 들어가 사용할 수 없게 되는 문제를 해결했습니다. 열 유형에 대한 자세한 내용은 innodb-row-format을 참조하세요.
- 바이너리 로그가 켜져 있고 하나 이상의 바이너리 로그 소비자가 연결되어 있는 클러스터에서 기존 연결 제한 시간이 초과되고 새 연결을 설정할 수 없어 애플리케이션과 소비자 간에 리소스 경합이 발생하는 문제를 해결했습니다.
- 사용 가능한 메모리는 `FreeableMemory` CloudWatch 지표로 표시됩니다. 자세한 내용은 Amazon Aurora에 대한 Amazon CloudWatch 지표를 참조하세요.
  - 바이너리 로그 복제가 활성화된 경우 사용 가능한 메모리 감소로 인해 DB 인스턴스가 다시 시작되거나 장애 조치가 발생할 수 있는 문제를 해결했습니다.
  - 세션 변수를 설정할 때 사용 가능한 메모리 감소로 인해 DB 인스턴스가 다시 시작되거나 장애 조치가 발생할 수 있는 문제를 해결했습니다.
  - 데이터베이스 프로세스가 기존 파일을 열 때 사용 가능한 메모리 감소로 인해 DB 인스턴스가 다시 시작되거나 장애 조치가 발생할 수 있는 문제를 해결했습니다.
- 드문 경우지만 스냅샷에서 복원된 클러스터의 `AUTO_INCREMENT` 열을 포함하는 테이블에 새 행을 삽입할 때 중복 입력 오류가 발생할 수 있는 문제를 해결했습니다.
- `INSERT INTO SELECT` 및 `FROM`과 같은 쿼리를 실행할 때 불일치를 일으킬 수 있는 문제로 인해 이 Aurora MySQL 버전에서는 빠른 삽입이 활성화되지 않습니다. 빠른 삽입 최적화에 대한 자세한 내용은 Amazon Aurora MySQL 성능 향상을 참조하세요.

일반적인 개선 사항

- `SHOW VOLUME STATUS` 명령을 사용할 때 볼륨 상태가 표시되지 않는 문제를 해결했습니다. 자세한 내용은 AuroraMySQL.Managing.VolumeStatus를 참조하세요.
- MySQL 서버 ERROR 3512로 인해 mysql_rds_import_binlog_ssl_material에 대한 직접 호출이 실패하는 문제를 해결했습니다.
- 삭제된 Aurora 리더 인스턴스에 대해 Aurora 복제 지연이 잘못 보고되는 문제를 해결했습니다.

업그레이드 및 마이그레이션:

- ibdata 파일 및 테이블스페이스를 Aurora 스토리지로 복사할 때 발생하는 문제로 인해 MySQL 8.0.x 데이터베이스를 Aurora MySQL 버전 3으로 마이그레이션하지 못하는 문제를 해결했습니다.
- 데이터베이스 테이블에 대량의 데이터가 포함된 경우 Aurora MySQL 버전 2에서 Aurora MySQL 버전 3으로의 클러스터 업그레이드가 실패할 수 있는 문제를 해결했습니다.
- 테이블에 대해 직렬화된 데이터 딕셔너리 정보(SDI)를 생성하지 못해 클러스터를 Aurora MySQL 버전 2에서 Aurora MySQL 버전 3으로 복원할 때 오류가 발생할 수 있는 문제를 해결했습니다.
- RDS 시스템 테이블에 대한 업그레이드 사전 확인에서 보고된 스키마 불일치 오류로 인해 Aurora MySQL 버전 2에서 Aurora MySQL 버전 3으로의 업그레이드가 실패할 수 있는 문제를 해결했습니다.
- RDS 관리형 저장 프로시저의 잘못된 구문으로 인해 RDS for MySQL 8.0 또는 Aurora MySQL 버전 2에서 Aurora MySQL 버전 3 데이터베이스로 마이그레이션하거나 복원할 때 오류가 발생할 수 있는 문제를 해결했습니다.
- 일반 로그 및 느린 로그 테이블에 대한 업그레이드 사전 확인에서 보고된 스키마 불일치 오류로 인해 Aurora MySQL 2에서 Aurora MySQL 3으로의 업그레이드가 실패할 수 있는 문제를 해결했습니다.

MySQL 커뮤니티 에디션 버그 픽스 통합

이 릴리스에는 아래 사항 외에도 8.0.23까지의 모든 커뮤니티 버그 수정이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 3.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그를 참조하세요.

- 저장 프로시저 내에서 커서에 사용되는 임시 테이블을 잘못 처리하여 예기치 않은 서버 동작 mysqld-8-0-24-bug가 발생할 수 있는 문제를 해결했습니다. (버그 #32416811)

Aurora MySQL 데이터베이스 엔진 업데이트 2022년 4월 15일(버전 3.01.1)(사용되지 않음)

버전: 3.01.1

Aurora MySQL 3.01.1이 정식 출시되었습니다. Aurora MySQL 3.01 버전은 MySQL 8.0.23과 호환되고, Aurora MySQL 2.x 버전은 MySQL 5.7과 호환되며, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

Aurora MySQL 버전 3의 새로운 기능과 Aurora MySQL 버전 3과 Aurora MySQL 버전 2 또는 커뮤니티 MySQL 8.0의 차이점에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

현재 지원되는 Aurora MySQL 버전 2 클러스터에서 Aurora MySQL 3.01.1로 스냅샷을 복원할 수 있습니다.

Aurora MySQL 버전 3으로의 업그레이드 계획에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3에 대한 업그레이드 계획을 참조하세요. 업그레이드 절차 자체는 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3으로 업그레이드를 참조하세요. Aurora MySQL 업그레이드에 대한 일반 정보는 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터 업그레이드를 참조하세요.

문제 해결 정보는 Aurora MySQL 버전 3의 업그레이드 문제 해결을 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

Aurora MySQL 버전 3.01.1은 정식 출시되었으며 일반적으로 커뮤니티 MySQL 8.0.23과 호환됩니다.

MySQL 8.0 호환 Aurora 데이터베이스로의 업그레이드 및 마이그레이션에는 Aurora MySQL 버전 3.01.1을 사용하는 것이 좋습니다.

아래 나열된 보안 문제 및 CVE 해결:

관리된 환경에서 처리를 미세 조정하기 위한 수정 사항 및 기타 향상된 기능 아래의 추가 CVE 수정:

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

가용성 향상:

- 사용 가능한 메모리는 `FreeableMemory` CloudWatch 지표로 표시됩니다. 자세한 내용은 Amazon Aurora에 대한 Amazon CloudWatch 지표를 참조하세요.

  - 바이너리 로그 복제가 활성화된 경우 사용 가능한 메모리 감소로 인해 DB 인스턴스가 다시 시작되거나 장애 조치가 발생할 수 있는 문제를 해결했습니다.
  - 세션 변수를 설정할 때 사용 가능한 메모리 감소로 인해 DB 인스턴스가 다시 시작되거나 장애 조치가 발생할 수 있는 문제를 해결했습니다.
  - 데이터베이스 프로세스가 기존 파일을 열 때 사용 가능한 메모리 감소로 인해 DB 인스턴스가 다시 시작되거나 장애 조치가 발생할 수 있는 문제를 해결했습니다.
- 드문 경우지만 스냅샷에서 복원된 클러스터의 `AUTO_INCREMENT` 열을 포함하는 테이블에 새 행을 삽입할 때 중복 입력 오류가 발생할 수 있는 문제를 해결했습니다.
- `INSERT INTO SELECT` 및 `FROM`과 같은 쿼리를 실행할 때 불일치를 일으킬 수 있는 문제로 인해 이 Aurora MySQL 버전에서는 빠른 삽입이 활성화되지 않습니다. 빠른 삽입 최적화에 대한 자세한 내용은 Amazon Aurora MySQL 성능 향상을 참조하세요.

일반적인 개선 사항

- `SHOW VOLUME STATUS` 명령을 사용할 때 볼륨 상태가 표시되지 않는 문제를 해결했습니다. 자세한 내용은 AuroraMySQL.Managing.VolumeStatus를 참조하세요.
- MySQL 서버 ERROR 3512로 인해 mysql_rds_import_binlog_ssl_material에 대한 직접 호출이 실패하는 문제를 해결했습니다.
- 삭제된 Aurora 리더 인스턴스에 대해 Aurora 복제 지연이 잘못 보고되는 문제를 해결했습니다.

업그레이드 및 마이그레이션:

- ibdata 파일 및 테이블스페이스를 Aurora 스토리지로 복사할 때 발생하는 문제로 인해 MySQL 8.0.x 데이터베이스를 Aurora MySQL 버전 3으로 마이그레이션하지 못하는 문제를 해결했습니다.
- 데이터베이스 테이블에 대량의 데이터가 포함된 경우 Aurora MySQL 버전 2에서 Aurora MySQL 버전 3으로의 클러스터 업그레이드가 실패할 수 있는 문제를 해결했습니다.
- 테이블에 대해 직렬화된 데이터 딕셔너리 정보(SDI)를 생성하지 못해 클러스터를 Aurora MySQL 버전 2에서 Aurora MySQL 버전 3으로 복원할 때 오류가 발생할 수 있는 문제를 해결했습니다.
- RDS 시스템 테이블에 대한 업그레이드 사전 확인에서 보고된 스키마 불일치 오류로 인해 Aurora MySQL 버전 2에서 Aurora MySQL 버전 3으로의 업그레이드가 실패할 수 있는 문제를 해결했습니다.
- RDS 관리형 저장 프로시저의 잘못된 구문으로 인해 RDS for MySQL 8.0 또는 Aurora MySQL 버전 2에서 Aurora MySQL 버전 3 데이터베이스로 마이그레이션하거나 복원할 때 오류가 발생할 수 있는 문제를 해결했습니다.

- 일반 로그 및 느린 로그 테이블에 대한 업그레이드 사전 확인에서 보고된 스키마 불일치 오류로 인해 Aurora MySQL 2에서 Aurora MySQL 3으로의 업그레이드가 실패할 수 있는 문제를 해결했습니다.

MySQL 커뮤니티 에디션 버그 픽스 통합

이 릴리스에는 아래 사항 외에도 8.0.23까지의 모든 커뮤니티 버그 수정이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 3.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그를 참조하세요.

- 저장 프로시저 내에서 커서에 사용되는 임시 테이블을 잘못 처리하여 예기치 않은 서버 동작 mysqld-8-0-24-bug가 발생할 수 있는 문제를 해결했습니다. (버그 #32416811)

Aurora MySQL 데이터베이스 엔진 업데이트 2021년 11월 18일(버전 3.01.0)(사용되지 않음)

버전: 3.01.0

Aurora MySQL 3.01.0이 정식 버전입니다. Aurora MySQL 3.01 버전은 MySQL 8.0.23과 호환되고, Aurora MySQL 2.x 버전은 MySQL 5.7과 호환되며, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

Aurora MySQL 버전 3의 새로운 기능과 Aurora MySQL 버전 3과 Aurora MySQL 버전 2 또는 커뮤니티 MySQL 8.0의 차이점에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 2와 Aurora MySQL 버전 3의 비교를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

또한 현재 지원되는 Aurora MySQL 버전 2 클러스터에서 Aurora MySQL 3.01.0으로 스냅샷을 복원할 수 있습니다.

Aurora MySQL 버전 3으로의 업그레이드 계획에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3에 대한 업그레이드 계획을 참조하세요. 업그레이드 절차 자체는 Amazon Aurora 사용 설명서의 Aurora MySQL 버전 3으로 업그레이드를 참조하세요. Aurora MySQL 업그레이드에 대한 일반 정보는 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터 업그레이드를 참조하세요.

문제 해결 정보는 Aurora MySQL 버전 3의 업그레이드 문제 해결을 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

Aurora MySQL 버전 3.01.0은 일반적으로 커뮤니티 MySQL 8.0.23과 호환됩니다. 이 버전에는 커뮤니티 MySQL 8.0.23의 일반적인 취약점 및 노출 (CVE) 문제에 대한 보안 수정이 포함되어 있습니다.

Aurora MySQL 버전 3.01.0에는 Aurora MySQL 버전 2.10.0을 통한 모든 Aurora 관련 버그 수정이 포함되어 있습니다.

Aurora MySQL 버전 3의 새로운 기능에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 커뮤니티 MySQL 8.0 기능 및 새로운 병렬 쿼리 최적화를 참조하세요.

가용성 향상:

- `INSERT INTO SELECT` 및 `FROM`과 같은 쿼리를 실행할 때 불일치를 일으킬 수 있는 문제로 인해 이 Aurora MySQL 버전에서는 빠른 삽입이 활성화되지 않습니다. 빠른 삽입 최적화에 대한 자세한 내용은 Amazon Aurora MySQL 성능 향상을 참조하세요.

Amazon Aurora MySQL 버전 2에 대한 데이터베이스 엔진 업데이트

> **Important**
>
> 2024년 10월 31일에 Aurora MySQL 버전 2의 표준 지원이 종료되었습니다. 자세한 내용은 Amazon Aurora MySQL 호환 버전 2 표준 지원 종료 준비를 참조하세요.

다음은 Amazon Aurora MySQL 버전 2 데이터베이스 엔진 업데이트입니다.

- Aurora MySQL 데이터베이스 엔진 업데이트 2025년 4월 9일(버전 2.12.5, MySQL 5.7.44와 호환) - RDS 추가 지원 버전
- Aurora MySQL 데이터베이스 엔진 업데이트 2024년 10월 23일(버전 2.12.4, MySQL 5.7.44와 호환) - RDS 추가 지원 버전
- Aurora MySQL 데이터베이스 엔진 업데이트 2024년 7월 9일(버전 2.12.3, MySQL 5.7.44와 호환) - RDS 추가 지원 버전
- Aurora MySQL 데이터베이스 엔진 업데이트 2024년 3월 19일(버전 2.12.2, MySQL 5.7.44와 호환) - RDS 추가 지원 버전
- Aurora MySQL 데이터베이스 엔진 업데이트 2023년 12월 28일(버전 2.12.1, MySQL 5.7.40과 호환) - RDS 추가 지원 버전
- Aurora MySQL 데이터베이스 엔진 업데이트 2023년 10월 25일(버전 2.12.0.1, MySQL 5.7.40과 호환) - RDS 추가 지원 버전(베타)
- Aurora MySQL 데이터베이스 엔진 업데이트 2023년 7월 25일(버전 2.12.0, MySQL 5.7.40과 호환) - RDS 추가 지원 버전
- Aurora MySQL 데이터베이스 엔진 업데이트 2024년 7월 19일(버전 2.11.6, MySQL 5.7.12와 호환) - RDS 추가 지원 버전
- Aurora MySQL 데이터베이스 엔진 업데이트 2024년 3월 26일(버전 2.11.5, MySQL 5.7.12와 호환) - RDS 추가 지원 버전(기본)
- Aurora MySQL 데이터베이스 엔진 업데이트 2023년 10월 17일(버전 2.11.4, MySQL 5.7.12와 호환) - RDS 추가 지원 버전
- Aurora MySQL 데이터베이스 엔진 업데이트 2023년 6월 9일(버전 2.11.3, MySQL 5.7.12와 호환) - RDS 추가 지원 버전

- Aurora MySQL 데이터베이스 엔진 업데이트 2023년 3월 24일(버전 2.11.2, MySQL 5.7.12와 호환) - RDS 추가 지원 버전
- Aurora MySQL 데이터베이스 엔진 업데이트 2023년 2월 14일(버전 2.11.1, MySQL 5.7.12와 호환) - RDS 추가 지원 버전
- Aurora MySQL 데이터베이스 엔진 업데이트 2022년 10월 25일(버전 2.11.0, MySQL 5.7.12와 호환) - RDS 추가 지원 버전
- Aurora MySQL 데이터베이스 엔진 업데이트 2022년 11월 1일(버전 2.10.3)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2022년 1월 26일(버전 2.10.2)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2021년 10월 21일(버전 2.10.1)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2021년 5월 25일(버전 2.10.0)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2021년 11월 12일(버전 2.09.3)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2021년 2월 26일(버전 2.09.2)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2020년 12월 11일(버전 2.09.1)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2020년 9월 17일(버전 2.09.0)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2022년 1월 6일(버전 2.08.4)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2020년 11월 12일(버전 2.08.3)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2020년 8월 28일(버전 2.08.2)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2020년 6월 18일(버전 2.08.1)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2020년 6월 2일(버전 2.08.0)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2023년 8월 15일(버전 2.07.10)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2023년 5월 4일(버전 2.07.9)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2022년 6월 16일(버전 2.07.8)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2021년 11월 24일(버전 2.07.7)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2021년 9월 2일(버전 2.07.6)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2021년 7월 6일(버전 2.07.5)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2021년 3월 4일(버전 2.07.4)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2020년 11월 10일(버전 2.07.3)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2020년 4월 17일(버전 2.07.2)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2019년 12월 23일(버전 2.07.1)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2019년 11월 25일(버전 2.07.0)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2019년 11월 22일(버전 2.06.0)(사용되지 않음)

- Aurora MySQL 데이터베이스 엔진 업데이트 2019년 11월 11일(버전 2.05.0)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2020년 8월 14일(버전 2.04.9)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2019년 11월 20일(버전 2.04.8)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2019년 11월 14일(버전 2.04.7)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2019년 9월 19일(버전 2.04.6)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2019년 7월 8일(버전 2.04.5)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2019년 5월 29일(버전 2.04.4)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2019년 5월 9일(버전 2.04.3)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2019년 5월 2일(버전 2.04.2)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2019년 3월 25일(버전 2.04.1)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2019년 3월 25일(버전 2.04.0)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2019년 2월 7일(버전 2.03.4)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2019년 1월 18일(버전 2.03.3)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2019년 1월 9일(버전 2.03.2)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2018년 10월 24일(버전 2.03.1)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2018년 10월 11일(버전 2.03)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2018년 10월 8일(버전 2.02.5)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2018년 9월 21일(버전 2.02.4)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2018년 8월 23일(버전 2.02.3)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2018년 6월 4일(버전 2.02.2)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2018년 5월 3일(버전 2.02)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2018년 3월 13일(버전 2.01.1)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2018년 2월 6일(버전 2.01)(사용되지 않음)

Aurora MySQL 데이터베이스 엔진 업데이트 2025년 4월 9일(버전 2.12.5, MySQL 5.7.44와 호환) - RDS 추가 지원 버전

버전: 2.12.5

Aurora MySQL 2.12.5가 정식 버전입니다. Aurora MySQL 2.12 버전은 MySQL 5.7.44까지 호환됩니다. 커뮤니티 변경 사항에 대한 자세한 내용은 Changes in MySQL 5.7.44(2022-10-11, General Availability)를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 2.11.*, 2.12.*, 3.04.*, 3.05.*, 3.06.*, 3.07.* 및 3.08.*입니다.

기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.12.5으로 업그레이드할 수 있습니다. 또한 현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.12.5으로 스냅샷을 복원할 수 있습니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

보안 문제 및 CVE 해결:

이 릴리스에는 MySQL 5.7.44까지의 모든 커뮤니티 CVE 수정 사항이 포함되어 있습니다. 다음 CVE 수정 사항이 포함되어 있습니다.

- CVE-2025-21543
- CVE-2025-21500
- CVE-2025-21491
- CVE-2025-21490
- CVE-2025-21540
- CVE-2025-21559
- CVE-2025-21555
- CVE-2025-21497
- CVE-2025-21520
- CVE-2025-21501
- CVE-2024-37371

- CVE-2024-11053
- CVE-2024-21201
- CVE-2024-21241
- CVE-2024-21230
- CVE-2023-44487

가용성 향상:

- 네트워크 중단으로 인해 라이터와의 연결이 올바르게 재설정되지 않는 복제본 문제를 수정했습니다.

Aurora MySQL 데이터베이스 엔진 업데이트 2024년 10월 23일(버전 2.12.4, MySQL 5.7.44와 호환) - RDS 추가 지원 버전

버전: 2.12.4

Aurora MySQL 2.12.4가 정식 버전입니다. Aurora MySQL 2.12 버전은 MySQL 5.7.44까지 호환됩니다. 커뮤니티 변경 사항에 대한 자세한 내용은 Changes in MySQL 5.7.44(2022-10-11, General Availability)를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 2.11.*, 2.12.*, 3.04.*, 3.05.*, 3.06.* 및 3.07.*입니다.

기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.12.4으로 업그레이드할 수 있습니다. 또한 현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.12.4으로 스냅샷을 복원할 수 있습니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

보안 문제 및 CVE 해결:

이 릴리스에는 MySQL 5.7.44까지의 모든 커뮤니티 CVE 수정 사항이 포함되어 있습니다. 다음 CVE 수정 사항이 포함되어 있습니다.

- CVE-2023-44487
- CVE-2024-21142
- CVE-2024-21177
- CVE-2024-25062

MySQL Community Edition 버그 수정 통합

이 릴리스에는 MySQL 5.7.44까지의 모든 커뮤니티 버그 수정 사항이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 2.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그 섹션을 참조하세요.

Aurora MySQL 버전 2에서 지원되지 않는 기능

다음 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환)에서는 현재 지원되지 않습니다.

- 배치화 스캔

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- `CREATE TABLESPACE` SQL 문
- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인

- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- X 프로토콜

Aurora MySQL 데이터베이스 엔진 업데이트 2024년 7월 9일(버전 2.12.3, MySQL 5.7.44와 호환) - RDS 추가 지원 버전

버전: 2.12.3

Aurora MySQL 2.12.3이 정식 버전입니다. Aurora MySQL 2.12 버전은 MySQL 5.7.44까지 호환됩니다. 커뮤니티 변경 사항에 대한 자세한 내용은 Changes in MySQL 5.7.44(2022-10-11, General Availability)를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 2.11.*, 2.12.*, 3.03.*, 3.04.*, 3.05.*, 3.06.* 및 3.07.*입니다.

기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.12.3으로 업그레이드할 수 있습니다. 또한 현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.12.3으로 스냅샷을 복원할 수 있습니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

보안 문제 및 CVE 해결:

- MySQL 저장 프로시저의 보안 문제를 수정했습니다.

이 릴리스에는 MySQL 5.7.44까지의 모든 커뮤니티 CVE 수정 사항이 포함되어 있습니다. 다음 CVE 수정 사항이 포함되어 있습니다.

- CVE-2023-21912
- CVE-2023-44487
- CVE-2024-0853
- CVE-2024-20963
- CVE-2024-20998
- CVE-2024-21008
- CVE-2024-21009
- CVE-2024-21013
- CVE-2024-21047
- CVE-2024-21054
- CVE-2024-21055
- CVE-2024-21057
- CVE-2024-21062
- CVE-2024-21069
- CVE-2024-21096
- CVE-2024-21097

가용성 향상:

- 병렬 쿼리를 실행할 때 Aurora MySQL DB 인스턴스가 다시 시작되는 문제를 수정했습니다.
- 원활한 조정, 제로 가동 중지 재시작(ZDR) 및 제로 가동 중지 패치 적용(ZDP) 중에 연결 리소스의 동시 액세스로 인해 데이터베이스 서버가 다시 시작될 수 있는 문제를 수정했습니다.
- 로그 애플리케이션에 사용되는 메모리를 비울 때 리더 DB 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 백그라운드 프로세스에서 백그라운드 작업이 임시 인덱스를 삭제하는 동안 쿼리 실행이 장기화되거나 실패하는 문제를 수정했습니다.
- 메타데이터 불일치로 인해 라이터 DB 인스턴스가 다시 시작될 수 있는 시작 루틴 문제를 수정했습니다.

- 트랜잭션 복구 진행 상황에 대한 지표를 추가했습니다. 이렇게 하면 트랜잭션 복구를 완료하는 데 시간이 오래 걸리는 드문 상황에서 잠재적인 가용성이 방지됩니다.
- 라이터 DB 인스턴스에서 변경되거나 삭제되는 테이블을 읽을 때 리더 DB 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- `thread_stack` 파라미터 값이 낮으면 데이터베이스가 반복적으로 다시 시작되는 문제를 수정했습니다. 부팅 성공을 보장하고 시작 문제를 방지하기 위해 최소 허용 `thread_stack` 값이 131,072에서 136,192로 증가했습니다.
- 병렬 쿼리를 실행할 때 리더 DB 인스턴스가 다시 시작되는 문제를 수정했습니다.
- 라이터 DB 인스턴스에서 드문 트랜잭션 커밋 주문이 발생하는 경우 Aurora 읽기 전용 복제본이 다시 시작될 수 있는 문제를 수정했습니다.
- 드문 경우지만 읽기 전용 트랜잭션이 공유 잠금을 획득할 때 DB 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 전달된 암시적 커밋 문에 오류가 발생할 때 쓰기 전달을 사용하는 리더 DB 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.

일반적인 개선 사항

- `AUTO_INCREMENT` 프라이머리 키 열과 고유 키 열이 있는 테이블에서 동시 `INSERT` 문을 수행할 때와 `INSERT` 문에 서로 다른 행에 고유한 키 위반이 있을 때 SQL 문에 예기치 않은 프라이머리 키 위반 오류 또는 경고가 발생할 수 있는 문제를 수정했습니다.
- ZDR이 쿼리에서 힌트로 설정된 세션 변수를 잘못 복원할 때 잘못된 쿼리 결과가 발생할 수 있는 문제를 수정했습니다.
- 기본 제공 `LPAD` 및 `RPAD` 문자열 함수를 사용할 때 불완전한 결과 집합이 반환되는 병렬 쿼리 문제를 수정했습니다.
- 외래 키가 있는 테이블에 대해 라이터 DB 인스턴스에서 `ALTER TABLE RENAME COLUMN` 문을 실행할 때 리더 DB 인스턴스에서 외래 키 인덱스가 누락되는 문제를 수정했습니다.
- 쓰기 전달을 비활성화하는 프로세스를 완료하지 못할 수 있는 문제를 해결했습니다.
- Aurora Serverless v1 조정 중 조정점을 찾는 동안 내부 데이터 구조에 대한 잘못된 액세스로 인해 DB 인스턴스가 다시 시작되는 문제를 수정했습니다.
- db.t4g.medium 및 db.t4g.large DB 인스턴스에 대해 Performance Insights 자동 관리가 켜져 있을 때 성능 스키마가 활성화되지 않는 문제를 수정했습니다.
- Amazon SageMaker AI에 대한 Aurora 기계 학습 작업의 요청 제한 시간이 3초에서 30초로 늘어났습니다. 이는 더 큰 배치 크기를 사용할 때 Aurora 기계 학습에서 Amazon SageMaker AI 에 대한 요청 재시도 또는 실패 횟수가 증가하는 문제를 해결하는 데 도움이 됩니다.

- 느린 SELECT 쿼리가 선행되지 않는 한 MySQL Event Scheduler에서 실행하는 느린 INSERT, DELETE 및 UPDATE 쿼리가 느린 쿼리 로그에 기록되지 않는 문제를 수정했습니다.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 MySQL 5.7.44까지의 모든 커뮤니티 버그 수정 사항이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 2.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그 섹션을 참조하세요.

- 문을 실행하는 동안 임시 테이블이 트리거되어 예기치 않은 DB 엔진 재시작이 발생할 수 있는 문제를 수정했습니다.
- 인덱싱된 표현식을 사용하는 단일 테이블 UPDATE 및 DELETE 문이 준비된 문으로 실행될 때 서버가 종료될 수 있는 결함을 수정했습니다. (버그 #29257254)

Aurora MySQL 버전 2에서 지원되지 않는 기능

다음 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환)에서는 현재 지원되지 않습니다.

- 배치화 스캔

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- CREATE TABLESPACE SQL 문
- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인

- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- X 프로토콜

Aurora MySQL 데이터베이스 엔진 업데이트 2024년 3월 19일(버전 2.12.2, MySQL 5.7.44와 호환) - RDS 추가 지원 버전

버전: 2.12.2

Aurora MySQL 2.12.2가 정식 버전입니다. Aurora MySQL 2.12 버전은 MySQL 5.7.44까지 호환됩니다. 커뮤니티 변경 사항에 대한 자세한 내용은 Changes in MySQL 5.7.44(2022-10-11, General Availability)를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 2.11.*, 2.12.*, 3.03.*, 3.04.*, 3.05.* 및 3.06.*입니다.

기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.12.2으로 업그레이드할 수 있습니다. 또한 현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.12.2으로 스냅샷을 복원할 수 있습니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

아래 나열된 보안 문제 및 CVE 해결:

이 릴리스에는 MySQL 5.7.44까지의 모든 커뮤니티 CVE 수정 사항이 포함되어 있습니다. 다음 CVE 수정 사항이 포함되어 있습니다.

- CVE-2024-20963

- CVE-2023-39975
- CVE-2023-38545

보안 문제:

- 소스가 MASTER_SSL 설정과 관계없이 암호화된 연결을 지원하는 경우 이진 로그 복제본이 기본적으로 SSL/TLS를 사용하도록 하는 수정 사항이 추가되었습니다.

가용성 향상:

- 라이터 인스턴스에 워크로드가 많은 경우 읽기 전용 복제본 인스턴스가 성공적으로 시작되지 않는 문제를 수정했습니다.
- Aurora 스토리지와 통신하는 구성 요소의 결함으로 인해 Aurora MySQL 데이터베이스 라이터 인스턴스가 장애 조치될 수 있는 문제를 수정했습니다. 결함은 Aurora 스토리지 인스턴스의 소프트웨어 업데이트 후 데이터베이스 인스턴스와 기본 스토리지 간의 통신이 분석된 결과로 발생합니다.
- 드문 경우지만 리더 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 권한이 있는 사용자가 사용자인 rdsadmin과 연결된 리소스 제한을 수정할 수 있는 문제를 수정했습니다. 잘못 설정하면 이러한 리소스 제한이 RDS 모니터링 에이전트의 데이터베이스 인스턴스 상태 모니터링 기능을 방해하여 데이터베이스를 사용할 수 없게 될 수 있습니다.

업그레이드 및 마이그레이션:

- Amazon RDS MySQL 5.7에서 마이그레이션하고 지원되지 않는 저장 프로시저가 포함된 Aurora MySQL 클러스터에 대한 이진 로그 복제를 시작하려고 할 때 발생하는 문제를 수정했습니다.
- Aurora MySQL 버전 3으로 메이저 버전을 업그레이드하는 동안 데이터베이스 이벤트 스케줄러를 비활성화했습니다. 이 업데이트는 메이저 버전 업그레이드가 진행되는 동안 이벤트 실행에 의한 데이터베이스 변경을 방지하는 데 도움이 됩니다.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 MySQL 5.7.44까지의 모든 커뮤니티 버그 수정 사항이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 2.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그를 참조하세요.

Aurora MySQL 버전 2에서 지원되지 않는 기능

다음 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환)에서는 현재 지원되지 않습니다.

- 배치화 스캔

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- `CREATE TABLESPACE` SQL 문
- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- X 프로토콜

Aurora MySQL 데이터베이스 엔진 업데이트 2023년 12월 28일(버전 2.12.1, MySQL 5.7.40과 호환) - RDS 추가 지원 버전

버전: 2.12.1

Aurora MySQL 2.12.1이 정식 출시되었습니다. Aurora MySQL 2.12 버전은 MySQL 5.7.40까지 호환됩니다. 커뮤니티 변경 사항에 대한 자세한 내용은 Changes in MySQL 5.7.40(2022-10-11, General Availability)를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.*, 3.04.* 및 3.05.*입니다.

기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.12.1로 업그레이드할 수 있습니다. 또한 현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.12.1로 스냅샷을 복원할 수 있습니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

아래 나열된 보안 문제 및 CVE 해결:

이 릴리스에는 MySQL 5.7.44까지의 모든 커뮤니티 CVE 수정 사항이 포함되어 있습니다.

관리된 환경에서 처리를 미세 조정하기 위한 수정 사항 및 기타 향상된 기능 추가 CVE 수정은 다음과 같습니다.

- 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
- 전체 텍스트 검색(FTS) 구문 분석기 플러그인에 의한 단일 문자 토큰 처리를 수정했습니다. (버그 #35432973)
- 감사 로그 교체를 처리하는 동안 보고된 이벤트가 감사 로그에 기록되지 않는 문제를 해결했습니다.

새로운 기능:

- 가능한 경우 binlog 복제본의 SQL 스레드가 바이너리 로그 이벤트를 병렬로 적용하는 다중 스레드 바이너리 로그(binlog) 복제에 대한 지원을 추가했습니다. Aurora 사용 설명서에서 다중 스레드 복제를 미세 조정하는 데 도움이 되는 구성 옵션에 대해 자세히 알아보세요.

가용성 향상:

- 병렬 쿼리를 사용하는 Aurora MySQL 데이터베이스 인스턴스에서 많은 수의 동시 병렬 쿼리를 실행할 때 데이터베이스가 다시 시작될 수 있는 문제를 해결했습니다.
- 감사 로깅 스레드로 인해 발생하는 잠금 경합에 따라 CPU 사용률이 높아지고 클라이언트 애플리케이션 제한 시간이 초과될 수 있는 문제를 해결했습니다.
- 삭제된 테이블에 속한 데이터베이스 페이지를 읽으려고 할 때 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 라이터 인스턴스에서 데이터베이스 볼륨이 160GB의 배수로 증가하면 리더 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 격리 수준을 `READ_COMMITED` 또는 `READ_UNCOMMITED`로 설정하고 XA 트랜잭션을 사용하거나 바이너리 로그(binlog)를 활성화한 상태에서 2단계 커밋을 처리할 때 잠금 관리자에서 재시작 또는 장애 조치가 발생할 수 있는 문제를 해결했습니다.
- 데이터베이스가 내부 시스템 테이블에서 트리거를 생성하거나 삭제하는 동안 라이터 인스턴스가 다시 시작되면 데이터베이스 클러스터를 사용할 수 없게 되는 문제를 해결했습니다.
- 데이터베이스 연결 수가 `max_connections` 파라미터로 설정된 값에 가까워지면 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 전체 텍스트 인덱스가 포함된 테이블에 대해 데이터 조작 언어(DML) 쿼리를 실행할 때 Aurora 리더 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- `INSERT INTO SELECT` 및 `FROM`과 같은 쿼리를 실행할 때 불일치를 일으킬 수 있는 문제로 인해 이 Aurora MySQL 버전에서는 빠른 삽입이 활성화되지 않습니다. 빠른 삽입 최적화에 대한 자세한 내용은 Amazon Aurora MySQL 성능 향상을 참조하세요.

일반적인 개선 사항

- Aurora 클러스터 볼륨에서 데이터를 읽는 동안 일시적인 네트워크 문제로 인해 병렬 쿼리가 실패할 수 있는 문제를 해결했습니다.
- 다운로드 또는 교체 시 로그 파일에 액세스할 수 없고 경우에 따라 CPU 사용률이 증가할 수 있는 감사 로그 파일 관리 관련 문제를 해결했습니다.

- 2.11.* 미만 버전에서 업그레이드한 후 소규모 읽기 전용 복제본 인스턴스에서 복제 지연이 증가할 수 있는 문제를 해결했습니다.
- 저장 루틴과 관련된 요청을 확인하기 위해 `procs_priv` 권한 부여 테이블을 참조할 때 과도한 로그 메시지가 생성될 수 있는 문제를 해결했습니다.
- 해시 조인 최적화를 사용하여 쿼리를 실행하는 동안 데이터베이스 인스턴스가 과도한 메모리를 사용할 수 있게 되는 메모리 관리 문제를 해결했습니다.
- 쓰기 전달을 사용할 때 `information_schema` 및 `performance_schema` 글로벌 상태 테이블에서 변수 `Threads_running`의 값이 잘못 생성될 수 있는 문제를 해결했습니다.
- 파티셔닝된 테이블(이전 `ha_partition` 파티션 핸들러를 지원하는 MySQL 버전에서 생성됨)을 사용하여 `SELECT` 문을 실행하고 쿼리 플래너가 병렬 쿼리를 선택할 때 데이터베이스가 다시 시작되는 문제를 해결했습니다.
- 쓰기 전달이 활성화된 경우 데이터베이스에 새 클라이언트 연결이 설정되지 않는 문제를 해결했습니다.
- Aurora MySQL binlog 복제본이 `USE` 명령으로 정의된 기본 데이터베이스 없이 소스의 binlog 파일에 기록된 `QUERY` 이벤트를 실행할 때 바이너리 로그(binlog) 복제 지연을 줄였습니다.
- `innodb_flush_log_at_trx_commit` 파라미터가 1로 설정되지 않은 경우 `CommitLatency` CloudWatch 지표가 잘못 보고될 수 있는 문제를 해결했습니다.
- 데이터베이스 연결이 설정되기 전에 종료될 수 있는 문제를 해결했습니다. 이 문제는 빠른 속도로 연결을 시작하고 종료하는 데이터베이스 인스턴스에 영향을 미칠 가능성이 높습니다.
- 연결된 바이너리 로그(binlog) 소비자가 중복된 binlog 복제 서버 ID를 사용할 때 데이터베이스가 다시 시작될 수 있는 문제를 해결했습니다.
- `replica_parallel_type` 변수가 `LOGICAL_CLOCK`으로 설정되고 `replica_preserve_commit_order` 변수가 `ON`으로 설정된 경우 멀티스레드 바이너리 로그 복제가 중지될 수 있는 문제를 해결했습니다. 이 문제는 500MB보다 큰 트랜잭션이 소스에서 실행될 때 발생할 수 있습니다.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 아래 사항 외에도 5.7.40까지의 모든 커뮤니티 버그 수정이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 2.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그를 참조하세요.

- `SHOW PROCESSLIST` 문과 동시에 실행될 때 기존 및 새 원격 연결이 중단될 수 있는 문제를 해결했습니다. (커뮤니티 버그 #34857411)
- 복제: 일부 바이너리 로그 이벤트가 경우에 따라 올바르게 처리되지 않았습니다. (버그 #34617506)

Aurora MySQL 버전 2에서 지원되지 않는 기능

다음 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환)에서는 현재 지원되지 않습니다.

- 배치화 스캔

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- `CREATE TABLESPACE` SQL 문
- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- X 프로토콜

Aurora MySQL 데이터베이스 엔진 업데이트 2023년 10월 25일(버전 2.12.0.1, MySQL 5.7.40과 호환) - RDS 추가 지원 버전(베타)

버전: 2.12.0.1

Aurora MySQL 2.12.0.1이 미국 동부(버지니아 북부), 미국 동부(오하이오), 미국 서부(캘리포니아 북부), 미국 서부(오레곤), AWS GovCloud(미국 동부) 및 AWS GovCloud(미국 서부) 리전에서 정식 출시

되었습니다. 이는 보안 수정 사항만 포함된 초기 릴리스입니다. 다음 패치 릴리스인 2.12.1에서는 이러한 수정 사항이 모든 리전에 더 광범위하게 배포될 예정입니다. Aurora MySQL 2.12 버전은 MySQL 5.7.40과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 2.07.*, 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.* 및 3.04.*입니다.

기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.12.0.1로 업그레이드할 수 있습니다. 또한 현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.12.0.1로 스냅샷을 복원할 수 있습니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

아래 나열된 보안 문제 및 CVE 해결:

이 릴리스에는 MySQL 5.7.40까지의 모든 커뮤니티 CVE 수정 사항이 포함되어 있습니다.

- CVE-2023-38545

가용성 향상:

- `INSERT INTO SELECT` 및 `FROM`과 같은 쿼리를 실행할 때 불일치를 일으킬 수 있는 문제로 인해 이 Aurora MySQL 버전에서는 빠른 삽입이 활성화되지 않습니다. 빠른 삽입 최적화에 대한 자세한 내용은 Amazon Aurora MySQL 성능 향상을 참조하세요.

Aurora MySQL 데이터베이스 엔진 업데이트 2023년 7월 25일(버전 2.12.0, MySQL 5.7.40과 호환) - RDS 추가 지원 버전

버전: 2.12.0

Aurora MySQL 2.12.0이 정식 출시되었습니다. Aurora MySQL 2.12 버전은 MySQL 5.7.40까지 호환됩니다. 커뮤니티 변경 사항에 대한 자세한 내용은 Changes in MySQL 5.7.40(2022-10-11, General Availability)를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 2.11.*, 2.12.*, 3.01.*, 3.02.* 및 3.03.*입니다.

기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.12.0으로 업그레이드할 수 있습니다. 또한 현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.12.0으로 스냅샷을 복원할 수 있습니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를AWS사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

아래 나열된 보안 문제 및 CVE 해결:

이 릴리스에는 MySQL 5.7.40까지의 모든 커뮤니티 CVE 수정 사항이 포함되어 있습니다.

- 보안 수준이 낮은 DES-CBC3-SHA 값을 SSL_CIPHER 데이터베이스 파라미터에서 제외하도록 Aurora MySQL에서 사용하는 기본 SSL 암호를 업데이트했습니다. DES-CBC3-SHA 암호 제거로 인해 SSL 연결 문제가 발생하는 경우 Aurora MySQL DB 클러스터 연결을 위한 암호 그룹 구성 목록에 나온 적용 가능한 보안 암호를 사용하세요. MySQL 클라이언트 연결 암호 구성에 대한 자세한 내용은 MySQL 설명서에서 확인할 수 있습니다.
- 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

가용성 향상:

- 데이터베이스 활동 스트림 이벤트 암호화에서 데이터베이스 다시 시작을 일으킬 수 있는 문제를 해결했습니다.
- 데이터 정의 언어(DDL) 쿼리를 실행하는 동안 데이터베이스가 다시 시작될 경우 다시 시작의 실패를 유발할 수 있는 두 가지 문제를 해결했습니다.
- 연결 급증으로 인해 쿼리 지연 시간이 증가하거나 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 드문 경우지만 라이터 인스턴스의 대규모 업데이트 작업 또는 데이터 정의 언어(DDL) 워크로드와 Aurora 복제본의 동일한 테이블 세트에 대한 읽기 작업을 동시에 실행하는 동안 Aurora 복제본이 다시 시작될 수 있는 문제를 해결했습니다.
- 연결 급증으로 인해 연결 설정 프로세스가 완료되는 데 더 오랜 시간이 걸리거나 시간 초과 오류가 발생하여 연결 설정이 실패할 수 있는 문제를 해결했습니다.
- 고급 감사 로그 교체로 인해 사용 가능한 메모리가 줄어들어 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- Aurora 병렬 쿼리 실행 계획을 사용하는 쿼리를 실행하는 동안 Aurora MySQL 리더 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 전체 텍스트 검색(FTS) 인덱스가 있는 테이블에 대해 `OPTIMIZE TABLE` 쿼리를 실행하는 동안 라이터 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- Aurora 글로벌 데이터베이스 보조AWS리전에서 글로벌 쓰기 전달을 사용하여 `SELECT FOR UPDATE` 쿼리를 실행할 때 Aurora 글로벌 데이터베이스 기본 리전의 라이터 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 전달된 암시적 커밋 문에 오류가 발생할 때 글로벌 쓰기 전달을 사용하는 Aurora 글로벌 데이터베이스 보조AWS리전 리더 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- `INSERT INTO SELECT` 및 `FROM`과 같은 쿼리를 실행할 때 불일치를 일으킬 수 있는 문제로 인해 이 Aurora MySQL 버전에서는 빠른 삽입이 활성화되지 않습니다. 빠른 삽입 최적화에 대한 자세한 내용은 Amazon Aurora MySQL 성능 향상을 참조하세요.

일반적인 개선 사항:

- 릴레이 로그 파일에 쓸 때 경합을 줄일 수 있도록 binlog 복제본에 파일 관리 성능 최적화 기능을 도입했습니다.
- `information_schema` 지표에서 `buffer_pool_read_requests` 카운터가 잘못 보고될 수 있는 문제를 해결했습니다.

- `LOAD FROM S3` 또는 `SELECT INTO S3` 작업을 수행할 때 로컬 스토리지가 가득 찰 수 있게 되는 문제를 해결했습니다. 또한 이 문제로 인해 CPU 사용률이 높아지고, 메모리 부족으로 인해 데이터베이스가 다시 시작되고, 이러한 쿼리의 지연 시간이 늘어날 수 있습니다.
- 여러 바이너리 로그 복제 소비자가 연결된 경우 바이너리 로그 복제를 사용하는 DB 인스턴스에서 CPU 사용률이 증가하고 연결 실패가 발생할 수 있는 문제를 해결했습니다.
- SSL 서버 상태 변수가 채워지지 않는 문제를 해결했습니다.
- 중복 쓰기를 실행하는 데이터 조작 언어(DML) 문에서 과도한 오류 로깅이 발생하고 쿼리 지연 시간이 길어지는 문제를 해결했습니다.
- 표준 시간대 정의를 IANA 2023c 버전으로 업그레이드했습니다.
- 세션 수준 바이너리 로깅 활성화 및 비활성화에 대한 지원을 추가했습니다. Amazon Aurora 사용 설명서의 저장 프로시저 - 복제를 참조하세요.
- 세션 수준 바이너리 로그 형식 설정에 대한 지원을 추가했습니다. Amazon Aurora 사용 설명서의 저장 프로시저 - 복제를 참조하세요.
- `aurora_disable_hash_join` 파라미터를 `1` 또는 `ON`으로 설정하면 최적화 프로그램이 해시 조인을 사용하는 것을 방지할 수 없는 문제를 해결했습니다.
- `GROUP BY` 절과 `aurora_parallel_query` 파라미터가 `ON`으로 설정된 상태에서 `SELECT` 쿼리를 실행할 때 부정확한 결과가 반환될 수 있는 인덱스 스캔 관련 문제를 해결했습니다.
- 드문 경우지만 라이터 인스턴스에서 대규모 업데이트 또는 데이터 정의 언어(DDL) 작업이 동시에 실행되는 테이블에 액세스할 때 Amazon Aurora 리더 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- `information_schema` 지표에서 `buffer_pool_read_requests` 카운터가 잘못 보고될 수 있는 문제를 해결했습니다.
- 소스의 시스템 변수 server uuid가 누락되었거나 잘못된 값을 갖는 경우 binlog 복제본이 다시 시작될 수 있는 문제를 해결했습니다.
- InnoDB 통계가 오래되어 쿼리 실행 계획이 최적화되지 않음으로 인해 쿼리 실행 시간이 길어질 수 있는 문제를 해결했습니다.
- `AuroraGlobalDBRPOLag` CloudWatch 지표가 사용자 워크로드에 관계없이 항상 0으로 표시되는 문제를 해결했습니다.

업그레이드 및 마이그레이션:

- Aurora MySQL 버전 2.07 또는 2.11에서 Aurora MySQL 버전 2.12 이상으로 마이너 버전 업그레이드를 수행하려면 엔진 버전을 수정하여 Aurora MySQL 업그레이드를 참조하세요.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 아래 사항 외에도 5.7.40까지의 모든 커뮤니티 버그 수정이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 2.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그를 참조하세요.

- 백그라운드 TLS 인증서 교체로 인해 CPU 사용률이 높아질 수 있는 문제를 해결했습니다. (커뮤니티 버그 수정 #34284186)

Aurora MySQL 버전 2에서 지원되지 않는 기능

다음 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환)에서는 현재 지원되지 않습니다.

- 배치화 스캔.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- `CREATE TABLESPACE` SQL 문
- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- X 프로토콜

Aurora MySQL 데이터베이스 엔진 업데이트 2024년 7월 19일(버전 2.11.6, MySQL 5.7.12와 호환) - RDS 추가 지원 버전

버전: 2.11.6

Aurora MySQL 2.11.6이 정식 버전입니다. Aurora MySQL 2.11 버전은 MySQL 5.7.12와 호환됩니다. 커뮤니티 변경 사항에 대한 자세한 내용은 Changes in MySQL 5.7.12(2016-04-11, General Availability)를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.*, 3.04.*, 3.05.*, 3.06.* 및 3.07.*입니다.

기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.11.6로 업그레이드할 수 있습니다. 또한 현재 지원되는 Aurora MySQL 버전 2 이하 릴리스에서 Aurora MySQL 2.11.6로 스냅샷을 복원할 수 있습니다.

Aurora MySQL Global Database를 버전 2.11.*로 업그레이드하려는 경우 기본 및 보조 DB 클러스터를 패치 수준을 포함하여 정확히 동일한 버전으로 업그레이드해야 합니다. Aurora 글로벌 데이터베이스의 마이너 버전 업그레이드에 대한 자세한 내용은 마이너 버전 업그레이드를 참조하세요.

Aurora MySQL 2.11.*으로의 인플레이스 엔진 버전 업그레이드가 수행된 직후, 해당 인스턴스가 이전 운영 체제 버전을 실행하는 경우 db.r4, db.r5, db.t2 및 db.t3 DB 인스턴스 클래스의 영향을 받는 모든 인스턴스에 운영 체제 업그레이드가 자동으로 적용됩니다. 다중 AZ DB 클러스터에서는 모든 리더 인스턴스에 먼저 운영 체제 업그레이드가 적용됩니다. 첫 번째 리더 인스턴스의 운영 체제 업그레이드가 완료되면 장애 조치가 수행되고 이전 라이터 인스턴스가 업그레이드됩니다.

> **Note**
>
> 메이저 버전 업그레이드 중에는 Aurora 글로벌 데이터베이스에 운영 체제 업그레이드가 자동으로 적용되지 않습니다.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

보안 문제 및 CVE 해결:

- MySQL 저장 프로시저의 보안 문제를 수정했습니다.

이 릴리스에는 MySQL 5.7.12까지의 모든 커뮤니티 CVE 수정 사항이 포함되어 있습니다. 이 릴리스에는 다음 CVE 수정 사항이 포함되어 있습니다.

- CVE-2023-21912
- CVE-2023-38545
- CVE-2023-39975
- CVE-2023-44487
- CVE-2024-0853
- CVE-2024-20963
- CVE-2024-20963
- CVE-2024-20998
- CVE-2024-21008
- CVE-2024-21009
- CVE-2024-21013
- CVE-2024-21047
- CVE-2024-21054
- CVE-2024-21055
- CVE-2024-21057
- CVE-2024-21062
- CVE-2024-21069
- CVE-2024-21096
- CVE-2024-21097

가용성 향상:

- 원활한 조정, 제로 가동 중지 재시작(ZDR) 및 제로 가동 중지 패치 적용(ZDP) 중에 연결 리소스의 동시 액세스로 인해 데이터베이스 서버가 다시 시작될 수 있는 문제를 수정했습니다.
- 로그 애플리케이션에 사용되는 메모리를 비울 때 리더 DB 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 병렬 쿼리를 실행할 때 리더 DB 인스턴스가 다시 시작되는 문제를 수정했습니다.
- 전달된 암시적 커밋 문에 오류가 발생할 때 쓰기 전달을 사용하는 리더 DB 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.

일반적인 개선 사항

- `AUTO_INCREMENT` 프라이머리 키 열과 고유 키 열이 있는 테이블에서 동시 `INSERT` 문을 수행할 때와 `INSERT` 문에 서로 다른 행에 고유한 키 위반이 있을 때 SQL 문에 예기치 않은 프라이머리 키 위반 오류 또는 경고가 발생할 수 있는 문제를 수정했습니다.
- ZDR이 쿼리에서 힌트로 설정된 세션 변수를 잘못 복원할 때 잘못된 쿼리 결과가 발생할 수 있는 문제를 수정했습니다.
- Aurora Serverless v1 조정 중 조정점을 찾는 동안 내부 데이터 구조에 대한 잘못된 액세스로 인해 DB 인스턴스가 다시 시작되는 문제를 수정했습니다.
- 느린 `SELECT` 쿼리가 선행되지 않는 한 MySQL Event Scheduler에서 실행하는 느린 `INSERT`, `DELETE` 및 `UPDATE` 쿼리가 느린 쿼리 로그에 기록되지 않는 문제를 수정했습니다.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 MySQL 5.7.12까지의 모든 커뮤니티 버그 수정 사항이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 2.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그 섹션을 참조하세요.

Aurora MySQL 버전 2에서 지원되지 않는 기능

다음 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환)에서는 현재 지원되지 않습니다.

- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2024년 3월 26일(버전 2.11.5, MySQL 5.7.12와 호환) - RDS 추가 지원 버전(기본)

버전: 2.11.5

Aurora MySQL 2.11.5가 정식 버전입니다. Aurora MySQL 2.11 버전은 MySQL 5.7.12와 호환됩니다. 커뮤니티 변경 사항에 대한 자세한 내용은 Changes in MySQL 5.7.12(2016-04-11, General Availability)를 참조하세요.

버전 2.11.5는 DB 클러스터를 생성할 때 Aurora MySQL 버전 2의 현재 기본 버전입니다. 자세한 내용은 기본 Amazon Aurora 버전을 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.*, 3.04.*, 3.05.* 및 3.06.*입니다.

기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.11.5로 업그레이드할 수 있습니다. 또한 현재 지원되는 Aurora MySQL 버전 2 이하 릴리스에서 Aurora MySQL 2.11.5로 스냅샷을 복원할 수 있습니다.

Aurora MySQL Global Database를 버전 2.11.*로 업그레이드하려는 경우 기본 및 보조 DB 클러스터를 패치 수준을 포함하여 정확히 동일한 버전으로 업그레이드해야 합니다. Aurora 글로벌 데이터베이스의 마이너 버전 업그레이드에 대한 자세한 내용은 마이너 버전 업그레이드를 참조하세요.

Aurora MySQL 2.11.*으로의 인플레이스 엔진 버전 업그레이드가 수행된 직후, 해당 인스턴스가 이전 운영 체제 버전을 실행하는 경우 db.r4, db.r5, db.t2 및 db.t3 DB 인스턴스 클래스의 영향을 받는 모든 인스턴스에 운영 체제 업그레이드가 자동으로 적용됩니다. 다중 AZ DB 클러스터에서는 모든 리더 인스턴스에 먼저 운영 체제 업그레이드가 적용됩니다. 첫 번째 리더 인스턴스의 운영 체제 업그레이드가 완료되면 장애 조치가 수행되고 이전 라이터 인스턴스가 업그레이드됩니다.

> **Note**
>
> 메이저 버전 업그레이드 중에는 Aurora 글로벌 데이터베이스에 운영 체제 업그레이드가 자동으로 적용되지 않습니다.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

보안 문제 및 CVE 해결:

이 릴리스에는 다음 CVE 수정 사항이 포함되어 있습니다.

- 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

가용성 향상:

- Aurora 스토리지와 통신하는 구성 요소의 결함으로 인해 Aurora MySQL 라이터 DB 인스턴스가 장애 조치될 수 있는 문제를 수정했습니다. 결함은 소프트웨어 업데이트 후 DB 인스턴스와 기본 스토리지 간의 통신이 분석된 결과로 발생합니다.
- 드문 경우지만 리더 DB 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 감사 로깅 스레드로 인해 발생하는 잠금 경합에 따라 CPU 사용률이 높아지고 클라이언트 애플리케이션 제한 시간이 초과될 수 있는 문제를 해결했습니다.

일반적인 개선 사항

- Aurora DB 클러스터 볼륨에서 데이터를 읽는 동안 일시적인 네트워크 문제로 인해 병렬 쿼리가 실패할 수 있는 문제를 해결했습니다.
- 다운로드 또는 교체 시 로그 파일에 액세스할 수 없고 경우에 따라 CPU 사용량이 증가할 수 있는 감사 로그 파일 관리 관련 문제를 해결했습니다.
- 쓰기 전달을 사용할 때 `information_schema` 및 `performance_schema` 글로벌 상태 테이블에서 `Threads_running` 변수의 값이 잘못 생성될 수 있는 문제를 해결했습니다.

업그레이드 및 마이그레이션:

- RDS for MySQL 5.7에서 마이그레이션된 Aurora MySQL DB 클러스터에서 이진 로그 복제를 시작하지 못하는 문제를 수정했습니다.
- Aurora MySQL 버전 3으로 메이저 버전을 업그레이드하는 동안 데이터베이스 이벤트 스케줄러를 비활성화했습니다. 이렇게 하면 메이저 버전 업그레이드가 진행되는 동안 이벤트 실행을 통해 데이터베이스가 변경되는 것을 방지할 수 있습니다.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 아래 사항 외에도 5.7.12까지의 모든 커뮤니티 버그 수정이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 2.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그를 참조하세요.

Aurora MySQL 버전 2에서 지원되지 않는 기능

다음 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환)에서는 현재 지원되지 않습니다.

- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2023년 10월 17일(버전 2.11.4, MySQL 5.7.12와 호환) - RDS 추가 지원 버전

버전: 2.11.4

Aurora MySQL 2.11.4가 정식 출시되었습니다. Aurora MySQL 2.11 버전은 MySQL 5.7.12와 호환됩니다. 커뮤니티 변경 사항에 대한 자세한 내용은 Changes in MySQL 5.7.12(2016-04-11, General Availability)를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.* 및 3.04.* 입니다.

기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.111.4로 업그레이드할 수 있습니다. 또한 현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.14.4로 스냅샷을 복원할 수 있습니다.

Aurora MySQL Global Database를 버전 2.11.*로 업그레이드하려는 경우 기본 및 보조 DB 클러스터를 패치 수준을 포함하여 정확히 동일한 버전으로 업그레이드해야 합니다. Aurora 글로벌 데이터베이스의 마이너 버전 업그레이드에 대한 자세한 내용은 마이너 버전 업그레이드를 참조하세요.

Aurora MySQL 2.11.*으로의 인플레이스 엔진 버전 업그레이드가 수행된 직후, 해당 인스턴스가 이전 운영 체제 버전을 실행하는 경우 db.r4, db.r5, db.t2 및 db.t3 DB 인스턴스 클래스의 영향을 받는 모든 인스턴스에 운영 체제 업그레이드가 자동으로 적용됩니다. 다중 AZ DB 클러스터에서는 모든 리더 인스턴스에 먼저 운영 체제 업그레이드가 적용됩니다. 첫 번째 리더 인스턴스의 운영 체제 업그레이드가 완료되면 장애 조치가 수행되고 이전 라이터 인스턴스가 업그레이드됩니다.

> **Note**
>
> 메이저 버전 업그레이드 중에는 Aurora 글로벌 데이터베이스에 운영 체제 업그레이드가 자동으로 적용되지 않습니다.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

아래 나열된 보안 문제 및 CVE 해결:

- 감사 로그 교체를 처리하는 동안 보고된 이벤트가 감사 로그에 기록되지 않는 문제를 해결했습니다.
- CVE-2022-24407

가용성 향상:

- 병렬 쿼리를 사용하는 Aurora MySQL 데이터베이스 인스턴스에서 많은 수의 동시 병렬 쿼리를 실행할 때 데이터베이스가 다시 시작될 수 있는 문제를 해결했습니다.
- I/O 집약적인 읽기 워크로드를 실행하는 동안 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 삭제된 테이블에 속한 데이터베이스 페이지를 읽으려고 할 때 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 라이터 인스턴스에서 데이터베이스 볼륨이 160GB의 배수로 증가하면 리더 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 데이터베이스가 내부 시스템 테이블에서 트리거를 생성하거나 삭제하는 동안 라이터 인스턴스가 다시 시작되면 데이터베이스 클러스터를 사용할 수 없게 되는 문제를 해결했습니다.
- 전체 텍스트 인덱스가 포함된 테이블에 대해 데이터 조작 언어(DML) 쿼리를 실행할 때 리더 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- Aurora 병렬 쿼리 실행 계획을 사용하는 쿼리를 실행하는 동안 리더 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 전체 텍스트 검색(FTS) 인덱스가 있는 테이블에 대해 `OPTIMIZE TABLE` 쿼리를 실행하는 동안 라이터 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- `INSERT INTO SELECT` 및 `FROM`과 같은 쿼리를 실행할 때 불일치를 일으킬 수 있는 문제로 인해 이 Aurora MySQL 버전에서는 빠른 삽입이 활성화되지 않습니다. 빠른 삽입 최적화에 대한 자세한 내용은 Amazon Aurora MySQL 성능 향상을 참조하세요.

일반적인 개선 사항

- 2.11.* 미만 버전에서 업그레이드한 후 소규모 읽기 전용 복제본 인스턴스에서 복제 지연이 증가할 수 있는 문제를 해결했습니다.
- 저장 루틴과 관련된 요청을 확인하기 위해 procs_priv 권한 부여 테이블을 참조할 때 과도한 로그 메시지가 생성될 수 있는 문제를 해결했습니다.

- 해시 조인 최적화를 사용하여 쿼리를 실행하는 동안 데이터베이스 인스턴스가 과도한 메모리를 사용할 수 있게 되는 메모리 관리 문제를 해결했습니다.
- 파티셔닝된 테이블(이전 `ha_partition` 파티션 핸들러를 지원하는 MySQL 버전에서 생성됨)을 사용하여 `SELECT` 문을 실행하고 쿼리 플래너가 병렬 쿼리를 선택할 때 데이터베이스가 다시 시작되는 문제를 해결했습니다.
- 쓰기 전달이 활성화된 경우 데이터베이스에 새 클라이언트 연결이 설정되지 않는 문제를 해결했습니다.
- Aurora MySQL binlog 복제본이 `USE` 명령으로 정의된 기본 데이터베이스 없이 소스의 binlog 파일에 기록된 `QUERY` 이벤트를 실행할 때 바이너리 로그(binlog) 복제 지연을 줄였습니다.
- `GROUP BY` 절과 `aurora_parallel_query` 파라미터가 `ON`으로 설정된 상태에서 `SELECT` 쿼리를 실행할 때 부정확한 결과가 반환될 수 있는 인덱스 스캔 관련 문제를 해결했습니다.
- 세션 수준 바이너리 로깅 활성화 및 비활성화에 대한 지원을 추가했습니다. Amazon Aurora 사용 설명서의 저장 프로시저 - 복제를 참조하세요.
- 소스의 시스템 변수 server_uuid가 누락되었거나 잘못된 값을 갖는 경우 binlog 복제본이 다시 시작될 수 있는 문제를 해결했습니다.
- 세션 수준 바이너리 로그 형식 설정에 대한 지원을 추가했습니다. Amazon Aurora 사용 설명서의 저장 프로시저 - 복제를 참조하세요.
- `innodb_flush_log_at_trx_commit` 파라미터가 1로 설정되지 않은 경우 `CommitLatency` CloudWatch 지표가 잘못 보고될 수 있는 문제를 해결했습니다.
- InnoDB 통계가 오래되어 쿼리 실행 계획이 최적화되지 않음으로 인해 쿼리 실행 시간이 길어질 수 있는 문제를 해결했습니다.
- 연결된 바이너리 로그(binlog) 소비자가 중복된 binlog 복제 서버 ID를 사용할 때 데이터베이스가 다시 시작될 수 있는 문제를 해결했습니다.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 아래 사항 외에도 5.7.12까지의 모든 커뮤니티 버그 수정이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 2.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그를 참조하세요.

- 복제: 일부 바이너리 로그 이벤트가 경우에 따라 올바르게 처리되지 않았습니다. (버그 #34617506)
- 백그라운드 TLS 인증서 교체로 인해 CPU 사용률이 높아질 수 있는 문제를 해결했습니다. (커뮤니티 버그 수정 #34284186)
- 준비된 문에서 일부 유형의 하위 쿼리로 인해 서버가 종료될 수 있습니다. (버그 #33100586)

Aurora MySQL 버전 2에서 지원되지 않는 기능

다음 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환)에서는 현재 지원되지 않습니다.

- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2023년 6월 9일(버전 2.11.3, MySQL 5.7.12와 호환) - RDS 추가 지원 버전

버전: 2.11.3

Aurora MySQL 2.11.3이 정식 출시되었습니다. Aurora MySQL 2.11 버전은 MySQL 5.7.12와 호환됩니다. 커뮤니티 변경 사항에 대한 자세한 내용은 Changes in MySQL 5.7.12(2016-04-11, General Availability)를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 2.07.*, 2.11.*, 3.01.*, 3.02.* 및 3.03.*입니다.

기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.11.3으로 업그레이드할 수 있습니다. 또한 현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.11.3으로 스냅샷을 복원할 수 있습니다.

Aurora MySQL Global Database를 버전 2.11.*로 업그레이드하려는 경우 기본 및 보조 DB 클러스터를 패치 수준을 포함하여 정확히 동일한 버전으로 업그레이드해야 합니다. Aurora 글로벌 데이터베이스의 마이너 버전 업그레이드에 대한 자세한 내용은 마이너 버전 업그레이드를 참조하세요.

Aurora MySQL 2.11.*으로의 인플레이스 엔진 버전 업그레이드가 수행된 직후, 해당 인스턴스가 이전 운영 체제 버전을 실행하는 경우 db.r4, db.r5, db.t2 및 db.t3 DB 인스턴스 클래스의 영향을 받는 모든 인스턴스에 운영 체제 업그레이드가 자동으로 적용됩니다. 다중 AZ DB 클러스터에서는 모든 리더 인스턴스에 먼저 운영 체제 업그레이드가 적용됩니다. 첫 번째 리더 인스턴스의 운영 체제 업그레이드가 완료되면 장애 조치가 수행되고 이전 라이터 인스턴스가 업그레이드됩니다.

> **Note**
>
> 메이저 버전 업그레이드 중에는 Aurora 글로벌 데이터베이스에 운영 체제 업그레이드가 자동으로 적용되지 않습니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

아래 나열된 보안 문제 및 CVE 해결:

- 보안 수준이 낮은 `DES-CBC3-SHA` 값을 SSL_CIPHER 데이터베이스 파라미터에서 제외하도록 Aurora MySQL에서 사용하는 기본 SSL 암호를 업데이트했습니다. `DES-CBC3-SHA` 암호 제거로 인

해 SSL 연결 문제가 발생하는 경우 ConfiguringCipherSuites 목록의 적용 가능한 보안 암호를 사용하세요. MySQL 클라이언트 연결 암호 구성에 대한 자세한 내용은 MySQL 설명서에서 확인할 수 있습니다.

- CVE-2023-21963
- CVE-2023-21912
- CVE-2023-0215
- CVE-2022-43551
- CVE-2022-37434

가용성 향상:

- 데이터베이스 활동 스트림(DAS) 이벤트 암호화에서 데이터베이스 다시 시작을 일으킬 수 있는 문제를 해결했습니다.
- 데이터 정의 언어(DDL) 쿼리를 실행하는 동안 데이터베이스가 다시 시작될 경우 다시 시작의 실패를 유발할 수 있는 두 가지 문제를 해결했습니다.
- `INSERT INTO SELECT` 및 `FROM`과 같은 쿼리를 실행할 때 불일치를 일으킬 수 있는 문제로 인해 이 Aurora MySQL 버전에서는 빠른 삽입이 활성화되지 않습니다. 빠른 삽입 최적화에 대한 자세한 내용은 Amazon Aurora MySQL 성능 향상을 참조하세요.

일반적인 개선 사항

- 릴레이 로그 파일에 쓸 때 경합을 줄일 수 있도록 binlog 복제본에 파일 관리 성능 최적화 기능을 도입했습니다.
- `aurora_disable_hash_join` 파라미터를 1 또는 ON으로 설정하면 최적화 프로그램이 해시 조인을 사용하는 것을 방지할 수 없는 문제를 해결했습니다.
- `information_schema` 지표에서 `buffer_pool_read_requests` 카운터가 잘못 보고될 수 있는 문제를 해결했습니다.
- `LOAD FROM S3` 또는 `SELECT INTO S3` 작업을 수행할 때 로컬 스토리지가 가득 찰 수 있게 되는 문제를 해결했습니다. 또한 이 문제로 인해 CPU 사용률이 높아지고, 메모리 부족으로 인해 데이터베이스가 다시 시작되고, 이러한 쿼리의 지연 시간이 늘어날 수 있습니다.

Aurora MySQL 버전 2에서 지원되지 않는 기능

다음 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환)에서는 현재 지원되지 않습니다.

- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2023년 3월 24일(버전 2.11.2, MySQL 5.7.12와 호환) - RDS 추가 지원 버전

버전: 2.11.2

Aurora MySQL 2.11.2가 정식 출시되었습니다. Aurora MySQL 2.11 버전은 MySQL 5.7.12와 호환됩니다. 커뮤니티 변경 사항에 대한 자세한 내용은 Changes in MySQL 5.7.12(2016-04-11, General Availability)를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 2.07.*, 2.11.*, 3.01.*, 3.02.* 및 3.03.*입니다.

기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.11.2로 업그레이드할 수 있습니다. 또한 현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.11.2로 스냅샷을 복원할 수 있습니다.

Aurora MySQL Global Database를 버전 2.11.*로 업그레이드하려는 경우 기본 및 보조 DB 클러스터를 패치 수준을 포함하여 정확히 동일한 버전으로 업그레이드해야 합니다. Aurora 글로벌 데이터베이스의 마이너 버전 업그레이드에 대한 자세한 내용은 마이너 버전 업그레이드를 참조하세요.

Aurora MySQL 2.11.*으로의 인플레이스 엔진 버전 업그레이드가 수행된 직후, 해당 인스턴스가 이전 운영 체제 버전을 실행하는 경우 db.r4, db.r5, db.t2 및 db.t3 DB 인스턴스 클래스의 영향을 받는 모든 인스턴스에 운영 체제 업그레이드가 자동으로 적용됩니다. 다중 AZ DB 클러스터에서는 모든 리더 인스턴스에 먼저 운영 체제 업그레이드가 적용됩니다. 첫 번째 리더 인스턴스의 운영 체제 업그레이드가 완료되면 장애 조치가 수행되고 이전 라이터 인스턴스가 업그레이드됩니다.

> **Note**
>
> 메이저 버전 업그레이드 중에는 Aurora 글로벌 데이터베이스에 운영 체제 업그레이드가 자동으로 적용되지 않습니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

일반적인 개선 사항

- 여러 바이너리 로그 복제 소비자가 연결된 경우 바이너리 로그 복제를 사용하는 DB 인스턴스에서 CPU 사용률이 증가하고 연결 실패가 발생할 수 있는 문제를 해결했습니다.

- 기본 데이터베이스 라이터가 Aurora MySQL 버전 2.10을 사용하는 경우 Aurora MySQL 버전 2.11로 업그레이드한 후 글로벌 데이터베이스 보조 리전의 리더 인스턴스가 동기화되지 않을 수 있는 문제를 해결했습니다.

가용성 향상:

- `INSERT INTO SELECT` 및 `FROM`과 같은 쿼리를 실행할 때 불일치를 일으킬 수 있는 문제로 인해 이 Aurora MySQL 버전에서는 빠른 삽입이 활성화되지 않습니다. 빠른 삽입 최적화에 대한 자세한 내용은 Amazon Aurora MySQL 성능 향상을 참조하세요.

Aurora MySQL 버전 2에서 지원되지 않는 기능

다음 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환)에서는 현재 지원되지 않습니다.

- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링

- CREATE TABLESPACE SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2023년 2월 14일(버전 2.11.1, MySQL 5.7.12와 호환) - RDS 추가 지원 버전

버전: 2.11.1

Aurora MySQL 2.11.1이 정식 출시되었습니다. Aurora MySQL 2.11 버전은 MySQL 5.7.12와 호환됩니다. 커뮤니티 변경 사항에 대한 자세한 내용은 Changes in MySQL 5.7.12(2016-04-11, General Availability)를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.07.*, 2.09.*, 2.10.*, 2.11.*, 3.01.* 및 3.02.*입니다.

기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.11.1로 업그레이드할 수 있습니다. Aurora MySQL 버전 1을 실행하는 클러스터의 경우 기존 Aurora MySQL 1.23 이상의 클러스터를 2.11.1로 직접 업그레이드할 수 있습니다. 또한 현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.11.1로 스냅샷을 복원할 수 있습니다.

Aurora MySQL Global Database를 버전 2.11.*로 업그레이드하고 쓰기 전달을 활성화한 경우 쓰기 전달을 계속 사용하려면 기본 및 보조 DB 클러스터를 패치 수준을 포함하여 정확히 동일한 버전으로 업그레이드해야 합니다. Aurora 글로벌 데이터베이스의 마이너 버전 업그레이드에 대한 자세한 내용은 마이너 버전 업그레이드를 참조하세요.

Aurora MySQL 2.11.*으로의 인플레이스 엔진 버전 업그레이드가 수행된 직후, 해당 인스턴스가 이전 운영 체제 버전을 실행하는 경우 db.r4, db.r5, db.t2 및 db.t3 DB 인스턴스 클래스의 영향을 받는 모든 인스턴스에 운영 체제 업그레이드가 자동으로 적용됩니다. 다중 AZ DB 클러스터에서는 모든 리더 인스턴스에 먼저 운영 체제 업그레이드가 적용됩니다. 첫 번째 리더 인스턴스의 운영 체제 업그레이드가 완료되면 장애 조치가 수행되고 이전 라이터 인스턴스가 업그레이드됩니다.

> **Note**
>
> 메이저 버전 업그레이드 중에는 Aurora 글로벌 데이터베이스에 운영 체제 업그레이드가 자동으로 적용되지 않습니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

아래 나열된 보안 문제 및 CVE 해결:

관리된 환경에서 처리를 미세 조정하기 위한 수정 사항 및 기타 향상된 기능 아래의 추가 CVE 수정:

- CVE-2022-32221
- CVE-2021-36222
- CVE-2021-22926
- CVE-2021-2169

가용성 향상:

- 연결 급증으로 인해 쿼리 지연 시간이 증가하거나 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 드문 경우지만 라이터 인스턴스의 대규모 업데이트 작업 또는 데이터 정의 언어(DDL) 워크로드와 Aurora 복제본의 동일한 테이블 세트에 대한 읽기 작업을 동시에 실행하는 동안 Aurora 복제본이 다시 시작될 수 있는 문제를 해결했습니다.
- 연결 급증으로 인해 연결 설정 프로세스가 완료되는 데 더 오랜 시간이 걸리거나 시간 초과 오류가 발생하여 연결 설정이 실패할 수 있는 문제를 해결했습니다.
- 고급 감사 로그 교체로 인해 사용 가능한 메모리가 줄어들어 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- `INSERT INTO SELECT` 및 `FROM`과 같은 쿼리를 실행할 때 불일치를 일으킬 수 있는 문제로 인해 이 Aurora MySQL 버전에서는 빠른 삽입이 활성화되지 않습니다. 빠른 삽입 최적화에 대한 자세한 내용은 Amazon Aurora MySQL 성능 향상을 참조하세요.

일반적인 개선 사항

- SSL 서버 상태 변수가 채워지지 않는 문제를 해결했습니다.

- 중복 쓰기를 실행하는 데이터 조작 언어(DML) 문에서 과도한 오류 로깅이 발생하고 쿼리 지연 시간이 길어지는 문제를 해결했습니다.

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- AWS Lambda 함수를 비동기식으로 호출하기 위한 네이티브 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링

- CREATE TABLESPACE SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2022년 10월 25일(버전 2.11.0, MySQL 5.7.12와 호환) - RDS 추가 지원 버전

버전: 2.11.0

Aurora MySQL 2.11.0이 정식 출시되었습니다. Aurora MySQL 2.x 버전은 MySQL 5.7.12와 호환됩니다. 커뮤니티 변경 사항에 대한 자세한 내용은 Changes in MySQL 5.7.12(2016-04-11, General Availability)를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 2.11.*, 3.01.* 및 3.02.*입니다.

기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.11.0으로 업그레이드할 수 있습니다. Aurora MySQL 버전 1을 실행하는 클러스터의 경우 기존 Aurora MySQL 1.23 이상의 클러스터를 2.11.0으로 직접 업그레이드할 수 있습니다. 또한 현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.11.0으로 스냅샷을 복원할 수 있습니다.

Aurora MySQL Global Database를 버전 2.11.*로 업그레이드하고 쓰기 전달을 활성화한 경우 쓰기 전달을 계속 사용하려면 기본 및 보조 DB 클러스터를 패치 수준을 포함하여 정확히 동일한 버전으로 업그레이드해야 합니다. Aurora 글로벌 데이터베이스의 마이너 버전 업그레이드에 대한 자세한 내용은 마이너 버전 업그레이드를 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

보안 문제 및 CVE 해결:

- SSL/TLS 공급자를 OpenSSL에서 AWS-LC로 변경했습니다.

  이렇게 하면 덜 안전한 DHE-RSA-* 암호에 대한 지원 제거를 포함하되 이에 국한되지 않는 여러 가지 변경 사항이 발생합니다.

  자세한 내용은 Aurora MySQL DB 클러스터에서 TLS 사용을 참조하세요.

이 릴리스에는 다음 CVE 수정 사항이 포함되어 있습니다.

- 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

새로운 기능:

- Aurora MySQL 버전 2.11이 릴리스됨에 따라 새로운 운영 체제 업그레이드를 사용할 수 있습니다. 버전 2.11로 업그레이드한 후 보류 중인 이 OS 업데이트를 모든 Aurora MySQL 데이터베이스 인스턴스에 적용하는 것이 좋습니다. 자세한 내용은 운영 체제 업데이트 작업을 참조하세요.
- 새로운 동적 구성 옵션인 innodb_deadlock_detect를 사용하여 교착 상태 감지를 비활성화할 수 있습니다. 동시성이 높은 시스템에서 수많은 스레드가 동일한 잠금을 기다릴 때 교착 감지로 인해 속도가 느려질 수 있습니다. 때로는 교착 상태 감지를 비활성화하고 교착 상태가 발생할 경우 innodb_lock_wait_timeout 설정을 사용하여 트랜잭션을 롤백하는 것이 더 효율적일 수 있습니다. (버그 #23477773) Innodb 교착 상태 감지에 대한 자세한 내용은 MySQL 설명서에서 확인할 수 있습니다.
- MySQL 8.0의 `UUID_TO_BIN`, `BIN_TO_UUID` 및 `IS_UUID` 함수를 추가했습니다. 이러한 함수를 사용하는 방법에 대한 자세한 내용은 MySQL Miscellaneous function에서 확인할 수 있습니다.
- 테이블별 또는 쿼리별로 Aurora MySQL 병렬 쿼리를 활성화하거나 비활성화할 수 있도록 최적화 프로그램 힌트에 대한 지원을 추가했습니다.
  - Amazon Aurora MySQL용 Parallel Query 처리
  - Aurora MySQL 힌트
- R3 인스턴스 유형 지원을 제거했습니다.
- R6i 인스턴스에 대한 지원을 추가했습니다.

가용성 향상:

- 오류 로그에 잘못된 binlog 파일 및 위치가 기록되어 데이터베이스 클러스터에서 크로스 리전 논리적 복제를 수행할 수 없게 되는 문제를 해결했습니다. 이 문제는 DDL 문을 실행한 후 엔진을 다시 시작할 때 발생할 수 있습니다.
- 드문 경우지만 라이터 인스턴스에서 GRANT 및 FLUSH와 같은 액세스 제어 목록(ACL) 문을 실행할 때 Aurora 리더 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다. 이 문제는 많은 수의 사용자와 ACL 작업(예: 권한 변경)이 있는 리더 인스턴스에 영향을 미칠 가능성이 높습니다.
- 드문 경우지만 다른 트랜잭션에 의해 삭제되는 행에 트랜잭션이 액세스할 때 라이터 인스턴스가 다시 시작되거나 장애 조치될 수 있는 문제를 해결했습니다.
- 전체 텍스트 인덱스가 있는 테이블에서 구문을 검색하는 데 걸리는 시간을 크게 줄일 수 있도록 전체 텍스트 구문 검색 성능을 개선했습니다.
- 라이터 인스턴스가 재시작된 후 느린 복구 상태에서 멈췄다가 다시 재시작되는 문제를 해결했습니다. 이 문제는 처음 재시작될 때 데이터베이스에 커밋되지 않은 행이 많을 경우 발생합니다.

- 드문 경우지만 교착 상태 감지기 스레드가 중단되면 긴 세마포어 대기로 인해 데이터베이스 서버가 다시 시작될 수 있는 문제를 해결했습니다.
- 드문 경우지만 I/O 스레드가 교착 상태에 빠지면 긴 세마포어 대기로 인해 데이터베이스가 다시 시작될 수 있는 문제를 해결했습니다.
- `INSERT INTO SELECT` 및 `FROM`과 같은 쿼리를 실행할 때 불일치를 일으킬 수 있는 문제로 인해 이 Aurora MySQL 버전에서는 빠른 삽입이 활성화되지 않습니다. 빠른 삽입 최적화에 대한 자세한 내용은 Amazon Aurora MySQL 성능 향상을 참조하세요.

일반적인 개선 사항

- 다음 조건이 모두 충족될 때 데이터베이스 서버가 다시 시작될 수 있는 문제를 해결했습니다.
  - ALLOW_INVALID_DATES가 SQL MODE에서 비활성화되어 있습니다.
  - 데이터베이스 서버가 DATETIME 유형의 값이 잘못된(월이 1에서 12 사이가 아님) INSERT, UPDATE, DELETE 또는 SELECT 문을 처리 중입니다.
- log-bin이 OFF로 설정된 경우 바이너리 로그 보존 기간이 준수되지 않아 스토리지 사용률이 예상보다 높아지는 문제를 해결했습니다. 이 수정 후에는 보존 기간에 따라 바이너리 로그가 제거됩니다. 바이너리 로그 보존 기간을 구성하는 방법에 대한 자세한 내용은 Aurora MySQL 사용 설명서에서 확인할 수 있습니다.
- GRANT, FLUSH PRIVILEGES 등과 같은 특정 데이터 제어 언어(DCL) SQL 문이 해당 인스턴스에서 실행될 때 데이터베이스 인스턴스에서 사용 가능한 메모리가 줄어들 수 있는 문제를 해결했습니다. 이러한 문을 자주 사용하면 사용 가능한 메모리가 계속 줄어들고 메모리 부족 문제로 인해 데이터베이스 인스턴스가 다시 시작될 수 있습니다. 라이터 인스턴스에서 이러한 문을 사용하면 리더 인스턴스에서 사용 가능한 메모리가 줄어들 수도 있습니다.
- 읽기 I/O 작업 수를 최소화하여 I/O 스레드와 SQL 스레드 간의 경합을 줄이기 위해 릴레이 로그에서 수행되는 읽기에 대해 더 큰 읽기 버퍼 크기를 도입했습니다.
- "Table 'mysql.slow_log_backup' doesn't exist"라는 오류 메시지와 함께 mysql.rds_rotate_slow_log 저장 프로시저가 실패할 수 있는 문제를 해결했습니다.
- 과도한 쿼리 캐시 무효화로 인해 읽기 전용 복제본이 쿼리 캐시 대신 디스크에서 데이터를 읽어야 하기 때문에 읽기 전용 복제본의 CPU 사용량과 지연 시간이 예상보다 높아지는 문제를 해결했습니다.
- 리더 인스턴스에서 INSTALL PLUGIN 및 UNINSTALL PLUGIN 명령을 실행할 수 있도록 허용하여 LOCK_plugin, LOCK_system_variables_hash, LOCK_global_system_variables에서 교착 상태가 발생할 수 있는 문제를 해결했습니다. 이제 데이터베이스 클러스터의 라이터 인스턴스에서만 이러한 명령문을 실행할 수 있습니다.

- 바이너리 로깅이 활성화되어 있을 때 클러스터의 커밋 지연 시간이 예상보다 길어질 수 있는 문제를 해결했습니다. 이는 대규모 binlog 이벤트(500MB 초과)를 생성하는 모든 트랜잭션에 영향을 줍니다.
- INFORMATION_SCHEMA.INNODB_METRICS 테이블의 trx_active_transactions 지표에 잘못된 값이 있을 수 있는 문제를 해결했습니다.
- 대규모 트랜잭션의 경우 저장점으로 롤백을 실행하는 동안 binlog 파일이 일관되지 않아 논리적 복제가 중지될 수 있는 문제를 해결했습니다.
- 기본적으로 일관된 마스크 보안 암호를 사용하여 general-log, slow-query-log 및 audit-log에서 보안 인증 해시를 마스킹했습니다. 이는 aurora_mask_password_hashes_type 파라미터를 통해 구성할 수 있습니다.
- 고객이 관찰하는 이벤트에서 제로 가동 중지 다시 시작(ZDR) 기간이 잘못 보고되는 문제를 해결했습니다.
- MySQL 서버 ERROR 1457로 인해 mysql_rds_import_binlog_ssl_material에 대한 직접 호출이 실패할 수 있는 문제를 해결했습니다.
- 덤프 스레드 초기화가 바이너리 로그 제거 스레드로 인해 교착 상태에 빠질 수 있는 문제를 해결했습니다. 이로 인해 활성 binlog 파일이 교체되지 않고 계속 커지거나 새 binlog 복제본 연결에 문제가 발생할 수 있습니다.
- 쿼리 캐시에서 Aurora 리더 복제본의 오래된 결과가 반환될 수 있는 문제를 해결했습니다.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 아래 사항 외에도 5.7까지의 모든 커뮤니티 버그 수정이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 2.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그를 참조하세요.

- 성능 스키마 문 이벤트 테이블(예: events_statements_current)에서 문자 집합 정보를 읽는 코드가 해당 문자 집합 정보에 대한 동시 쓰기를 방지하지 못하는 문제를 해결했습니다. 결과적으로 SQL 쿼리 텍스트 문자 집합이 유효하지 않아 서버가 종료될 수 있었습니다. 이 수정을 사용하면 잘못된 문자 집합으로 인해 SQL_TEXT 열이 잘리며 서버가 종료되지 않습니다. (버그 #23540008)
- InnoDB: 커뮤니티 버그 #25189192, 버그 #84038에 대한 수정 백포트. 테이블을 다른 스키마로 이동한 RENAME TABLE 작업 이후 InnoDB가 INNODB_SYS_DATAFILES 데이터 딕셔너리 테이블을 업데이트하지 못하는 문제를 해결했습니다. 이로 인해 다시 시작 시 테이블스페이스 데이터 파일을 찾을 수 없다는 오류가 발생했습니다.
- InnoDB: 새 인덱스를 추가할 때 서버가 내부적으로 정의된 외래 키 인덱스를 삭제하고 가상 생성 열에 정의된 보조 인덱스를 외래 키 인덱스로 사용하려고 시도하여 서버가 종료되는 문제를 해결했습니다. InnoDB는 이제 외래 키 제약 조건이 가상 생성 열에 정의된 보조 인덱스를 참조하도록 허용합니다. (버그 #23533396)

- INSERT ... ON DUPLICATE KEY UPDATE 작업을 동시에 실행하는 두 세션에서 교착 상태가 발생하는 문제를 해결했습니다. 튜플을 부분적으로 롤백하는 동안 다른 세션에서 튜플을 업데이트할 수 있습니다. 이 버그에 대한 수정은 버그 #11758237, 버그 #17604730 및 버그 #20040791에 대한 수정을 되돌립니다. (버그 #25966845)
- 커뮤니티 버그 #27407480에 대한 수정 백포트: automatic_sp_privileges가 활성화된 경우에도 루틴 생성자에게 EXECUTE 및 ALTER ROUTINE 권한이 올바르게 부여되지 않는 문제를 해결했습니다.
- 커뮤니티 버그#24671968에 대한 수정 백포트: WHERE 절에 종속 하위 쿼리가 포함되어 있고, 테이블에서 선택 목록의 열과 뒤따르는 하위 쿼리의 열에 보조 인덱스가 있고, `GROUP BY` 또는 `DISTINCT`가 쿼리에서 느슨한 인덱스 스캔을 사용하도록 허용하는 경우 쿼리가 잘못된 결과를 생성할 수 있는 문제를 해결했습니다.
- 외래 키가 있는 여러 테이블에 대해 다중 테이블 삭제 문을 실행하면 복제가 중단되는 문제를 해결했습니다. (버그 #80821)
- slave_skip_errors가 활성화된 상태에서도 특수한 경우에 특정 슬레이브 오류가 무시되지 않는 문제를 해결했습니다. 테이블 열기 및 잠금이 실패하거나 행 기반 복제를 실행하는 서버에서 필드 변환이 실패할 경우 오류가 심각한 것으로 간주되어 slave_skip_errors 상태가 무시됩니다. 이 수정을 통해 slave_skip_errors가 활성화된 상태에서 트랜잭션을 적용하는 동안 보고된 모든 오류가 올바르게 처리됩니다. (버그 #70640, 버그 #17653275)
- `SET PASSWORD` 문이 MySQL 5.6 마스터에서 MySQL 5.7 슬레이브로 복제되거나 log_builtin_as_identified_by_password 시스템 변수가 ON으로 설정된 MySQL 5.7 마스터에서 MySQL 5.7 슬레이브로 복제되며 암호 해시 자체도 슬레이브에 저장되기 전에 해시되는 문제를 해결했습니다. 이제 이 문제가 해결되었으며 복제된 암호 해시가 원래 슬레이브에 전달된 대로 저장됩니다. (버그 #24687073)
- 여러 수준의 JSON 배열, 객체 또는 둘 다로 래핑되어 있는 대규모 하위 문서로 구성된 JSON 값의 직렬화를 완료하는 데 때때로 과도한 시간이 소요되는 문제를 해결했습니다. (버그 #23031146)
- 구문 오류 등으로 인해 구문 분석할 수 없는 문은 더 이상 느린 쿼리 로그에 기록되지 않습니다. (버그 #33732907)

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.

- AWS Lambda 함수를 비동기식으로 호출하기 위한 네이티브 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2022년 11월 1일(버전 2.10.3)(사용되지 않음)

버전: 2.10.3

Aurora MySQL 2.10.3이 정식 출시되었습니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 2.11.*, 3.01.* 및 3.02.*입니다.

기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.10.3으로 업그레이드할 수 있습니다. Aurora MySQL 버전 1을 실행하는 클러스터의 경우 기존 Aurora MySQL 1.23 이상의 클러스터를 2.10.3으로 직접 업그레이드할 수 있습니다. 또한 현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.10.3으로 스냅샷을 복원할 수 있습니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

아래 나열된 보안 문제 및 CVE 해결:

관리된 환경에서 처리를 미세 조정하기 위한 수정 사항 및 기타 향상된 기능 아래의 추가 CVE 수정:

- CVE-2022-21444
- CVE-2022-21344
- CVE-2022-21304
- CVE-2022-21245
- CVE-2021-36222
- CVE-2021-22926

일반적인 개선 사항

- 드문 경우지만 교착 상태 감지기 스레드가 중단되면 긴 세마포어 대기로 인해 데이터베이스 서버가 다시 시작될 수 있는 문제를 해결했습니다.
- GRANT, FLUSH PRIVILEGES 등과 같은 특정 데이터 제어 언어(DCL) SQL 문이 해당 인스턴스에서 실행될 때 데이터베이스 인스턴스에서 사용 가능한 메모리가 줄어들 수 있는 문제를 해결했습니다.

이러한 문을 자주 사용하면 사용 가능한 메모리가 계속 줄어들고 메모리 부족 문제로 인해 데이터베이스 인스턴스가 다시 시작될 수 있습니다. 라이터 인스턴스에서 이러한 문을 사용하면 리더 인스턴스에서 사용 가능한 메모리가 줄어들 수도 있습니다.

- 'wait/io/aurora_respond_to_client' performance_schema 대기 이벤트가 활성화된 상태에서 데이터베이스 인스턴스가 과부하 상태일 때 'performance_schema.events_waits_summary_global_by_event_name' 테이블에 대한 쿼리가 느려질 수 있는 문제를 해결했습니다.
- 드문 경우지만 보조 인덱스의 제약 조건 위반으로 인해 트랜잭션이 부분적으로 롤백될 때 데이터베이스 서버가 중단되고 다시 시작될 수 있는 문제를 해결했습니다.
- 드문 경우지만 다른 트랜잭션에 의해 삭제되는 행에 트랜잭션이 액세스할 때 라이터 인스턴스가 다시 시작되거나 장애 조치될 수 있는 문제를 해결했습니다.
- 드문 경우지만 I/O 스레드가 교착 상태에 빠지면 긴 세마포어 대기로 인해 데이터베이스가 다시 시작될 수 있는 문제를 해결했습니다.
- Unix 소켓 잠금 파일을 사용 중일 때 드물지만 장애 조치 동안 읽기 전용 복제본이 다시 시작될 수 있는 문제를 해결했습니다.
- 과도한 쿼리 캐시 무효화로 인해 읽기 전용 복제본이 쿼리 캐시 대신 디스크에서 데이터를 읽어야 하기 때문에 읽기 전용 복제본의 CPU 사용량과 지연 시간이 예상보다 높아지는 문제를 해결했습니다.

MySQL Community Edition 버그 수정 통합

이 릴리스에는 아래 사항 외에도 5.7까지의 모든 커뮤니티 버그 수정이 포함되어 있습니다. 자세한 내용은 Aurora MySQL 2.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그를 참조하세요.

- 성능 스키마 문 이벤트 테이블(예: events_statements_current)에서 문자 집합 정보를 읽는 코드가 해당 문자 집합 정보에 대한 동시 쓰기를 방지하지 못하는 문제를 해결했습니다. 결과적으로 SQL 쿼리 텍스트 문자 집합이 유효하지 않아 서버가 종료될 수 있었습니다. 이 수정을 사용하면 잘못된 문자 집합으로 인해 SQL_TEXT 열이 잘리며 서버가 종료되지 않습니다. (버그 #23540008)
- UPDATE에 1,024바이트보다 큰 프라이머리 키가 있는 임시 테이블이 필요하고 해당 테이블이 InnoDB를 사용하여 생성된 경우 서버가 종료될 수 있는 문제를 해결했습니다. (버그 #25153670)
- INSERT ... ON DUPLICATE KEY UPDATE 작업을 동시에 실행하는 두 세션에서 교착 상태가 발생하는 문제를 해결했습니다. 튜플을 부분적으로 롤백하는 동안 다른 세션에서 튜플을 업데이트할 수 있습니다. 이 버그에 대한 수정은 버그 #11758237, 버그 #17604730 및 버그 #20040791에 대한 수정을 되돌립니다. (버그 #25966845)

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- AWS Lambda 함수를 비동기식으로 호출하기 위한 네이티브 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2022년 1월 26일(버전 2.10.2)(사용되지 않음)

버전: 2.10.2

Aurora MySQL 2.10.2가 정식 출시되었습니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.10.0으로 업그레이드할 수 있습니다. Aurora MySQL 버전 1을 실행하는 클러스터의 경우 기존 Aurora MySQL 1.23 이상의 클러스터를 2.10.0으로 직접 업그레이드할 수 있습니다. 또한 현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.10.0으로 스냅샷을 복원할 수 있습니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

아래 나열된 보안 문제 및 CVE 해결:

관리된 환경에서 처리를 미세 조정하기 위한 수정 사항 및 기타 향상된 기능 아래의 추가 CVE 수정:

- 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

일반적인 개선 사항

- 24XL 인스턴스 클래스에서 데이터베이스 IO 대기 시간을 줄이는 데 도움이 되는 성능 최적화를 추가했습니다.
- ECDHE SSL 암호에 대한 지원을 추가합니다. 이러한 SSL 암호를 사용하도록 클라이언트를 구성하는 방법에 대한 자세한 내용은 MySQL 설명서, 암호화된 연결 프로토콜 암호를 참조하세요.
- Amazon S3, Amazon ML, AWS Lambda 등의 기타 AWS 서비스와 Aurora MySQL를 통합하는 것과 관련된 보안 문제를 수정했습니다.
- 데이터베이스에 약 1GB 이상의 사용자 및 권한 조합이 있는 경우 데이터베이스 인스턴스 재시작이 실패할 수 있는 문제를 수정했습니다.
- GROUP BY 절과 범위 술어가 포함된 WHERE 절을 사용하여 쿼리를 실행할 때 데이터베이스가 잘못된 그룹화 또는 정렬 순서를 반환할 수 있는 병렬 쿼리 문제를 수정했습니다.
- Aurora-MySQL 1.x(MySQL 5.6과 호환)에서 Aurora-MySQL 2.x(MySQL 5.7과 호환)로 인플레이스 메이저 버전을 업그레이드한 후 general_log 및 slow_log 테이블에 액세스할 수 없게 되는 문제를 수정했습니다.
- 드문 경우지만 데이터베이스의 작업 부하가 많은 동안 innodb_trx, innodb_lock 또는 innodb_lockwaits 테이블을 쿼리할 때 데이터베이스 인스턴스가 다시 시작되는 문제를 수정했습니다. 성능 개선 도우미와 같은 모니터링 도구는 이러한 테이블을 쿼리할 수 있습니다.
- 다음 조건이 모두 충족되면 기존 행의 TIMESTAMP 열 값이 최신 타임스탬프로 업데이트되는 문제가 해결되었습니다.

  1. 테이블에 대한 트리거가 존재합니다.
  2. INSERT는 ON DUPLICATE KEY UPDATE가 있는 테이블에서 수행됩니다.
  3. 삽입된 행은 UNIQUE 인덱스 또는 PRIMARY KEY에서 중복 값 위반이 발생합니다
  4. 하나 이상의 열은 TIMESTAMP 데이터 유형이며 기본값은 CURRENT_TIMESTAMP입니다.

- 드문 경우이지만 binlog 복제본이 binlog가 사용 설정된 인스턴스에 연결되지 않는 문제를 수정했습니다.

- 드문 경우지만 binlog가 사용 설정된 인스턴스에서 실행할 때 트랜잭션을 커밋할 수 없는 문제를 수정했습니다.
- binlog가 사용 설정된 인스턴스에 대한 새 연결을 설정할 수 없는 문제를 수정했습니다.
- 가동 중지 없는 패치 및 재시작을 시도할 때 과도한 내부 로깅을 유발하여 로컬 스토리지가 가득 차게 하는 문제를 수정했습니다.
- 특정 DDL 및 DCL 문을 복제할 때 HA_ERR_FOUND_DUPP_KEY 오류가 발생하면서 binlog 복제본이 중지되는 문제를 수정했습니다. 이 문제는 원본 인스턴스가 혼합 바이너리 로깅 형식과 READ COMMITED 또는 READ UNCOMMITED 격리 수준으로 구성된 경우에 발생합니다.
- 다중 스레드 복제가 사용 설정된 경우 binlog 복제 I/O 스레드가 프라이머리 인스턴스를 따라갈 수 없는 문제를 수정했습니다.
- 드문 경우지만 데이터베이스 인스턴스에 대한 많은 수의 활성 연결로 인해 CloudWatch CommitLatency 지표가 잘못 보고될 수 있는 문제를 수정했습니다.
- LOAD FROM S3 또는 SELECT INTO S3를 수행할 때 Graviton 인스턴스의 로컬 스토리지가 가득 차게 하는 문제를 수정했습니다.
- 외래 키로 테이블을 쿼리하고 다음 조건이 모두 충족될 때 잘못된 쿼리 결과를 유발할 수 있는 문제를 수정했습니다.

  1. 쿼리 캐시가 사용 설정됨
  2. 해당 테이블에 대한 계단식 삭제 또는 업데이트가 있는 트랜잭션이 롤백됨
- 드문 경우지만 Aurora 리더 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다. 트랜잭션 롤백 수가 증가할수록 이 문제가 발생할 가능성이 높아집니다.
- 세션을 열고 닫을 때 성능 스키마에서 뮤텍스 'Lock_epoch_ID_master' 발생 횟수가 증가하는 문제를 수정했습니다.
- 동일한 행 세트를 동시에 업데이트하는 많은 트랜잭션이 있는 워크로드에 대해 교착 상태가 증가할 수 있는 문제를 수정했습니다.
- 드문 경우지만 데이터베이스 볼륨이 160GB의 배수로 증가하면 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- LIMIT 절을 사용하여 SQL 문을 실행할 때 데이터베이스가 다시 시작될 수 있는 병렬 쿼리 문제를 수정했습니다.
- 드문 경우지만 READ COMMITED 격리 수준으로 XA 트랜잭션을 사용할 때 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 다시 시작하는 동안 DDL 워크로드가 많을 경우 Aurora 읽기 인스턴스가 다시 시작된 후 또 다시 시작될 수 있는 문제를 수정했습니다.

- Aurora 리더 복제 지연 시간을 잘못 보고하는 문제를 수정했습니다.
- 드문 경우지만 메모리 내 데이터 무결성 검사에 실패할 때 작성기 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 드물게 세션 처리가 완료되고 유휴 상태인 경우에도 PI (성능 개선 도우미) 세션의 "데이터베이스 로드" 차트가 CPU를 적극적으로 사용하는 것으로 표시되는 문제를 수정했습니다.
- 드문 경우지만 병렬 쿼리를 사용하여 쿼리를 처리할 때 데이터베이스 서버가 다시 시작될 수 있는 문제를 수정했습니다.
- 드문 경우지만 글로벌 데이터베이스 복제 중 경합 조건으로 인해 프라이머리 글로벌 데이터베이스 클러스터의 라이터 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 데이터베이스 인스턴스를 다시 시작하는 동안 발생하여 재시작을 여러 번 유발할 수 있는 문제를 수정했습니다.

MySQL Community Edition 버그 수정 통합

- 테이블 통계와 관련된 코드의 오류로 인해 dict0stats.cc 소스 파일에서 어설션이 발생하는 InnoDB의 문제를 수정했습니다. (버그 #24585978)
- 인덱스가 온라인으로 빌드될 때 가상 열에 대한 세컨더리 인덱스가 손상되는 문제를 수정했습니다. UPDATE 문의 경우 이를 다음과 같이 수정합니다. 인덱스 레코드의 가상 열 값이 NULL로 설정되면 클러스터 인덱스 레코드에서 이 값을 생성합니다. (버그 #30556595))
- 표시된 행을 삭제하면 부분 롤백이 완료되기 전에 외부 읽기 잠금을 획득할 수 있던 InnoDB 문제를 수정했습니다. 외부 읽기 잠금으로 인해 부분 롤백 중 암시적 잠금이 명시적 잠금으로 변환되지 않아 어설션 오류가 발생했습니다. (버그 #29195848)
- 계정의 빈 호스트 이름으로 인해 서버가 오작동할 수 있는 문제를 수정했습니다. (버그 #28653104)
- 잠금 대기 중 쿼리 중단으로 인해 오류가 발생하는 InnoDB 문제를 수정했습니다. (버그 #28068293)
- 트랜잭션 격리 수준이 REPEATABLE READ로 설정된 경우 인터리브 트랜잭션이 때때로 슬레이브 적용자를 교착 상태로 만들 수 있는 복제 문제를 수정했습니다. (버그 #25040331)
- 잠금 대기 시간 초과로 인해 binlog 복제본이 중단될 수 있는 문제를 수정했습니다. (버그 #27189701)

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- AWS Lambda 함수를 비동기식으로 호출하기 위한 네이티브 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2021년 10월 21일(버전 2.10.1)(사용되지 않음)

버전: 2.10.1

Aurora MySQL 2.10.1이 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.10.0으로 업그레이드할 수 있습니다. Aurora MySQL 버전 1을 실행하는 클러스터의 경우 기존 Aurora MySQL 1.23 이상의 클러스터를 2.10.0으로 직접 업그레이드할 수 있습니다. 또한 현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.10.0으로 스냅샷을 복원할 수 있습니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

아래 나열된 보안 문제 및 CVE 해결:

관리된 환경에서 처리를 미세 조정하기 위한 수정 사항 및 기타 향상된 기능 아래의 추가 CVE 수정:

- 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

가용성 향상:

- 향후 메이저 버전 업그레이드를 위해 클러스터를 완전히 종료하는 기능이 추가되었습니다.

일반적인 개선 사항

- 내부 진단 로그 파일에 정보 메시지의 과도한 로깅으로 인해 리더 인스턴스에서 CPU 사용량을 높일 수 있는 문제를 수정했습니다.
- 다음 조건이 모두 충족되면 기존 행의 TIMESTAMP 열 값이 최신 타임스탬프로 업데이트되는 문제가 해결되었습니다.
  1. 테이블에 대한 트리거가 존재합니다.
  2. INSERT는 ON DUPLICATE KEY UPDATE가 있는 테이블에서 수행됩니다.
  3. 삽입된 행은 UNIQUE 인덱스 또는 PRIMARY KEY에서 중복 값 위반이 발생합니다
  4. 하나 이상의 열은 TIMESTAMP 데이터 유형이며 기본값은 CURRENT_TIMESTAMP입니다.
- 특정 경우에 json_merge 함수를 사용하여 오류 코드를 발생시키는 버전 2.10.0에 도입된 문제를 수정했습니다. 특히, 생성된 열을 포함하는 DDL에서 json_merge 함수를 사용하면 오류 코드 1305이 반환될 수 있습니다.
- 드문 경우지만 읽기 전용 복제본에서 트랜잭션의 읽기 보기에 대해 대규모 객체의 업데이트 내역을 검증할 때 읽기 전용 복제본이 다시 시작되는 문제를 수정했습니다.
- 드문 경우지만 인 메모리 데이터 무결성 검사가 실패할 때 작성기 인스턴스가 다시 시작되는 문제를 수정했습니다.

MySQL 커뮤니티 에디션 버그 픽스 통합

- CURRENT_TIMESTAMP가 트리거에서 0을 생성합니다. (버그 #25209512)

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.

- AWS Lambda 함수를 비동기식으로 호출하기 위한 네이티브 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2021년 5월 25일(버전 2.10.0)(사용되지 않음)

버전: 2.10.0

Aurora MySQL 2.10.0이 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.10.0으로 업그레이드할 수 있습니다. Aurora MySQL 버전 1을 실행하는 클러스터의 경우 기존 Aurora MySQL 1.23 이상의 클러스터를 2.10.0으로 직접 업그레이드할 수 있습니다. 또한 현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.10.0으로 스냅샷을 복원할 수 있습니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

아래 나열된 보안 문제 및 CVE 해결:

관리된 환경에서 처리를 미세 조정하기 위한 수정 사항 및 기타 향상된 기능 아래의 추가 CVE 수정:

- 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

새로운 기능:

- 이제 `db.t3.large` 인스턴스 클래스가 Aurora MySQL에 대해 지원됩니다.
- 바이너리 로그 복제:
  - 라이터 스레드와 덤프 스레드 간의 경합을 줄여 binlog 성능을 개선하기 위해 binlog I/O 캐시가 도입되었습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 바이너리 로그 복제 최적화를 참조하세요.
  - Aurora MySQL 버전 2.08에는 대규모 트랜잭션이 관련된 경우 충돌 복구 시간과 커밋 시간 지연을 줄이기 위해 개선된 binlog 처리가 도입되었습니다. 이제 GTID가 사용되는 클러스터에 대해 이러한 개선 사항이 지원됩니다.
- 리더 인스턴스 가용성 개선:
  - 이전에는 라이터 인스턴스가 다시 시작될 때 Aurora MySQL 클러스터의 모든 리더 인스턴스도 다시 시작되었습니다. 오늘 출시를 통해 리전 내 리더 인스턴스는 라이터 인스턴스가 다시 시작되는 동안 계속해서 읽기 요청을 처리하므로 클러스터의 읽기 가용성이 개선됩니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 클러스터 재부팅(버전 2.10 이상)을 참조하세요.

    > **Important**
    >
    > Aurora MySQL 2.10으로 업그레이드한 후에는 라이터 인스턴스를 재부팅해도 전체 클러스터가 재부팅되지 않습니다. 전체 클러스터를 재부팅하려면 라이터 인스턴스를 재부팅한 후 클러스터의 리더 인스턴스를 재부팅합니다.

- 논리적 미리 읽기(LRA) 기술에 의해 요청된 미리 읽기 페이지 읽기 성능이 개선되었습니다. 이 성능 개선은 Aurora 스토리지로 전송된 단일 요청의 여러 페이지 읽기를 배치 처리하여 달성되었습니다. 따라서 LRA 최적화를 사용하는 쿼리가 최대 3배 더 빠르게 실행됩니다.
- 제로 가동 중지 다시 시작 및 패치 적용:
  - 바이너리 로깅이 사용되는 경우에 대한 추가 지원을 포함하여 광범위한 시나리오에서 ZDR 및 ZDP를 지원하도록 제로 다운타임 다시 시작(ZDR) 및 제로 다운타임 패치 적용(ZDP)이 개선되었습니다. 또한 ZDR 및 ZDP 이벤트에 대한 가시성이 향상되었습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL의 제로 다운타임 다시 시작(ZDR) 및 제로 가동 중지 패치 적용 기능 사용을 참조하세요.

가용성 향상:

- 데이터베이스에 이전에 중단된 DDL 활동 중에 생성된 임시 인덱스 및 테이블이 많은 경우 더 빠르게 시작되도록 개선되었습니다.
- `DROP TRIGGER`, `ALTER TABLE` 및 특히 테이블의 분할 유형 또는 파티션 수를 수정하는 `ALTER TABLE` 등, 중단된 DDL 활동의 충돌 복구 중에 다시 시작이 반복되는 것과 관련된 여러 문제가 해결되었습니다.
- 데이터베이스 활동 스트림(DAS) 로그 처리 중에 서버가 다시 시작되는 문제가 수정되었습니다.
- 시스템 테이블에서 `ALTER` 쿼리를 처리하는 동안 오류 메시지가 인쇄되는 문제가 수정되었습니다.

일반적인 개선 사항

- 쿼리 캐시에서 리더 인스턴스의 오래된 결과가 반환될 수 있는 문제가 수정되었습니다.
- 시스템 변수 `innodb_flush_log_at_trx_commit`이 0 또는 2로 설정되었을 때 일부 Aurora 커밋 지표가 업데이트되지 않는 문제가 수정되었습니다.
- 쿼리 캐시에 저장된 쿼리 결과가 다중 문 트랜잭션에 의해 새로 고쳐지지 않는 문제가 수정되었습니다.
- 바이너리 로그 파일의 마지막으로 수정된 타임스탬프가 올바르게 업데이트되지 않는 문제가 수정되었습니다. 이 문제로 인해 고객이 구성한 보존 기간에 도달하기 전에 바이너리 로그 파일이 조기에 제거될 수 있었습니다.
- 충돌 복구 후 InnoDB에서 잘못 보고된 binlog 파일 이름과 위치가 수정되었습니다.
- `binlog_checksum` 파라미터가 `NONE`인 경우 대규모 트랜잭션이 잘못된 binlog 이벤트를 생성할 수 있는 문제를 수정했습니다.
- 복제된 트랜잭션에 DDL 문이 포함되어 있고 행 변경이 많은 경우 binlog 복제본이 오류와 함께 중지되던 문제가 수정되었습니다.
- 테이블을 삭제할 때 리더 인스턴스가 다시 시작되는 문제가 수정되었습니다.
- 대규모 트랜잭션이 있는 binlog 파일을 사용하려고 할 때 오픈 소스 커넥터가 실패하는 문제가 수정되었습니다.
- 큰 지오메트리 값이 있는 테이블에 공간 인덱스를 생성한 후 대형 지오메트리 열에 잘못된 쿼리 결과가 발생할 수 있는 문제가 수정되었습니다.
- 이제 데이터베이스가 다시 시작되는 동안 임시 테이블스페이스를 다시 생성하여 연결된 스토리지 공간을 확보하고 회수할 수 있습니다.
- Aurora 리더 인스턴스에서 `performance_schema` 테이블이 잘리지 않던 문제가 수정되었습니다.

- HA_ERR_KEY_NOT_FOUND 오류가 발생하면서 binlog 복제본이 중지될 수 있는 문제를 수정했습니다.
- FLUSH TABLES WITH READ LOCK 문을 실행할 때 데이터베이스가 다시 시작되는 문제가 수정되었습니다.
- Aurora 리더 인스턴스에서 사용자 수준 잠금 기능을 사용할 수 없던 문제가 수정되었습니다.

MySQL 커뮤니티 에디션 버그 픽스 통합

- 트랜잭션 격리 수준이 REPEATABLE READ로 설정된 경우 인터리브 트랜잭션에서 슬레이브 애플리케이션이 교착 상태에 빠질 수 있습니다. (버그 #25040331)
- 저장 프로시저에 다른 보기를 참조하는 보기를 참조하는 문이 포함되어 있으면 프로시저를 두 번 이상 성공적으로 호출할 수 없습니다. (버그 #87858, 버그 #26864199)
- OR 조건이 많은 쿼리에서 이제 옵티마이저의 메모리 효율성이 향상되어 range_optimizer_max_mem_size 시스템 변수에 의해 적용되는 메모리 제한을 초과할 가능성이 적어집니다. 또한 해당 변수의 기본값이 1,536,000에서 8,388,608로 늘어났습니다. (버그 #79450, 버그 #22283790)
- 복제: 복제본의 SQL 스레드에서 호출되어 릴레이 로그의 다음 이벤트를 읽는 next_event() 함수에서 SQL 스레드가 오류(예: 닫힌 릴레이 로그)로 인해 가져온 relaylog.log_lock을 해제하지 않아 릴레이 로그의 잠금을 가져오기 위해 대기 중인 다른 모든 스레드가 중단되는 문제가 있었습니다. 이 수정을 적용하면 이 상황에서 SQL 스레드가 함수에서 나가기 전에 잠금이 해제됩니다. (버그 #21697821)
- 가상 열이 있는 ALTER TABLE에 대한 메모리 손상이 해결되었습니다. (버그 #24961167, 버그 #24960450)
- 복제: 다중 스레드 복제본에서 slave_pending_jobs_size_max보다 큰 트랜잭션을 처리해야 하는 경우 이 크기를 사용하여 작은 대기열 크기로 다중 스레드 복제본을 구성할 수 없었습니다. slave_pending_jobs_size_max보다 큰 모든 패킷은 패킷이 slave_max_allowed_packet으로 설정된 제한보다 작은 경우에도 ER_MTS_EVENT_BIGGER_PENDING_JOBS_SIZE_MAX 오류와 함께 거부되었습니다. 이 수정을 적용하면 slave_pending_jobs_size_max가 하드 제한이 아닌 소프트 제한이 됩니다. 패킷 크기가 slave_pending_jobs_size_max를 초과하지만 slave_max_allowed_packet보다 작은 경우 모든 복제본 작업자의 대기열이 빌 때까지 트랜잭션이 보류된 후 처리됩니다. 모든 후속 트랜잭션은 대규모 트랜잭션이 완료될 때까지 보류됩니다. 따라서 복제본 작업자의 대기열 크기는 제한되지만 경우에 따라 더 큰 트랜잭션을 허용할 수 있습니다. (버그 #21280753, 버그 #77406)
- 복제: 다중 스레드 복제본을 사용할 때 적용자 오류에 표시되는 작업자 ID 데이터가 성능 스키마 복제 테이블에 표시된 데이터와 일치하지 않았습니다. (버그 #25231367)

- 복제: -gtid-mode=ON, -log-bin=OFF이고 -slave-skip-errors를 사용하여 실행되는 GTID 기반 복제 복제본에서 Exec_Master_Log_Pos가 올바르게 업데이트되지 않는 무시할 수 있는 오류가 발생하는 경우 Exec_Master_Log_Pos가 Read_master_log_pos와 느슨하게 동기화되었습니다. GTID_NEXT가 지정되지 않은 경우 복제본은 단일 문 트랜잭션에서 롤백할 때 GTID 상태를 업데이트하지 않습니다. 트랜잭션이 완료되었지만 GTID 상태가 다르게 표시되므로 Exec_Master_Log_Pos가 업데이트되지 않습니다. 이 수정은 GTID_NEXT가 지정된 경우에만 트랜잭션 롤백 시 GTID 상태 업데이트 제한을 제거합니다. (버그 #22268777)
- 복제: 바이너리 로깅이 사용 중지된 경우 부분적으로 실패한 문에 자동 생성되거나 지정된 GTID가 올바르게 사용되지 않았습니다. 이 수정을 적용하면 이진 로깅이 사용 중지된 경우 부분적으로 실패한 DROP TABLE, 부분적으로 실패한 DROP USER 또는 부분적으로 실패한 DROP VIEW에 관련 GTID가 각각 사용되고 @@GLOBAL.GTID_EXECUTED 및 mysql.gtid_executed 테이블에 저장됩니다. (버그 #21686749)
- 복제: MySQL 5.7을 실행하는 복제본에서 MySQL 5.5의 일부가 아닌 server_uuid 검색 중에 발생하는 오류로 인해 MySQL 5.5 소스에 연결할 수 없었습니다. 이 문제는 server_uuid 검색 방법의 변경으로 인해 발생했습니다. (버그 #22748612)
- 복제: 이전에 실행된 GTID 트랜잭션을 자동으로 건너뛰는 GTID 트랜잭션 건너뛰기 메커니즘이 XA 트랜잭션에 대해 올바르게 작동하지 않았습니다. (버그 #25041920)
- 잘못된 트랜잭션 ID가 지정되어 실패한 ">XA ROLLBACK 문이 올바른 트랜잭션 ID로 바이너리 로그에 기록되었기 때문에 복제 복제본의 작업에 사용될 수 있었습니다. 이제 바이너리 로깅이 수행되기 전에 오류 상황이 확인되고 실패한 XA ROLLBACK 문이 로깅되지 않습니다. (버그 #26618925)
- 복제: 소스 로그 파일 이름 및 소스 로그 위치를 지정하지 않은 CHANGE MASTER TO 문을 사용하여 설정된 복제본이 START SLAVE 실행 전에 중단되고 -relay-log-recovery 설정 옵션을 사용하여 다시 시작되는 경우 복제가 시작되지 않았습니다. 이 문제는 릴레이 로그 복구가 시도되기 전에 수신기 스레드가 시작되지 않아 릴레이 로그에서 소스 로그 파일 이름과 소스 로그 위치를 제공하는 데 필요한 로그 회전 이벤트를 사용할 수 없기 때문에 발생했습니다. 이제 이 상황에서 복제본은 릴레이 로그 복구를 건너뛰고 경로를 로깅한 다음 복제를 시작합니다. (버그 #28996606, 버그 #93397)
- 복제: 행 기반 복제에서 utf8mb3 열이 있는 테이블을 utf8mb4 문자 집합으로 정의된 열이 있는 동일한 정의의 테이블로 복제할 때 필드 길이가 잘못 표시된 메시지가 반환되었습니다. (버그 #25135304, 버그 #83918)
- 복제: GTID를 사용 중인 복제 복제본에서 RESET SLAVE 문이 실행되는 경우 기존 릴레이 로그 파일이 제거되지만 채널에 대해 수신된 GTID 세트가 지워지기 전에 새로운 대체 릴레이 로그 파일이 생성되었습니다. 따라서 이전 GTID 세트가 새 릴레이 로그 파일에 PREVIOUS_GTIDS 이벤트로 기록되고 이로 인해 두 서버에 설정된 gtid_executed가 모두 비어 있음에도 불구하고 복제본에 소스보다 많은 GTID가 있다는 내용의 치명적 오류가 복제에서 발생합니다. 이제 RESET SLAVE가 실행되

면 새 릴레이 로그 파일이 생성되기 전에 수신된 GTID 세트가 지워지므로 이러한 상황이 발생하지 않습니다. (버그 #27411175)

- 복제: GTID를 복제에 사용하는 경우 동일한 GTID의 빈 트랜잭션 또는 대체 트랜잭션을 삽입하는 권장 방법으로 구문 분석 오류(ER_PARSE_ERROR)를 야기한 문이 포함된 트랜잭션을 수동으로 건너뛸 수 없습니다. 이 작업을 수행하면 복제본에서 GTID를 이미 사용된 것으로 식별하여 해당 GTID를 공유하는 원치 않는 트랜잭션을 건너뛰어야 합니다. 그러나 구문 분석 오류의 경우 GTID를 건너뛰어야 하는지 여부를 확인하기 전에 문이 구문 분석되므로 의도가 트랜잭션을 건너뛰는 것이라 할지라도 구문 분석 오류로 인해 복제 적용자 스레드가 중지되었습니다. 이 수정에서는 GTID가 이미 사용되었으므로 해당 트랜잭션을 건너뛰어야 하는 경우 복제 적용자 스레드가 구문 분석 오류를 무시합니다. 이 동작 변경은 `mysqlbinlog`에 의해 생성된 바이너리 로그 출력으로 구성되는 워크로드에는 적용되지 않습니다. 이 상황에서는 건너뛴 트랜잭션 바로 뒤에 오는 트랜잭션에 구문 분석 오류가 있는 경우 오류가 생성되어야 하지만 자동으로 건너뛸 위험이 있습니다. (버그 #27638268)
- 복제: GTID에 대한 SQL 스레드에서 부분 트랜잭션을 건너뛸 수 있습니다. (버그 #25800025)
- 복제: `WAIT_UNTIL_SQL_THREAD_AFTER_GTIDS()`에 제공된 제한 시간 파라미터가 음수 또는 분수인 경우 서버가 예기치 않은 방식으로 동작합니다. 이 수정에서는 다음과 같이 수정됩니다.
  - 분수 제한 시간 값을 반올림 없이 있는 그대로 읽습니다.
  - 서버가 엄격한 SQL 모드에 있는 경우 음수 제한 시간 값이 오류와 함께 거부됩니다. 서버가 엄격한 SQL 모드에 있지 않은 경우 함수가 대기 없이 즉시 `NULL`을 반환한 다음 경고를 생성합니다. (버그 #24976304, 버그 #83537)
- 복제: 분수 값이 포함된 제한 시간(예: 1.5)에 `WAIT_FOR_EXECUTED_GTID_SET()` 함수가 사용된 경우 제한 시간이 근사한 정수(초)로 반내림되고 값이 1초 미만(예: 0.1)인 경우 0으로 반내림되었음을 의미하는 캐스팅 로직 오류가 발생했습니다. 이제 캐스팅 로직이 수정되어 제한 시간 값이 원래 지정된 대로 반올림/반내림 없이 적용됩니다. Dirkjan Bussink 씨의 기여에 감사드립니다. (버그 #29324564, 버그 #94247)
- GTID를 사용하는 경우 다중 문 트랜잭션 내의 연결이 끊긴 XA 트랜잭션의 XA COMMIT에서 어설션이 발생했습니다. (버그 #22173903)
- 복제: gtid_next 값이 수동으로 설정된 경우 알 수 없는 트랜잭션 식별자에 대해 XA ROLLBACK 문이 실행되면 디버그 빌드에서 어설션이 발생했습니다. 이제 XA `ROLLBACK` 문이 오류와 함께 실패하는 경우 서버에서 GTID 상태 업데이트를 시도하지 않습니다. (버그 #27928837, 버그 #90640)
- 여러 `CASE` 함수가 `ORDER BY` 절에 사용되는 경우 잘못된 정렬 순서 문제가 수정됩니다 (Bug#22810883).
- 정렬을 사용한 일부 쿼리는 최적화 중에 초기화되지 않은 열에 액세스하여 서버가 종료될 수 있습니다. (버그 #27389294)

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- AWS Lambda 함수를 비동기식으로 호출하기 위한 네이티브 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2021년 11월 12일(버전 2.09.3)(사용되지 않음)

버전: 2.09.3

Aurora MySQL 2.09.3이 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.10.0으로 업그레이드할 수 있습니다. Aurora MySQL 버전 1을 실행하는 클러스터의 경우 기존 Aurora MySQL 1.23 이상의 클러스터를 2.10.0으로 직접 업그레이드할 수 있습니다. 또한 현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.10.0으로 스냅샷을 복원할 수 있습니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 AWS Management Console, AWS CLI 또는 Amazon RDS API를 통해 엔진 버전을 지정합니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

보안 수정:

관리된 환경에서 처리를 미세 조정하기 위한 수정 사항 및 기타 향상된 기능 아래의 추가 CVE 수정:

- 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

가용성 향상:

- information_schema의 테이블에서 실행되는 쿼리에 대한 경합을 줄일 수 있는 최적화를 도입했습니다.
- ECDHE SSL 암호에 대한 지원을 추가합니다.

일반적인 개선 사항

- 드문 경우지만 메모리 내 데이터 무결성 검사에 실패할 때 작성기 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 드문 경우지만 바이너리 로깅을 사용하는 동안 클러스터 볼륨이 확장될 때 데이터베이스 인스턴스가 다시 시작되던 문제를 수정했습니다.
- 데이터베이스 인스턴스를 다시 시작하는 동안 드문 경합 상태가 수정되어 두 개 이상의 재시작을 발생시키던 문제를 수정했습니다.
- 데이터베이스에 사용자 및 권한 조합이 많은 경우 데이터베이스 인스턴스 재시작이 실패할 수 있는 문제를 수정했습니다.
- LIMIT 절을 사용하여 SQL 문을 실행할 때 데이터베이스가 다시 시작될 수 있는 병렬 쿼리 문제를 수정했습니다.

- aurora 복제 지연을 잘못 보고하는 문제가 해결되었습니다.
- Aurora-MySQL 1.x(MySQL 5.6 기반)에서 Aurora-MySQL 2.x(MySQL 5.7 기반)로 인플레이스 메이저 버전을 업그레이드한 후 general_log 및 slow_log 테이블에 액세스할 수 없게 되는 문제를 수정했습니다.
- 드문 경우지만 데이터베이스의 작업 부하가 많은 동안 innodb_trx, innodb_lock 또는 innodb_lockwaits 테이블을 쿼리할 때 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다. 성능 개선 도우미와 같은 모니터링 도구 및 기능은 이러한 테이블을 쿼리할 수 있습니다.
- 'FLUSH TABLES WITH READ LOCK' SQL 문이 실행될 때 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 읽기 프로그램 인스턴스를 삭제하는 동안 InnoDB 제거 프로세스가 일시 중지되어 히스토리 목록 길이가 일시적으로 증가하는 문제를 수정했습니다.
- 가상 열이 포함된 테이블에 대해 SQL 문을 실행할 때 데이터베이스가 다시 시작될 수 있는 병렬 쿼리 문제를 수정했습니다.
- GROUP BY 절과 범위 술어가 포함된 WHERE 절을 사용하여 쿼리를 실행할 때 데이터베이스가 잘못된 그룹화 또는 정렬 순서를 반환할 수 있는 병렬 쿼리 문제를 수정했습니다.
- 드문 경우지만 JSON 함수를 사용하여 SQL 문을 실행할 때 데이터베이스가 다시 시작될 수 있는 병렬 쿼리의 문제가 해결되었습니다.
- 드문 경우지만 전역 데이터베이스 복제 중 경합 조건으로 인해 기본 글로벌 데이터베이스 클러스터의 라이터 인스턴스가 다시 시작될 수 있는 문제가 해결되었습니다.
- 특정 DDL 및 DCL 문을 복제할 때 HA_ERR_FOUND_DUPP_KEY 오류가 발생하면서 Binlog 복제본이 중지될 수 있는 문제를 수정했습니다. 이 문제는 원본 인스턴스가 혼합 바이너리 로깅 형식과 READ COMMITED 또는 READ UNCOMMITED 격리 수준으로 구성된 경우에 발생합니다.
- 드문 경우지만 READ COMMITED 격리 수준에서 XA 트랜잭션을 사용할 때 데이터베이스 인스턴스가 다시 시작될 수 있는 문제가 해결되었습니다.
- 다음 조건을 모두 충족할 때 기존 행의 TIMESTAMP 열 값이 최신 타임스탬프로 업데이트되는 문제를 수정했습니다. 1. 테이블에 대한 트리거가 있습니다. 2. ON DUPLATE KEY UPDATE 절이 있는 테이블에 대해 INSERT가 수행됩니다. 3. 삽입된 행이 UNIQUE 인덱스 또는 PRIMARY KEY에서 중복 값을 위반합니다. 4. 하나 이상의 열은 TIMESTAMP 데이터 형식이며 기본값은 CURRENT_TIMESTAMP입니다.
- 드문 경우지만 잘못된 확인 처리로 인해 리더 인스턴스가 다시 시작될 수 있는 문제가 해결되었습니다.
- 라이터 인스턴스가 데이터베이스 볼륨을 확장하여 특정 볼륨 크기 경계를 넘을 때 리더 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.

- 복제된 클러스터 볼륨을 사용하여 데이터베이스 인스턴스의 재시작 시간이 더 길어질 수 있는 문제를 수정했습니다.
- 라이터 인스턴스에서 TRUNCATE TABLE 작업을 수행한 후 데이터베이스 인스턴스 재시작이 한 번 이상 실패할 수 있는 문제를 수정했습니다.
- 드문 경우지만 데이터베이스 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.
- 드문 경우지만 데이터베이스 볼륨이 160GB의 배수로 증가하면 라이터 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.

MySQL 커뮤니티 에디션 버그 픽스 통합

- 버그 #23533396 - 새 인덱스를 추가할 때 서버는 내부적으로 정의된 외래 키 인덱스를 삭제하고 가상 생성 열에 정의된 보조 인덱스를 외래 키 인덱스로 사용하려고 시도하여 서버가 종료되었습니다. InnoDB는 이제 외래 키 제약 조건이 가상 생성 열에 정의된 보조 인덱스를 참조하도록 허용합니다.
- 버그 #29550513 - 복제: WAIT_FOR_EXECUTED_GTID_SET() 함수의 잠금 문제로 인해 특정 상황에서 서버가 멈출 수 있습니다. 이제 문제가 수정되었습니다.

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- AWS Lambda 함수를 비동기식으로 호출하기 위한 네이티브 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2021년 2월 26일(버전 2.09.2)(사용되지 않음)

버전: 2.09.2

Aurora MySQL 2.09.2가 정식 출시되었습니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.09.2으로 업그레이드할 수 있습니다. Aurora MySQL 버전 1을 실행하는 클러스터의 경우 기존 Aurora MySQL 1.23 이상의 클러스터를 2.09.2으로 직접 업그레이드할 수 있습니다. 또한 현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.09.2으로 스냅샷을 복원할 수 있습니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 AWS Management Console, AWS CLI 또는 Amazon RDS API를 통해 엔진 버전을 지정합니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

새로운 기능:

- Aurora MySQL 클러스터는 이제 Arm 기반 AWS Graviton2 프로세서로 구동되는 다음 EC2 R6g 인스턴스를 지원합니다. `r6g.large`, `r6g.xlarge`, `r6g.2xlarge`, `r6g.4xlarge`, `r6g.8xlarge`, `r6g.12xlarge`, `r6g.16xlarge` 자세한 내용은 Amazon RDS 사용 설명서의 Aurora DB 인스턴스 클래스를 참조하세요.

보안 수정:

관리된 환경에서 처리를 미세 조정하기 위한 수정 사항 및 기타 향상된 기능 아래의 추가 CVE 수정:

- 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

가용성 향상:

- 2.09.0에서 클러스터 스토리지 볼륨을 확장하는 동안 쓰기 지연 시간이 길어질 수 있는 문제가 해결되었습니다.
- Aurora 읽기 복제본이 재시작될 수 있는 동적 크기 조정 기능의 문제가 해결되었습니다.
- 1.23.*에서 2.09.*로 업그레이드하는 동안 가동 중지 시간이 길어지는 문제를 수정했습니다.
- DDL 또는 DML로 인해 페이지 미리 가져오기 요청 중에 엔진이 재시작될 수 있는 문제가 해결되었습니다.
- 복제된 트랜잭션에 DDL 문이 포함되어 있고 행 변경이 많은 경우 binlog 복제본이 오류와 함께 중지되던 문제를 수정했습니다.
- mysql `time_zone` 테이블에서 DDL 이벤트를 복제하는 동안 binlog 복제본 역할을 하는 데이터베이스가 재시작되던 문제가 해결되었습니다.
- `binlog_checksum` 파라미터가 `NONE`인 경우 대규모 트랜잭션이 잘못된 binlog 이벤트를 생성할 수 있는 문제를 수정했습니다.
- `HA_ERR_KEY_NOT_FOUND` 오류가 발생하면서 binlog 복제본이 중지될 수 있는 문제를 수정했습니다.
- 전반적인 안정성이 향상되었습니다.

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- AWS Lambda 함수를 비동기식으로 호출하기 위한 네이티브 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2020년 12월 11일(버전 2.09.1)(사용되지 않음)

버전: 2.09.1

Aurora MySQL 2.09.1이 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.09.1으로 업그레이드할 수 있습니다. Aurora MySQL 버전 1을 실행하는 클러스터의 경우 기존 Aurora MySQL 1.23 이상의 클러스터를 2.09.1으로 직접 업그레이드할 수 있습니다. 또한 현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.09.1으로 스냅샷을 복원할 수 있습니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 AWS Management Console, AWS CLI 또는 RDS API를 통해 엔진 버전을 지정합니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

보안 수정:

관리된 환경에서 처리를 미세 조정하기 위한 수정 사항 및 기타 향상된 기능 아래의 추가 CVE 수정:

- 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

호환되지 않는 변경 사항:

이 버전에서는 mysqldump 명령의 동작에 영향을 주는 권한 변경을 도입했습니다. 사용자는 PROCESS 테이블에 액세스하려면 INFORMATION_SCHEMA.FILES 권한이 있어야 합니다. 변경 없이 mysqldump 명령을 실행하려면 PROCESS 명령이 연결되는 데이터베이스 사용자에게 mysqldump 권한을 부여합니다. mysqldump 옵션을 사용하여 --no-tablespaces 명령을 실행할 수도 있습니다. 이 옵션을 사용하면 mysqldump 출력에 CREATE LOGFILE GROUP 또는 CREATE TABLESPACE 문이 포함되지 않습니다. 이 경우 mysqldump 명령은 INFORMATION_SCHEMA.FILES 테이블에 액세스하지 않으므로 PROCESS 권한을 부여할 필요가 없습니다.

가용성 향상:

- 데이터베이스 엔진에서 네트워크에서 읽거나 쓰는 동안 오류가 발생하면 클라이언트 세션이 정지될 수 있는 문제가 해결되었습니다.
- 2.09.0에서 도입된 동적 크기 조정 기능의 메모리 누수가 수정되었습니다.

글로벌 데이터베이스:

- 기본 글로벌 작성기가 이전 릴리스 버전에 있는 동안 릴리스 2.09.0으로 업그레이드하면 글로벌 데이터베이스 보조 지역의 복제본이 다시 시작될 수 있는 여러 문제가 해결되었습니다.

MySQL 커뮤니티 에디션 버그 픽스 통합

- 복제: 트랜잭션 격리 수준이 반복 읽기로 설정된 경우 인터리브 트랜잭션이 슬레이브 응용 프로그램이 교착 상태에 빠질 수 있습니다. (버그 #25040331)
- 기본값 CURRENT_TIMESTAMP가 있는 TIMESTAMP 또는 DATETIME 열을 포함하는 테이블의 경우 테이블에 BEFORE INSERT 트리거가 있으면 열을 0000-00-00 00:00:00로 초기화할 수 있습니다. (버그 #25209512, 버그 #84077)
- VALUES 목록에서 조인을 포함하는 하위 쿼리를 사용하여 두 번째 또는 이후 행에 대한 값을 생성하는 삽입문의 경우 필요한 권한을 확인하지 못한 후 서버가 종료될 수 있습니다. (버그 #23762382)

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.

- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- AWS Lambda 함수를 비동기식으로 호출하기 위한 네이티브 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2020년 9월 17일(버전 2.09.0)(사용되지 않음)

버전: 2.09.0

Aurora MySQL 2.09.0이 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

Aurora MySQL1.23.*에서 Aurora MySQL 2.09.0으로 스냅샷을 복원할 수 있습니다. 기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.09.0으로 업그레이드할 수 있는 옵션도 있습니다. 기존 Aurora MySQL 1.23.* 클러스터를 2.09.0으로 직접 업그레이드할 수 없습니다. 하지만 이 클러스터의 스냅샷을 Aurora MySQL 2.09.0으로 복원할 수 있습니다.

> **Important**
>
> 이 버전의 Aurora 저장소가 개선되어 사용 가능한 업그레이드 경로가 Aurora MySQL 1.*에서 Aurora MySQL 2.09로 제한됩니다. Aurora MySQL 1.* 클러스터를 2.09로 업그레이드하는 경우 Aurora MySQL 1.23에서 업그레이드해야 합니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 AWS Management Console, AWS CLI 또는 RDS API를 통해 엔진 버전을 지정합니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

새로운 기능:

- 이 릴리스에서는 최대 128테비바이트(TiB)의 스토리지가 포함된 Amazon Aurora MySQL 데이터베이스 인스턴스를 생성할 수 있습니다. 새 스토리지 제한은 이전 64TiB에서 증가한 것입니다. 128TiB 스토리지 크기는 더 큰 데이터베이스를 지원합니다. 작은 인스턴스 크기(db.t2 또는 db.t3)에서는 이

기능이 지원되지 않습니다. 단일 테이블스페이스는 16KB 페이지 크기의 InnoDB 한도로 인해 64TiB 이상으로 증가할 수 없습니다.

Aurora은 클러스터 볼륨 크기가 128TiB에 가까울 때 경고를 표시하므로 크기 제한에 도달하기 전에 조치를 취할 수 있습니다. 경고는 mysql 로그와 AWS Management Console의 RDS 이벤트에 나타납니다.

- 이제 DB 클러스터 파라미터 `aurora_parallel_query`의 값을 변경하여 기존 클러스터에 대해 병렬 쿼리를 설정하거나 해제할 수 있습니다. 클러스터를 생성할 때 `parallelquery` 파라미터에 대한 `--engine-mode` 설정을 사용할 필요가 없습니다.

  이제 Aurora MySQL을 사용 가능한 모든 리전에서 병렬 쿼리를 사용할 수 있습니다.

  Aurora 클러스터에서 병렬 쿼리를 업그레이드하고 활성화하는 절차에 대한 기타 여러 가지 기능 향상 및 변경 사항이 많이 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL용 Parallel Query 처리를 참조하세요.
- Aurora는 클러스터 스토리지 공간의 크기를 동적으로 조정합니다. 동적 크기 조정이 사용되면 Aurora DB 클러스터에서 데이터를 제거할 때 DB 클러스터의 스토리지 공간이 자동으로 줄어듭니다. 자세한 내용은 Amazon Aurora 사용 설명서의 스토리지 조정을 참조하세요.

  > **Note**
  >
  > 동적 크기 조정 기능은 Aurora를 사용할 수 있는 AWS 리전에 단계적으로 배포되고 있습니다. 클러스터가 있는 리전에 따라 이 기능을 아직 사용하지 못할 수도 있습니다. 자세한 내용은 새로운 소식 공지를 참조하세요.

높은 우선 순위:

- 백포트 커뮤니티 버그 #27659490: 동적 범위 및 인덱스 병합을 사용하여 선택하면 너무 많은 메모리를 사용합니다(OOM).
- 버그 #26881508: MYSQL #1: AUTH_COMMON.H에서 DISABLE_ABORT_ON_ERROR가 발생합니다.
- 백포트 커뮤니티 버그 #24437124: 테이블 생성 시 버퍼 오버플로가 발생할 수 있습니다.
- 백포트 버그 #27158030: INNODB 온라인 변경이 동시 DML과 충돌합니다.
- 버그 #29770705: 특정 WHERE 절을 사용하여 SELECT를 실행하는 동안 서버가 충돌했습니다.
- 백포트 버그 #26502135: MYSQLD SEGFAULTS IN MDL_CONTEXT::TRY_ACQUIRE_LOCK_IMPL

- 백포트 버그 #26935001: ALTER TABLE AUTO_INCREMENT가 폐기된 테이블스페이스에서 인덱스를 읽으려고 시도합니다.
- 버그 #28491099: [치명적] 메모리 블록이 유효하지 않음 | INNODB: 어설션 실패: UT0UT.CC:670
- 버그 #30499288: GCC 9.2.1 OS_FILE_GET_PARENT_DIR에 대한 새 경고가 보고됩니다.
- 버그 #29952565: 쿼리(UNION + ORDER BY + SUB-QUERY)를 실행하는 동안 MYSQLD에서 신호 11이 발생합니다.
- 버그 #30628268: 메모리 부족 충돌
- 버그 #30441969: 버그 #29723340: 데이터 세트가 있는 SQL 쿼리 후 MySQL Server 충돌
- 버그 #30569003: 5.7 권한 부여 관리와 함께 구문 오류가 발생하여 복제가 중단됩니다.
- 버그 #29915479: COM_BINLOG_DUMP 없이 COM_REGISTER_SLAVE를 실행하면 서버가 종료될 수 있습니다.
- 버그 #30569003: 5.7 권한 부여 관리와 함께 구문 오류가 발생하여 복제가 중단됩니다.
- 버그 #29915479: COM_BINLOG_DUMP 없이 COM_REGISTER_SLAVE를 실행하면 서버가 종료될 수 있습니다.
- Bug #20712046: SHOW PROCESSLIST 및 PERFORMANCE_SCHEMA TABLES가 쿼리에서 암호를 마스킹하지 않습니다.
- 백포트 버그 #18898433: 외부 조인 및 조인 버퍼로 인해 성능이 매우 느립니다(5.7.21에서 수정됨). 버그 #18898433: 조인 버퍼링을 사용하는 경우(예: 블록 중첩 루프 알고리즘 사용) 왼쪽 조인이 많은 쿼리가 느려졌습니다. (버그 #18898433, 버그 #72854)"
- 백포트 버그 #26402045: 쿼리에서 MYSQLD가 충돌합니다(MySQL 5.7.23에서 수정). 특정 하위 쿼리 구체화 사례로 인해 서버가 종료될 수 있습니다. 이러한 쿼리는 이제 구체화가 비활성화되었음을 나타내는 오류를 생성합니다. (버그 #26402045)
- [MySQL에서 백포트] rdsadmin 이외의 사용자는 읽기 전용 복제본의 pfs 테이블을 업데이트할 수 없습니다.
- 고객이 읽기 전용 복제본에서 perfschema를 업데이트할 수 없는 문제를 해결했습니다.
- 버그 #26666274: 성능 스키마 버퍼 컨테이너에서 무한 루프가 발생합니다.
- 버그 #26997096: relay_log_space 값이 동기화된 방식으로 업데이트되지 않으므로 값이 릴레이 로그에서 사용되는 실제 디스크 공간보다 훨씬 높습니다.
- 버그 #25082593: 외래 키 유효성 검사 시 커밋된 읽기에서 갭 잠금을 획득할 필요가 없습니다.
- 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

가용성 향상:

- 기본적으로 잠금 관리자 ABA 수정 사항을 활성화합니다.
- 잠금 관리자에서 교착 상태로 인해 두 트랜잭션에 의해 잠금이 공유되어 데이터베이스가 다시 시작되는 문제를 수정했습니다.
- 압축된 행 형식으로 임시 테이블을 만들면 데이터베이스가 다시 시작되는 문제가 해결되었습니다.
- 대규모 인스턴스 클래스(R4/R5-16XL, R5-12XL, R5-24XL)에서 반복적인 장애 조치 및 높은 CPU 사용률을 유발할 수 있는 16XL 및 24XL 인스턴스에서 기본값 `table_open_cache`를 수정합니다. 이는 2.07.x에 영향을 미쳤습니다.
- S3 백업에 `mysql.host` 테이블이 포함되지 않은 경우 Amazon S3에서 Aurora MySQL 버전 2.08.0으로 클러스터를 복원하는 데 예상보다 오래 걸리는 문제를 해결했습니다.
- 보조 인덱스가 있는 가상 열의 업데이트로 인해 장애 조치가 반복되는 문제를 해결했습니다.
- 장기 실행 쓰기 트랜잭션으로 인해 데이터베이스가 다시 시작되는 트랜잭션 잠금 메모리 관리와 관련된 문제를 수정했습니다.
- 패치 적용을 위해 안전 지점을 확인하는 동안 가동 중지 없는 패치 적용 중에 엔진이 충돌할 수 있는 여러 문제를 해결했습니다.
- 이전에 충돌을 일으킨 임시 테이블에 대한 다시 실행 로깅을 건너뛰는 문제를 해결했습니다.

- 연결/쿼리 종료와 세션 종료 사이에 잠금 관리자의 경합 상태를 해결했습니다.
- 데이터베이스가 binlog 복제본이고 MySQL `time_zone` 테이블을 통해 DDL 이벤트를 받는 경우 데이터베이스가 충돌할 수 있던 문제를 해결했습니다.

글로벌 데이터베이스:

- 보조 리전의 MySQL `INFORMATION_SCHEMA.REPLICA_HOST_STATUS` 보기에 이제 해당 리전에 속한 복제본의 항목이 표시됩니다.
- 기본 리전과 보조 리전 간의 일시적인 네트워크 연결 문제 이후 글로벌 DB 보조 리전에서 발생할 수 있는 예기치 않은 쿼리 실패를 해결했습니다.
-

병렬 쿼리:

- 간단한 단일 테이블 쿼리의 경우 병렬 쿼리에 대한 `EXPLAIN` 계획이 부정확한 문제를 해결했습니다.
- 병렬 쿼리가 활성화되어 있을 때 발생할 수 있는 자체 데드 래치 문제를 해결했습니다.

일반적인 개선 사항

- 이제 S3로 내보내기에서 `ENCRYPTION` 키워드를 지원합니다.
- 이제 `aurora_binlog_replication_max_yield_seconds` 파라미터의 최대 값이 36,000입니다. 이전에 허용된 최대 값은 45였습니다. 이 파라미터는 파라미터 `aurora_binlog_use_large_read_buffer`가 1로 설정된 경우에만 작동합니다.
- `MIXED`를 실행할 때 `binlog_format` `ROW`을 `STATEMENT` 대신 `LOAD DATA FROM INFILE | S3`에 매핑하는 동작을 변경했습니다.
- Aurora MySQL binlog 기본이 `LOAD DATA FROM S3`를 실행하고 `binlog_format`이 `STATEMENT`로 설정되어 있을 때 이 기본에 연결된 binlog 복제본이 불완전한 데이터를 표시하는 문제를 해결했습니다.
- 감사 시스템 변수 `server_audit_incl_users` 및 `server_audit_excl_users`의 최대 허용 길이가 1024바이트에서 2000바이트로 증가했습니다.
- 현재 연결이 설정되는 값보다 클 때 파라미터 그룹에서 `max_connections` 파라미터를 낮추면 사용자가 데이터베이스에 액세스할 수 없던 문제를 해결했습니다.
- 데이터 활동 스트림에서 작은 따옴표와 백슬래시가 제대로 이스케이프되지 않는 문제를 해결했습니다.

MySQL 커뮤니티 에디션 버그 픽스 통합

- 버그 #27659490: 동적 범위 및 인덱스 병합을 사용하여 선택하면 너무 많은 메모리를 사용합니다(OOM).
- 버그 #26881508: MYSQL #1: AUTH_COMMON.H에서 DISABLE_ABORT_ON_ERROR가 발생합니다.
- 버그 #24437124: 테이블 생성 시 버퍼 오버플로가 발생할 수 있습니다.
- 버그 #27158030: INNODB 온라인 변경이 동시 DML과 충돌합니다.
- 버그 #29770705: 특정 WHERE 절을 사용하여 SELECT를 실행하는 동안 서버가 충돌했습니다.
- 버그 #26502135: MYSQLD SEGFAULTS IN MDL_CONTEXT::TRY_ACQUIRE_LOCK_IMPL
- 버그 #26935001: ALTER TABLE AUTO_INCREMENT가 폐기된 테이블스페이스에서 인덱스를 읽으려고 시도합니다.
- 버그 #28491099: [치명적] 메모리 블록이 유효하지 않음 | INNODB: 어설션 실패: UT0UT.CC:670
- 버그 #30499288: GCC 9.2.1 OS_FILE_GET_PARENT_DIR에 대한 새 경고가 보고됩니다.
- 버그 #29952565: 쿼리(UNION + ORDER BY + SUB-QUERY)를 실행하는 동안 MYSQLD에서 신호 11이 발생합니다.
- 버그 #30628268: 메모리 부족 충돌
- 버그 #30441969: 버그 #29723340: 데이터 세트가 있는 SQL 쿼리 후 MySQL Server 충돌
- 버그 #30569003: 5.7 권한 부여 관리와 함께 구문 오류가 발생하여 복제가 중단됩니다.
- 버그 #29915479: COM_BINLOG_DUMP 없이 COM_REGISTER_SLAVE를 실행하면 서버가 종료될 수 있습니다.
- 버그 #30569003: 5.7 권한 부여 관리와 함께 구문 오류가 발생하여 복제가 중단됩니다.
- 버그 #29915479: COM_BINLOG_DUMP 없이 COM_REGISTER_SLAVE를 실행하면 서버가 종료될 수 있습니다.
- Bug #20712046: SHOW PROCESSLIST 및 PERFORMANCE_SCHEMA TABLES가 쿼리에서 암호를 마스킹하지 않습니다.
- 버그 #18898433: 외부 조인 및 조인 버퍼로 인해 성능이 매우 느립니다(5.7.21에서 수정됨).
- 버그 #26402045: 쿼리에서 MYSQLD가 충돌합니다(MySQL 5.7.23에서 수정).
- 버그 #23103937: PS_TRUNCATE_ALL_TABLES()이 SUPER_READ_ONLY MODE 모드에서 작동하지 않습니다.
- 버그 #26666274: 성능 스키마 버퍼 컨테이너에서 무한 루프가 발생합니다.

- 버그 #26997096: relay_log_space 값이 동기화된 방식으로 업데이트되지 않으므로 값이 릴레이 로그에서 사용되는 실제 디스크 공간보다 훨씬 높습니다. (https://github.com/mysql/mysql-server/commit/78f25d2809ad457e81f90342239c9bc32a36cdfa)
- 버그 #25082593: 외래 키 유효성 검사 시 커밋된 읽기에서 갭 잠금을 획득할 필요가 없습니다.
- 버그 #24764800: XAER_RMFAIL 오류와 함께 슬레이브에서 복제가 실패합니다.
- 버그 #81441: SKIP-NAME-RESOLVE를 사용할 때 로컬 호스트에 대한 경고가 표시됩니다.

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- AWS Lambda 함수를 비동기식으로 호출하기 위한 네이티브 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩

- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- CREATE TABLESPACE SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2022년 1월 6일(버전 2.08.4)(사용되지 않음)

버전: 2.08.4

Aurora MySQL 2.08.4가 정식 출시되었습니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.10.0으로 업그레이드할 수 있습니다. Aurora MySQL 버전 1을 실행하는 클러스터의 경우 기존 Aurora MySQL 1.23 이상의 클러스터를 2.10.0으로 직접 업그레이드할 수 있습니다. 또한 현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.10.0으로 스냅샷을 복원할 수 있습니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 AWS Management Console, AWS CLI 또는 Amazon RDS API를 통해 엔진 버전을 지정합니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

보안 수정 사항 및 일반 개선 사항:

- Amazon S3, Amazon ML, AWS Lambda 등의 기타 AWS 서비스와 Aurora MySQL를 통합하는 것과 관련된 보안 문제를 수정했습니다.
- 다음 조건을 모두 충족할 때 기존 행의 TIMESTAMP 열 값이 최신 타임스탬프로 업데이트되는 문제를 수정했습니다. 1. 테이블에 대한 트리거가 있습니다. 2. ON DUPLATE KEY UPDATE 절이 있는 테이블에 대해 INSERT가 수행됩니다. 3. 삽입된 행이 UNIQUE 인덱스 또는 PRIMARY KEY에서 중복 값을 위반합니다. 4. 하나 이상의 열은 TIMESTAMP 데이터 형식이며 기본값은 CURRENT_TIMESTAMP입니다.
- 드문 경우지만 인 메모리 데이터 무결성 검사가 실패할 때 라이터 인스턴스가 다시 시작되는 문제를 수정했습니다.
- LIMIT 절을 사용하여 SQL 문을 실행할 때 데이터베이스가 다시 시작될 수 있는 병렬 쿼리 문제를 수정했습니다.

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- AWS Lambda 함수를 비동기식으로 호출하기 위한 네이티브 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2020년 11월 12일(버전 2.08.3)(사용되지 않음)

버전: 2.08.3

Aurora MySQL 2.08.3이 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

기존 Aurora MySQL 2.* 데이터베이스 클러스터를 직접 Aurora MySQL 2.08.3으로 업그레이드할 수 있습니다. 기존 Aurora MySQL 1.* 클러스터를 2.07.3 이상으로 직접 업그레이드한 다음 2.08.3으로 직접 업그레이드할 수 있습니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 AWS Management Console, AWS CLI 또는 RDS API를 통해 엔진 버전을 지정합니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

보안 수정:

관리된 환경에서 처리를 미세 조정하기 위한 수정 사항 및 기타 향상된 기능 아래의 추가 CVE 수정:

- 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

호환되지 않는 변경 사항:

이 버전에서는 mysqldump 명령의 동작에 영향을 주는 권한 변경을 도입했습니다. 사용자는 PROCESS 테이블에 액세스하려면 INFORMATION_SCHEMA.FILES 권한이 있어야 합니다. 변경 없이 mysqldump 명령을 실행하려면 PROCESS 명령이 연결되는 데이터베이스 사용자에게 mysqldump 권한을 부여합니다. mysqldump 옵션을 사용하여 --no-tablespaces 명령을 실행할 수도 있습니다. 이 옵션을 사용하면 mysqldump 출력에 CREATE LOGFILE GROUP 또는 CREATE TABLESPACE 문이 포함되지 않습니다. 이 경우 mysqldump 명령은 INFORMATION_SCHEMA.FILES 테이블에 액세스하지 않으므로 PROCESS 권한을 부여할 필요가 없습니다.

MySQL 커뮤니티 에디션 버그 픽스 통합

- 버그 #23762382 - 선택 항목에 조인된 값 쿼리를 입력하면 잘못된 동작이 발생합니다.
- 버그 #25209512 - 현재_타임 스탬프가 트리거에서 0을 생성합니다.

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- AWS Lambda 함수를 비동기식으로 호출하기 위한 네이티브 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2020년 8월 28일(버전 2.08.2)(사용되지 않음)

버전: 2.08.2

Aurora MySQL 2.08.2가 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.08.2로 스냅샷을 복원할 수 있습니다. 또한 기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.08.2로 업그레이드할 수 있는 옵션도 있습니다. 기존 Aurora MySQL 1.* 클러스터를 2.08.2로 직접 업그레이드할 수 없습니다. 하지만 이 클러스터의 스냅샷을 Aurora MySQL 2.08.2로 복원할 수 있습니다. 스냅샷 복원에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 DB 클러스터 스냅샷에서 복원을 참조하세요.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 AWS Management Console, AWS CLI 또는 RDS API를 통해 엔진 버전을 지정합니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

심각한 수정 사항:

- 예기치 않은 중단을 일으키고 데이터베이스 가용성에 영향을 줄 수 있는 문제를 수정했습니다.

가용성 수정 사항:

- Aurora MySQL 데이터베이스가 binlog 복제본이고 mysql `time_zone` 테이블을 통해 DDL 이벤트를 복제하는 경우 데이터베이스가 다시 시작될 수 있던 문제를 수정했습니다.

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- AWS Lambda 함수를 비동기식으로 호출하기 위한 네이티브 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2020년 6월 18일(버전 2.08.1)(사용되지 않음)

버전: 2.08.1

Aurora MySQL 2.08.1은 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.08.1로 스냅샷을 복원할 수 있습니다. 또한 기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.08.1로 업그레이드할 수 있는 옵션도 있습니다. 기존 Aurora MySQL 1.* 클러스터를 2.08.1로 직접 업그레이드할 수 없습니다. 하지만 이 클러스터의 스냅샷을 Aurora MySQL 1.08.1로 복원할 수 있습니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 AWS Management Console, AWS CLI 또는 RDS API를 통해 엔진 버전을 지정합니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

새로운 기능:

- 글로벌 데이터베이스 쓰기 전달입니다. Aurora 글로벌 데이터베이스에서는 이제 보조 클러스터에 연결된 동안 DML 문과 같은 특정 쓰기 작업을 수행할 수 있습니다. 쓰기 작업은 기본 클러스터로 전달되고 모든 변경 사항은 보조 클러스터로 다시 복제됩니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora 글로벌 데이터베이스에서 쓰기 전달 사용을 참조하세요.

일반적인 안정성 수정 사항:

- S3 백업에 `mysql.host` 테이블이 포함되지 않은 경우 Amazon S3에서 Aurora MySQL 버전 2.08.0으로 클러스터를 복원하는 데 예상보다 오래 걸리는 문제가 해결되었습니다.

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.

- AWS Lambda 함수를 비동기식으로 호출하기 위한 네이티브 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2020년 6월 2일(버전 2.08.0)(사용되지 않음)

버전: 2.08.0

Aurora MySQL 2.08.0이 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.08.0으로 스냅샷을 복원할 수 있습니다. 또한 기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.08.0으로 업그레이드할 수 있는 옵션도 있습니다. 기존 Aurora MySQL 1.* 클러스터를 2.08.0으로 직접 업그레이드할 수 없습니다. 하지만 이 클러스터의 스냅샷을 Aurora MySQL 0.08.0으로 복원할 수 있습니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 AWS Management Console, AWS CLI 또는 RDS API를 통해 엔진 버전을 지정합니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

새로운 기능:

- 대규모 트랜잭션이 관련된 경우 충돌 복구 시간과 커밋 시간 지연을 줄이기 위해 바이너리 로그(binlog) 처리를 개선했습니다.
- Aurora MySQL의 DAS(데이터베이스 활동 스트림) 기능 시작. 이 기능은 관계형 데이터베이스에서 데이터베이스 활동에 대해 거의 실시간으로 데이터 스트림을 제공하여 활동을 모니터링하는 데 도움을 줍니다. 자세한 내용은 Amazon Aurora 사용 설명서의 데이터베이스 활동 스트림을 사용하여 Amazon Aurora 모니터링을 참조하세요.
- 최신 브라질 표준 시간대 변경을 지원하기 위해 표준 시간대 파일을 업데이트했습니다.
- SQL에서 특정 테이블 및/또는 내부 테이블에 대한 해시 조인 기능을 실행하도록 새 키워드를 도입했습니다(`HASH_JOIN`, `HASH_JOIN_PROBING`, `HASH_JOIN_BUILDING`) 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 힌트를 참조하세요.

- Aurora MySQL 5.7에서 MySQL 8.0 기능을 백포팅하여 조인 순서 힌트 지원을 도입했습니다. 새로운 힌트는 `JOIN_FIXED_ORDER`, `JOIN_ORDER`, `JOIN_PREFIX`, `JOIN_SUFFIX`입니다. 조인 순서 힌트 지원에 대한 자세한 설명서는 WL#9158: Join Order Hints를 참조하십시오.
- Aurora Machine Learning이 이제 반환 유형으로 `MEDIUMINT`가 있는 사용자 정의 함수를 지원합니다.
- `lambda_async()` 저장 프로시저가 이제 모든 MySQL `utf8` 문자를 지원합니다.

높은 우선 순위:

- 라이터 DB 인스턴스에서 `INFORMATION_SCHEMA.INNODB_SYS_TABLES` 테이블을 쿼리한 후 리더 DB 인스턴스가 FTS 쿼리에 대해 불완전한 결과를 반환할 수 있는 문제를 수정했습니다.
- CVE-2019-5443
- CVE-2019-3822

가용성 향상:

- 쿼리 캐시가 활성화된 상태에서 여러 테이블 또는 데이터베이스에 액세스하는 다중 쿼리 문이 실행된 후 데이터베이스가 다시 시작되는 문제를 수정했습니다.
- 트랜잭션 롤백 중에 데이터베이스 재시작 또는 장애 조치로 이어지는 잠금 관리자의 교착 상태를 수정했습니다.
- 여러 연결이 전체 텍스트 검색 인덱스로 동일한 테이블을 업데이트하려고 하면 데이터베이스 재시작 또는 장애 조치를 트리거한 문제를 수정했습니다.
- `kill session` 명령 중 데이터베이스 재시작 또는 장애 조치를 트리거할 수 있는 문제를 수정했습니다. 이 문제가 발생하면 AWS Support 팀에 문의하여 인스턴스에서 이 수정 사항을 활성화하세요.
- `SELECT`가 활성화된 라이터 DB 인스턴스에서 다중 `AUTOCOMMIT` 문과 과도한 쓰기 워크로드가 있는 다중 문 트랜잭션 중에 리더 DB 인스턴스가 다시 시작되던 문제를 수정했습니다.
- 라이터 DB 인스턴스에 OLTP 쓰기 워크로드가 많을 때 장기 실행 쿼리를 실행한 후 리더 DB 인스턴스가 다시 시작되던 문제를 수정했습니다.

일반적인 개선 사항

- binlog가 활성화되었을 때 장기 실행 트랜잭션의 데이터베이스 복구 시간 및 커밋 지연 시간을 개선했습니다.

- 데이터 분포가 왜곡된 열을 포함하여 인덱싱된 열의 고유 값 수를 추정하기 위한 더 나은 통계를 생성하도록 알고리즘을 개선했습니다.
- MyISAM 임시 테이블에 액세스하는 조인 쿼리의 응답 시간 및 CPU 사용률이 감소했으며 결과가 로컬 스토리지로 유출됩니다.
- 공백이 포함된 데이터베이스 또는 테이블 이름의 Aurora MySQL 5.6 스냅샷이 새 Aurora MySQL 5.7 클러스터로 복원되지 않던 문제를 수정했습니다.
- 교착 상태가 `show engine innodb status`에서 해결되었을 때 피해 대상 트랜잭션 정보를 포함했습니다.
- 여러 버전의 클라이언트가 동일한 데이터베이스에 연결되어 있고 쿼리 캐시에 액세스할 때 연결이 중단되는 문제를 수정했습니다.
- 데이터베이스 인스턴스의 수명 기간 동안 ZDP(제로 가동 중지 패치) 또는 ZDR(제로 가동 중지 재시작) 워크플로우를 여러 번 호출하여 발생하는 메모리 누수를 수정했습니다.
- ZDP(제로 가동 중지 패치) 또는 ZDR(제로 가동 중지 재시작) 작업에서 자동 커밋 플래그가 해제되면 마지막 트랜잭션이 중단되었다는 오류 메시지를 수정했습니다.
- 새 데이터베이스 프로세스에서 사용자 세션 변수를 복원할 때 서버 오류 메시지가 나타날 수 있는 ZDP(제로 가동 중지 패치) 작업 문제를 수정했습니다.
- 패치 적용 중에 장기 실행 쿼리가 있으면 간헐적인 데이터베이스 오류가 발생할 수 있는 ZDP(제로 가동 중지 패치) 작업의 문제를 수정했습니다.
- Amazon SageMaker AI 및 Amazon Comprehend와 같은 기계 학습 서비스에서 잘못 처리한 오류 응답으로 인해 Aurora Machine Learning 함수를 포함한 쿼리에서 빈 오류 메시지를 반환한 문제를 해결했습니다.
- `table_definition_cache` 파라미터의 사용자 지정 값을 따르지 않는 메모리 부족 모니터링 기능의 문제를 수정했습니다.
- Aurora Machine Learning 쿼리가 중단되면 “쿼리 실행이 중단되었습니다”라는 오류 메시지가 반환됩니다. 이전에는 “ML 요청을 처리하는 중 내부 오류”라는 일반 메시지가 대신 반환되었습니다.
- `slave_net_timeout` 파라미터가 `aurora_binlog_replication_max_yield_seconds` 파라미터보다 작고 binlog 마스터 클러스터에서 워크로드가 적을 때 binlog 작업자에 연결 시간 초과가 발생할 수 있는 문제를 수정했습니다.
- 분당 하나의 메시지 빈도로 오류 로그에 정보 메시지를 출력하여 binlog 복구 진행률을 모니터링하는 기능을 개선했습니다.
- 활성 트랜잭션이 `SHOW ENGINE INNODB STATUS` 쿼리에서 보고되지 않을 수 있는 문제를 수정했습니다.

MySQL 커뮤니티 에디션 버그 픽스 통합

- 버그 #25289359: 전체 텍스트 캐시 크기가 전체 텍스트 캐시 크기 제한을 초과한 경우에 데이터가 동기화될 때 취해진 전체 텍스트 캐시 잠금이 해제되지 않았습니다.
- 버그 #29138644: MySQL 서버가 실행되는 동안 시스템 시간을 수동으로 변경하면 페이지 클리너 스레드가 지연됩니다.
- 버그 #25222337: 가상 인덱스의 NULL 가상 열 필드 이름으로 인해 외래 키 제약 조건의 영향을 받는 가상 열을 채우면서 필드 이름을 비교하는 동안 서버가 종료되었습니다.
- 버그 #25053286: 보기에 액세스한 쿼리가 포함된 저장 프로시저를 실행하면 세션이 종료될 때까지 비워지지 않은 메모리를 할당할 수 있습니다.
- 버그 #25586773: 특정 SELECT 문의 내용에서 테이블을 만든 문을 포함하는 저장 프로시저를 실행하면 메모리 누수가 발생할 수 있습니다.
- 버그 #28834208: 로그 애플리케이션 중에 OPTIMIZE TABLE 작업 후 InnoDB가 가상 열 인덱스 업데이트를 확인하기 전에 가상 열을 채우지 않았습니다.
- 버그 #26666274: 32비트 부호 없는 정수 오버플로우로 인해 성능 스키마 버퍼 컨테이너의 무한 루프가 발생합니다.

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- AWS Lambda 함수를 비동기식으로 호출하기 위한 네이티브 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2023년 8월 15일(버전 2.07.10)(사용되지 않음)

버전: 2.07.10

Aurora MySQL 2.07.10이 정식 출시되었습니다. Aurora MySQL 2.07 버전은 MySQL 5.7.12와 호환됩니다. 커뮤니티 변경 사항에 대한 자세한 내용은 Changes in MySQL 5.7.12(2016-04-11, General Availability)를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 2.07.*, 2.11.*, 3.01.*, 3.02.*, 3.03.* 및 3.04.*입니다.

현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.07.10으로 스냅샷을 복원할 수 있습니다. 기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.07.10으로 업그레이드할 수 있는 옵션도 있습니다. Aurora MySQL 1.* 클러스터에서 Aurora MySQL 2.*로 인플레이스 업그레이드가 가능합니다(Aurora MySQL 1.x에서 2.x로 업그레이드 참조). 또한 Aurora MySQL 2.* 클러스터에서

Aurora MySQL 3.*로도 인플레이스 업그레이드가 가능합니다(Aurora MySQL 2.x에서 3.x로 업그레이드 참조).

Aurora MySQL 2.07.10으로의 인플레이스 엔진 버전 업그레이드가 수행된 직후, 해당 인스턴스가 이전 운영 체제 버전을 실행하는 경우 db.r4, db.r5, db.t2 및 db.t3 DB 인스턴스 클래스의 영향을 받는 모든 인스턴스에 운영 체제 업그레이드가 자동으로 적용됩니다. 다중 AZ DB 클러스터에서는 모든 리더 인스턴스에 먼저 운영 체제 업그레이드가 적용됩니다. 첫 번째 리더 인스턴스의 운영 체제 업그레이드가 완료되면 장애 조치가 수행되고 이전 라이터 인스턴스가 업그레이드됩니다.

> **Note**
>
> 메이저 버전 업그레이드 중에는 Aurora 글로벌 데이터베이스에 운영 체제 업그레이드가 자동으로 적용되지 않습니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

아래 나열된 보안 문제 및 CVE 해결:

관리된 환경에서 처리를 미세 조정하기 위한 수정 사항 및 기타 향상된 기능 아래의 추가 CVE 수정:

- CVE-2023-21963
- CVE-2023-21912
- CVE-2023-0215
- CVE-2022-43551
- CVE-2022-37434
- 감사 로그 교체를 처리하는 동안 보고된 이벤트가 감사 로그에 기록되지 않는 문제를 해결했습니다.
- 보안 수준이 낮은 DES-CBC3-SHA 값을 SSL_CIPHER 데이터베이스 파라미터에서 제외하도록 Aurora MySQL에서 사용하는 기본 SSL 암호를 업데이트했습니다. DES-CBC3-SHA 암호 제거로 인해 SSL 연결 문제가 발생하는 경우 Aurora MySQL DB 클러스터 연결을 위한 암호 그룹 구성에 나온 적용 가능한 보안 암호를 사용하세요.

- OpenSSL이 1.0.2zh 버전으로 업그레이드되었습니다.

일반적인 개선 사항:

- 암호화에 더 작은 크기의 키를 사용하는 ECDHE-RSA SSL 암호에 대한 지원을 추가했습니다.
- 해시 조인을 사용하여 쿼리를 실행할 때 발생하는 메모리 관리 문제를 해결했습니다.

Aurora MySQL 버전 2에서 지원되지 않는 기능

다음 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환)에서는 현재 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP).
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 단원을 참조하십시오.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- `CREATE TABLESPACE` SQL 문
- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- X 프로토콜

Aurora MySQL 데이터베이스 엔진 업데이트 2023년 5월 4일(버전 2.07.9)(사용되지 않음)

버전: 2.07.9

Aurora MySQL 2.07.9가 정식 출시되었습니다. Aurora MySQL 2.07 버전은 MySQL 5.7.12와 호환됩니다. 커뮤니티 변경 사항에 대한 자세한 내용은 Changes in MySQL 5.7.12(2016-04-11, General Availability)를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 2.07.*, 2.11.*, 3.01.*, 3.02.* 및 3.03.*입니다.

현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.07.9로 스냅샷을 복원할 수 있습니다. 기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.07.9로 업그레이드할 수 있는 옵션도 있습니다. Aurora MySQL 1.* 클러스터에서 Aurora MySQL 2.*로 인플레이스 업그레이드가 가능합니다(Aurora MySQL 1.x에서 2.x로 업그레이드 참조). 또한 Aurora MySQL 2.* 클러스터에서 Aurora MySQL 3.*로도 인플레이스 업그레이드가 가능합니다(Aurora MySQL 2.x에서 3.x로 업그레이드 참조).

Aurora MySQL 2.07.9로의 인플레이스 엔진 버전 업그레이드가 수행된 직후, 해당 인스턴스가 이전 운영 체제 버전을 실행하는 경우 db.r4, db.r5, db.t2 및 db.t3 DB 인스턴스 클래스의 영향을 받는 모든 인스턴스에 운영 체제 업그레이드가 자동으로 적용됩니다. 다중 AZ DB 클러스터에서는 모든 리더 인스턴스에 먼저 운영 체제 업그레이드가 적용됩니다. 첫 번째 리더 인스턴스의 운영 체제 업그레이드가 완료되면 장애 조치가 수행되고 이전 라이터 인스턴스가 업그레이드됩니다.

> **Note**
>
> 메이저 버전 업그레이드 중에는 Aurora 글로벌 데이터베이스에 운영 체제 업그레이드가 자동으로 적용되지 않습니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

아래 나열된 보안 문제 및 CVE 해결:

관리된 환경에서 처리를 미세 조정하기 위한 수정 사항 및 기타 향상된 기능 아래의 추가 CVE 수정:

- CVE-2022-32221

가용성 향상:

- 고급 감사 로그 교체로 인해 사용 가능한 메모리가 줄어들어 DB 인스턴스가 다시 시작될 수 있는 문제를 해결했습니다.
- 데이터베이스가 다시 시작되는 동안 발생하여 장기간 데이터베이스가 제대로 시작되지 않는 결과를 초래하는 문제를 해결했습니다.

일반적인 개선 사항:

- 드문 경우지만 데이터베이스 볼륨이 160GB의 배수로 증가하면 인스턴스가 다시 시작될 수 있는 문제를 수정했습니다.

Aurora MySQL 버전 2에서 지원되지 않는 기능

다음 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환)에서는 현재 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP).
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 단원을 참조하십시오.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩

- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2022년 6월 16일(버전 2.07.8)(사용되지 않음)

버전: 2.07.8

Aurora MySQL 2.07.8이 정식 출시되었습니다. Aurora MySQL 2.* 버전은 MySQL 5.7과 호환되고 Aurora MySQL 1.* 버전은 MySQL 5.6과 호환됩니다.

> **Note**
>
> 이 버전은 장기 지원(LTS) 릴리스로 지정됩니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL LTS(장기 지원) 릴리스를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.07.8로 스냅샷을 복원할 수 있습니다. 기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.07.8로 업그레이드할 수 있는 옵션도 있습니다. Aurora MySQL 1.* 클러스터에서 Aurora MySQL 2.*로 인플레이스 업그레이드가 가능합니다(Aurora MySQL 1.x에서 2.x로 업그레이드 참조). 또한 Aurora MySQL 2.* 클러스터에서 Aurora MySQL 3.*로도 인플레이스 업그레이드가 가능합니다(Aurora MySQL 2.x에서 3.x로 업그레이드 참조).

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 AWS Management Console, AWS CLI 또는 RDS API를 통해 엔진 버전을 지정합니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

보안 수정:

관리된 환경에서 처리를 미세 조정하기 위한 수정 사항 및 기타 향상된 기능 아래의 추가 CVE 수정:

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

일반적인 개선 사항

- 드문 경우지만 경합 상태로 인해 교착 상태 감지기 스레드가 중단되면 데이터베이스 서버가 다시 시작될 수 있는 문제를 해결했습니다.

MySQL 커뮤니티 에디션 버그 픽스 통합

- UPDATE에 1,024바이트보다 큰 프라이머리 키가 있는 임시 테이블이 필요하고 해당 테이블이 InnoDB를 사용하여 생성된 경우 서버가 종료될 수 있습니다. (버그 #25153670)

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.

- AWS Lambda 함수를 비동기식으로 호출하기 위한 네이티브 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2021년 11월 24일(버전 2.07.7)(사용되지 않음)

버전: 2.07.7

Aurora MySQL 2.07.7이 정식 버전입니다. Aurora MySQL 2.* 버전은 MySQL 5.7과 호환되고 Aurora MySQL 1.* 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.07.7로 스냅샷을 복원할 수 있습니다. 기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.07.7로 업그레이드할 수 있는 옵션도 있습니다. 기존 Aurora MySQL 1.* 클러스터를 2.07.7로 직접 업그레이드할 수 없습니다. 하지만 이 클러스터의 스냅샷을 Aurora MySQL 2.07.7로 복원할 수 있습니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 AWS Management Console, AWS CLI 또는 RDS API를 통해 엔진 버전을 지정합니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

보안 수정:

관리된 환경에서 처리를 미세 조정하기 위한 수정 사항 및 기타 향상된 기능 아래의 추가 CVE 수정:

- CVE-2019-17543
- CVE-2019-2960

일반적인 개선 사항

- Amazon S3, Amazon ML, Lambda와 같은 다른 AWS 서비스와 Aurora MySQL를 통합하는 것과 관련된 보안 문제가 수정되었습니다.
- Aurora 복제 지연을 잘못 보고하는 문제가 해결되었습니다.
- 데이터베이스에 사용자 및 권한 조합이 많은 경우 데이터베이스 인스턴스 재시작이 실패할 수 있는 문제를 수정했습니다.

- Aurora-MySQL 1.x(MySQL 5.6 기반)에서 Aurora-MySQL 2.x(MySQL 5.7 기반)로 인플레이스 메이저 버전을 업그레이드한 후 general_log 및 slow_log 테이블에 액세스할 수 없게 되는 문제를 수정했습니다.
- 드문 경우지만 잘못된 확인 처리로 인해 리더 인스턴스가 다시 시작될 수 있는 문제가 해결되었습니다.
- 드물게 세션 처리가 완료되고 유휴 상태인 경우에도 PI (Performance Insights) 세션의 '데이터베이스 로드' 차트가 CPU를 적극적으로 사용하는 것으로 표시되는 문제가 해결되었습니다.
- LIMIT 절을 사용하여 SQL 문을 실행할 때 데이터베이스가 다시 시작될 수 있는 병렬 쿼리 문제를 수정했습니다.
- 다음 조건이 모두 충족되면 기존 행의 TIMESTAMP 열 값이 최신 타임스탬프로 업데이트되는 문제가 해결되었습니다. 1. 테이블에 대한 트리거가 존재합니다. 2. INSERT는 ON DUPLICE KEY UPDATE 절이 있는 테이블에서 수행됩니다. 3. 삽입된 행은 UNIQUE 인덱스 또는 PRIMARY KEY에서 중복 값 위반이 발생할 수 있으며, 4. 하나 이상의 열이 TIMESTAMP 데이터 유형이고 기본값은 CURRENT_TIMESTAMP입니다.
- 드문 경우지만 READ COMMITED 격리 수준에서 XA 트랜잭션을 사용할 때 데이터베이스 인스턴스가 다시 시작될 수 있는 문제가 해결되었습니다.

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- AWS Lambda 함수를 비동기식으로 호출하기 위한 네이티브 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2021년 9월 2일(버전 2.07.6)(사용되지 않음)

버전: 2.07.6

Aurora MySQL 2.07.6이 정식 출시되었습니다. Aurora MySQL 2.* 버전은 MySQL 5.7과 호환되고 Aurora MySQL 1.* 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.07.6으로 스냅샷을 복원할 수 있습니다. 기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.07.6로 업그레이드할 수 있는 옵션도 있습니다. 기존 Aurora MySQL 1.* 클러스터를 2.07.6으로 직접 업그레이드할 수 없습니다. 하지만 이 클러스터의 스냅샷을 Aurora MySQL 2.07.6으로 복원할 수 있습니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 AWS Management Console, AWS CLI 또는 RDS API를 통해 엔진 버전을 지정합니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

MySQL 커뮤니티 에디션 버그 픽스 통합

- 64K 크기의 레코드를 삽입하는 데 너무 많은 시간이 걸립니다. (버그#23031146)

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- AWS Lambda 함수를 비동기식으로 호출하기 위한 네이티브 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인

- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2021년 7월 6일(버전 2.07.5)(사용되지 않음)

버전: 2.07.5

Aurora MySQL 2.07.5가 정식 버전입니다. Aurora MySQL 2.* 버전은 MySQL 5.7과 호환되고 Aurora MySQL 1.* 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.07.5로 스냅샷을 복원할 수 있습니다. 기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.07.5로 업그레이드할 수 있는 옵션도 있습니다. 기존 Aurora MySQL 1.* 클러스터를 2.07.5로 직접 업그레이드할 수 없습니다. 하지만 이 클러스터의 스냅샷을 Aurora MySQL 2.07.5로 복원할 수 있습니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 AWS Management Console, AWS CLI 또는 RDS API를 통해 엔진 버전을 지정합니다.

> **Note**
>
> 이 버전은 장기 지원(LTS) 릴리스로 지정됩니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL LTS(장기 지원) 릴리스를 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

가용성 향상:

- Aurora 복제본에서 사용자 수준 잠금이 허용되지 않던 문제를 해결했습니다.
- `READ COMMITTED` 격리 수준에서 XA 트랜잭션을 사용할 때 데이터베이스 재시작을 유발할 수 있는 문제를 해결했습니다
- `server_audit_incl_users` 및 `server_audit_excl_users` 글로벌 파라미터에 대한 최대 허용 길이를 2,000으로 확장했습니다.

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- AWS Lambda 함수를 비동기식으로 호출하기 위한 네이티브 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2021년 3월 4일(버전 2.07.4)(사용되지 않음)

버전: 2.07.4

Aurora MySQL 2.07.2가 정식 출시되었습니다. Aurora MySQL 2.* 버전은 MySQL 5.7과 호환되고 Aurora MySQL 1.* 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.07.4로 스냅샷을 복원할 수 있습니다. 기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.07.4로 업그레이드할 수 있는 옵션도 있습니다. 기존 Aurora MySQL 1.* 클러스터를 2.07.4로 직접 업그레이드할 수 없습니다. 하지만 이 클러스터의 스냅샷을 Aurora MySQL 2.07.4로 복원할 수 있습니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 AWS Management Console, AWS CLI 또는 RDS API를 통해 엔진 버전을 지정합니다.

> **Note**
>
> 이 버전은 장기 지원(LTS) 릴리스로 지정됩니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL LTS(장기 지원) 릴리스를 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

보안 수정:

- 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

가용성 향상:

- 네트워크 패킷을 읽거나 쓰는 동안 네트워크 오류가 발생할 경우 클라이언트가 정지될 수 있는 문제를 수정했습니다.
- 일부 경우에 DDL이 중단된 후 엔진 재시작 시간이 개선되었습니다.
- DDL 또는 DML로 인해 페이지 미리 가져오기 요청 중에 엔진이 재시작될 수 있는 문제가 해결되었습니다.
- Aurora 읽기 전용 복제본에서 테이블 또는 인덱스의 역방향 검색을 수행하는 동안 복제본이 재시작될 수 있는 문제가 해결되었습니다.
- 복제 클러스터 작업에서 복제본 생성 시간이 더 오래 걸리는 문제를 수정했습니다.
- 지리 공간 열에 병렬 쿼리 최적화를 사용할 때 데이터베이스가 다시 시작될 수 있는 문제를 해결했습니다.
- `HA_ERR_KEY_NOT_FOUND` 오류가 발생하면서 binlog 복제본이 중지될 수 있는 문제를 수정했습니다.

MySQL 커뮤니티 에디션 버그 픽스 통합

- ' '(공백), '%' 또는 ', '가 포함된 토큰을 처리할 때 전체 텍스트 ngram 파서에서 발생하던 문제가 수정되었습니다. ngram 파서를 사용하는 경우 고객은 FTS 인덱스를 재구축해야 합니다. (버그 #25873310)
- 중첩된 SQL 뷰를 사용하여 쿼리를 실행하는 동안 엔진이 재시작될 수 있는 문제가 해결되었습니다. (버그 #27214153, 버그 #26864199)

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- AWS Lambda 함수를 비동기식으로 호출하기 위한 네이티브 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩

- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- CREATE TABLESPACE SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2020년 11월 10일(버전 2.07.3)(사용되지 않음)

버전: 2.07.3

Aurora MySQL 2.07.3이 정식 버전입니다. Aurora MySQL 2.* 버전은 MySQL 5.7과 호환되고 Aurora MySQL 1.* 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.07.3으로 스냅샷을 복원할 수 있습니다. 기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.07.3로 업그레이드할 수 있는 옵션도 있습니다. 기존 Aurora MySQL 1.* 클러스터를 2.07.3으로 직접 업그레이드할 수 없습니다. 하지만 이 클러스터의 스냅샷을 Aurora MySQL 2.07.3으로 복원할 수 있습니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 AWS Management Console, AWS CLI 또는 RDS API를 통해 엔진 버전을 지정합니다.

> **Note**
>
> 이 버전은 장기 지원(LTS) 릴리스로 지정됩니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL LTS(장기 지원) 릴리스를 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

보안 수정:

관리된 환경에서 처리를 미세 조정하기 위한 수정 사항 및 기타 향상된 기능

- 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

호환되지 않는 변경 사항:

이 버전에서는 mysqldump 명령의 동작에 영향을 주는 권한 변경을 도입했습니다. 사용자는 PROCESS 테이블에 액세스하려면 INFORMATION_SCHEMA.FILES 권한이 있어야 합니다. 변경 없이 mysqldump 명령을 실행하려면 PROCESS 명령이 연결되는 데이터베이스 사용자에게 mysqldump 권한을 부여합니다. mysqldump 옵션을 사용하여 --no-tablespaces 명령을 실행할 수도 있습니다. 이 옵션을 사용하면 mysqldump 출력에 CREATE LOGFILE GROUP 또는 CREATE TABLESPACE 문이 포함되지 않습니다. 이 경우 mysqldump 명령은 INFORMATION_SCHEMA.FILES 테이블에 액세스하지 않으므로 PROCESS 권한을 부여할 필요가 없습니다.

가용성 향상:

- 잠금 관리자에서 연결/쿼리 종료 및 데이터베이스 재시작 결과 세션 종료 간의 경쟁 조건을 수정했습니다.

- 쿼리 캐시가 활성화된 상태에서 여러 테이블 또는 데이터베이스에 액세스하는 다중 쿼리 문이 실행된 후 데이터베이스가 다시 시작되는 문제를 수정했습니다.
- 보조 인덱스가 있는 가상 열의 업데이트로 인해 재시작이 반복되는 문제를 해결했습니다.

MySQL 커뮤니티 에디션 버그 픽스 통합

- InnoDB: 마스터의 XA 준비 단계에 성공적으로 실행된 동시 XA 트랜잭션이 슬레이브에서 재생될 때 충돌하여 응용 프로그램 스레드에서 잠금 대기 시간 초과가 발생합니다. 충돌은 트랜잭션이 슬레이브에서 순차적으로 재생될 때 다른 GAP 잠금 범위 때문에 발생합니다. 이러한 유형의 충돌을 방지하기 위해 XA 트랜잭션이 준비 단계에 도달하면 READ COMMITTED 격리 수준에서 XA 트랜잭션에 의해 수행된 GAP 잠금이 해제되고 더 이상 상속되지 않습니다. (버그 #27189701, 버그 #25866046)
- InnoDB: READ COMMITED 격리 수준을 사용하는 동안 외래 키 유효성 검사 중 불필요하게 간격 잠금이 발생했습니다. (버그 #25082593)
- 복제: XA 트랜잭션을 사용할 때 복제 슬레이브의 적용자 (SQL) 스레드에 대한 잠금 대기 시간 초과 또는 교착 상태가 발생한 경우 자동 재시도가 작동하지 않습니다. 원인은 SQL 스레드가 롤백을 수행하는 동안 XA 트랜잭션을 롤백하지 않는다는 것입니다. 즉 트랜잭션을 다시 시도할 때 발생하는 첫 번째 이벤트는 XA 트랜잭션이 이미 진행 중일 때 유효하지 않은 XA START이며 XAER_RMFAIL 오류가 발생합니다. (버그 #24764800)
- 복제: 트랜잭션 격리 수준이 반복 읽기로 설정된 경우 인터리브 트랜잭션이 슬레이브 응용 프로그램이 교착 상태에 빠질 수 있습니다. (버그 #25040331)
- 복제: 기존의 모든 릴레이 로그 파일 (Relay_Log_Space) 의 총 합산 크기에 대해 SHOW 슬레이브 상태문에서 반환된 값이 릴레이 로그 파일에 사용된 실제 디스크 공간보다 훨씬 커질 수 있습니다. I/O 스레드가 값을 업데이트하는 동안 변수를 잠그지 않았으므로 SQL 스레드가 릴레이 로그 파일을 자동으로 삭제하고 I/O 스레드가 값 업데이트를 완료하기 전에 감소된 값을 쓸 수 있습니다. 그런 다음 I/O 스레드는 SQL 스레드의 업데이트를 무시하고 삭제된 파일의 공간을 다시 추가하여 원래 크기 계산을 작성했습니다. 이제 Relay_Log_Space 값이 업데이트 중에 잠겨 동시 업데이트를 방지하고 정확한 계산을 보장합니다. (버그 #26997096, 버그 #87832)
- VALUES 목록에서 조인을 포함하는 하위 쿼리를 사용하여 두 번째 또는 이후 행에 대한 값을 생성하는 INSERT 문의 경우 필요한 권한을 확인하지 못한 후 서버가 종료될 수 있습니다. (버그 #23762382)
- 기본값 CURRENT_TIMESTAMP가 있는 TIMESTAMP 또는 DATETIME 열을 포함하는 테이블의 경우 테이블에 `BEFORE INSERT` 트리거가 있으면 열을 `0000-00-00 00:00:00`로 초기화할 수 있습니다. (버그 #25209512, 버그 #84077)
- 다중 스레드가 메타데이터 성능 스키마 개체를 등록 및 등록 취소하려고 동시에 시도하면 서버 종료가 발생할 수 있습니다. (버그 #26502135)

- 특정 SELECT문의 내용에서 테이블을 만든 문을 포함하는 저장 프로시저를 실행하면 메모리 누수가 발생할 수 있습니다. (버그 #25586773)
- 보기에 액세스한 쿼리가 포함된 저장 프로시저를 실행하면 세션이 종료될 때까지 비워지지 않은 메모리를 할당할 수 있습니다. (버그 #25053286)
- 특정 하위 쿼리 구체화 사례로 인해 서버가 종료될 수 있습니다. 이러한 쿼리는 이제 구체화가 비활성화되었음을 나타내는 오류를 생성합니다. (버그 #26402045)
- 버그 #18898433: 조인 버퍼링을 사용하는 경우(예: 블록 중첩 루프 알고리즘 사용) 왼쪽 조인이 많은 쿼리가 느려졌습니다. (버그 #18898433, 버그 #72854)
- 최적화 프로그램은 두 번째 열에 대해 `LIKE` 절이 있는 내부 조인을 실행할 때 복합 인덱스의 두 번째 열을 건너 뜁니다. (버그 #28086754)

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- AWS Lambda 함수를 비동기식으로 호출하기 위한 네이티브 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2020년 4월 17일(버전 2.07.2)(사용되지 않음)

버전: 2.07.2

Aurora MySQL 2.07.2가 정식 버전입니다. Aurora MySQL 2.* 버전은 MySQL 5.7과 호환되고 Aurora MySQL 1.* 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.07.2로 스냅샷을 복원할 수 있습니다. 또한 기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.07.2로 업그레이드할 수 있는 옵션도 있습니다. 기존 Aurora MySQL 1.* 클러스터를 2.07.2로 직접 업그레이드할 수 없습니다. 하지만 이 클러스터의 스냅샷을 Aurora MySQL 2.07.2로 복원할 수 있습니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 , AWS CLI또는 RDS API AWS Management Console를 통해 엔진 버전을 지정하십시오.

> **Note**
>
> 이 버전은 장기 지원(LTS) 릴리스로 지정됩니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL LTS(장기 지원) 릴리스를 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

보안 수정:

- CVE-2016-8287
- CVE-2016-5634

높은 우선 순위:

- 쓰기 로드가 많은 일부 데이터베이스 클러스터에서 복제가 더 오래 걸리는 문제를 해결했습니다.
- 보조 인덱스를 사용하는 실행 계획이 있는 리더 DB 인스턴스에서 쿼리가 커밋되지 않은 데이터를 반환할 수 있는 문제를 수정했습니다. 이 문제는 기본 또는 보조 인덱스 키 열을 수정하는 데이터 조작 언어(DML) 작업의 영향을 받는 데이터로 제한됩니다.

일반적인 개선 사항:

- 전체 텍스트 검색(FTS) 인덱스를 포함하는 Aurora 1.x DB 클러스터를 Aurora 2.x DB 클러스터로 복원하는 속도가 느려지는 문제를 수정했습니다.
- 테이블 이름에 특수 문자가 있는 분할된 테이블을 포함하는 Aurora 1.x 데이터베이스 스냅샷을 Aurora 2.x DB 클러스터로 복원하는 속도가 느려지는 문제를 수정했습니다.
- 리더 DB 인스턴스에서 느린 쿼리 로그와 일반 로그를 쿼리할 때 오류를 야기하는 문제를 수정했습니다.

MySQL 커뮤니티 에디션 버그 픽스 통합

- 버그 #23104498: 사용된 총 메모리를 보고할 때 성능 스키마의 문제를 수정했습니다(https://github.com/mysql/mysql-server/commit/20b6840df5452f47313c6f9a6ca075bfbc00a96b).
- 버그 #22551677: 데이터베이스를 오프라인으로 전환하려고 할 때 데이터베이스 엔진 충돌을 유발할 수 있는 성능 스키마의 문제를 수정했습니다(https://github.com/mysql/mysql-server/commit/05e2386eccd32b6b444b900c9f8a87a1d8d531e9).

- 버그 #23550835, 버그 #23298025, 버그 #81464: 내부 버퍼 용량 초과로 인해 데이터베이스 엔진 충돌이 발생하는 성능 스키마의 문제를 수정했습니다(https://github.com/mysql/mysql-server/commit/b4287f93857bf2f99b18fd06f555bbe5b12debfc).

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- 동기식 호출 함수를 위한 기본 AWS Lambda 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정

- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2019년 12월 23일(버전 2.07.1)(사용되지 않음)

버전 2.07.1

Aurora MySQL 2.07.1이 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.07.1로 스냅샷을 복원할 수 있습니다. 또한 기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.07.1로 업그레이드할 수 있는 옵션도 있습니다. 기존 Aurora MySQL 1.* 클러스터를 2.07.1로 직접 업그레이드할 수 없습니다. 하지만 이 클러스터의 스냅샷을 Aurora MySQL 1.07.1로 복원할 수 있습니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 , AWS CLI또는 RDS API AWS Management Console를 통해 엔진 버전을 지정하십시오.

> **Note**
>
> 현재이 버전은 AWS GovCloud(미국 동부) [us-gov-east-1], AWS GovCloud(미국 서부) [us-gov-west-1], 중국(닝샤) [cn-northwest-1], 아시아 태평양(홍콩) [ap-east-1], 중동(바레인) [me-south-1] AWS 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼 및 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

높은 우선 순위:

- Aurora 특정 데이터베이스 추적 및 로깅 하위 시스템에서 사용 가능한 메모리를 낮추는 느린 메모리 누수 문제가 수정되었습니다.

일반적인 안정성 수정 사항:

- 중간 테이블을 내부적으로 사용하는 다중 테이블 조인 및 집계가 포함된 복잡한 쿼리를 실행하는 동안 발생하는 충돌 문제가 수정되었습니다.

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2019년 11월 25일(버전 2.07.0)(사용되지 않음)

버전: 2.07.0

Aurora MySQL 2.07.0이 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.07.0으로 스냅샷을 복원할 수 있습니다. 또한 기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.07.0으로 업그레이드할 수 있는 옵션도 있습니다. 기존 Aurora MySQL 1.* 클러스터를 2.07.0으로 직접 업그레이드할 수 없습니다. 하지만 이 클러스터의 스냅샷을 Aurora MySQL 0.07.0으로 복원할 수 있습니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 , AWS CLI또는 RDS API AWS Management Console를 통해 엔진 버전을 지정하십시오.

> **Note**
>
> 현재이 버전은 AWS GovCloud(미국 동부) [us-gov-east-1], AWS GovCloud(미국 서부) [us-gov-west-1], 중국(닝샤) [cn-northwest-1], 아시아 태평양(홍콩) [ap-east-1], 중동(바레인) [me-

south-1], 남아메리카(상파울루) [sa-east-1] AWS 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼 및 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

새로운 기능:

- 글로벌 데이터베이스를 사용하면 이제 다음 리전에 배포된 데이터베이스 클러스터에 보조 읽기 전용 복제본 AWS 리전을 추가할 수 있습니다. 리전: 미국 동부(버지니아 북부) [us-east-1], 미국 동부(오하이오)[us-east-2], 미국 서부(캘리포니아 북부) [us-west-1], 미국 서부(오레곤) [us-west-2], 유럽(아일랜드)[eu-west-1], 유럽(런던)[eu-west-2], 유럽(파리)[eu-west-3], 아시아 태평양(도쿄) [ap-northeast-1], 아시아 태평양(서울) [ap-northeast-2], 아시아 태평양(싱가포르) [ap-southeast-1], 아시아 태평양(시드니) [ap-southeast-2], 캐나다(중부) [ca-central-1], 유럽(프랑크푸르트)[eu-central-1], 및 아시아 태평양(뭄바이) [ap-south-1].
- Amazon Aurora 기계 학습은 Aurora MySQL 데이터베이스와 AWS 기계 학습(ML) 서비스 간의 고도로 최적화된 통합입니다. Aurora Machine Learning에서는 개발자가 사용자 지정 통합을 구축하거나 별도의 도구를 배울 필요 없이 데이터베이스 개발을 위해 이미 사용 중인 익숙한 SQL 프로그래밍 언어를 사용해 ML 모델을 호출하여 다양한 ML 기반 예측을 자체 데이터베이스 애플리케이션에 추가할 수 있게 허용합니다. 자세한 내용은 Amazon Aurora에서 기계 학습(ML) 기능 사용 단원을 참조하십시오.
- 읽기 전용 복제본의 ANSI `READ COMMITTED` 격리 수준에 대한 지원을 추가하였습니다. 이 격리 수준을 통해 읽기 전용 복제본의 장기 실행 쿼리는 라이터 노드의 높은 쓰기 처리량에 영향을 미치지 않는 상태로 실행됩니다. 자세한 내용은 Aurora MySQL 격리 수준을 참조하십시오.

심각한 수정 사항:

- CVE-2019-2922
- CVE-2019-2923
- CVE-2019-2924
- CVE-2019-2910

우선 순위가 높은 수정 사항:

- 데이터베이스 가동 중지 시간이 늘어나는 원인이 되는 DDL 복구 관련 문제를 해결하였습니다. 다중 테이블 drop 문(예: `DROP TABLE t1, t2, t3`) 실행 이후 사용할 수 없게 되는 클러스터는 이 버전으로 업데이트해야 합니다.
- 데이터베이스 가동 중지 시간이 늘어나는 원인이 되는 DDL 복구 관련 문제를 해결하였습니다. `INPLACE ALTER TABLE` DDL 문 실행 이후 사용할 수 없게 되는 클러스터는 이 버전으로 업데이트해야 합니다.

일반적인 안정성 수정 사항:

- `information_schema.replica_host_status` 테이블에 일관성 없는 데이터를 생성한 문제를 해결하였습니다.

MySQL 커뮤니티 에디션 버그 픽스 통합

- 버그 #26251621: 트리거 및 GCOL로 인한 잘못된 동작
- 버그 #22574695: ASSERTION `!TABLE || (!TABLE->READ_SET || BITMAP_IS_SET(TABLE->READ_SET, FIEL
- 버그 #25966845: 복제 키의 INSERT로 인해 교착 상태 발생
- 버그 #23070734: 동시 TRUNCATE 테이블로 인해 중단 발생
- 버그 #26191879: 외래 키 CASCADE에서 과도한 메모리 사용
- 버그 #20989615: INNODB AUTO_INCREMENT에서 동일한 값을 두 번 산출

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

Aurora MySQL 2.07.0은 MySQL 5.7과 연결성이 호환되며, JSON 지원, 공간 인덱스, 생성된 열과 같은 기능을 포함합니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

현재 Aurora MySQL 2.07.0은 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2019년 11월 22일(버전 2.06.0)(사용되지 않음)

버전: 2.06.0

Aurora MySQL 2.06.0이 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 2.01.*, 2.02.*, 2.03.*, 2.04.*, 2.05.*, 2.06.*입니다.

현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.06.0으로 스냅샷을 복원할 수 있습니다. 또한 기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.06.0으로 업그레이드할 수 있는 옵션도 있습니다. 기존 Aurora MySQL 1.* 클러스터를 2.06.0으로 직접 업그레이드할 수 없습니다. 하지만 이 클러스터의 스냅샷을 Aurora MySQL 2.06.0으로 복원할 수 있습니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 , AWS CLI또는 RDS API AWS Management Console를 통해 엔진 버전을 지정하십시오.

> **Note**
>
> 현재이 버전은 AWS GovCloud(미국 동부) [us-gov-east-1], AWS GovCloud(미국 서부) [us-gov-west-1], 중국(닝샤) [cn-northwest-1], 아시아 태평양(홍콩) [ap-east-1], 중동(바레인) [me-south-1] AWS 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

새로운 기능:

- Aurora MySQL 클러스터는 이제 db.r5.8xlarge, db.r5.16xlarge, db.r5.24xlarge 인스턴스 유형을 지원합니다. Aurora MySQL 클러스터의 인스턴스 유형에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora DB 인스턴스 클래스를 참조하세요.
- 이제 해시 조인 기능은 정식 버전이며 Aurora 랩 모드 설정이 ON 상태일 필요가 없습니다. 이 기능은 동등 조인을 사용하여 많은 양의 데이터를 조인해야 하는 경우 이 기능으로 쿼리 성능을 향상시킬 수

있습니다. 이 기능에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora Serverless에 데이터 API 사용을 참조하세요.

- 이제 핫 행 경합 기능은 정식 버전이며 Aurora 랩 모드 설정이 ON 상태일 필요가 없습니다. 이 기능은 동일한 페이지의 행에 대해 경합하는 트랜잭션이 많은 워크로드의 처리량을 크게 향상시킵니다.
- Aurora MySQL 2.06 이상에서는 백업에서 데이터를 복원하지 않고도 특정 시간으로 DB 클러스터 "되감기"하는 기능을 지원합니다. 역추적이라고 하는 이 기능을 사용하면 잘못된 테이블이나 잘못된 행을 삭제하는 등 사용자가 실수를 저지른 경우 빠르게 복구할 수 있습니다. 역추적은 대규모 데이터베이스라 할지라도 몇 초 내에 완료됩니다. AWS 블로그에서 개요를 읽고 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora DB 클러스터 역추적을 참조하세요.
- Aurora 2.06 이상은 네이티브 함수를 통한 동기 AWS Lambda 호출을 지원합니다`lambda_sync()`. 또한 비동기식 Lambda 호출의 기존 저장 프로시저의 대안으로 사용할 수 있는 네이티브 함수 `lambda_async()`도 사용할 수 있습니다. Lambda 함수 호출에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.

심각한 수정 사항:

없음.

우선 순위가 높은 수정 사항:

보안 수정

- 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

연결 처리

- DDL을 한 개 이상 실행하는 동안 클라이언트 연결 급증에 더 잘 대처하기 위해 데이터베이스 가용성을 개선하였습니다. 이러한 급증은 필요 시 추가 스레드를 일시적으로 생성하여 처리합니다. DDL 처리 중 연결 급증에 뒤이어 데이터베이스가 응답하지 않는 경우 업그레이드하는 것이 좋습니다.

엔진 다시 시작

- 엔진을 다시 시작하는 중에 사용 불가 상태가 지속되는 문제를 해결하였습니다. 이를 통해 버퍼 풀 초기화 관련 문제가 해결됩니다. 이 문제는 드물게 발생하지만 지원되는 릴리스에 영향을 미칠 가능성이 있습니다.
- 대량의 쓰기 워크로드가 실행 중일 때 바이너리 로그(binlog) 마스터로 구성된 데이터베이스가 다시 시작하는 원인이 되는 문제를 해결하였습니다.

일반적인 안정성 수정 사항:

- 캐시되지 않은 데이터에 액세스하는 쿼리가 평상시보다 더 느려질 수 있는 문제점을 개선하였습니다. 캐시되지 않은 데이터에 액세스하는 동안 알 수 없는 이유로 읽기 지연 시간이 늘어나는 고객은 이러한 문제를 겪을 때 업그레이드하는 것이 좋습니다.
- 데이터베이스 스냅샷에서 분할된 테이블을 복원하지 못하는 문제를 해결하였습니다. Aurora MySQL 1.* 데이터베이스의 스냅샷에서 복원된, 데이터베이스 내 분할된 테이블에 액세스할 때 오류가 발생하는 문제를 겪는 고객은 이 버전을 사용하는 것이 좋습니다.
- DDL 쿼리가 라이터 DB 인스턴스에서 진행 중일 때 읽기 쿼리를 처리하는 스레드와 스키마 변경 사항을 적용하는 쿼리 사이의 잠금 경합을 해결하여 Aurora 복제본의 안정성을 높였습니다.
- DDL 작업을 통해 트리거되는 `mysql.innodb_table_stats` 테이블 업데이트와 관련된 안정성 문제를 해결하였습니다.
- Aurora 복제본의 임시 테이블에 대해 중첩된 쿼리가 실행될 때 `ERROR 1836`을 잘못 보고하는 문제를 해결하였습니다.

성능 개선 사항:

- 쿼리 캐시가 binlog 작업자에서 비활성화된 경우 캐시에 대한 불필요한 API 호출을 방지함으로써 binlog 복제의 성능을 높였습니다.

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

Aurora MySQL 2.06.0은 MySQL 5.7과 연결성이 호환되며, JSON 지원, 공간 인덱스, 생성된 열과 같은 기능을 포함합니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

현재 Aurora MySQL 2.06.0은 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2019년 11월 11일(버전 2.05.0)(사용되지 않음)

버전: 2.05.0

Aurora MySQL 2.05.0이 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 2.01.*, 2.02.*, 2.03.*, 2.04.*입니다.

현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.05.0으로 스냅샷을 복원할 수 있습니다. 또한 최고 2.04.6까지, 기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.05.0으로 업그레이드할 수 있는 옵션도 있습니다. 기존 Aurora MySQL 1.* 클러스터를 2.05.0으로 직접 업그레이드할 수 없습니다. 하지만 이 클러스터의 스냅샷을 Aurora MySQL 2.05.0으로 복원할 수 있습니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 , AWS CLI또는 RDS API AWS Management Console를 통해 엔진 버전을 지정하십시오.

> **Note**
>
> 현재이 버전은 AWS GovCloud(미국 동부) [us-gov-east-1], AWS GovCloud(미국 서부) [us-gov-west-1], 중국(닝샤) [cn-northwest-1], 아시아 태평양(홍콩) [ap-east-1], 유럽(스톡홀름) [eu-north-1], 중동(바레인) [me-south-1] AWS 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

심각한 수정 사항:

- 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
- `sync_binlog` 파라미터의 값이 1로 설정되지 않은 경우 마스터의 현재 binlog 파일에 있는 이벤트가 작업자에서 복제되지 않는 문제를 수정했습니다.

우선 순위가 높은 수정 사항:

- 데이터베이스 크기가 64테비바이트(TiB)에 가까운 고객은 이 버전으로 업그레이드하여 Aurora 스토리지 한도에 가까운 볼륨에 영향을 미치는 안정성 버그로 인한 가동 중지를 방지할 것을 적극 권장합니다.
- 복제 지연 시간이 늘어나는 것을 방지하여 binlog 마스터에서 포그라운드 쿼리 성능을 향상하기 위해 `aurora_binlog_replication_max_yield_seconds` 파라미터의 기본값을 0으로 변경하였습니다.

MySQL 버그 수정 통합

- Bug#23054591: PURGE BINARY LOGS TO가 전체 binlog 파일을 읽고 있어 MySql이 지연되고 있습니다.

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- 동기식 호출 함수를 위한 기본 AWS Lambda 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

Aurora MySQL 2.05.0은 MySQL 5.7과 연결성으로 호환되며, JSON 지원, 공간 인덱스, 생성된 열과 같은 기능을 포함합니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

현재 Aurora MySQL 2.05.0은 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2020년 8월 14일(버전 2.04.9)(사용되지 않음)

버전: 2.04.9

Aurora MySQL 2.04.9가 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 , AWS CLI또는 RDS API AWS Management Console를 통해 엔진 버전을 지정하십시오.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> 현재이 버전은 AWS GovCloud(미국 동부) [us-gov-east-1], AWS GovCloud(미국 서부) [us-gov-west-1], 아시아 태평양(홍콩) [ap-east-1], 중동(바레인) [me-south-1] AWS 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

우선 순위가 높은 수정 사항:

심각한 수정 사항:

- 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

가용성 향상:

- `kill session` 명령 실행으로 인해 데이터베이스가 다시 시작되거나 장애 조치가 수행되던 문제를 해결했습니다. 이 문제가 발생하면 AWS 지원팀에 문의하여 인스턴스에서이 수정 사항을 활성화합니다.
- 중간 테이블을 내부적으로 사용하는 다중 테이블 조인 및 집계가 포함된 복잡한 쿼리를 실행하는 동안 데이터베이스가 다시 시작되는 문제를 해결했습니다.
- 다중 테이블에서 `DROP TABLE`이 중단되어 데이터베이스가 다시 시작되는 문제를 해결했습니다.
- 데이터베이스 복구 중에 데이터베이스 장애 조치가 발생하는 문제를 해결했습니다.
- 감사 및 느린 쿼리 로그가 활성화된 경우 threads_running의 잘못된 보고로 인해 데이터베이스가 다시 시작되는 문제를 해결했습니다.
- 실행 중에 `kill query` 명령이 멈출 수 있는 문제를 해결했습니다.
- 트랜잭션 롤백 중에 데이터베이스 재시작 또는 장애 조치로 이어지는 잠금 관리자의 교착 상태를 수정했습니다.
- 여러 연결이 전체 텍스트 검색 인덱스로 동일한 테이블을 업데이트하려고 하면 데이터베이스 재시작 또는 장애 조치를 트리거한 문제를 수정했습니다.
- 인덱스를 제거할 때 장애 조치 또는 다시 시작으로 인해 데드 래치가 발생할 수 있는 문제를 해결했습니다.

일반적인 개선 사항:

- 읽기 전용 복제본에 대한 쿼리가 커밋되지 않은 트랜잭션의 데이터를 사용하도록 할 수 있는 문제를 해결했습니다. 이 문제는 데이터베이스가 다시 시작된 직후에 시작되는 트랜잭션으로 제한됩니다.
- 트리거가 정의된 테이블에 대해 `INPLACE ALTER TABLE`이 진행되는 동안과 DDL에 `RENAME` 절이 포함되지 않은 경우 발생하는 문제를 해결했습니다.
- 쓰기 로드가 많은 일부 데이터베이스 클러스터에서 복제가 더 오래 걸리는 문제를 해결했습니다.
- 분할된 테이블의 이름에 공백이 포함된 경우 업그레이드하는 동안 발생하는 문제를 해결했습니다.
- 읽기 전용 복제본이 쓰기에서 최근에 커밋된 트랜잭션의 일부 결과를 일시적으로 볼 수 있는 문제를 해결했습니다.
- FTS 테이블에 대한 읽기 전용 복제본에 대한 쿼리가 기한 경과 결과를 생성할 수 있는 문제를 해결했습니다. 이 문제는 읽기 전용 복제본의 FTS 쿼리가 쓰기의 동일한 FTS 테이블에 대해 `INFORMATION_SCHEMA.INNODB_SYS_TABLES`에 대한 쿼리를 긴밀히 따르는 경우에만 발생합니다.
- 전체 텍스트 검색(FTS) 인덱스를 포함하는 Aurora 1.x 데이터베이스 클러스터를 Aurora 2.x 데이터베이스 클러스터로 복원하는 속도가 느려지는 문제를 해결했습니다.

- `server_audit_incl_users` 및 `server_audit_excl_users` 전역 파라미터에 대한 최대 허용 길이를 2000으로 확장했습니다.
- Aurora 1.x에서 Aurora 2.x로 복원이 완료되는 데 시간이 오래 걸리는 문제를 해결했습니다.
- 저장 프로시저를 통한 `lambda_async` 호출이 유니코드에서 작동하지 않는 문제를 해결했습니다.
- 공간 인덱스가 레코드 외 지오메트리 열을 제대로 처리하지 못할 때 발생하는 문제를 해결했습니다.
- 리더 DB 인스턴스에서 쿼리가 실패하고 '작업이 종료됨(내부 오류)' 메시지와 함께 `InternalFailureException` 오류가 발생할 수 있는 문제를 해결했습니다.

MySQL 버그 수정 통합

- 버그 #23070734, 버그 #80060: 동시 TRUNCATE 테이블로 인해 중단이 발생합니다.
- 버그 #23103937: PS_TRUNCATE_ALL_TABLES()이 SUPER_READ_ONLY MODE 모드에서 작동하지 않습니다.
- 버그 #22551677: 서버를 오프라인으로 전환할 때 성능 스키마 내의 경합 상태로 인해 서버가 종료될 수 있습니다.
- 버그 #27082268: 잘못된 FTS 동기화.
- BUG #12589870: 쿼리 캐시가 활성화된 경우 다중 쿼리 문으로 다시 시작되는 문제를 해결했습니다.
- 버그 #26402045: 특정 하위 쿼리 구체화 사례로 인해 서버가 종료될 수 있습니다. 이러한 쿼리는 이제 구체화가 비활성화되었음을 나타내는 오류를 생성합니다.
- 버그 #18898433: 조인 버퍼링을 사용하는 경우(예: 블록 중첩 루프 알고리즘 사용) 왼쪽 조인이 많은 쿼리가 느려졌습니다.
- 버그 #25222337: 가상 인덱스의 NULL 가상 열 필드 이름으로 인해 외래 키 제약 조건의 영향을 받는 가상 열을 채우면서 필드 이름을 비교하는 동안 서버가 종료되었습니다.(https://github.com/mysql/mysql-server/commit/273d5c9d7072c63b6c47dbef6963d7dc491d5131)
- 버그 #25053286: 보기에 액세스한 쿼리가 포함된 저장 프로시저를 실행하면 세션이 종료될 때까지 비워지지 않은 메모리를 할당할 수 있습니다.(https://github.com/mysql/mysql-server/commit/d7b37d4d141a95f577916448650c429f0d6e193d)
- 버그 #25586773: 특정 SELECT(https://dev.mysql.com/doc/refman/5.7/en/select.html) 문의 내용에서 테이블을 만든 문을 포함하는 저장 프로시저를 실행하면 메모리 누수가 발생할 수 있습니다.(https://github.com/mysql/mysql-server/commit/88301e5adab65f6750f66af284be410c4369d0c1)
- 버그 #26666274: 성능 스키마 버퍼 컨테이너의 무한 루프가 발생합니다.

- 버그 #23550835, 버그 #23298025, 버그 #81464: 내부 버퍼가 꽉 찼을 때 SELECT 성능 스키마 테이블로 인해 서버가 종료될 수 있습니다.

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- 동기식 호출 함수를 위한 기본 AWS Lambda 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

Aurora MySQL 2.04.9는 MySQL 5.7과 연결성이 호환되며, JSON 지원, 공간 인덱스, 생성된 열과 같은 기능을 포함합니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

Aurora MySQL 2.04.9는 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인

- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- CREATE TABLESPACE SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2019년 11월 20일(버전 2.04.8)(사용되지 않음)

버전: 2.04.8

Aurora MySQL 2.04.8이 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

2.* Aurora MySQL 릴리스의 스냅샷을 Aurora MySQL 2.04.8로 복원할 수 있습니다. 또한 기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.04.8로 업그레이드할 수 있는 옵션도 있습니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 , AWS CLI또는 RDS API AWS Management Console를 통해 엔진 버전을 지정하십시오.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> 현재이 버전은 AWS GovCloud(미국 동부) [us-gov-east-1], AWS GovCloud(미국 서부) [us-gov-west-1], 아시아 태평양(홍콩) [ap-east-1], 중동(바레인) [me-south-1] AWS 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

새로운 기능:

- 읽기 전용 복제본 개선 사항:
  - 데이터를 효율적으로 전송함으로써 Aurora DB 클러스터 내 라이터 인스턴스에서 리더 인스턴스로 이동하는 네트워크 트래픽을 줄였습니다. 이 개선 사항은 복제본이 속도가 느려 다시 시작되는 것을 방지하는 데 도움이 되므로 기본적으로 활성화되어 있습니다. 이 기능의 파라미터는 `aurora_enable_repl_bin_log_filtering`입니다.
  - Aurora DB 클러스터 내 라이터 인스턴스에서 리더 인스턴스로 전송되는 네트워크 트래픽을 압축을 사용해 줄였습니다. 이 개선 사항은 8xlarge 및 16xlarge 인스턴스 클래스에 대해서만 기본적으로 활성화되어 있습니다. 그 이유는 이 인스턴스가 압축을 위한 추가 CPU 오버헤드를 허용할 수 있기 때문입니다. 이 기능의 파라미터는 `aurora_enable_replica_log_compression`입니다.

우선 순위가 높은 수정 사항:

- Aurora DB 클러스터 내에 리더 인스턴스가 있는 경우 대량 워크로드 처리 중 메모리 부족으로 인해 라이터가 다시 시작하는 것을 방지하는 Aurora 라이터 인스턴스의 메모리 관리를 개선하였습니다.
- 성능 스키마 객체에 동시에 액세스하는 동안 엔진이 재시작되는 스케줄러의 비결정적 조건이 수정되었습니다.

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- 동기식 호출 함수를 위한 기본 AWS Lambda 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.

- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

Aurora MySQL 2.04.8은 MySQL 5.7과 연결성이 호환되며, JSON 지원, 공간 인덱스, 생성된 열과 같은 기능을 포함합니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

Aurora MySQL 2.04.8은 현재, 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2019년 11월 14일(버전 2.04.7)(사용되지 않음)

버전: 2.04.7

Aurora MySQL 2.04.7이 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

현재 지원되는 Aurora MySQL 릴리스에서 Aurora MySQL 2.04.7로 스냅샷을 복원할 수 있습니다. 또한 기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.04.7로 업그레이드할 수 있는

옵션도 있습니다. 기존 Aurora MySQL 1.* 클러스터를 2.04.7로 직접 업그레이드할 수 없습니다. 하지만 이 클러스터의 스냅샷을 Aurora MySQL 2.04.7로 복원할 수 있습니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 , AWS CLI또는 RDS API AWS Management Console를 통해 엔진 버전을 지정하십시오.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼 및 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> 현재이 버전은 AWS GovCloud(미국 동부) [us-gov-east-1], AWS GovCloud(미국 서부) [us-gov-west-1], 아시아 태평양(홍콩) [ap-east-1], 중동(바레인) [me-south-1] AWS 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

우선 순위가 높은 수정 사항:

연결 처리

- DDL을 한 개 이상 실행하는 동안 클라이언트 연결 급증에 더 잘 대처하기 위해 데이터베이스 가용성을 개선하였습니다. 이러한 급증은 필요 시 추가 스레드를 일시적으로 생성하여 처리합니다. DDL 처리 중 연결 급증에 뒤이어 데이터베이스가 응답하지 않는 경우 업그레이드하는 것이 좋습니다.
- `Threads_running` 전역 상태 변수의 값이 잘못되는 문제를 해결했습니다.

엔진 다시 시작

- 엔진을 다시 시작하는 중에 사용 불가 상태가 지속되는 문제를 해결하였습니다. 이를 통해 버퍼 풀 초기화 관련 문제가 해결됩니다. 이 문제는 드물게 발생하지만 지원되는 릴리스에 영향을 미칠 가능성이 있습니다.

일반적인 안정성 수정 사항:

- 캐시되지 않은 데이터에 액세스하는 쿼리가 평상시보다 더 느려질 수 있는 문제점을 개선하였습니다. 캐시되지 않은 데이터에 액세스하는 동안 알 수 없는 이유로 읽기 지연 시간이 늘어나는 고객은 이러한 문제를 겪을 때 업그레이드하는 것이 좋습니다.

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- 동기식 호출 함수를 위한 기본 AWS Lambda 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

Aurora MySQL 2.04.7은 MySQL 5.7과 연결성이 호환되며, JSON 지원, 공간 인덱스, 생성된 열과 같은 기능을 포함합니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

Aurora MySQL 2.04.7은 현재, 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인

- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2019년 9월 19일(버전 2.04.6)(사용되지 않음)

버전: 2.04.6

Aurora MySQL 2.04.6이 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.04.6으로 업그레이드할 수 있는 옵션이 있습니다. Aurora MySQL 1.* 클러스터의 현재 위치 업그레이드는 허용되지 않습니다. 이러한 제한은 나중에 Aurora MySQL 2.* 버전에서 해제될 예정입니다. Aurora MySQL 스냅샷 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 2.01.*, 2.02.*, 2.03.*, 2.04.*의 스냅샷을 Aurora MySQL 2.04.6으로 복원할 수 있습니다.

이전 버전의 Aurora MySQL을 사용하려면 AWS Management Console AWS CLI, 또는 Amazon RDS API를 통해 엔진 버전을 지정하여 새 데이터베이스 클러스터를 생성할 수 있습니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> 이 버전은 현재 유럽(런던) [eu-west-2], AWS GovCloud(미국 동부) [us-gov-east-1], AWS GovCloud(미국 서부) [us-gov-west-1], 중국(닝샤) [cn-northwest-1], 아시아 태평양(홍콩) [ap-east-1] AWS 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

- `sync_binlog` 파라미터의 값이 1로 설정되지 않은 경우 마스터의 현재 binlog 파일에 있는 이벤트가 작업자에서 복제되지 않는 문제를 수정했습니다.
- 복제 지연 시간이 늘어나는 것을 방지하여 binlog 마스터에서 포그라운드 쿼리 성능을 향상하기 위해 `aurora_binlog_replication_max_yield_seconds` 파라미터의 기본값을 0으로 변경하였습니다.

MySQL 버그 수정 통합

- Bug#23054591: PURGE BINARY LOGS TO가 전체 binlog 파일을 읽고 있어 MySql이 지연되고 있습니다.

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.

- 동기식 호출 함수를 위한 기본 AWS Lambda 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

Aurora MySQL 2.04.6은 MySQL 5.7과 연결성이 호환되며, JSON 지원, 공간 인덱스, 생성된 열과 같은 기능을 포함합니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

현재 Aurora MySQL 2.04.6은 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2019년 7월 8일(버전 2.04.5)(사용되지 않음)

버전: 2.04.5

Aurora MySQL 2.04.5가 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

기존 Aurora MySQL 2.* 데이터베이스 클러스터를 Aurora MySQL 2.04.5로 업그레이드할 수 있는 옵션이 있습니다. Aurora MySQL 1.* 클러스터의 현재 위치 업그레이드는 허용되지 않습니다. 이러한 제한은 나중에 Aurora MySQL 2.* 버전에서 해제될 예정입니다. Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 2.01.*, 2.02.*, 2.03.* 및 2.04.*의 스냅샷을 Aurora MySQL 2.04.5로 복원할 수 있습니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

보안 수정:

- CVE-2016-3518

일반 수정 사항:

- 스토리지 볼륨을 늘리는 과정에서 데이터베이스가 다시 시작되는 원인이었던 경합 조건이 수정되었습니다.
- 볼륨을 여는 과정에서 데이터베이스가 다시 시작되는 원인이었던 내부 통신 장애가 수정되었습니다.
- 분할된 테이블에서 `ALTER TABLE ALGORITHM=INPLACE`에 대한 DDL 복구 지원 기능이 추가되었습니다.
- 데이터베이스가 다시 시작되는 원인이었던 `ALTER TABLE ALGORITHM=COPY`의 DDL 복구 문제가 수정되었습니다.
- 라이터에서 삭제 워크로드가 지나치게 높을 때 Aurora 복제본 안정성이 개선되었습니다.

- 전체 텍스트 검색 인덱스 동기화를 실행하는 스레드와 사전 캐시에서 전체 텍스트 검색 테이블을 제거하는 스레드 사이의 데드 래치로 인한 데이터베이스 재시작 문제가 수정되었습니다.
- DDL 복제 과정에서 Binlog 마스터에 대한 연결이 불안정할 때 Binlog 작업자에서 발생하는 안정성 문제가 수정되었습니다.
- 전체 텍스트 검색 코드에서 데이터베이스가 다시 시작되는 원인이었던 메모리 부족 문제가 수정되었습니다.
- 64테비바이트(TiB) 볼륨을 모두 사용할 때 Aurora 라이터가 다시 시작되는 문제를 해결했습니다.
- 성능 스키마 기능에서 데이터베이스가 다시 시작되는 원인이었던 경합 조건이 수정되었습니다.
- 네트워크 프로토콜 관리의 오류를 처리할 때 중단된 연결로 인한 문제를 해결했습니다.

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- 동기식 호출 함수를 위한 기본 AWS Lambda 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

Aurora MySQL 2.04.5는 MySQL 5.7과 연결성이 호환되며, JSON 지원, 공간 인덱스, 생성된 열과 같은 기능을 포함합니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

Aurora MySQL 2.04.5는 현재, 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2019년 5월 29일(버전 2.04.4)(사용되지 않음)

버전: 2.04.4

Aurora MySQL 2.04.4가 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

새 Aurora MySQL DB 클러스터를 생성할 때(스냅샷 복원 포함) MySQL 5.7 또는 MySQL 5.6과의 호환성을 선택할 수 있습니다. Aurora MySQL 1.* 클러스터를 인플레이스 업그레이드하거나 Amazon S3 백업에서 Aurora MySQL 1.* 클러스터를 Aurora MySQL 2.04.4로 복원하는 작업은 허용되지 않습니다. 나중에 Aurora MySQL 2.* 릴리스에서 이러한 제한 사항을 제거할 계획입니다.

Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 2.01.*, 2.02.*, 2.03.*, 2.04.*의 스냅샷을 Aurora MySQL 2.04.4로 복원할 수 있습니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼 및 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> 현재이 버전은 AWS GovCloud(미국 서부) [us-gov-west-1], 유럽(스톡홀름) [eu-north-1], 중국(닝샤) [cn-northwest-1], 아시아 태평양(홍콩) [ap-east-1] AWS 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

- S3에서 Aurora로 데이터를 로드할 때 오류를 일으킬 수 있는 문제가 해결되었습니다.
- Aurora에서 S3로 데이터를 업로드할 때 오류를 일으킬 수 있는 문제가 해결되었습니다.
- 네트워크 프로토콜 관리의 오류를 처리할 때 중단된 연결로 인한 문제를 해결했습니다.
- 분할된 테이블을 처리할 때 충돌을 일으킬 수 있는 문제가 해결되었습니다.
- 일부 리전에서 성능 개선 도우미 기능을 사용할 수 없는 문제가 해결되었습니다.

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- 동기식 호출 함수를 위한 기본 AWS Lambda 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.

- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

Aurora MySQL 2.04.4는 MySQL 5.7과 연결성이 호환되며, JSON 지원, 공간 인덱스, 생성된 열과 같은 기능을 포함합니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

Aurora MySQL 2.04.4는 현재, 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2019년 5월 9일(버전 2.04.3)(사용되지 않음)

버전: 2.04.3

Aurora MySQL 2.04.3이 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

새 Aurora MySQL DB 클러스터를 생성할 때(스냅샷 복원 포함) MySQL 5.7 또는 MySQL 5.6과의 호환성을 선택할 수 있습니다. Aurora MySQL 1.* 클러스터를 인플레이스 업그레이드하거나 Amazon S3

백업에서 Aurora MySQL 1.* 클러스터를 Aurora MySQL 2.04.3으로 복원하는 작업은 허용되지 않습니다. 나중에 Aurora MySQL 2.* 릴리스에서 이러한 제한 사항을 제거할 계획입니다.

Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 2.01.*, 2.02.*, 2.03.*, 2.04.*의 스냅샷을 Aurora MySQL 2.04.3으로 복원할 수 있습니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼 및 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> 이 버전은 현재 AWS GovCloud(미국 서부) [us-gov-west-1] 및 중국(닝샤) [cn-northwest-1] AWS 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

- binlog 작업자로 구성된 Aurora 인스턴스에서 문제를 일으킬 수 있는 binlog 복제의 버그를 수정했습니다.
- 대규모 저장 루틴 처리 시 발생하는 메모리 부족 상태를 수정했습니다.
- 특정 유형의 `ALTER TABLE` 명령을 처리할 때 발생하는 오류를 수정했습니다.
- 네트워크 프로토콜 관리 시 발생한 오류로 인해 중단된 연결과 관련된 문제를 해결했습니다.

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.

- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- 동기식 호출 함수를 위한 기본 AWS Lambda 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

Aurora MySQL 2.04.3은 MySQL 5.7과 연결성이 호환되며, JSON 지원, 공간 인덱스, 생성된 열과 같은 기능을 포함합니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

Aurora MySQL 2.04.3은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2019년 5월 2일(버전 2.04.2)(사용되지 않음)

버전: 2.04.2

Aurora MySQL 2.04.2가 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

새 Aurora MySQL DB 클러스터를 생성할 때(스냅샷 복원 포함) MySQL 5.7 또는 MySQL 5.6과의 호환성을 선택할 수 있습니다. Aurora MySQL 1.* 클러스터를 인플레이스 업그레이드하거나 Amazon S3 백업에서 Aurora MySQL 1.* 클러스터를 Aurora MySQL 2.04.2로 복원하는 작업은 허용되지 않습니다. 나중에 Aurora MySQL 2.* 릴리스에서 이러한 제한 사항을 제거할 계획입니다.

Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 2.01.*, 2.02.*, 2.03.*, 2.04.0, 2.04.1의 스냅샷을 Aurora MySQL 2.04.2로 복원할 수 있습니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼 및 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> 이 버전은 현재 AWS GovCloud(미국 서부) [us-gov-west-1] 및 중국(닝샤) [cn-northwest-1] AWS 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

> **Note**
>
> Aurora MySQL 데이터베이스 클러스터를 업그레이드하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

- 사용자 지정 인증서를 사용하는 SSL binlog 복제에 대한 지원을 추가하였습니다. Aurora MySQL에서 SSL binlog 복제를 사용하는 방법에 관한 자세한 내용은 mysql_rds_import_binlog_ssl_material 단원을 참조하십시오.
- 전체 텍스트 검색 인덱스가 있는 테이블이 최적화되는 중에 Aurora 기본 인스턴스에서 발생하는 데드 래치를 수정했습니다.

- SELECT(*)를 사용하는 특정 쿼리의 성능이 보조 인덱스가 있는 테이블에서 영향을 받을 수 있는 Aurora 복제본의 문제를 수정했습니다.
- 오류 1032가 게시되게 한 조건을 수정했습니다.
- 여러 가지 데드 래치를 수정하여 Aurora 복제본의 안정성을 개선하였습니다.

MySQL 버그 수정 통합

- 버그 #24829050 - INDEX_MERGE_INTERSECTION 최적화로 인해 잘못된 쿼리 결과 산출

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- 동기식 호출 함수를 위한 기본 AWS Lambda 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

Aurora MySQL 2.04.2는 MySQL 5.7과 연결성이 호환되며, JSON 지원, 공간 인덱스, 생성된 열과 같은 기능을 포함합니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

Aurora MySQL 2.04.2는 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인

- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2019년 3월 25일(버전 2.04.1)(사용되지 않음)

버전: 2.04.1

Aurora MySQL 2.04.1이 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

새 Aurora MySQL DB 클러스터를 생성할 때(스냅샷 복원 포함) MySQL 5.7 또는 MySQL 5.6과의 호환성을 선택할 수 있습니다. Aurora MySQL 1.* 클러스터를 인플레이스 업그레이드하거나 Amazon S3 백업에서 Aurora MySQL 1.* 클러스터를 Aurora MySQL 2.04.1로 복원하는 작업은 허용되지 않습니다. 나중에 Aurora MySQL 2.* 릴리스에서 이러한 제한 사항을 제거할 계획입니다.

Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 2.01.*, 2.02.*, 2.03.*, 2.04.0의 스냅샷을 Aurora MySQL 2.04.1로 복원할 수 있습니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼 및 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> 이 버전은 현재 AWS GovCloud(미국 서부) [us-gov-west-1] 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

> **Note**
>
> DB 클러스터를 업그레이드하는 절차가 변경되었습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

- 1.16보다 낮은 버전에 대한 Aurora MySQL 5.6 스냅샷을 최신 Aurora MySQL 5.7 클러스터로 복원할 수 없는 문제를 수정했습니다.

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- 동기식 호출 함수를 위한 기본 AWS Lambda 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2019년 3월 25일(버전 2.04.0)(사용되지 않음)

버전: 2.04

Aurora MySQL 2.04가 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

새 Aurora MySQL DB 클러스터를 생성할 때(스냅샷 복원 포함) MySQL 5.7 또는 MySQL 5.6과의 호환성을 선택할 수 있습니다. Aurora MySQL 1.* 클러스터를 인플레이스 업그레이드하거나 Amazon S3 백업에서 Aurora MySQL 1.* 클러스터를 Aurora MySQL 2.04.0으로 복원하는 작업은 허용되지 않습니다. 나중에 Aurora MySQL 2.* 릴리스에서 이러한 제한 사항을 제거할 계획입니다.

Aurora MySQL 1.19.*, 2.01.*, 2.02.*, 2.03.*의 스냅샷을 Aurora MySQL 2.04.0으로 복원할 수 있습니다. Aurora MySQL 1.14.* 이하, 1.15.*, 1.16.*, 1.17.*, 1.18.*의 스냅샷을 Aurora MySQL 2.04.0으로 복원할 수 없습니다. 이 제한 사항은 Aurora MySQL 2.04.1에서는 제거되었습니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> 이 버전은 현재 AWS GovCloud(미국 서부) [us-gov-west-1] 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

> **Note**
>
> DB 클러스터를 업그레이드하는 절차가 변경되었습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

- GTID 기반 복제를 지원합니다. Aurora MySQL에서 GTID 기반 복제를 사용하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL에 대한 GTID 기반 복제 사용을 참조하세요.
- 임시 테이블에 있는 행을 삭제 또는 업데이트하는 문에 InnoDB 하위 쿼리가 포함된 경우 Aurora 복제본에서 `Running in read-only mode` 오류를 잘못 표시하는 문제를 수정했습니다.

MySQL 버그 수정 통합

- 버그 #26225783: 생성 테이블에서 MYSQL 충돌(반복될 수 있음) -> INNODB: 세마포어 대기에 따름.

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- 동기식 호출 함수를 위한 기본 AWS Lambda 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

이 Aurora MySQL 버전은 MySQL 5.7과 연결성이 호환되며 JSON 지원, 공간 인덱스 및 생성된 열과 같은 기능이 포함되어 있습니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

이 Aurora MySQL 버전은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문

Aurora MySQL 데이터베이스 엔진 업데이트 2019년 2월 7일(버전 2.03.4)(사용되지 않음)

(버전 2.03.4)

Aurora MySQL 2.03.4가 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

새 Aurora MySQL DB 클러스터(스냅샷 복원 포함)를 생성할 때 MySQL 5.7 또는 MySQL 5.6과의 호환성을 선택할 수 있습니다.

Aurora MySQL 1.* 클러스터를 Aurora MySQL 2.03.4로 인플레이스 업그레이드하거나 Amazon S3 백업에서 Aurora MySQL 2.03.4로 복원하는 작업은 허용되지 않습니다. 나중에 Aurora MySQL 2.* 릴리스에서 이러한 제한 사항을 제거할 계획입니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> 현재이 버전은 AWS GovCloud(미국 서부) [us-gov-west-1] 및 중국(베이징) [cn-north-1] 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

> **Note**
>
> DB 클러스터를 업그레이드하는 절차가 변경되었습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

- UTF8MB4 Unicode 9.0 액센트 구분 및 대소문자 구분 콜레이션 지원, `utf8mb4_0900_as_ci`.

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- 동기식 호출 함수를 위한 기본 AWS Lambda 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

Aurora MySQL 2.03.4는 MySQL 5.7과 연결성이 호환되며, JSON 지원, 공간 인덱스, 생성된 열과 같은 기능을 포함합니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

Aurora MySQL 2.03.4는 현재, 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 글로벌 트랜잭션 식별자(GTID) Aurora MySQL은 버전 2.04 이상에서 GTID를 지원합니다.
- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링

- CREATE TABLESPACE SQL 문
- X 프로토콜

Aurora MySQL 데이터베이스 엔진 업데이트 2019년 1월 18일(버전 2.03.3)(사용되지 않음)

(버전 2.03.3)

Aurora MySQL 2.03.3이 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

새 Aurora MySQL DB 클러스터(스냅샷 복원 포함)를 생성할 때 MySQL 5.7 또는 MySQL 5.6과의 호환성을 선택할 수 있습니다.

Aurora MySQL 1.* 클러스터를 Aurora MySQL 2.03.3으로 인플레이스 업그레이드하거나 Amazon S3 백업에서 Aurora MySQL 2.03.3으로 복원하는 작업은 허용되지 않습니다. 나중에 Aurora MySQL 2.* 릴리스에서 이러한 제한 사항을 제거할 계획입니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> 현재이 버전은 AWS GovCloud(미국 서부) [us-gov-west-1] 및 중국(베이징) [cn-north-1] 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

> **Note**
>
> DB 클러스터를 업그레이드하는 절차가 변경되었습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

CVE 수정 사항

- CVE-2016-5436

심각한 수정 사항:

- Aurora 복제본이 인덱스에서 역방향 스캔을 실행할 때 데드 래치 상태가 될 수 있는 문제를 해결했습니다.
- Aurora 기본 인스턴스가 분할된 테이블에서 현재 위치 DDL 작업을 실행할 때 Aurora 복제본이 다시 시작되는 문제를 해결했습니다.
- Aurora 기본 인스턴스에 대한 DDL 작업 후 쿼리 캐시 무효화 중에 Aurora 복제본이 다시 시작될 수 있는 문제를 해결했습니다.
- Aurora 기본 인스턴스가 테이블에서 잘림을 실행하는 동안 이 테이블에서 `SELECT` 쿼리 중에 Aurora 복제본이 다시 시작될 수 있는 문제를 해결했습니다.
- 인덱싱된 열만 액세스할 수 있는 MyISAM 임시 테이블에서 잘못된 결과가 나오는 문제를 해결했습니다.
- 약 40,000건의 쿼리를 수행한 후 주기적으로 `query_time` 및 `lock_time`에 대해 잘못된 대규모의 값을 생성하는 느린 로그의 문제를 해결했습니다.
- 'tmp'라는 스키마로 인해 RDS for MySQL에서 Aurora MySQL로의 마이그레이션이 정체될 수 있는 문제를 해결했습니다.
- 로그 순환 중에 감사 로그에 이벤트가 누락되는 문제를 해결했습니다.
- 랩 모드의 Fast DDL 기능이 활성화될 때 Aurora 5.6 스냅샷에서 복원된 Aurora 기본 인스턴스가 다시 시작되는 문제를 수정했습니다.
- 사전 통계 스레드로 인해 CPU 사용량이 100%가 되는 문제를 해결했습니다.
- `CHECK TABLE` 문 실행 중에 Aurora 복제본이 다시 시작될 수 있는 문제를 해결했습니다.

MySQL 버그 수정 통합

- 버그 #25361251: SP에 있는 복제 키의 INSERT로 인한 오작동
- 버그 #26734162: BLOB INSERT + 복제 키 업데이트로 인한 오작동
- 버그 #27460607: INSERT SELECT의 원본 테이블이 비어 있을 때 IODKU 오작동 발생
- `DISTINCT` 또는 `GROUP BY` 절을 사용하는 쿼리는 잘못된 결과를 반환할 수 있습니다. (MySQL 5.7 버그 #79591, 버그 #22343910)
- 오류 1093으로 인해 `DELETE` 절의 파생된 테이블 사용을 사용하는 조인 테이블의 `WHERE` 실패 (버그 #23074801).

- GCOLS: 문자 집합 변경으로 인한 오작동(버그 #25287633).

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- 동기식 호출 함수를 위한 기본 AWS Lambda 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

Aurora MySQL 2.03.3은 MySQL 5.7과 연결성이 호환되며, JSON 지원, 공간 인덱스, 생성된 열과 같은 기능을 포함합니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

Aurora MySQL 2.03.3은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 글로벌 트랜잭션 식별자(GTID) Aurora MySQL은 버전 2.04 이상에서 GTID를 지원합니다.
- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정

- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- CREATE TABLESPACE SQL 문
- X 프로토콜

Aurora MySQL 데이터베이스 엔진 업데이트 2019년 1월 9일(버전 2.03.2)(사용되지 않음)

(버전 2.03.2)

Aurora MySQL 2.03.2가 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

새 Aurora MySQL DB 클러스터(스냅샷 복원 포함)를 생성할 때 MySQL 5.7 또는 MySQL 5.6과의 호환성을 선택할 수 있습니다.

Aurora MySQL 1.* 클러스터를 Aurora MySQL 2.03.2로 인플레이스 업그레이드하거나, Amazon S3 백업에서 Aurora MySQL 2.03.2로 복원하는 작업은 허용되지 않습니다. 나중에 Aurora MySQL 2.* 릴리스에서 이러한 제한 사항을 제거할 계획입니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> 현재이 버전은 AWS GovCloud(미국 서부) [us-gov-west-1] 및 중국(베이징) [cn-north-1] 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

> **Note**
>
> DB 클러스터를 업그레이드하는 절차가 변경되었습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

- Aurora 버전 선택기 – Aurora MySQL 2.03.2부터는 AWS Management Console에서 MySQL 5.7 호환 Aurora의 여러 버전 간에 선택할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 AWS를 통해 Aurora MySQL 엔진 버전 확인 또는 지정을 참조하세요.

심각한 수정 사항:

- CVE-2016-3495

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- 동기식 호출 함수를 위한 기본 AWS Lambda 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

Aurora MySQL 2.03.2는 MySQL 5.7과 연결성이 호환되며, JSON 지원, 공간 인덱스, 생성된 열과 같은 기능을 포함합니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

Aurora MySQL 2.03.2는 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 글로벌 트랜잭션 식별자(GTID) Aurora MySQL은 버전 2.04 이상에서 GTID를 지원합니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- CREATE TABLESPACE SQL 문
- X 프로토콜

Aurora MySQL 데이터베이스 엔진 업데이트 2018년 10월 24일(버전 2.03.1)(사용되지 않음)

(버전 2.03.1)

Aurora MySQL 2.03.1이 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

스냅샷에서 복원한 클러스터를 포함하여 새로운 Aurora MySQL DB 클러스터를 생성할 때는 MySQL 5.7 또는 MySQL 5.6과의 호환성을 선택할 수 있습니다. MySQL 5.6 호환 스냅샷을 복원할 때, MySQL 5.7 또는 MySQL 5.6과의 호환성을 선택할 수 있습니다.

Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 2.01.*, 2.02.*, 2.03의 스냅샷을 Aurora MySQL 2.03.1로 복원할 수 있습니다.

Aurora MySQL 1.* 클러스터를 Aurora MySQL 2.03.1로 인플레이스 업그레이드하거나, Amazon S3 백업에서 Aurora MySQL 2.03.1로 복원하는 작업은 허용되지 않습니다. 나중에 Aurora MySQL 2.* 릴리스에서 이러한 제한 사항을 제거할 계획입니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> 현재이 버전은 AWS GovCloud(미국 서부) [us-gov-west-1] 및 중국(베이징) [cn-north-1] 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

개선 사항

- 트랜잭션 교착(deadlock) 감지 실행 시 Aurora Writer가 다시 시작될 수 있는 문제를 수정했습니다.

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- 동기식 호출 함수를 위한 기본 AWS Lambda 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

Aurora MySQL 2.03.1은 MySQL 5.7과 연결성이 호환되며, JSON 지원, 공간 인덱스, 생성된 열과 같은 기능을 포함합니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

Aurora MySQL 2.03.1은 현재, 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 글로벌 트랜잭션 식별자(GTID) Aurora MySQL은 버전 2.04 이상에서 GTID를 지원합니다.

- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- CREATE TABLESPACE SQL 문
- X 프로토콜

Aurora MySQL 데이터베이스 엔진 업데이트 2018년 10월 11일(버전 2.03)(사용되지 않음)

(버전 2.03)

Aurora MySQL 2.03이 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

스냅샷에서 복원한 클러스터를 포함하여 새로운 Aurora MySQL DB 클러스터를 생성할 때는 MySQL 5.7 또는 MySQL 5.6과의 호환성을 선택할 수 있습니다. MySQL 5.6 호환 스냅샷을 복원할 때, MySQL 5.7 또는 MySQL 5.6과의 호환성을 선택할 수 있습니다.

Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 2.01.*, 2.02.*의 스냅샷을 Aurora MySQL 2.03으로 복원할 수 있습니다.

Aurora MySQL 1.* 클러스터를 Aurora MySQL 2.03으로 인플레이스 업그레이드하거나, Amazon S3 백업에서 Aurora MySQL 2.03으로 복원하는 작업은 허용되지 않습니다. 나중에 Aurora MySQL 2.* 릴리스에서 이러한 제한 사항을 제거할 계획입니다.

> **Note**
>
> 현재이 버전은 AWS GovCloud(미국 서부) [us-gov-west-1] 및 중국(베이징) [cn-north-1] 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

- 성능 스키마를 사용할 수 있습니다.
- 종료됨(killed) 상태의 좀비 세션이 CPU를 더 사용할 수 있는 문제를 수정했습니다.
- 읽기 전용 트랜잭션이 Aurora Writer의 레코드에 대한 잠금을 획득한 경우 발생하는 데드 래치 문제를 수정했습니다.
- 고객 워크로드가 없는 Aurora 복제본의 CPU 사용률이 높을 수 있는 문제를 수정했습니다.
- Aurora 복제본 또는 Aurora Writer를 다시 시작하게 만들 수 있는 여러 문제를 수정했습니다.
- 디스크 처리량 한도에 도달한 경우 진단 로깅을 건너뛰는 기능을 추가했습니다.
- Aurora Writer에서 binlog가 활성화된 경우 발생하는 메모리 누수 문제를 수정했습니다.

MySQL 커뮤니티 에디션 버그 픽스 통합

- REVERSE SCAN ON A PARTITIONED TABLE DOES ICP - ORDER BY DESC(버그 #24929748).
- JSON_OBJECT CREATES INVALID JSON CODE(버그 #26867509).
- INSERTING LARGE JSON DATA TAKES AN INORDINATE AMOUNT OF TIME(버그 #22843444).
- PARTITIONED TABLES USE MORE MEMORY IN 5.7 THAN 5.6(버그 #25080442).

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- 동기식 호출 함수를 위한 기본 AWS Lambda 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.

- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

Aurora MySQL 2.03은 MySQL 5.7과 연결성이 호환되며, JSON 지원, 공간 인덱스, 생성된 열과 같은 기능을 포함합니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

Aurora MySQL 2.03은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 글로벌 트랜잭션 식별자(GTID) Aurora MySQL은 버전 2.04 이상에서 GTID를 지원합니다.
- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문
- X 프로토콜

Aurora MySQL 데이터베이스 엔진 업데이트 2018년 10월 8일(버전 2.02.5)(사용되지 않음)

(버전 2.02.5)

Aurora MySQL 2.02.5가 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

스냅샷에서 복원한 클러스터를 포함하여 새로운 Aurora MySQL DB 클러스터를 생성할 때는 MySQL 5.7 또는 MySQL 5.6과의 호환성을 선택할 수 있습니다. MySQL 5.6 호환 스냅샷을 복원할 때, MySQL 5.7 또는 MySQL 5.6과의 호환성을 선택할 수 있습니다.

Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 2.01.*, 2.02.*의 스냅샷을 Aurora MySQL 2.02.5로 복원할 수 있습니다. 또한 Aurora MySQL 2.01.* 또는 2.02.*에서 Aurora MySQL 2.02.5로 인플레이스 업그레이드를 수행할 수 있습니다.

Aurora MySQL 1.* 클러스터를 Aurora MySQL 2.02.5로 인플레이스 업그레이드하거나, Amazon S3 백업에서 Aurora MySQL 2.02.5로 복원하는 작업은 허용되지 않습니다. 나중에 Aurora MySQL 2.* 릴리스에서 이러한 제한 사항을 제거할 계획입니다.

Aurora MySQL 5.7의 이 릴리스에서는 성능 스키마가 비활성화됩니다. 성능 스키마를 지원하려면 Aurora 2.03으로 업그레이드하십시오.

> **Note**
>
> 현재이 버전은 AWS GovCloud(미국 서부) [us-gov-west-1] 및 중국(베이징) [cn-north-1] 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

- Aurora 복제본이 테이블에서 역방향 스캔을 수행할 때 다시 시작되는 문제를 수정했습니다.

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.

- 동기식 호출 함수를 위한 기본 AWS Lambda 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

Aurora MySQL 2.02.5는 MySQL 5.7과 연결성이 호환되며, JSON 지원, 공간 인덱스, 생성된 열과 같은 기능을 포함합니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

Aurora MySQL 2.02.5는 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 글로벌 트랜잭션 식별자(GTID) Aurora MySQL은 버전 2.04 이상에서 GTID를 지원합니다.
- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문
- X 프로토콜

Aurora MySQL 데이터베이스 엔진 업데이트 2018년 9월 21일(버전 2.02.4)(사용되지 않음)

(버전 2.02.4)

Aurora MySQL 2.02.4가 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

스냅샷에서 복원한 클러스터를 포함하여 새로운 Aurora MySQL DB 클러스터를 생성할 때는 MySQL 5.7 또는 MySQL 5.6과의 호환성을 선택할 수 있습니다. MySQL 5.6 호환 스냅샷을 복원할 때, MySQL 5.7 또는 MySQL 5.6과의 호환성을 선택할 수 있습니다.

Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 2.01.*, 2.02.*의 스냅샷을 Aurora MySQL 2.02.4로 복원할 수 있습니다. 또한 Aurora MySQL 2.01.* 또는 2.02.*에서 Aurora MySQL 2.02.4로 인플레이스 업그레이드를 수행할 수 있습니다.

Aurora MySQL 1.* 클러스터를 Aurora MySQL 2.02.4로 인플레이스 업그레이드하거나, Amazon S3 백업에서 Aurora MySQL 2.02.4로 복원하는 작업은 허용되지 않습니다. 나중에 Aurora MySQL 2.* 릴리스에서 이러한 제한 사항을 제거할 계획입니다.

Aurora MySQL 5.7의 이 릴리스에서는 성능 스키마가 비활성화됩니다. 성능 스키마를 지원하려면 Aurora 2.03으로 업그레이드하십시오.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

- Aurora MySQL 5.6 스냅샷에서 복원된 테이블에서 전체 텍스트 검색 인덱스와 관련된 안정성 문제를 수정했습니다.

MySQL 커뮤니티 에디션 버그 픽스 통합

- 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 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- 동기식 호출 함수를 위한 기본 AWS Lambda 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

MySQL 5.7 호환성

Aurora MySQL 2.02.4는 MySQL 5.7과 연결성이 호환되며, JSON 지원, 공간 인덱스, 생성된 열과 같은 기능을 포함합니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

Aurora MySQL 2.02.4는 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 글로벌 트랜잭션 식별자(GTID) Aurora MySQL은 버전 2.04 이상에서 GTID를 지원합니다.
- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문
- X 프로토콜

Aurora MySQL 데이터베이스 엔진 업데이트 2018년 8월 23일(버전 2.02.3)(사용되지 않음)

(버전 2.02.3)

Aurora MySQL 2.02.3이 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

스냅샷에서 복원한 클러스터를 포함하여 새로운 Aurora MySQL DB 클러스터를 생성할 때는 MySQL 5.7 또는 MySQL 5.6과의 호환성을 선택할 수 있습니다. MySQL 5.6 호환 스냅샷을 복원할 때, MySQL 5.7 또는 MySQL 5.6과의 호환성을 선택할 수 있습니다.

Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 2.01.*, 2.02.*의 스냅샷을 Aurora MySQL 2.02.3으로 복원할 수 있습니다. 또한 Aurora MySQL 2.01.* 또는 2.02.*에서 Aurora MySQL 2.02.3으로 인플레이스 업그레이드를 수행할 수 있습니다.

Aurora MySQL 1.* 클러스터를 Aurora MySQL 2.02.3으로 인플레이스 업그레이드하거나, Amazon S3 백업에서 Aurora MySQL 2.02.3으로 복원하는 작업은 허용되지 않습니다. 나중에 Aurora MySQL 2.* 릴리스에서 이러한 제한 사항을 제거할 계획입니다.

Aurora MySQL 5.7의 이 릴리스에서는 성능 스키마가 비활성화됩니다. 성능 스키마를 지원하려면 Aurora 2.03으로 업그레이드하십시오.

> **Note**
>
> 현재이 버전은 AWS GovCloud(미국 서부) [us-gov-west-1] 및 중국(베이징) [cn-north-1] 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

Aurora MySQL 버전 1과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 버전 1(MySQL 5.6과 호환됨)에서 지원되지만 현재 이 기능은 Aurora MySQL 버전 2(MySQL 5.7과 호환됨)에서는 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- 동기식 호출 함수를 위한 기본 AWS Lambda 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

현재, Aurora MySQL 2.01은 Aurora MySQL 버전 1.16 이상에서 추가된 기능을 지원하지 않습니다. Aurora MySQL 버전 1.16에 대한 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 단원을 참조하십시오.

MySQL 5.7 호환성

Aurora MySQL 2.02.3은 MySQL 5.7과 연결성이 호환되며, JSON 지원, 공간 인덱스, 생성된 열과 같은 기능을 포함합니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

Aurora MySQL 2.02.3은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 글로벌 트랜잭션 식별자(GTID) Aurora MySQL은 버전 2.04 이상에서 GTID를 지원합니다.
- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문
- X 프로토콜

Aurora MySQL 2.x와 Aurora MySQL 1.x의 CLI 차이점

- Aurora MySQL 2.x의 엔진 이름은 `aurora-mysql`이고 Aurora MySQL 1.x의 엔진 이름은 계속해서 `aurora`입니다.
- Aurora MySQL 2.x의 기본 파라미터 그룹은 `default.aurora-mysql5.7`이고 Aurora MySQL 1.x의 기본 파라미터 그룹은 계속해서 `default.aurora5.6`입니다.
- Aurora MySQL 2.x의 DB 클러스터 파라미터 그룹 패밀리 이름은 `aurora-mysql5.7`이고 Aurora MySQL 1.x의 DB 클러스터 파라미터 그룹 패밀리 이름은 계속해서 `aurora5.6`입니다.

전체 CLI 명령 세트 및 Aurora MySQL 2.x와 Aurora MySQL 1.x의 차이점은 Aurora 설명서를 참조하세요.

개선 사항

- 레코드를 읽는 동안 낙관적 커서 사용이 복원될 때 Aurora 복제본이 다시 시작할 수 있는 문제를 수정했습니다.
- 인덱스 통계를 향상하기 위해 `innodb_stats_persistent_sample_pages` 파라미터 기본값을 128로 업데이트했습니다.
- Aurora 복제본이 Aurora 기본 인스턴스에서 동시 수정되는 작은 테이블에 액세스할 때 다시 시작될 수 있는 문제를 수정했습니다.
- 테이블 정의 캐시 비우기를 중지하기 위해 `ANALYZE TABLE`을 수정했습니다.
- 지역에 대한 포인트 쿼리를 검색 범위로 변환할 때 Aurora 기본 인스턴스 또는 Aurora 복제본이 다시 시작될 수 있는 문제를 수정했습니다.

Aurora MySQL 데이터베이스 엔진 업데이트 2018년 6월 4일(버전 2.02.2)(사용되지 않음)

(버전 2.02.2)

Aurora MySQL 2.02.2이 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

스냅샷에서 복원한 클러스터를 포함하여 새로운 Aurora MySQL DB 클러스터를 생성할 때는 MySQL 5.7 또는 MySQL 5.6과의 호환성을 선택할 수 있습니다. MySQL 5.6 호환 스냅샷을 복원할 때, MySQL 5.7 또는 MySQL 5.6과의 호환성을 선택할 수 있습니다.

Aurora MySQL 1.14*, 1.15*, 1.16*, 1.17*, 2.01*, 2.02의 스냅샷을 Aurora MySQL 2.02.2로 복원할 수 있습니다. 또한 Aurora MySQL 2.01* 또는 2.02에서 Aurora MySQL 2.02.2로 인플레이스 업그레이드를 수행할 수 있습니다.

Aurora MySQL 1.* 클러스터를 Aurora MySQL 2.02.2로 인플레이스 업그레이드하거나, Amazon S3 백업에서 Aurora MySQL 2.02.2로 복원하는 작업은 허용되지 않습니다. 나중에 Aurora MySQL 2.* 릴리스에서 이러한 제한 사항을 제거할 계획입니다.

Aurora MySQL 5.7의 이 릴리스에서는 성능 스키마가 비활성화됩니다. 성능 스키마를 지원하려면 Aurora 2.03으로 업그레이드하십시오.

> **Note**
>
> 현재이 버전은 AWS GovCloud(미국 서부) [us-gov-west-1] 및 중국(베이징) [cn-north-1] 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

심각한 수정 사항:

- CVE-2016-3486

Aurora MySQL 5.6과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 5.6에서 지원되지만, 이러한 기능은 Aurora MySQL 5.7에서 현재 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- 동기식 호출 함수를 위한 기본 AWS Lambda 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

현재, Aurora MySQL 2.01은 Aurora MySQL 버전 1.16 이상에서 추가된 기능을 지원하지 않습니다. Aurora MySQL 버전 1.16에 대한 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 단원을 참조하십시오.

MySQL 5.7 호환성

Aurora MySQL 2.02.2는 MySQL 5.7과 연결성이 호환되며, JSON 지원, 공간 인덱스, 생성된 열과 같은 기능을 포함합니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

Aurora MySQL 2.02.2는 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 글로벌 트랜잭션 식별자(GTID) Aurora MySQL은 버전 2.04 이상에서 GTID를 지원합니다.
- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문
- X 프로토콜

Aurora MySQL 2.x와 Aurora MySQL 1.x의 CLI 차이점

- Aurora MySQL 2.x의 엔진 이름은 `aurora-mysql`이고 Aurora MySQL 1.x의 엔진 이름은 계속해서 `aurora`입니다.
- Aurora MySQL 2.x의 기본 파라미터 그룹은 `default.aurora-mysql5.7`이고 Aurora MySQL 1.x의 기본 파라미터 그룹은 계속해서 `default.aurora5.6`입니다.
- Aurora MySQL 2.x의 DB 클러스터 파라미터 그룹 패밀리 이름은 `aurora-mysql5.7`이고 Aurora MySQL 1.x의 DB 클러스터 파라미터 그룹 패밀리 이름은 계속해서 `aurora5.6`입니다.

전체 CLI 명령 세트 및 Aurora MySQL 2.x와 Aurora MySQL 1.x의 차이점은 Aurora 설명서를 참조하세요.

개선 사항

- Aurora 복제 진행 상황을 추적할 때 Aurora Writer가 가끔 다시 시작할 수 있는 문제를 수정했습니다.
- Aurora Writer의 테이블에서 인덱스 생성 또는 삭제 문을 실행한 후 파티셔닝된 테이블에 액세스할 때 Aurora 복제본이 다시 시작하거나 오류를 발생시키는 문제를 수정했습니다.
- Aurora Writer에서 ALTER 테이블 ADD/DROP 열 문을 실행하여 발생하는 변경 내용을 적용하는 동안 Aurora 복제본의 테이블에 액세스할 수 없는 문제를 수정했습니다.

Aurora MySQL 데이터베이스 엔진 업데이트 2018년 5월 3일(버전 2.02)(사용되지 않음)

(버전 2.02)

Aurora MySQL 2.02가 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

스냅샷에서 복원한 클러스터를 포함하여 새로운 Aurora MySQL DB 클러스터를 생성할 때는 MySQL 5.7 또는 MySQL 5.6과의 호환성을 선택할 수 있습니다. MySQL 5.6 호환 스냅샷을 복원할 때, MySQL 5.7 또는 MySQL 5.6과의 호환성을 선택할 수 있습니다.

Aurora MySQL 1.14*, 1.15*, 1.16*, 1.17*, 2.01*의 스냅샷을 Aurora MySQL 2.02로 복원할 수 있습니다. 또한 Aurora MySQL 2.01*에서 Aurora MySQL 2.02로 인플레이스 업그레이드를 수행할 수 있습니다.

Aurora MySQL 1.x 클러스터를 Aurora MySQL 2.02로 인플레이스 업그레이드하거나, Amazon S3 백업에서 Aurora MySQL 2.02로 복원하는 작업은 허용되지 않습니다. 향후 Aurora MySQL 2.x 릴리스에서는 이러한 제한 사항을 없앨 계획입니다.

Aurora MySQL 5.7의 이 릴리스에서는 성능 스키마가 비활성화됩니다. 성능 스키마를 지원하려면 Aurora 2.03으로 업그레이드하십시오.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

Aurora MySQL 5.6과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 5.6에서 지원되지만, 이러한 기능은 Aurora MySQL 5.7에서 현재 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- 동기식 호출 함수를 위한 기본 AWS Lambda 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

현재, Aurora MySQL 2.01은 Aurora MySQL 버전 1.16 이상에서 추가된 기능을 지원하지 않습니다. Aurora MySQL 버전 1.16에 대한 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 단원을 참조하십시오.

MySQL 5.7 호환성

Aurora MySQL 2.02는 MySQL 5.7과 연결성이 호환되며, JSON 지원, 공간 인덱스, 생성된 열과 같은 기능을 포함합니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

Aurora MySQL 2.02는 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 글로벌 트랜잭션 식별자(GTID) Aurora MySQL은 버전 2.04 이상에서 GTID를 지원합니다.
- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링

- CREATE TABLESPACE SQL 문
- X 프로토콜

Aurora MySQL 2.x와 Aurora MySQL 1.x의 CLI 차이점

- Aurora MySQL 2.x의 엔진 이름은 aurora-mysql이고 Aurora MySQL 1.x의 엔진 이름은 계속해서 aurora입니다.
- Aurora MySQL 2.x의 기본 파라미터 그룹은 default.aurora-mysql5.7이고 Aurora MySQL 1.x의 기본 파라미터 그룹은 계속해서 default.aurora5.6입니다.
- Aurora MySQL 2.x의 DB 클러스터 파라미터 그룹 패밀리 이름은 aurora-mysql5.7이고 Aurora MySQL 1.x의 DB 클러스터 파라미터 그룹 패밀리 이름은 계속해서 aurora5.6입니다.

전체 CLI 명령 세트 및 Aurora MySQL 2.x와 Aurora MySQL 1.x의 차이점은 Aurora 설명서를 참조하세요.

개선 사항

- INSERT 문을 실행시키고 Fast Insert 최적화를 이용할 때 Aurora Writer가 재시작되는 문제를 수정했습니다.
- Aurora 복제본에서 ALTER DATABASE 문을 실행시킬 때 Aurora 복제본이 재시작되는 문제를 수정했습니다.
- Aurora Writer에서 방금 삭제된 테이블에 쿼리를 실행할 때 Aurora 복제본이 재시작되는 문제를 수정했습니다.
- Aurora 복제본에서 innodb_adaptive_hash_index를 OFF로 설정할 때 Aurora 복제본이 재시작되는 문제를 수정했습니다.
- Aurora Writer에서 TRUNCATE TABLE 쿼리를 실행할 때 Aurora 복제본이 재시작되는 문제를 수정했습니다.
- INSERT 문을 실행할 때 Aurora Writer가 특정 상황에서 정지하는 문제를 수정했습니다. 여러 노드 클러스터에서는 장애 조치가 시작될 수 있습니다.
- 설정 세션 변수와 관련된 메모리 누수 문제를 수정했습니다.
- 생성된 열이 있는 테이블의 제거 실행 취소와 관련된 특정 상황에서 Aurora Writer가 정지하는 문제를 수정했습니다.
- 이진수 로깅이 활성화되어 있을 때 Aurora Writer가 때때로 재시작하는 문제를 수정했습니다.

MySQL 버그 수정 통합

- 좌측 조인이 외부 측에 잘못된 결과를 반환합니다(버그 #22833364).

Aurora MySQL 데이터베이스 엔진 업데이트 2018년 3월 13일(버전 2.01.1)(사용되지 않음)

(버전 2.01.1)

Aurora MySQL 2.01.1이 정식 버전입니다. Aurora MySQL 2.x 버전은 MySQL 5.7과 호환 가능하고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

스냅샷에서 복원한 클러스터를 포함하여 새로운 Aurora MySQL DB 클러스터를 생성할 때는 MySQL 5.7 또는 MySQL 5.6과의 호환성을 선택할 수 있습니다. MySQL 5.6 호환 스냅샷을 복원할 때, MySQL 5.7 또는 MySQL 5.6과의 호환성을 선택할 수 있습니다.

Aurora MySQL 1.14*, 1.15*, 1.16*, 1.17*의 스냅샷을 Aurora MySQL 2.01.1로 복원할 수 있습니다.

Aurora MySQL 1.x 클러스터를 Aurora MySQL 2.01.1로 인플레이스 업그레이드하거나, Amazon S3 백업에서 Aurora MySQL 2.01.1로 복원하는 작업은 허용되지 않습니다. 향후 Aurora MySQL 2.x 릴리스에서는 이러한 제한 사항을 없앨 계획입니다.

Aurora MySQL 5.7의 이 릴리스에서는 성능 스키마가 비활성화됩니다. 성능 스키마를 지원하려면 Aurora 2.03으로 업그레이드하십시오.

Aurora MySQL 5.6과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 5.6에서 지원되지만, 이러한 기능은 Aurora MySQL 5.7에서 현재 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.
- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- 동기식 호출 함수를 위한 기본 AWS Lambda 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.

- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

현재, Aurora MySQL 2.01.1은 Aurora MySQL 버전 1.16 이상에서 추가된 기능을 지원하지 않습니다. Aurora MySQL 버전 1.16에 대한 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 단원을 참조하십시오.

MySQL 5.7 호환성

Aurora MySQL 2.01.1은 MySQL 5.7과 연결성이 호환되며, JSON 지원, 공간 인덱스, 생성된 열과 같은 기능을 포함합니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

Aurora MySQL 2.01.1은 현재, 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 글로벌 트랜잭션 식별자(GTID) Aurora MySQL은 버전 2.04 이상에서 GTID를 지원합니다.
- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- `CREATE TABLESPACE` SQL 문
- X 프로토콜

Aurora MySQL 2.x와 Aurora MySQL 1.x의 CLI 차이점

- Aurora MySQL 2.x의 엔진 이름은 `aurora-mysql`이고 Aurora MySQL 1.x의 엔진 이름은 계속해서 `aurora`입니다.
- Aurora MySQL 2.x의 기본 파라미터 그룹은 `default.aurora-mysql5.7`이고 Aurora MySQL 1.x의 기본 파라미터 그룹은 계속해서 `default.aurora5.6`입니다.

- Aurora MySQL 2.x의 DB 클러스터 파라미터 그룹 패밀리 이름은 `aurora-mysql5.7`이고 Aurora MySQL 1.x의 DB 클러스터 파라미터 그룹 패밀리 이름은 계속해서 `aurora5.6`입니다.

전체 CLI 명령 세트 및 Aurora MySQL 2.x와 Aurora MySQL 1.x의 차이점은 Aurora 설명서를 참조하세요.

개선 사항

- MySQL 5.6 호환 스냅샷이 MySQL 5.7 호환성으로 복원되었을 때 Aurora 지정 데이터베이스 권한이 잘못 생성된 스냅샷 관련 문제를 해결했습니다.
- 1.17 스냅샷 복원에 대한 지원이 추가되었습니다.

Aurora MySQL 데이터베이스 엔진 업데이트 2018년 2월 6일(버전 2.01)(사용되지 않음)

버전: 2.01

Aurora MySQL 2.01이 정식 버전입니다. 앞으로 Aurora MySQL 2.x 버전은 MySQL 5.7과 호환되고, Aurora MySQL 1.x 버전은 MySQL 5.6과 호환됩니다.

새 Aurora MySQL DB 클러스터(스냅샷에서 복원된 클러스터 포함)를 생성할 때 MySQL 5.7 또는 MySQL 5.6과의 호환성을 선택할 수 있습니다.

Aurora MySQL 1.14*, 1.15*, 1.16*의 스냅샷을 Aurora MySQL 2.01로 복원할 수 있습니다.

Aurora MySQL 1.x 클러스터를 Aurora MySQL 2.01로 인플레이스 업그레이드하거나, Amazon S3 백업에서 Aurora MySQL 2.01로 복원하는 작업은 허용되지 않습니다. 향후 Aurora MySQL 2.x 릴리스에서는 이러한 제한 사항을 없앨 계획입니다.

Aurora MySQL 5.7의 이 릴리스에서는 성능 스키마가 비활성화됩니다. 성능 스키마를 지원하려면 Aurora 2.03으로 업그레이드하십시오.

Aurora MySQL 5.6과의 비교

다음 Amazon Aurora MySQL 기능은 Aurora MySQL 5.6에서 지원되지만, 이러한 기능은 Aurora MySQL 5.7에서 현재 지원되지 않습니다.

- 비동기식 키 미리 가져오기(AKP). 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.

- 해시 조인. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- 동기식 호출 함수를 위한 기본 AWS Lambda 함수입니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 네이티브 함수로 Lambda 함수 호출을 참조하세요.
- 배치화 스캔. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 섹션을 참조하세요.
- Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 MySQL에서 데이터 마이그레이션을 참조하세요.

현재, Aurora MySQL 2.01은 Aurora MySQL 버전 1.16 이상에서 추가된 기능을 지원하지 않습니다. Aurora MySQL 버전 1.16에 대한 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음) 단원을 참조하십시오.

MySQL 5.7 호환성

Aurora MySQL 2.01은 MySQL 5.7과 연결성이 호환되며, JSON 지원, 공간 인덱스, 생성된 열과 같은 기능을 포함합니다. Aurora MySQL은 z축 곡선을 사용하는 공간 인덱싱의 기본 구현을 사용하여 MySQL 5.7보다 20배 이상 높은 쓰기 성능과 10배 이상 높은 읽기 성능을 공간 데이터 집합에 제공합니다.

Aurora MySQL 2.01은 현재 다음과 같은 MySQL 5.7 기능을 지원하지 않습니다.

- 글로벌 트랜잭션 식별자(GTID) Aurora MySQL은 버전 2.04 이상에서 GTID를 지원합니다.
- 그룹 복제 플러그인
- 페이지 크기 증가
- 시작 시 InnoDB 버퍼 풀 로딩
- InnoDB 풀 텍스트 구문 분석기 플러그인
- 멀티 소스 복제
- 온라인 버퍼 풀 크기 조정
- 암호 확인 플러그인
- 쿼리 다시 쓰기 플러그인
- 복제 필터링
- CREATE TABLESPACE SQL 문
- X 프로토콜

Aurora MySQL 2.x와 Aurora MySQL 1.x의 CLI 차이점

- Aurora MySQL 2.x의 엔진 이름은 `aurora-mysql`이고 Aurora MySQL 1.x의 엔진 이름은 계속해서 `aurora`입니다.
- Aurora MySQL 2.x의 기본 파라미터 그룹은 `default.aurora-mysql5.7`이고 Aurora MySQL 1.x의 기본 파라미터 그룹은 계속해서 `default.aurora5.6`입니다.
- Aurora MySQL 2.x의 DB 클러스터 파라미터 그룹 패밀리 이름은 `aurora-mysql5.7`이고 Aurora MySQL 1.x의 DB 클러스터 파라미터 그룹 패밀리 이름은 계속해서 `aurora5.6`입니다.

전체 CLI 명령 세트 및 Aurora MySQL 2.x와 Aurora MySQL 1.x의 차이점은 Aurora 설명서를 참조하세요.

Amazon Aurora MySQL 버전 1에 대한 데이터베이스 엔진 업데이트(사용되지 않음)

다음은 Amazon Aurora 버전 1 데이터베이스 엔진 업데이트입니다.

- Aurora MySQL 데이터베이스 엔진 업데이트 2021년 9월 30일(버전 1.23.4)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2021년 6월 28일(버전 1.23.3)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2021년 3월 18일(버전 1.23.2)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2020년 11월 24일(버전 1.23.1)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2020년 9월 2일(버전 1.23.0)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2021년 6월 3일(버전 1.22.5)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2021년 3월 4일(버전 1.22.4)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2020년 11월 9일(버전 1.22.3)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2020년 3월 5일(버전 1.22.2)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2019년 12월 23일(버전 1.22.1)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2019년 11월 25일(버전 1.22.0)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2019년 11월 25일(버전 1.21.0)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2020년 3월 5일(버전 1.20.1)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2019년 11월 11일(버전 1.20.0)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2020년 3월 5일(버전 1.19.6)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2019년 9월 19일(버전 1.19.5)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2019년 6월 5일(버전 1.19.2)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2019년 5월 9일(버전 1.19.1)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2019년 2월 7일(버전 1.19.0)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2018년 9월 20일(버전 1.18.0)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2020년 3월 5일(버전 1.17.9)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2019년 1월 17일(버전 1.17.8)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2018년 10월 8일(버전 1.17.7)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2018년 9월 6일(버전 1.17.6)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2018년 8월 14일(버전 1.17.5)(사용되지 않음)

- Aurora MySQL 데이터베이스 엔진 업데이트 2018년 8월 7일(버전 1.17.4)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2018년 6월 5일(버전 1.17.3)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2018년 4월 27일(버전 1.17.2)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2018년 3월 23일(버전 1.17.1)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2018년 3월 13일(버전 1.17)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2017년 11월 20일(버전 1.15.1)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트 2017년 10월 24일(버전 1.15)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트: 2018년 3월 13일(버전 1.14.4)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트: 2017년 9월 22일(버전 1.14.1)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트: 2017년 8월 7일(버전 1.14)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트: 2017년 5월 15일(버전 1.13)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트: 2017년 4월 5일(버전 1.12)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트: 2017년 2월 23일(버전 1.11)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트: 2017년 1월 12일(버전 1.10.1)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트: 2016년 12월 14일(버전 1.10)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트: 2016년 11월 10일(버전 1.9.0, 1.9.1)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트: 2016년 10월 26일(버전 1.8.1)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트: 2016년 10월 18일(버전 1.8)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트: 2016년 9월 20일(버전 1.7.1)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트: 2016년 8월 30일(버전 1.7.0)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트: 2016년 6월 1일(버전 1.6.5)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트: 2016년 4월 6일(버전 1.6)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트: 2016년 1월 11일(버전 1.5)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트: 2015년 12월 3일(버전 1.4)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트: 2015년 10월 16일(버전 1.2, 1.3)(사용되지 않음)
- Aurora MySQL 데이터베이스 엔진 업데이트: 2015년 8월 24일(버전 1.1)(사용되지 않음)

Aurora MySQL 데이터베이스 엔진 업데이트 2021년 9월 30일(버전 1.23.4)(사용되지 않음)

버전: 1.23.4

Aurora MySQL 1.23.4가 정식 출시되었습니다. Aurora MySQL 2.* 버전은 MySQL 5.7과 호환되고 Aurora MySQL 1.* 버전은 MySQL 5.6과 호환됩니다.

이 엔진 버전은 2023년 2월 28일에 사용 중지될 예정입니다. 자세한 내용은 Amazon Aurora MySQL 호환 버전 1 수명 종료 준비를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 RDS 콘솔, AWS CLI 또는 Amazon RDS API를 통해 엔진 버전을 지정합니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

일반적인 개선 사항

- 내부 진단 로그 파일에 정보 메시지의 과도한 로깅으로 인해 리더 인스턴스에서 CPU 사용량을 높일 수 있는 문제를 수정했습니다.

우선 순위가 높은 수정 사항:

- CVE-2021-2307
- CVE-2021-2226
- CVE-2021-2160
- CVE-2021-2154
- CVE-2021-2060
- CVE-2021-2032
- CVE-2021-2001

Aurora MySQL 데이터베이스 엔진 업데이트 2021년 6월 28일(버전 1.23.3)(사용되지 않음)

버전: 1.23.3

Aurora MySQL 1.23.3이 정식 출시되었습니다. Aurora MySQL 1.* 버전은 MySQL 5.6과 호환되고 Aurora MySQL 2.* 버전은 MySQL 5.7과 호환됩니다.

이 엔진 버전은 2023년 2월 28일에 사용 중지될 예정입니다. 자세한 내용은 Amazon Aurora MySQL 호환 버전 1 수명 종료 준비를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 RDS 콘솔, AWS CLI 또는 Amazon RDS API를 통해 엔진 버전을 지정합니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

일반적인 안정성 및 가용성 향상

보안 수정:

- CVE-2021-23841
- CVE-2021-3449
- CVE-2020-28196

Aurora MySQL 데이터베이스 엔진 업데이트 2021년 3월 18일(버전 1.23.2)(사용되지 않음)

버전: 1.23.2

Aurora MySQL 1.23.2가 정식 출시되었습니다. Aurora MySQL 1.* 버전은 MySQL 5.6과 호환되고 Aurora MySQL 2.* 버전은 MySQL 5.7과 호환됩니다.

이 엔진 버전은 2023년 2월 28일에 사용 중지될 예정입니다. 자세한 내용은 Amazon Aurora MySQL 호환 버전 1 수명 종료 준비를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 RDS 콘솔, AWS CLI 또는 Amazon RDS API를 통해 엔진 버전을 지정합니다.

> **Note**
>
> 이 버전은 현재 다음 리전에서 사용할 수 없습니다. AWS GovCloud(미국 동부) [us-gov-east-1], AWS GovCloud(미국-서부) [us-gov-west-1]. 사용 가능해지면 따로 발표할 예정입니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

높은 우선 순위:

- CVE-2020-14867
- CVE-2020-14812
- CVE-2020-14769
- CVE-2020-14765
- CVE-2020-14793
- CVE-2020-14672
- CVE-2020-1971
- CVE-2018-3143

가용성 향상:

- 리더 DB 인스턴스가 재시작될 수 있는 동적 클러스터 스토리지 크기 조정 기능의 문제가 해결되었습니다.
- `RESET QUERY CACHE` 문의 경합 조건으로 인한 장애 조치 문제를 수정했습니다.
- 쿼리 캐시가 있는 중첩된 저장 프로시저 호출의 충돌을 수정했습니다.

- 분할 또는 하위 분할 테이블의 불완전한 잘라내기로부터 복구할 때 mysqld의 반복적인 재시작이 불가능했던 문제를 수정했습니다.
- 온프레미스 또는 RDS for MySQL에서 Aurora MySQL로의 마이그레이션이 성공적으로 실행되지 못하는 문제를 수정했습니다.
- 스토리지 볼륨 크기를 조정하는 동안 데이터베이스 재시작을 유발할 수 있는 드문 경합 조건이 수정되었습니다.
- 잠금 관리자에서 교착 상태로 인해 두 트랜잭션에 의해 잠금이 공유되어 데이터베이스가 다시 시작되는 문제를 수정했습니다.
- 장기 실행 쓰기 트랜잭션으로 인해 데이터베이스가 다시 시작되는 트랜잭션 잠금 메모리 관리와 관련된 문제를 수정했습니다.
- 트랜잭션 롤백 중에 데이터베이스 재시작 또는 장애 조치로 이어지는 잠금 관리자의 교착 상태를 수정했습니다.
- 5.6의 랩 모드에서 테이블에 빠른 온라인 DDL이 활성화되어 있는 경우 5.6에서 5.7로 업그레이드하는 동안 발생하던 문제가 해결되었습니다.
- 패치 적용을 위해 데이터베이스 작업에서 중단된 지점을 확인하는 동안 가동 중지 없는 패치 적용 중에 엔진이 다시 시작될 수 있는 여러 문제를 수정했습니다.
- DROP TRIGGER, ALTER TABLE, 특히 테이블의 분할 유형 또는 분할 영역 수를 수정하는 ALTER TABLE 등, 중단된 DDL 작업으로 인해 발생하는 반복적 재시작과 관련한 여러 문제가 해결되었습니다.
- 대규모 인스턴스 클래스(R4/R5-16XL, R5-12XL, R5-24XL)에서 반복적인 재시작 및 높은 CPU 사용률을 방지하기 위해 16XL 및 24XL 인스턴스에서 기본값 table_open_cache를 업데이트했습니다. 이는 1.21.x와 1.22.x 릴리스에 영향을 미쳤습니다.
- HA_ERR_KEY_NOT_FOUND 오류가 발생하면서 binlog 복제본이 중지될 수 있는 문제를 수정했습니다.

MySQL 커뮤니티 에디션 버그 픽스 통합

- 복제: SHOW BINLOG EVENTS 이 실행되는 동안 모든 병렬 트랜잭션이 차단되었습니다. 이 수정 프로그램은 이제 SHOW BINLOG EVENTS 프로세스가 파일의 끝 위치를 계산하는 동안에만 잠기므로 병렬 트랜잭션이 오랜 기간 차단되지 않도록 합니다. (버그 #76618, 버그 #20928790)

Aurora MySQL 데이터베이스 엔진 업데이트 2020년 11월 24일(버전 1.23.1)(사용되지 않음)

버전: 1.23.1

Aurora MySQL 1.23.1이 정식 버전입니다. Aurora MySQL 1.* 버전은 MySQL 5.6과 호환되고 Aurora MySQL 2.* 버전은 MySQL 5.7과 호환됩니다.

이 엔진 버전은 2023년 2월 28일에 사용 중지될 예정입니다. 자세한 내용은 Amazon Aurora MySQL 호환 버전 1 수명 종료 준비를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 RDS 콘솔, AWS CLI 또는 Amazon RDS API를 통해 엔진 버전을 지정합니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

보안 수정:

관리된 환경에서 처리를 미세 조정하기 위한 수정 사항 및 기타 향상된 기능 아래의 추가 CVE 수정:

- CVE-2020-14559
- CVE-2020-14539

호환되지 않는 변경 사항:

이 버전에서는 `mysqldump` 명령의 동작에 영향을 주는 권한 변경을 도입했습니다. 사용자는 `PROCESS` 테이블에 액세스하려면 `INFORMATION_SCHEMA.FILES` 권한이 있어야 합니다. 변경 없이 `mysqldump` 명령을 실행하려면 `PROCESS` 명령이 연결되는 데이터베이스 사용자에게 `mysqldump` 권한을 부여합니다. `mysqldump` 옵션을 사용하여 `--no-tablespaces` 명령을 실행할 수도 있습니다. 이 옵션을 사용하면 `mysqldump` 출력에 `CREATE LOGFILE GROUP` 또는 `CREATE TABLESPACE` 문이 포함되지 않습니다. 이 경우 `mysqldump` 명령은 `INFORMATION_SCHEMA.FILES` 테이블에 액세스하지 않으므로 `PROCESS` 권한을 부여할 필요가 없습니다.

가용성 향상:

- 1.23.0을 실행하는 전역 데이터베이스 보조 클러스터에서 Aurora 판독기 인스턴스가 반복적으로 다시 시작되던 문제를 수정했습니다.
- 기본 리전 라이터가 이전 릴리스 버전을 사용하는 동안 릴리스 1.23.0으로 업그레이드하면, 글로벌 데이터베이스 보조 리전의 복제본이 다시 시작될 수 있는 문제가 해결되었습니다.
- Aurora MySQL 1.23.0에 도입된 동적 크기 조정 기능에서 메모리 누수가 수정되었습니다.
- 병렬 쿼리 기능을 사용하여 쿼리를 실행하는 동안 서버가 다시 시작될 수 있는 문제가 해결되었습니다.
- 데이터베이스 엔진에서 네트워크에서 읽거나 쓰는 동안 오류가 발생하면 클라이언트 세션이 정지될 수 있는 문제가 해결되었습니다.

Aurora MySQL 데이터베이스 엔진 업데이트 2020년 9월 2일(버전 1.23.0)(사용되지 않음)

버전: 1.23.0

Aurora MySQL 1.23.0이 정식 버전입니다. Aurora MySQL 1.* 버전은 MySQL 5.6과 호환되고 Aurora MySQL 2.* 버전은 MySQL 5.7과 호환됩니다.

이 엔진 버전은 2023년 2월 28일에 사용 중지될 예정입니다. 자세한 내용은 Amazon Aurora MySQL 호환 버전 1 수명 종료 준비를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

Aurora MySQL 1.* 데이터베이스의 스냅샷을 Aurora MySQL 1.23.0으로 복원할 수 있습니다.

> **Important**
>
> 이 버전의 Aurora 스토리지 개선 사항으로 인해 사용 가능한 업그레이드 경로가 Aurora MySQL 1.23에서 Aurora MySQL 2.*로 제한됩니다. Aurora MySQL 1.23 클러스터를 2.*로 업그레이드하는 경우 Aurora MySQL 2.09.0 이상으로 업그레이드해야 합니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 RDS 콘솔, AWS CLI 또는 Amazon RDS API를 통해 엔진 버전을 지정하십시오.

> **Note**
>
> 이 버전은 현재 다음 리전에서 사용할 수 없습니다. AWS GovCloud(미국 동부) [us-gov-east-1], AWS GovCloud(미국-서부) [us-gov-west-1]. 사용 가능해지면 따로 발표할 예정입니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

새로운 기능:

- 이제 DB 클러스터 파라미터 `aurora_parallel_query`의 값을 변경하여 기존 클러스터에 대해 병렬 쿼리를 설정하거나 해제할 수 있습니다. 클러스터를 생성할 때 `parallelquery` 파라미터에 대한 `--engine-mode` 설정을 사용할 필요가 없습니다.

  이제 Aurora MySQL을 사용 가능한 모든 리전에서 병렬 쿼리를 사용할 수 있습니다.

  Aurora 클러스터에서 병렬 쿼리를 업그레이드하고 활성화하는 절차에 대한 기타 여러 가지 기능 향상 및 변경 사항이 많이 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL용 Parallel Query 처리를 참조하세요.
- 이 릴리스에서는 최대 128테비바이트(TiB)의 스토리지가 포함된 Amazon Aurora MySQL 데이터베이스 인스턴스를 생성할 수 있습니다. 새 스토리지 제한은 이전 64TiB에서 증가한 것입니다. 128TiB 스토리지 크기는 더 큰 데이터베이스를 지원합니다. 작은 인스턴스 크기(db.t2 또는 db.t3)에서는 이 기능이 지원되지 않습니다. 단일 테이블스페이스는 16KB 페이지 크기의 InnoDB 한도로 인해 64TiB 이상으로 증가할 수 없습니다.

  Aurora은 클러스터 볼륨 크기가 128TiB에 가까울 때 경고를 표시하므로 크기 제한에 도달하기 전에 조치를 취할 수 있습니다. 경고는 mysql 로그와 AWS Management Console의 RDS 이벤트에 나타납니다.
- 대규모 트랜잭션이 관련된 경우 충돌 복구 시간과 커밋 시간 지연을 줄이기 위해 바이너리 로그(binlog) 처리를 개선했습니다.
- Aurora는 클러스터 스토리지 공간의 크기를 동적으로 조정합니다. 동적 크기 조정이 사용되면 Aurora DB 클러스터에서 데이터를 제거할 때 DB 클러스터의 스토리지 공간이 자동으로 줄어듭니다. 자세한 내용은 Amazon Aurora 사용 설명서의 스토리지 조정을 참조하세요.

> **Note**
>
> 동적 크기 조정 기능은 Aurora를 사용할 수 있는 AWS 리전에 단계적으로 배포되고 있습니다. 클러스터가 있는 리전에 따라 이 기능을 아직 사용하지 못할 수도 있습니다. 자세한 내용은 새로운 소식 공지를 참조하세요.

높은 우선 순위:

- CVE-2019-2911
- CVE-2019-2537
- CVE-2018-2787
- CVE-2018-2784
- CVE-2018-2645
- CVE-2018-2640

가용성 향상:

- 잠금 관리자에서 교착 상태로 인해 두 트랜잭션에 의해 잠금이 공유되어 데이터베이스가 다시 시작되는 문제를 수정했습니다.
- 장기 실행 쓰기 트랜잭션으로 인해 데이터베이스가 다시 시작되는 트랜잭션 잠금 메모리 관리와 관련된 문제를 수정했습니다.
- 트랜잭션 롤백 중에 데이터베이스 재시작 또는 장애 조치로 이어지는 잠금 관리자의 교착 상태를 수정했습니다.
- Fast DDL이 활성화된 테이블에서 `innodb_file_format`이 변경되었을 때 5.6에서 5.7로 업그레이드하는 동안 문제를 수정했습니다.
- 패치 적용을 위해 데이터베이스 작업에서 중단된 지점을 확인하는 동안 가동 중지 없는 패치 적용 중에 엔진이 다시 시작될 수 있는 여러 문제를 수정했습니다.
- 중단된 `DROP TRIGGER` 작업을 복구하는 동안 DB 인스턴스의 재시작에 영향을 주는 DDL 복구와 관련된 문제를 수정했습니다.
- 특정 파티셔닝 작업을 실행하는 동안 충돌이 발생할 경우 데이터베이스를 사용할 수 없게 되는 버그를 수정했습니다. 특히, 파티셔닝 유형 또는 테이블의 파티션 수를 수정하는 중단된 `ALTER TABLE` 작업입니다.

- 대규모 인스턴스 클래스(R4/R5-16XL, R5-12XL, R5-24XL)에서 반복적인 장애 조치 및 높은 CPU 사용률을 유발할 수 있는 16XL 및 24XL 인스턴스에서 기본값 `table_open_cache`를 수정합니다. 이는 1.21.x와 1.22.x에 영향을 미쳤습니다.

글로벌 데이터베이스:

- Aurora 글로벌 데이터베이스의 기본 및 보조 AWS 리전에 대한 MySQL `INFORMATION_SCHEMA.REPLICA_HOST_STATUS` 뷰에 누락된 데이터를 채웁니다.
- 기본 리전과 보조 리전 간의 일시적인 네트워크 연결 문제 이후 기본 리전에서 UNDO 레코드의 가비지 수집으로 인해 글로벌 DB 보조 리전에서 발생할 수 있는 예기치 않은 쿼리 실패를 수정했습니다.

병렬 쿼리:

- 병렬 쿼리로 인해 장기 실행 쿼리가 빈 결과를 반환할 수 있는 문제를 수정했습니다.
- Aurora 읽기 전용 복제본의 작은 테이블에 대한 쿼리가 1초 이상 걸릴 수 있는 문제를 수정했습니다.
- 병렬 쿼리와 DML 문이 과중한 워크로드 하에서 동시에 실행될 때 다시 시작될 수 있는 문제를 수정했습니다.

일반적인 개선 사항:

- 이미 공간 값이 큰 테이블에 공간 인덱스가 생성된 경우 공간 인덱스를 사용하는 쿼리가 부분 결과를 반환할 수 있는 문제를 수정했습니다.
- 감사 시스템 변수 `server_audit_incl_users` 및 `server_audit_excl_users`의 최대 허용 길이가 1024바이트에서 2000바이트로 증가했습니다.
- Aurora MySQL binlog 기본이 `statement` `binlog_format` 아래의 S3에서 데이터를 로드할 때 Aurora MySQL binlog 기본에 연결된 binlog 복제본이 불완전한 데이터를 표시하는 문제를 수정했습니다.
- 커뮤니티 동작을 준수하여 데이터를 로드하는 `mixed` 대신 `row`에 `statement` binlog_format을 매핑합니다.
- 사용자가 연결을 종료하고 세션이 임시 테이블을 사용 중일 때 binlog 복제가 작동을 중지하는 문제를 수정했습니다.
- MyISAM 임시 테이블을 포함하는 쿼리의 응답 시간이 개선되었습니다.
- binlog 작업자가 네이티브 lambda 함수를 실행할 때의 권한 문제를 수정했습니다.

- 느린 로그 또는 일반 로그를 쿼리하거나 회전하려고 할 때 Aurora 읽기 전용 복제본의 문제를 수정했습니다.
- `binlog_checksum` 파라미터가 마스터 및 복제본에서 서로 다른 값으로 설정된 경우 논리적 복제가 끊어지는 문제를 해결했습니다.
- 읽기 전용 복제본이 쓰기에서 최근에 커밋된 트랜잭션의 일부 결과를 일시적으로 볼 수 있는 문제를 해결했습니다.
- 교착 상태가 해결될 때 `show engine innodb status`의 롤백 트랜잭션에 대한 트랜잭션 정보를 포함합니다.

MySQL 커뮤니티 에디션 버그 픽스 통합

- `ALTER TABLE ADD COLUMN ALGORITHM=QUICK`이 있는 Binlog 이벤트는 커뮤니티 에디션과 호환되도록 `ALGORITHM=DEFAULT`로 다시 작성됩니다.
- 버그 #22350047: 이전 STMTS가 커밋된 저장점으로 롤백한 후 클라이언트가 중단된 경우
- 버그 #29915479: COM_BINLOG_DUMP 없이 COM_REGISTER_SLAVE를 실행하면 서버가 종료될 수 있습니다.
- 버그 #30441969: 버그 #29723340: 데이터 세트가 있는 SQL 쿼리 후 MySQL Server 충돌
- 버그 #30628268: 메모리 부족 충돌
- 버그 #27081349: 공간 기능을 사용하여 삭제할 때 예기치 않은 동작
- 버그 #27230859: 잘못된 폴리곤을 처리하는 동안 예기치 않은 동작
- 버그 #27081349: 공간을 사용하여 삭제할 때 예기치 않은 동작
- 버그 #26935001: ALTER TABLE AUTO_INCREMENT가 폐기된 테이블스페이스에서 인덱스를 읽으려고 시도합니다.
- 버그 #29770705: 특정 WHERE 절을 사용하여 SELECT를 실행하는 동안 서버가 충돌했습니다.
- 버그 #27659490: 동적 범위 및 인덱스 병합을 사용하여 선택하면 너무 많은 메모리를 사용합니다(OOM).
- 버그 #24786290: 마스터에서 버그 #74145가 발생한 후 복제가 중단됩니다.
- 버그 #27703912: 많은 준비 과정에서 과도한 메모리 사용
- 버그 #20527363: 임시 테이블 충돌 자르기: !DICT_TF2_FLAG_IS_SET(TABLE, DICT_TF2_TEMPORARY)
- 버그 #23103937: PS_TRUNCATE_ALL_TABLES()이 SUPER_READ_ONLY 모드에서 작동하지 않습니다.

- 버그 #25053286: 절차에서 조건이 있는 뷰를 사용하면 잘못된 동작이 발생합니다(5.6.36에서 수정됨).
- 버그 #25586773: SP의 루프에서 테이블 선택 생성에 대한 잘못된 동작(5.6.39에서 수정됨)
- 버그 #27407480: AUTOMATIC_SP_PRIVILEGES에는 MySQL 사용자 테이블에 대한 삽입 권한이 필요합니다.
- 버그 #26997096: `relay_log_space` 값이 동기화된 방식으로 업데이트되지 않으므로 값이 릴레이 로그에서 사용되는 실제 디스크 공간보다 훨씬 높습니다.
- 버그 #15831300: SLAVE_TYPE_CONVERSIONS=ALL_NON_LOSSY가 예상대로 작동하지 않습니다.
- SSL 버그 백포트 버그 #17087862, 버그 #20551271
- 버그 #16894092: 5.6.6 이상에서 INSERT INTO ... SELECT ... FROM에 대한 성능 회귀(5.6.15에서 수정됨)
- `SLAVE_TYPE_CONVERSIONS`와 관련된 버그 수정을 포팅합니다.

Aurora MySQL 데이터베이스 엔진 업데이트 2021년 6월 3일(버전 1.22.5)(사용되지 않음)

버전: 1.22.5

Aurora MySQL 1.22.5가 정식 버전입니다. Aurora MySQL 1.* 버전은 MySQL 5.6과 호환되고 Aurora MySQL 2.* 버전은 MySQL 5.7과 호환됩니다.

이 엔진 버전은 2023년 2월 28일에 사용 중지될 예정입니다. 자세한 내용은 Amazon Aurora MySQL 호환 버전 1 수명 종료 준비를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 RDS 콘솔, AWS CLI 또는 Amazon RDS API를 통해 엔진 버전을 지정합니다.

> **Note**
>
> 이 버전은 장기 지원(LTS) 릴리스로 지정됩니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL LTS(장기 지원) 릴리스를 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

가용성 향상:

- 내부 정리 스레드 간의 동시성 충돌로 인해 데이터베이스가 중단되고 이후에 다시 시작되거나 장애 조치될 수 있는 문제가 해결되었습니다.
- XA 트랜잭션을 준비된 상태로 유지하는 동안 데이터베이스가 다시 시작되고 해당 트랜잭션이 커밋되거나 롤백되기 전에 또 다시 시작되는 경우 클러스터를 사용할 수 없게 되는 문제가 해결되었습니다. 이 수정 전에는 첫 번째 다시 시작 전에 클러스터를 특정 시점으로 복원하여 문제를 해결할 수 있습니다.
- DDL 문을 처리하는 동안 데이터베이스가 다시 시작되면 InnoDB 제거가 차단되는 문제가 해결되었습니다. 그 결과 InnoDB 기록 목록 길이가 늘어나고 클러스터 스토리지 볼륨이 채워질 때까지 계속 증가하여 데이터베이스를 사용할 수 없게 됩니다. 이 수정 전에는 데이터베이스를 다시 시작하여 제거를 차단 해제함으로써 문제를 완화할 수 있습니다.

Aurora MySQL 데이터베이스 엔진 업데이트 2021년 3월 4일(버전 1.22.4)(사용되지 않음)

버전: 1.22.4

Aurora MySQL 1.22.4가 정식 출시되었습니다. Aurora MySQL 1.* 버전은 MySQL 5.6과 호환되고 Aurora MySQL 2.* 버전은 MySQL 5.7과 호환됩니다.

이 엔진 버전은 2023년 2월 28일에 사용 중지될 예정입니다. 자세한 내용은 Amazon Aurora MySQL 호환 버전 1 수명 종료 준비를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 RDS 콘솔, AWS CLI 또는 Amazon RDS API를 통해 엔진 버전을 지정합니다.

> **Note**
>
> 이 버전은 장기 지원(LTS) 릴리스로 지정됩니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL LTS(장기 지원) 릴리스를 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

보안 수정:

관리된 환경에서 처리를 미세 조정하기 위한 수정 사항 및 기타 향상된 기능 아래의 추가 CVE 수정:

- CVE-2020-14867
- CVE-2020-14812
- CVE-2020-14793
- CVE-2020-14769
- CVE-2020-14765
- CVE-2020-14672
- CVE-2020-1971

가용성 향상:

- `kill session` 명령 중 데이터베이스 재시작 또는 장애 조치를 트리거할 수 있는 문제를 수정했습니다. 이 문제가 발생하면 AWS Support 팀에 문의하여 인스턴스에서 이 수정 사항을 활성화하세요.
- 대규모 트랜잭션이 관련된 경우 충돌 복구 시간과 커밋 지연 시간을 줄이기 이진 로깅을 개선했습니다.
- `HA_ERR_KEY_NOT_FOUND` 오류가 발생하면서 binlog 복제본이 중지될 수 있는 문제를 수정했습니다.

Aurora MySQL 데이터베이스 엔진 업데이트 2020년 11월 9일(버전 1.22.3)(사용되지 않음)

버전: 1.22.3

Aurora MySQL 1.22.3이 정식 버전입니다. Aurora MySQL 1.* 버전은 MySQL 5.6과 호환되고 Aurora MySQL 2.* 버전은 MySQL 5.7과 호환됩니다.

이 엔진 버전은 2023년 2월 28일에 사용 중지될 예정입니다. 자세한 내용은 Amazon Aurora MySQL 호환 버전 1 수명 종료 준비를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 RDS 콘솔, AWS CLI 또는 Amazon RDS API를 통해 엔진 버전을 지정합니다.

> **Note**
>
> 이 버전은 장기 지원(LTS) 릴리스로 지정됩니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL LTS(장기 지원) 릴리스를 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼이나 AWS Support를 통해 AWS Support 팀에 도움을 요청할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

보안 수정:

관리된 환경에서 처리를 미세 조정하기 위한 수정 사항 및 기타 향상된 기능 아래의 추가 CVE 수정:

- CVE-2020-14559
- CVE-2020-14539
- CVE-2020-2579
- CVE-2020-2812
- CVE-2020-2780
- CVE-2020-2763

호환되지 않는 변경 사항:

이 버전에서는 mysqldump 명령의 동작에 영향을 주는 권한 변경을 도입했습니다. 사용자는 PROCESS 테이블에 액세스하려면 INFORMATION_SCHEMA.FILES 권한이 있어야 합니다. 변경 없이 mysqldump 명령을 실행하려면 PROCESS 명령이 연결되는 데이터베이스 사용자에게 mysqldump 권한을 부여합니다. mysqldump 옵션을 사용하여 --no-tablespaces 명령을 실행할 수도 있습니다. 이 옵션을 사용하면 mysqldump 출력에 CREATE LOGFILE GROUP 또는 CREATE TABLESPACE 문이 포함되지 않습니다. 이 경우 mysqldump 명령은 INFORMATION_SCHEMA.FILES 테이블에 액세스하지 않으므로 PROCESS 권한을 부여할 필요가 없습니다.

가용성 향상:

- 수행하지 않은 DDL 문을 복구하는 동안 서버 재시작 문제가 해결되었습니다.
- 잠금 관리자에서 서버를 다시 시작할 수 있는 경쟁 조건이 수정되었습니다.
- 대규모 트랜잭션을 복구하는 동안 모니터링 에이전트가 서버를 재시작할 수 있는 문제가 해결되었습니다.

일반적인 개선 사항

- MIXED를 실행할 때 binlog_format ROW을 STATEMENT 대신 LOAD DATA FROM INFILE | S3에 매핑하는 동작을 변경했습니다.
- Aurora MySQL binlog 기본이 LOAD DATA FROM S3를 실행하고 binlog_format이 STATEMENT로 설정되어 있을 때 이 기본에 연결된 binlog 복제본이 불완전한 데이터를 표시하는 문제를 해결했습니다.

MySQL 커뮤니티 에디션 버그 픽스 통합

- 버그 #26654685: 외래 키 검사 중에 손상된 인덱스 ID가 어설션을 일으켰습니다.
- 버그 #15831300: 기본적으로 마스터의 작은 유형에서 슬레이브의 더 큰 유형으로 정수를 프로모션할 때 (예: 마스터의 SmallINT 열에서 슬레이브의 BIGINT 열로) 프로모션된 값은 서명된 것처럼 처리됩니다. 이제 이러한 경우 slave_type_conversions 서버 시스템 변수에 지정된 값 집합에 속한 ALL_SIGNED과 ALL_UNSIGNED를 하나 또는 두 가지 모두 사용하여 이 동작을 수정하거나 재정의할 수 있습니다. 자세한 내용은 행 기반 복제: 특성 수준 프로모션 및 수준 디모션과 변수 설명을 참조하십시오.
- 버그 #17449901: foreign_key_checks=0을 사용할 때 InnoDB는 외래 키 제약 조건에 필요한 인덱스를 삭제하여 테이블을 일관성 없는 상태로 배치하고 테이블 로드시 발생하는 외래 키 검사가 실

패하는 원인을 제공했습니다. InnoDB는 이제 외래 키 제약 조건에 필요한 인덱스를 삭제하는 것을 방지하며 foreign_key_checks=0도 포함합니다. 외래 키 인덱스를 삭제하기 전에 외래 키 제약 조건을 제거해야 합니다.

- 버그 #20768847: 테이블 변경 ... 외래 키 종속성이 있는 테이블에 대한 DROP INDEX 작업은 어설션을 일으켰습니다.

Aurora MySQL 데이터베이스 엔진 업데이트 2020년 3월 5일(버전 1.22.2)(사용되지 않음)

버전: 1.22.2

Aurora MySQL 1.22.2가 정식 버전입니다. Aurora MySQL 1.* 버전은 MySQL 5.6과 호환되고 Aurora MySQL 2.* 버전은 MySQL 5.7과 호환됩니다.

이 엔진 버전은 2023년 2월 28일에 사용 중지될 예정입니다. 자세한 내용은 Amazon Aurora MySQL 호환 버전 1 수명 종료 준비를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 RDS 콘솔, AWS CLI 또는 Amazon RDS API를 통해 엔진 버전을 지정하십시오.

> **Note**
>
> 이 버전은 현재 다음 리전에서 사용할 수 없습니다. AWS GovCloud(미국 동부) [us-gov-east-1], AWS GovCloud(미국-서부) [us-gov-west-1]. 사용 가능해지면 따로 발표할 예정입니다.
>
> 이 버전은 장기 지원(LTS) 릴리스로 지정됩니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL LTS(장기 지원) 릴리스를 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

높은 우선 순위:

- 인증서 교체 후 간헐적인 연결 실패 문제를 해결했습니다.
- 쓰기 로드가 많은 일부 데이터베이스 클러스터에서 복제가 더 오래 걸리는 문제를 해결했습니다.
- `binlog_checksum` 파라미터가 마스터 및 복제본에서 서로 다른 값으로 설정된 경우 논리적 복제가 끊어지는 문제를 해결했습니다.
- 읽기 전용 복제본에서 느린 로그 및 일반 로그가 제대로 회전하지 않을 수 있는 문제를 해결했습니다.
- ANSI 읽기 커밋된 격리 수준 동작의 문제를 해결했습니다.

Aurora MySQL 데이터베이스 엔진 업데이트 2019년 12월 23일(버전 1.22.1)(사용되지 않음)

버전: 1.22.1

Aurora MySQL 1.22.1이 정식 버전입니다. Aurora MySQL 1.* 버전은 MySQL 5.6과 호환되고 Aurora MySQL 2.* 버전은 MySQL 5.7과 호환됩니다.

이 엔진 버전은 2023년 2월 28일에 사용 중지될 예정입니다. 자세한 내용은 Amazon Aurora MySQL 호환 버전 1 수명 종료 준비를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 AWS Management Console, 또는 RDS API를 AWS CLI 통해 엔진 버전을 지정하십시오. 기존 Aurora MySQL 1.* 데이터베이스 클러스터를 Aurora MySQL 1.22.1로 업그레이드할 수 있는 옵션이 있습니다.

> **Note**
>
> 현재이 버전은 AWS GovCloud(미국 동부) [us-gov-east-1], AWS GovCloud(미국 서부) [us-gov-west-1], 중국(닝샤) [cn-northwest-1], 아시아 태평양(홍콩) [ap-east-1], 중동(바레인) [me-south-1 AWS 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> DB 클러스터를 업그레이드하는 절차가 변경되었습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

심각한 수정 사항:

- 테이블 잠금 및 임시 테이블과 관련된 엔진 복구를 방해했던 문제가 해결되었습니다.
- 임시 테이블을 사용할 때 이진 로그의 안정성이 향상되었습니다.

높은 우선 순위:

- Aurora 특정 데이터베이스 추적 및 로깅 하위 시스템에서 사용 가능한 메모리를 낮추는 느린 메모리 누수 문제가 수정되었습니다.

Aurora MySQL 데이터베이스 엔진 업데이트 2019년 11월 25일(버전 1.22.0)(사용되지 않음)

버전: 1.22.0

Aurora MySQL 1.22.0이 정식 버전입니다. Aurora MySQL 1.* 버전은 MySQL 5.6과 호환되고 Aurora MySQL 2.* 버전은 MySQL 5.7과 호환됩니다.

이 엔진 버전은 2023년 2월 28일에 사용 중지될 예정입니다. 자세한 내용은 Amazon Aurora MySQL 호환 버전 1 수명 종료 준비를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 AWS Management Console, 또는 RDS API를 AWS CLI 통해 엔진 버전을 지정하십시오. 기존 Aurora MySQL 1.* 데이터베이스 클러스터를 Aurora MySQL 1.22.0으로 업그레이드할 수 있는 옵션이 있습니다.

> **Note**
>
> 현재이 버전은 AWS GovCloud(미국 동부) [us-gov-east-1], AWS GovCloud(미국 서부) [us-gov-west-1], 중국(닝샤) [cn-northwest-1], 아시아 태평양(홍콩) [ap-east-1], 중동(바레인) [me-south-1], 남아메리카(상파울루) [sa-east-1] AWS 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> DB 클러스터를 업그레이드하는 절차가 변경되었습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

새로운 기능:

- Aurora MySQL 클러스터는 이제 r5.8xlarge, r5.16xlarge 및 r5.24xlarge 인스턴스 유형을 지원합니다.
- Binlog는 매우 큰 트랜잭션이 수반될 때 커밋 시간 지연을 줄이기 위한 새로운 개선 사항입니다.
- 현재 Aurora MySQL은 커밋 시 대규모 트랜잭션의 이벤트가 binlog에 쓰이는 기간을 최소화하는 메커니즘을 갖추고 있습니다. 이 메커니즘을 통해 이 기간 중 데이터베이스 충돌이 발생할 때 초래되는 오프라인 복구 지연을 효과적으로 방지할 수 있습니다. 또한 이 기능은 binlog 커밋 시 대규모 트랜잭션이 소규모 트랜잭션을 차단하는 문제도 해결합니다. 이 기능은 기본적으로 꺼져 있으며 워크로드에 필요한 경우 서비스 팀이 활성화할 수 있습니다. 활성화된 후에는 트랜잭션 크기가 500MB를 초과하면 트리거됩니다.
- 읽기 전용 복제본의 ANSI `READ COMMITTED` 격리 수준에 대한 지원을 추가하였습니다. 이 격리 수준을 통해 읽기 전용 복제본의 장기 실행 쿼리는 라이터 노드의 높은 쓰기 처리량에 영향을 미치지 않는 상태로 실행됩니다. 자세한 내용은 Aurora MySQL 격리 수준을 참조하십시오.

- 글로벌 데이터베이스를 사용하면 이제 다음 리전에 배포된 데이터베이스 클러스터에 보조 읽기 전용 복제본 AWS 리전을 추가할 수 있습니다. 리전: 미국 동부(버지니아 북부) [us-east-1], 미국 동부(오하이오)[us-east-2], 미국 서부(캘리포니아 북부) [us-west-1], 미국 서부(오레곤) [us-west-2], 유럽(아일랜드)[eu-west-1], 유럽(런던)[eu-west-2], 유럽(파리)[eu-west-3], 아시아 태평양(도쿄) [ap-northeast-1], 아시아 태평양(서울) [ap-northeast-2], 아시아 태평양(싱가포르) [ap-southeast-1], 아시아 태평양(시드니) [ap-southeast-2], 캐나다(중부) [ca-central-1], 유럽(프랑크푸르트)[eu-central-1], 및 아시아 태평양(뭄바이) [ap-south-1].
- 이제 핫 행 경합 기능은 정식 버전이며 Aurora 랩 모드 설정이 ON 상태일 필요가 없습니다. 이 기능은 동일한 페이지의 행에 대해 경합하는 트랜잭션이 많은 워크로드의 처리량을 크게 향상시킵니다.
- 이 버전에서는 새 클러스터에 대해 최신 브라질 시간대 업데이트를 지원하도록 시간대 파일을 업데이트하였습니다.

심각한 수정 사항:

- CVE-2019-2922
- CVE-2019-2923
- CVE-2019-2924
- CVE-2019-2910

높은 우선 순위:

- 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
- 데이터베이스 가동 중지 시간이 늘어나는 원인이 되는 DDL 복구 구성 요소 관련 문제를 해결하였습니다. `TRUNCATE TABLE` 열이 있는 테이블에서 `AUTO_INCREMENT` 쿼리를 실행한 이후 사용할 수 없게 되는 클러스터는 업데이트해야 합니다.

- 데이터베이스 가동 중지 시간이 늘어나는 원인이 되는 DDL 복구 구성 요소 관련 문제를 해결하였습니다. 다중 테이블에서 `DROP TABLE` 쿼리를 병렬로 실행한 이후 사용할 수 없게 되는 클러스터는 업데이트해야 합니다.

일반적인 안정성 수정 사항:

- 장기 실행 트랜잭션 중에 읽기 전용 복제본이 다시 시작되는 원인이 되는 문제를 해결했습니다. 여유 메모리 하락 가속화와 동시에 복제본이 다시 시작하는 문제를 겪는 고객은 이 버전으로 업그레이드하는 것을 고려해야 합니다.
- 읽기 전용 복제본의 임시 테이블에 대해 중첩된 쿼리가 실행될 때 `ERROR 1836`을 잘못 보고하는 문제를 해결하였습니다.
- Aurora 라이터 인스턴스에서 많은 양의 쓰기 워크로드가 실행 중일 때 Aurora 리더 인스턴스의 병렬 쿼리가 중단되는 오류를 해결하였습니다.
- 대량의 쓰기 워크로드가 실행 중일 때 Binlog 마스터로 구성된 데이터베이스가 다시 시작하는 원인이 되는 문제를 해결하였습니다.
- 엔진을 다시 시작하는 중에 사용 불가 상태가 지속되는 문제를 해결하였습니다. 이를 통해 버퍼 풀 초기화 관련 문제가 해결됩니다. 이 문제는 드물게 발생하지만 지원되는 릴리스에 영향을 미칠 가능성이 있습니다.
- `information_schema.replica_host_status` 테이블에 일관성 없는 데이터를 생성한 문제를 해결하였습니다.
- 병렬 쿼리와 표준 실행 경로 간의 교착 상태로 인해 리더 노드가 간헐적으로 다시 시작하는 문제를 해결하였습니다.
- 클라이언트 연결의 수가 `max_connections` 파라미터 값을 초과할 때의 데이터베이스 안정성을 개선하였습니다.
- 지원되지 않는 DDL 및 `LOAD FROM S3` 쿼리를 차단하여 리더 인스턴스의 안정성을 개선하였습니다.

MySQL 커뮤니티 에디션 버그 픽스 통합

- 버그 #16346241 - ITEM_PARAM::QUERY_VAL_STR의 서버 충돌
- 버그 #17733850 - ITEM_NAME_CONST::ITEM_NAME_CONST()의 NAME_CONST() 충돌
- 버그 #20989615: INNODB AUTO_INCREMENT에서 동일한 값을 두 번 산출
- 버그 #20181776 - 액세스 제어에 와일드카드가 포함된 경우 가장 제한적인 호스트와 일치하지 않음
- Bug #27326796 - MYSQL이 PARS0PARS.CC 파일의 INNODB ASSERTION 실패와 충돌

- Bug #20590013 - FULLTEXT 인덱스가 있는데 이를 삭제하면 더 이상 온라인 DDL을 수행할 수 없음

Aurora MySQL 데이터베이스 엔진 업데이트 2019년 11월 25일(버전 1.21.0)(사용되지 않음)

버전: 1.21.0

Aurora MySQL 1.21.0이 정식 버전입니다. Aurora MySQL 1.* 버전은 MySQL 5.6과 호환되고 Aurora MySQL 2.* 버전은 MySQL 5.7과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 1.20.*, 1.21.*, 1.22.*, 2.01.*, 2.02.*, 2.03.*, 2.04.*, 2.05.*, 2.06.*, 2.07.*입니다. 이전 버전의 Aurora MySQL로 클러스터를 생성하려면 AWS Management Console, 또는 RDS API를 AWS CLI 통해 엔진 버전을 지정하십시오. 기존 Aurora MySQL 1.* 데이터베이스 클러스터를 Aurora MySQL 1.21.0으로 업그레이드할 수 있는 옵션이 있습니다.

> **Note**
>
> 현재이 버전은 AWS GovCloud(미국 동부) [us-gov-east-1], AWS GovCloud(미국 서부) [us-gov-west-1], 중국(닝샤) [cn-northwest-1], 아시아 태평양(홍콩) [ap-east-1], 유럽(스톡홀름) [eu-north-1], 중동(바레인) [me-south-1] AWS 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> DB 클러스터를 업그레이드하는 절차가 변경되었습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

심각한 수정 사항:

- CVE-2018-0734
- CVE-2019-2534
- CVE-2018-2612
- CVE-2017-3599
- CVE-2018-2562
- CVE-2017-3329
- CVE-2018-2696
- CVE-2015-4737

높은 우선 순위:

- 데이터베이스 크기가 64테비바이트(TiB)에 가까운 고객은 이 버전으로 업그레이드하여 Aurora 스토리지 한도에 가까운 볼륨에 영향을 미치는 안정성 버그로 인한 가동 중지를 방지할 것을 적극 권장합니다.

일반적인 안정성 수정 사항:

- Aurora 라이터 인스턴스에서 많은 양의 쓰기 워크로드가 실행 중일 때 Aurora 리더 인스턴스의 병렬 쿼리가 중단되는 오류를 해결했습니다.
- 라이터 인스턴스에 트랜잭션 커밋 트래픽이 가중되는 상황에서 트랜잭션이 장기간 실행 중이면 Aurora 리더 인스턴스의 여유 메모리가 줄어드는 문제를 해결했습니다.
- 이제 `aurora_disable_hash_join` 파라미터의 값은 데이터베이스 재시작 또는 호스트 대체 후에도 지속됩니다.
- Aurora 인스턴스의 메모리 부족 현상의 원인이 되는 전체 텍스트 검색 캐시 관련 문제를 해결했습니다. 전체 텍스트 검색을 사용하는 고객은 업그레이드해야 합니다.
- 해시 조인 기능이 활성화된 상태에서 인스턴스의 메모리가 얼마 남지 않았을 때를 대비해 데이터베이스의 안정성을 높였습니다. 해시 조인을 사용하는 고객은 업그레이드해야 합니다.
- “너무 많은 연결” 오류가 재부팅의 원인이 될 수 있는 쿼리 캐시 문제를 해결했습니다.
- 스왑 메모리 공간을 포함하여 불필요한 재부팅을 방지하도록 T2 인스턴스의 여유 메모리 계산 방식을 수정하였습니다.

MySQL 커뮤니티 에디션 버그 픽스 통합

- Bug #19929406: HANDLE_FATAL_SIGNAL (SIG=11) IN __MEMMOVE_SSSE3_BACK FROM STRING::COPY
- Bug #17059925: UNION 문에서 행 검토 값이 잘못 계산되었습니다. 이러한 오류는 성능 스키마 문 테이블(예: events_statements_current)의 `ROWS_EXAMINED` 열 값이 너무 커지는 것으로 나타났습니다.
- Bug #11827369: `SELECT ... FROM DUAL` 중첩 하위 쿼리가 포함된 일부 쿼리로 인해 어설션이 발생하였습니다.
- 버그 #16311231: `IN` 절에 XOR 작업이 포함된 `WHERE` 절에 하위 쿼리가 들어 있는 쿼리의 경우 잘못된 결과가 반환되었습니다.

Aurora MySQL 데이터베이스 엔진 업데이트 2020년 3월 5일(버전 1.20.1)(사용되지 않음)

버전: 1.20.1

Aurora MySQL 1.20.1이 정식 버전입니다. Aurora MySQL 1.* 버전은 MySQL 5.6과 호환되고 Aurora MySQL 2.* 버전은 MySQL 5.7과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 1.20.*, 1.21.*, 1.22.*, 2.01.*, 2.02.*, 2.03.*, 2.04.*, 2.05.*, 2.06.*, 2.07.*입니다. Aurora MySQL 1.* 데이터베이스의 스냅샷을 Aurora MySQL 1.20.1로 복원할 수 있습니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 RDS 콘솔, AWS CLI 또는 Amazon RDS API를 통해 엔진 버전을 지정하십시오.

> **Note**
>
> 이 버전은 현재 다음 리전에서 사용할 수 없습니다. AWS GovCloud(미국 동부) [us-gov-east-1], AWS GovCloud(미국-서부) [us-gov-west-1]. 사용 가능해지면 따로 발표할 예정입니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

높은 우선 순위:

- 인증서 교체 후 간헐적인 연결 실패 문제를 해결했습니다.
- 과중한 워크로드에서 장애 조치가 발생할 수 있는 연결 닫기 동시성과 관련된 문제를 해결했습니다.

일반적인 안정성 수정 사항:

- 중간 테이블을 내부적으로 사용하는 다중 테이블 조인 및 집계가 포함된 복잡한 쿼리를 실행하는 동안 발생하는 충돌 문제를 해결했습니다.

Aurora MySQL 데이터베이스 엔진 업데이트 2019년 11월 11일(버전 1.20.0)(사용되지 않음)

버전: 1.20.0

Aurora MySQL 1.20.0이 정식 버전입니다. Aurora MySQL 1.* 버전은 MySQL 5.6과 호환되고 Aurora MySQL 2.* 버전은 MySQL 5.7과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 1.20.*, 2.01.*, 2.02.*, 2.03.*, 2.04.*입니다. 이전 버전의 Aurora MySQL로 클러스터를 생성하려면 AWS Management Console, 또는 RDS API를 AWS CLI 통해 엔진 버전을 지정하십시오. 최고 1.19.5까지, 기존 Aurora MySQL1.* 데이터베이스 클러스터를 Aurora MySQL 1.20.0으로 업그레이드할 수 있는 옵션이 있습니다.

> **Note**
>
> 현재이 버전은 AWS GovCloud(미국 동부) [us-gov-east-1], AWS GovCloud(미국 서부) [us-gov-west-1], 중국(닝샤) [cn-northwest-1], 아시아 태평양(홍콩) [ap-east-1], 유럽(스톡홀름) [eu-north-1], 중동(바레인) [me-south-1] AWS 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> DB 클러스터를 업그레이드하는 절차가 변경되었습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

심각한 수정 사항:

- CVE-2018-0734
- CVE-2019-2534
- CVE-2018-2612
- CVE-2017-3599
- CVE-2018-2562
- CVE-2017-3329
- CVE-2018-2696
- CVE-2015-4737

높은 우선 순위:

- 데이터베이스 크기가 64테비바이트(TiB)에 가까운 고객은 이 버전으로 업그레이드하여 Aurora 스토리지 한도에 가까운 볼륨에 영향을 미치는 안정성 버그로 인한 가동 중지를 방지할 것을 적극 권장합니다.

일반적인 안정성 수정 사항:

- Aurora 라이터 인스턴스에서 많은 양의 쓰기 워크로드가 실행 중일 때 Aurora 리더 인스턴스의 병렬 쿼리가 중단되는 오류를 해결했습니다.
- 라이터 인스턴스에 트랜잭션 커밋 트래픽이 가중되는 상황에서 트랜잭션이 장기간 실행 중이면 Aurora 리더 인스턴스의 여유 메모리가 줄어드는 문제를 해결했습니다.
- 이제 `aurora_disable_hash_join` 파라미터의 값은 데이터베이스 재시작 또는 호스트 대체 후에도 지속됩니다.
- Aurora 인스턴스의 메모리 부족 현상의 원인이 되는 전체 텍스트 검색 캐시 관련 문제를 해결했습니다. 전체 텍스트 검색을 사용하는 고객은 업그레이드해야 합니다.

- 해시 조인 기능이 활성화된 상태에서 인스턴스의 메모리가 얼마 남지 않았을 때를 대비해 데이터베이스의 안정성을 높였습니다. 해시 조인을 사용하는 고객은 업그레이드해야 합니다.
- "너무 많은 연결" 오류가 재부팅의 원인이 될 수 있는 쿼리 캐시 문제를 해결했습니다.
- 스왑 메모리 공간을 포함하여 불필요한 재부팅을 방지하도록 T2 인스턴스의 여유 메모리 계산 방식을 수정하였습니다.

MySQL 커뮤니티 에디션 버그 픽스 통합

- Bug #19929406: HANDLE_FATAL_SIGNAL (SIG=11) IN __MEMMOVE_SSSE3_BACK FROM STRING::COPY
- Bug #17059925: UNION 문에서 행 검토 값이 잘못 계산되었습니다. 이러한 오류는 성능 스키마 문 테이블(예: events_statements_current)의 `ROWS_EXAMINED` 열 값이 너무 커지는 것으로 나타났습니다.
- Bug #11827369: `SELECT ... FROM DUAL` 중첩 하위 쿼리가 포함된 일부 쿼리로 인해 어설션이 발생하였습니다.
- 버그 #16311231: `IN` 절에 XOR 작업이 포함된 `WHERE` 절에 하위 쿼리가 들어 있는 쿼리의 경우 잘못된 결과가 반환되었습니다.

Aurora MySQL 데이터베이스 엔진 업데이트 2020년 3월 5일(버전 1.19.6)(사용되지 않음)

버전: 1.19.6

Aurora MySQL 1.19.6이 정식 버전입니다. Aurora MySQL 1.* 버전은 MySQL 5.6과 호환되고 Aurora MySQL 2.* 버전은 MySQL 5.7과 호환됩니다.

이 엔진 버전은 2023년 2월 28일에 사용 중지될 예정입니다. 자세한 내용은 Amazon Aurora MySQL 호환 버전 1 수명 종료 준비를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

Aurora MySQL 1.* 데이터베이스의 스냅샷을 Aurora MySQL 1.19.6으로 복원할 수 있습니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 RDS 콘솔, AWS CLI 또는 Amazon RDS API를 통해 엔진 버전을 지정하십시오.

> **Note**
>
> 이 버전은 현재 다음 리전에서 사용할 수 없습니다. AWS GovCloud(미국 동부) [us-gov-east-1], AWS GovCloud(미국-서부) [us-gov-west-1]. 사용 가능해지면 따로 발표할 예정입니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

높은 우선 순위:

- 인증서 교체 후 간헐적인 연결 실패 문제를 해결했습니다.

Aurora MySQL 데이터베이스 엔진 업데이트 2019년 9월 19일(버전 1.19.5)(사용되지 않음)

버전: 1.19.5

Aurora MySQL 1.19.5가 정식 버전입니다. Aurora MySQL 1.* 버전은 MySQL 5.6과 호환되고 Aurora MySQL 2.* 버전은 MySQL 5.7과 호환됩니다.

이 엔진 버전은 2023년 2월 28일에 사용 중지될 예정입니다. 자세한 내용은 Amazon Aurora MySQL 호환 버전 1 수명 종료 준비를 참조하세요.

현재 지원되는 Aurora MySQL 릴리스는 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* 및 3.02.*입니다.

기존 데이터베이스 클러스터를 Aurora MySQL 1.19.5으로 업그레이드할 수 있는 옵션이 있습니다. Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.1, 1.19.2의 스냅샷을 Aurora MySQL 1.19.5로 복원할 수 있습니다.

이전 버전의 Aurora MySQL을 사용하려면 AWS Management Console AWS CLI, 또는 RDS API를 통해 엔진 버전을 지정하여 새 데이터베이스 클러스터를 생성할 수 있습니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> 이 버전은 현재 유럽(런던) [eu-west-2], AWS GovCloud(미국 동부) [us-gov-east-1], AWS GovCloud(미국 서부) [us-gov-west-1], 중국(닝샤) [cn-northwest-1], 아시아 태평양(홍콩) [ap-east-1] AWS 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

> **Note**
>
> DB 클러스터를 업그레이드하는 절차가 변경되었습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

- 라이터 인스턴스에 트랜잭션 커밋 트래픽이 가중되는 상황에서 트랜잭션이 장기간 실행 중이면 Aurora 리더 인스턴스의 여유 메모리가 줄어드는 문제를 해결했습니다.
- Aurora 라이터 인스턴스에서 많은 양의 쓰기 워크로드가 실행 중일 때 Aurora 리더 인스턴스의 병렬 쿼리가 중단되는 오류를 해결했습니다.
- 이제 `aurora_disable_hash_join` 파라미터의 값은 데이터베이스 재시작 또는 호스트 대체 후에도 지속됩니다.
- Aurora 인스턴스의 메모리 부족 현상의 원인이 되는 전체 텍스트 검색 캐시 관련 문제를 해결했습니다.
- 볼륨 크기가 볼륨 한도인 64테비바이트(TiB)에 근접할 때를 대비해 복구 워크플로우가 장애 조치 없이 완료될 수 있도록 160GB의 공간을 예약하여 데이터베이스의 안정성을 높였습니다.
- 해시 조인 기능이 활성화된 상태에서 인스턴스의 메모리가 얼마 남지 않았을 때를 대비해 데이터베이스의 안정성을 높였습니다.
- 조기에 재부팅되는 현상을 유발하는 여유 메모리 계산 방식을 수정하여 T2 인스턴스의 스왑 메모리 공간을 포함하도록 하였습니다.
- “너무 많은 연결” 오류가 재부팅의 원인이 될 수 있는 쿼리 캐시 문제를 해결했습니다.

MySQL 커뮤니티 에디션 버그 픽스 통합

- CVE-2018-2696
- CVE-2015-4737
- Bug #19929406: HANDLE_FATAL_SIGNAL (SIG=11) IN __MEMMOVE_SSSE3_BACK FROM STRING::COPY
- Bug #17059925: UNION 문에서 행 검토 값이 잘못 계산되었습니다. 이러한 오류는 성능 스키마 문 테이블(예: events_statements_current)의 `ROWS_EXAMINED` 열 값이 너무 커지는 것으로 나타났습니다.
- Bug #11827369: `SELECT ... FROM DUAL` 중첩 하위 쿼리가 포함된 일부 쿼리로 인해 어설션이 발생하였습니다.
- 버그 #16311231: `IN` 절에 XOR 작업이 포함된 `WHERE` 절에 하위 쿼리가 들어 있는 쿼리의 경우 잘못된 결과가 반환되었습니다.

Aurora MySQL 데이터베이스 엔진 업데이트 2019년 6월 5일(버전 1.19.2)(사용되지 않음)

버전: 1.19.2

Aurora MySQL 1.19.2가 정식 버전입니다. 스냅샷에서 복원되는 클러스터를 포함하여 MySQL 5.6과 호환되는 새로운 Aurora MySQL 데이터베이스 클러스터는 모두 1.17.8, 1.19.0, 1.19.1 또는 1.19.2에서 생성됩니다. 기존 데이터베이스 클러스터를 Aurora MySQL 1.19.2로 업그레이드할 수 있지만 필수는 아닙니다. 이전 버전을 사용하려면 Aurora MySQL 1.14.4, Aurora MySQL 1.15.1, Aurora MySQL 1.16, Aurora MySQL 1.17.8 또는 Aurora MySQL 1.18에서 데이터베이스 클러스터를 새로 만드십시오. AWS CLI 또는 Amazon RDS API를 사용하여 엔진 버전을 지정할 수 있습니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> 이 버전은 현재 AWS GovCloud(미국 서부) [us-gov-west-1], 유럽(스톡홀름) [eu-north-1], 중국(닝샤) [cn-northwest-1], 아시아 태평양(홍콩) [ap-east-1] AWS 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

> **Note**
>
> DB 클러스터를 업그레이드하는 절차가 변경되었습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

- Amazon S3에서 Aurora로 데이터를 로드할 때 오류를 일으킬 수 있는 문제가 해결되었습니다.
- Aurora에서 Amazon S3로 데이터를 업로드할 때 오류를 일으킬 수 있는 문제가 해결되었습니다.
- 중단된 상태로 남아있는 좀비 세션을 생성한 문제가 해결되었습니다.
- 네트워크 프로토콜 관리의 오류를 처리할 때 중단된 연결로 인한 문제를 해결했습니다.
- 분할된 테이블을 처리할 때 충돌을 일으킬 수 있는 문제가 해결되었습니다.
- 트리거 생성의 binlog 복제와 관련된 문제가 수정되었습니다.

Aurora MySQL 데이터베이스 엔진 업데이트 2019년 5월 9일(버전 1.19.1)(사용되지 않음)

버전: 1.19.1

Aurora MySQL 1.19.1이 정식 버전입니다. 스냅샷에서 복원되는 클러스터를 포함하여 MySQL 5.6과 호환되는 새로운 Aurora MySQL 데이터베이스 클러스터는 모두 1.17.8, 1.19.0 또는 1.19.1에서 생성할 수 있습니다. 기존 데이터베이스 클러스터를 Aurora MySQL 1.19.1로 업그레이드할 수 있지만 필수는 아닙니다. 이전 버전을 사용하려면 Aurora MySQL 1.14.4, Aurora MySQL 1.15.1, Aurora MySQL 1.16, Aurora MySQL 1.17.8 또는 Aurora MySQL 1.18에서 데이터베이스 클러스터를 새로 만드십시오. AWS CLI 또는 Amazon RDS API를 사용하여 엔진 버전을 지정할 수 있습니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> 이 버전은 현재 AWS GovCloud(미국 서부) [us-gov-west-1] 및 중국(베이징) [cn-north-1] 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

> **Note**
>
> DB 클러스터를 업그레이드하는 절차가 변경되었습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

개선 사항

- binlog 작업자로 구성된 Aurora 인스턴스에서 문제를 일으킬 수 있는 binlog 복제의 버그를 수정했습니다.
- 특정 유형의 `ALTER TABLE` 명령을 처리할 때 발생하는 오류를 수정했습니다.
- 네트워크 프로토콜 관리 시 발생한 오류로 인해 중단된 연결과 관련된 문제를 해결했습니다.

Aurora MySQL 데이터베이스 엔진 업데이트 2019년 2월 7일(버전 1.19.0)(사용되지 않음)

버전: 1.19.0

Aurora MySQL 1.19.0이 정식 버전입니다. 스냅샷에서 복원되는 클러스터를 포함하여 MySQL 5.6과 호환되는 새로운 Aurora MySQL 데이터베이스 클러스터는 모두 1.17.8 또는 1.19.0에서 생성됩니다. 기존 데이터베이스 클러스터를 Aurora MySQL 1.19.0으로 업그레이드할 수 있지만 필수는 아닙니다. 이전 버전을 사용하려면 Aurora MySQL 1.14.4, Aurora MySQL 1.15.1, Aurora MySQL 1.16, Aurora MySQL 1.17.8 또는 Aurora MySQL 1.18.0에서 데이터베이스 클러스터를 새로 만드십시오. AWS CLI 또는 Amazon RDS API를 사용하여 엔진 버전을 지정할 수 있습니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> 이 버전은 현재 AWS GovCloud(미국 서부) [us-gov-west-1] 및 중국(베이징) [cn-north-1] 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

> **Note**
>
> DB 클러스터를 업그레이드하는 절차가 변경되었습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터의 부 버전 또는 패치 수준 업그레이드를 참조하세요.

특성

- Aurora 버전 선택기 - Aurora MySQL 1.19.0부터는 Amazon RDS 콘솔에서 MySQL 5.6과 호환되는 Aurora의 여러 버전 중에서 선택할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 AWS를 통해 Aurora MySQL 엔진 버전 확인 또는 지정을 참조하세요.

개선 사항

- Aurora 복제본의 `CHECK TABLE` 쿼리와 관련된 안정성 문제를 해결했습니다.
- 새로운 전역 사용자 변수인 `aurora_disable_hash_join`을 도입하여 해시 조인을 비활성화하였습니다.
- 다중 테이블 해시 조인 중에 출력을 생성할 때 발생하는 안정성 문제를 해결했습니다.
- 해시 조인 적용 가능성 점검 중 계획 변경으로 인해 잘못된 결과를 반환하는 문제를 해결했습니다.
- 가동 중단 제로화 패치 적용은 장기적인 실행 트랜잭션에서 지원됩니다. 이 개선 사항은 버전 1.19를 더 높은 버전으로 업그레이드할 때 효력이 발생합니다.
- 가동 중단 제로화 패치 적용은 현재 binlog가 활성화되어 있을 때 지원됩니다. 이 개선 사항은 버전 1.19를 더 높은 버전으로 업그레이드할 때 효력이 발생합니다.
- 워크로드와 관련이 없는 Aurora 복제본에서 CPU 사용률이 급증하는 원인이 되는 문제를 해결했습니다.
- 데이터베이스 재시작으로 이어지는 잠금 관리자의 교착 상태를 해결했습니다.
- Aurora 인스턴스의 안정성을 높이기 위해 잠금 관리자 구성 요소의 교착 상태를 해결했습니다.
- 잠금 관리자 구성 요소 내부에 있는 교착 상태 감지기의 안정성을 개선하였습니다.
- `INSERT`InnoDB에서 인덱스가 손상된 것을 감지한 경우 테이블에서 작업은 허용되지 않습니다.
- Fast DDL에서 안정성 문제를 해결했습니다.
- 단일 행 하위 쿼리에 대한 배치화 스캔 중 메모리 소비를 줄임으로써 Aurora 안정성을 개선하였습니다.
- 시스템 변수 `foreign_key_checks`가 0으로 설정된 상태에서 외래 키가 드롭된 후 발생한 안정성 문제를 해결하였습니다.

- `table_definition_cache` 값에 대한 사용자의 변경 사항을 잘못 재정의하는 메모리 부족 방지 기능의 문제를 해결했습니다.
- 메모리 부족 방지 기능의 안정성 문제를 해결했습니다.
- `query_time`의 `lock_time` 및 `slow_query_log`를 가비지 값으로 설정하는 문제를 해결했습니다.
- 내부적으로 문자열 콜레이션을 부적절하게 처리함으로 인해 트리거되는 병렬 쿼리 안정성 문제를 해결했습니다.
- 보조 인덱스 검색으로 인해 트리거되는 병렬 쿼리 안정성 문제를 해결했습니다.
- 다중 테이블 업데이트로 인해 트리거되는 병렬 쿼리 안정성 문제를 해결했습니다.

MySQL 커뮤니티 에디션 버그 픽스 통합

- 버그 #32917: ORPHAN TEMP-POOL 파일을 감지하여 정상적으로 처리
- 버그 #63144: 존재하지 않는 경우 테이블 생성, 메타데이터 잠금이 너무 제한적임

Aurora MySQL 데이터베이스 엔진 업데이트 2018년 9월 20일(버전 1.18.0)(사용되지 않음)

버전: 1.18.0

Aurora MySQL 1.18.0이 정식 버전입니다. 스냅샷에서 복원되는 클러스터를 포함하여, MySQL 5.6과 호환되는 새로운 Aurora MySQL 병렬 쿼리 클러스터는 모두 Aurora MySQL 1.18.0에서 생성됩니다. 기존 병렬 쿼리 클러스터를 Aurora MySQL 1.18.0으로 업그레이드할 수 있지만 필수는 아닙니다. Aurora MySQL 1.14.4, Aurora MySQL 1.15.1, Aurora MySQL 1.16 또는 Aurora MySQL 1.17.6에서 새 DB 클러스터를 생성할 수 있습니다. AWS CLI 또는 Amazon RDS API를 사용하여 엔진 버전을 지정할 수 있습니다.

Aurora MySQL 버전 1.18.0에서는 Aurora DB 클러스터의 모든 노드가 동시에 패치되는 클러스터 패치 적용 모델을 사용합니다.

> **Important**
>
> Aurora MySQL 1.18.0은 Aurora 병렬 쿼리 클러스터에만 적용됩니다. 프로비저닝된 5.6.10a 클러스터를 업그레이드하는 경우 결과 버전은 1.17.8입니다. 병렬 쿼리 5.6.10a 클러스터를 업그레이드하는 경우 결과 버전은 1.18.0입니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

특성

- 이 릴리스에서는 새 클러스터 및 복원된 스냅샷에 병렬 쿼리를 사용할 수 있습니다. Aurora MySQL 병렬 쿼리는 데이터 집약적인 쿼리 처리에 수반되는 I/O 및 컴퓨팅의 일부를 병렬화하는 최적화입니다. 병렬화되는 작업은 스토리지로부터 행 검색, 열 값 추출, 어떤 행이 `WHERE` 절 및 JOIN 절의 조건과 일치하는지 판단을 포함합니다. 이 데이터 집약적인 작업은 Aurora 분산 스토리지 계층의 여러 노드에 위임됩니다(데이터베이스 최적화 관점에서 볼 경우 아래로 밀어 내림). 병렬 쿼리가 없으면, 각 쿼리가 스캔한 모든 데이터를 Aurora MySQL 클러스터(헤드 노드) 내의 단일 노드로 가져오고 거기에서 모든 쿼리 처리를 수행합니다.
  - 병렬 쿼리 기능이 활성화되면, Aurora MySQL 엔진이 힌트 또는 테이블 속성과 같은 SQL 변경 필요 없이도 쿼리가 혜택을 얻을 수 있는 경우를 자동으로 결정합니다.

  자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL용 Parallel Query 처리를 참조하세요.
- OOM Avoidance: 이 기능은 시스템 메모리를 모니터링하고, 데이터베이스의 다양한 구성요소에서 사용한 메모리를 추적합니다. 시스템 메모리가 부족해지면 데이터베이스가 메모리 부족(OOM: Out of Memory) 상태로 실행되지 않도록 하기 위해, 추적된 다양한 구성 요소에서 메모리를 해제하는 작업 목록을 수행함으로써 데이터베이스가 다시 시작되는 문제를 방지합니다. 이 최선의 기능은 t2 인스턴스에 대해 기본적으로 활성화되며, `aurora_oom_response`라는 새 인스턴스 파라미터를 통해 다른 인스턴스 클래스에서도 활성화할 수 있습니다. 이 인스턴스 파라미터는 메모리가 부족할 때 인스턴스가 취해야 할 작업을 쉼표로 구분해 놓은 문자열을 받습니다. 유효한 작업으로는 "print", "tune", "decline", "kill_query" 등이 있으며 이러한 작업을 조합할 수 있습니다. 빈 문자열은 취해야 할 조치가 없음을 의미하므로 해당 기능을 비활성화합니다. 이 기능의 기본 작업은 "print, tune"입니다. 사용 예제:
  - "print" – 많은 양의 메모리를 사용하는 쿼리만 인쇄합니다.
  - "tune" – 내부 테이블 캐시를 조정하여 일부 메모리를 시스템으로 돌려줍니다.
  - "decline" – 인스턴스 메모리가 부족해지면 새 쿼리를 거부합니다.
  - "kill_query" – 인스턴스 메모리가 하한값 이상이 될 때까지 메모리 사용량이 많은 순서로 쿼리를 종료합니다. 데이터 정의 언어(DDL) 설명문이 종료되지 않습니다.
  - "print, tune" – "print" 및 "tune"에 대해 설명한 작업을 수행합니다.
  - "tune, decline, kill_query" – "tune", "decline", "kill_query"에 대해 설명한 작업을 수행합니다.

메모리 부족 상태 처리 및 기타 문제 해결 조언에 관한 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL 메모리 부족 문제를 참조하세요.

Aurora MySQL 데이터베이스 엔진 업데이트 2020년 3월 5일(버전 1.17.9)(사용되지 않음)

버전: 1.17.9

Aurora MySQL 1.17.9가 정식 버전입니다. Aurora MySQL 1.* 버전은 MySQL 5.6과 호환되고 Aurora MySQL 2.* 버전은 MySQL 5.7과 호환됩니다.

현재 지원되는 Aurora MySQL 릴리스는 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 1.20.*, 1.21.*, 1.22.*, 2.01.*, 2.02.*, 2.03.*, 2.04.*, 2.05.*, 2.06.*, 2.07.*입니다. Aurora MySQL 1.* 데이터베이스의 스냅샷을 Aurora MySQL 1.17.9로 복원할 수 있습니다.

이전 버전의 Aurora MySQL로 클러스터를 생성하려면 RDS 콘솔, AWS CLI 또는 Amazon RDS API를 통해 엔진 버전을 지정하십시오.

> **Note**
>
> 이 버전은 현재 다음 리전에서 사용할 수 없습니다. AWS GovCloud(미국 동부) [us-gov-east-1], AWS GovCloud(미국-서부) [us-gov-west-1]. 사용 가능해지면 따로 발표할 예정입니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

높은 우선 순위:

- 인증서 교체 후 간헐적인 연결 실패 문제를 해결했습니다.

Aurora MySQL 데이터베이스 엔진 업데이트 2019년 1월 17일(버전 1.17.8)(사용되지 않음)

버전: 1.17.8

Aurora MySQL 1.17.8이 정식 버전입니다. 스냅샷에서 복원되는 클러스터를 포함하여, MySQL 5.6과 호환되는 새로운 Aurora MySQL 데이터베이스 클러스터는 모두 Aurora MySQL 1.17.8에서 생성됩니다. 기존 데이터베이스 클러스터를 Aurora MySQL 1.17.8로 업그레이드할 수 있지만 필수는 아닙니다. 이전 버전을 사용하려면 Aurora MySQL 1.14.4, 1.15.1, 1.16, 1.17.7 등에서 데이터베이스 클러스터를 새로 만드십시오. AWS CLI 또는 Amazon RDS API를 사용하여 엔진 버전을 지정할 수 있습니다.

Aurora MySQL 버전 1.17.8에서는 Aurora DB 클러스터의 모든 노드가 동시에 패치되는 클러스터 패치 적용 모델을 사용합니다.

> **Note**
>
> 이 버전은 현재 AWS GovCloud(미국 서부) [us-gov-west-1] 및 중국(베이징) [cn-north-1] 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

- 재시작 후 Aurora 복제본에서 CPU 사용률이 증가하는 성능 문제를 수정했습니다.
- 해시 조인을 사용하는 SELECT 쿼리 관련 안정성 문제를 수정했습니다.

MySQL 커뮤니티 에디션 버그 픽스 통합

- 버그 #13418638: 존재하지 않는 경우 테이블 생성, 메타데이터 잠금이 너무 제한적임

Aurora MySQL 데이터베이스 엔진 업데이트 2018년 10월 8일(버전 1.17.7)(사용되지 않음)

버전: 1.17.7

Aurora MySQL 1.17.7이 정식 버전입니다. 스냅샷에서 복원되는 클러스터를 포함하여, MySQL 5.6과 호환되는 새로운 Aurora MySQL 데이터베이스 클러스터는 모두 Aurora MySQL 1.17.7에서 생성됩니다. 기존 데이터베이스 클러스터를 Aurora MySQL 1.17.7로 업그레이드할 수 있지만 필수는 아닙니다. 이전 버전을 사용하려면 Aurora MySQL 1.14.4, 1.15.1, 1.16, 1.17.6 등에서 데이터베이스 클러스터를 새로 만드십시오. AWS CLI 또는 Amazon RDS API를 사용하여 엔진 버전을 지정할 수 있습니다.

Aurora MySQL 버전 1.17.7에서는 Aurora DB 클러스터의 모든 노드가 동시에 패치되는 클러스터 패치 적용 모델을 사용합니다.

> **Note**
>
> 이 버전은 현재 AWS GovCloud(미국 서부) [us-gov-west-1] 및 중국(베이징) [cn-north-1] 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

- InnoDB 상태 변수 `innodb_buffer_pool_size`를 고객이 수정할 수 있도록 표시합니다.
- 장애 조치 과정에서 Aurora 클러스터에 발생하는 안정성 문제를 수정했습니다.
- `TRUNCATE` 작업 실패 후 발생하는 DDL 복구 문제를 수정하여 클러스터 가용성을 높였습니다.
- DDL 작업을 통해 트리거되는 `mysql.innodb_table_stats` 테이블 업데이트와 관련된 안정성 문제를 수정했습니다.
- DDL 연산 후 쿼리 캐시 무효화 과정에서 트리거되는 Aurora 복제본 안정성 문제를 수정했습니다.
- 백그라운드에서 정기적으로 실행되는 딕셔너리 캐시 제거 중 잘못된 메모리로 인해 트리거되는 안정성 문제를 수정했습니다.

MySQL 커뮤니티 에디션 버그 픽스 통합

- 버그 #16208542: 누락 테이블을 초래하는 외래 키 열에서 인덱스 삭제.
- 버그 #76349: add_derived_key()의 메모리 누수.
- Bug #16862316: 파티션 분할된 테이블의 경우, 인덱스 병합 사용 여부에 따라 쿼리가 다른 결과를 반환할 수 있습니다.

- Bug #17588348: 인덱스 병합 최적화(인덱스 병합 최적화 참조)를 사용하는 쿼리를 HASH로 파티션 분할된 테이블에서 실행할 경우 잘못된 결과가 반환될 수 있습니다.

Aurora MySQL 데이터베이스 엔진 업데이트 2018년 9월 6일(버전 1.17.6)(사용되지 않음)

버전: 1.17.6

Aurora MySQL 1.17.6이 정식 버전입니다. 스냅샷에서 복원되는 클러스터를 포함하여, MySQL 5.6과 호환되는 새로운 Aurora MySQL 데이터베이스 클러스터는 모두 Aurora MySQL 1.17.6에서 생성됩니다. 기존 데이터베이스 클러스터를 Aurora MySQL 1.17.6으로 업그레이드할 수 있지만 필수는 아닙니다. 이전 버전을 사용하려면 Aurora MySQL 1.14.4, 1.15.1, 1.16, 1.17.5 등에서 데이터베이스 클러스터를 새로 만드십시오. AWS CLI 또는 Amazon RDS API를 사용하여 엔진 버전을 지정할 수 있습니다.

Aurora MySQL 버전 1.17.6에서는 Aurora DB 클러스터의 모든 노드가 동시에 패치되는 클러스터 패치 적용 모델을 사용합니다.

> **Note**
>
> 이 버전은 현재 AWS GovCloud(미국 서부) [us-gov-west-1] 및 중국(베이징) [cn-north-1] 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

- Aurora Writer가 동일한 테이블에서 DDL 작업을 수행하는 동안 `SELECT` 쿼리에 대한 Aurora Reader의 안정성 문제를 수정했습니다.
- 힙/메모리 엔진을 사용하는 임시 테이블에 대한 DDL 로그 생성 및 삭제로 인해 발생하는 안정성 문제를 수정했습니다.
- Binlog 마스터에 대한 연결이 불안정할 때 DDL 문이 복제되는 경우 Binlog 작업자에서 발생하는 안정성 문제를 수정했습니다.
- 느린 쿼리 로그에 쓸 때 발생하는 안정성 문제를 수정했습니다.
- 올바르지 않은 Aurora Reader 지연 시간 정보를 표시하는 복제본 상태 테이블 문제를 수정했습니다.

MySQL 커뮤니티 에디션 버그 픽스 통합

- BINARY 열의 이름을 변경하거나 기본값을 변경한 ALTER TABLE 문의 경우, 인플레이스가 아닌 테이블 복사를 사용하여 변경이 수행되었습니다. (버그 #67141, 버그 #14735373, 버그 #69580, 버그 #17024290)
- 그룹인 정규 테이블과 파생 테이블 간의 외부 조인으로 인해 서버 종료가 발생할 수 있습니다. (버그 #16177639)

Aurora MySQL 데이터베이스 엔진 업데이트 2018년 8월 14일(버전 1.17.5)(사용되지 않음)

버전: 1.17.5

Aurora MySQL 1.17.5가 정식 버전입니다. 스냅샷에서 복원되는 클러스터를 포함하여, MySQL 5.6과 호환되는 새로운 Aurora MySQL 데이터베이스 클러스터는 모두 Aurora MySQL 1.17.5에서 생성됩니다. 기존 데이터베이스 클러스터를 Aurora MySQL 1.17.5로 업그레이드할 수 있지만 필수는 아닙니다. 이전 버전을 사용하려면 Aurora MySQL 1.14.4, 1.15.1, 1.16, 1.17.4 등에서 데이터베이스 클러스터를 새로 만드십시오. AWS CLI 또는 Amazon RDS API를 사용하여 엔진 버전을 지정할 수 있습니다.

Aurora MySQL 버전 1.17.5에서는 Aurora DB 클러스터의 모든 노드가 동시에 패치되는 클러스터 패치 적용 모델을 사용합니다.

> **Note**
>
> 이 버전은 현재 AWS GovCloud(미국 서부) [us-gov-west-1] 및 중국(베이징) [cn-north-1] 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

- 제로 가동 중지 패치 적용 기능을 사용하여 Aurora 클러스터에 패치를 적용한 후 Aurora Writer가 다시 시작될 수 있는 문제를 수정했습니다.

Aurora MySQL 데이터베이스 엔진 업데이트 2018년 8월 7일(버전 1.17.4)(사용되지 않음)

버전: 1.17.4

Aurora MySQL 1.17.4이 정식 버전입니다. 스냅샷에서 복원되는 클러스터를 포함하여, MySQL 5.6과 호환되는 새로운 Aurora MySQL 데이터베이스 클러스터는 모두 Aurora MySQL 1.17.4에서 생성됩니다. 기존 데이터베이스 클러스터를 Aurora MySQL 1.17.4로 업그레이드할 수 있지만 필수는 아닙니다. 이전 버전을 사용하려면 Aurora MySQL 1.14.4, 1.15.1, 1.16, 1.17.3 등에서 데이터베이스 클러스터를 새로 만드십시오. AWS CLI 또는 Amazon RDS API를 사용하여 엔진 버전을 지정할 수 있습니다.

Aurora MySQL 버전 1.17.4에서는 Aurora DB 클러스터의 모든 노드가 동시에 패치되는 클러스터 패치 적용 모델을 사용합니다.

> **Note**
>
> 이 버전은 현재 AWS GovCloud(미국 서부) [us-gov-west-1] 및 중국(베이징) [cn-north-1] 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

- 복제 개선:
  - binlog 레코드를 클러스터 복제본으로 전송하지 않아 네트워크 트래픽을 줄였습니다. 이 개선 사항은 기본적으로 활성화되어 있습니다.
  - 복제 메시지를 압축하여 네트워크 트래픽을 줄였습니다. 이 개선 사항은 8xlarge 및 16xlarge 인스턴스 클래스에 대해 기본적으로 활성화됩니다. 그런 대용량 인스턴스에서는 복제 메시지에 대한 높은 네트워크 트래픽을 발생하는 대용량의 쓰기 트래픽을 유지할 수 있습니다.
  - 복제본 쿼리 캐시에 대한 수정 사항
- `ORDER BY LOWER(col_name)` 데이터 정렬 중에 `utf8_bin`에서 잘못된 주문을 생성하던 문제가 해결되었습니다.
- 불안정성, 누락된 테이블을 비롯하여 DDL 문(특히, `TRUNCATE TABLE` 문)로 인해 Aurora 복제본에서 발생하던 문제를 해결했습니다.

- 스토리지 노드를 다시 시작할 때 소켓이 반개방 상태로 유지되던 문제를 해결했습니다.
- 다음과 같은 새로운 DB 클러스터 파라미터를 사용할 수 있습니다.
  - `aurora_enable_zdr` – Aurora 복제본에서 열려 있는 연결을 복제본 재시작 시 활성 상태로 유지합니다.
  - `aurora_enable_replica_log_compression` – 마스터와 Aurora 복제본 간의 네트워크 대역폭 활용률을 높이기 위해 복제 페이로드의 압축을 활성화합니다.
  - `aurora_enable_repl_bin_log_filtering` – Aurora 복제본이 마스터에서 사용할 수 없는 복제 레코드에 대해 필터링을 활성화합니다.

Aurora MySQL 데이터베이스 엔진 업데이트 2018년 6월 5일(버전 1.17.3)(사용되지 않음)

버전: 1.17.3

Aurora MySQL 1.17.3이 정식 버전입니다. 스냅샷에서 복원되는 클러스터를 포함하여, MySQL 5.6과 호환되는 새로운 Aurora MySQL 데이터베이스 클러스터는 모두 Aurora MySQL 1.17.3에서 생성됩니다. 기존 데이터베이스 클러스터를 Aurora MySQL 1.17.3으로 업그레이드할 수 있지만 필수는 아닙니다. Aurora MySQL 1.14.4, Aurora MySQL 1.15.1 또는 Aurora MySQL 1.16에서 새 DB 클러스터를 생성할 수 있습니다. AWS CLI 또는 Amazon RDS API를 사용하여 엔진 버전을 지정할 수 있습니다.

Aurora MySQL 버전 1.17.3에서는 Aurora DB 클러스터의 모든 노드가 동시에 패치되는 클러스터 패치 적용 모델을 사용합니다.

> **Note**
>
> 이 버전은 현재 AWS GovCloud(미국 서부) [us-gov-west-1] 및 중국(베이징) [cn-north-1] 리전에서 사용할 수 없습니다. 사용 가능해지면 따로 발표할 예정입니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

- 레코드를 읽는 동안 낙관적 커서 사용이 복원될 때 Aurora 복제본이 다시 시작할 수 있는 문제를 수정했습니다.

- 성능 스키마가 활성화된 MySQL 세션을 종료하려고 할 때(kill "*<## ID>*") Aurora Writer가 다시 시작하는 문제를 수정했습니다.
- 가비지 수집 임계값을 계산할 때 Aurora Writer가 다시 시작하는 문제를 수정했습니다.
- 로그 애플리케이션에서 Aurora 복제 진행 상황을 추적할 때 Aurora Writer가 가끔 다시 시작할 수 있는 문제를 수정했습니다.
- 자동 커밋이 꺼져 있을 때 잠재적으로 기한 경과 읽기가 발생할 수 있는 쿼리 캐시 문제를 수정했습니다.

Aurora MySQL 데이터베이스 엔진 업데이트 2018년 4월 27일(버전 1.17.2)(사용되지 않음)

버전: 1.17.2

Aurora MySQL 1.17.2가 정식 버전입니다. 스냅샷에서 복원되는 클러스터를 포함하여, MySQL 5.6과 호환되는 새로운 Aurora MySQL 데이터베이스 클러스터는 모두 Aurora MySQL 1.17.2에서 생성됩니다. 기존 데이터베이스 클러스터를 Aurora MySQL 1.17.2로 업그레이드할 수 있지만 필수는 아닙니다. Aurora MySQL 1.14.4, Aurora MySQL 1.15.1 또는 Aurora MySQL 1.16에서 새 DB 클러스터를 생성할 수 있습니다. AWS CLI 또는 Amazon RDS API를 사용하여 엔진 버전을 지정할 수 있습니다.

Aurora MySQL 버전 1.17.2에서는 Aurora DB 클러스터의 모든 노드가 동시에 패치되는 클러스터 패치 적용 모델을 사용합니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

- 특정 DDL 파티션 작업 동안 재시작을 초래하는 문제를 수정했습니다.
- 네이티브 Aurora MySQL AWS Lambda 함수를 통한 함수 호출 지원이 비활성화되는 문제를 수정했습니다.
- Aurora 복제본을 다시 시작하게 만드는 캐시 무효화 문제를 수정했습니다.
- 재시작을 초래하는 잠금 관리자의 문제를 수정했습니다.

Aurora MySQL 데이터베이스 엔진 업데이트 2018년 3월 23일(버전 1.17.1)(사용되지 않음)

버전: 1.17.1

Aurora MySQL 1.17.1이 정식 버전입니다. 스냅샷에서 복원되는 클러스터를 포함하여 새로운 데이터베이스 클러스터는 모두 Aurora MySQL 1.17.1에서 생성됩니다. 기존 데이터베이스 클러스터를 Aurora MySQL 1.17.1로 업그레이드할 수 있지만 필수는 아닙니다. Aurora MySQL 1.15.1, Aurora MySQL 1.16 또는 Aurora MySQL 1.17.에서 새 DB 클러스터를 생성할 수 있습니다. AWS CLI 또는 Amazon RDS API를 사용하여 엔진 버전을 지정할 수 있습니다.

Aurora MySQL 버전 1.17.1에서는 Aurora DB 클러스터의 모든 노드가 동시에 패치되는 클러스터 패치 적용 모델을 사용합니다. 이번 릴리스에서는 회귀는 물론 몇 가지 알려진 엔진 문제를 해결합니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

> **Note**
>
> 최신 버전의 Aurora MySQL 엔진에 문제가 있습니다. 1.17.1로 업그레이드한 후 엔진 버전이 `1.17`로 잘못 보고됩니다. 1.17.1로 업그레이드한 경우 AWS Management Console에서 DB 클러스터에 대한 유지 관리 열을 확인하여 업그레이드를 확인할 수 있습니다. `none`이라고 표시되면 엔진이 1.17.1로 업그레이드된 것입니다.

개선 사항

- 이진 로그가 자주 교체되는 경우 발생할 수 있는 대용량 이진 로드 인덱스 파일이 있는 상황에서 복구 시간이 더욱 길어지는 이진 로그 복원 문제를 해결했습니다.
- 분할된 테이블에 대한 비효율적인 쿼리 계획을 생성하는 쿼리 옵티마이저의 문제를 해결했습니다.
- 범위 쿼리로 인해 데이터베이스 엔진 재시작이 발생한 쿼리 옵티마이저의 문제를 해결했습니다.

Aurora MySQL 데이터베이스 엔진 업데이트 2018년 3월 13일(버전 1.17)(사용되지 않음)

버전: 1.17

Aurora MySQL 1.17이 정식 버전입니다. Aurora MySQL1.x 버전은 MySQL 5.6하고만 호환되며 MySQL 5.7과는 호환되지 않습니다. 스냅샷에서 복원되는 클러스터를 포함하여 새로운 5.6 호환 데이터베이스 클러스터는 모두 Aurora 1.17에서 생성됩니다. 기존 데이터베이스 클러스터를 Aurora 1.17로 업그레이드할 수 있지만 필수는 아닙니다. Aurora 1.14.1, Aurora 1.15.1 또는 Aurora 1.16.에서 새 DB 클러스터를 생성할 수 있습니다. AWS CLI 또는 Amazon RDS API를 사용하여 엔진 버전을 지정할 수 있습니다.

Aurora 버전 1.17에서는 Aurora DB 클러스터의 모든 노드가 동시에 패치되는 클러스터 패치 적용 모델을 사용합니다. 패치 적용 중에 최선을 다해 클라이언트 연결을 유지하도록 노력하는 제로 가동 중지 패치 적용 기능이 지원됩니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다.

제로 가동 중지 패치 적용

제로 가동 중지 패치 적용(ZDP) 기능은 최선을 다해 엔진 패치 도중 클라이언트 연결을 유지하기 위해 노력합니다. ZDP에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 제로 가동 중지 패치 적용 기능 사용을 참조하세요.

새로운 기능

- Aurora MySQL에서 이제 잠금 관리자의 메모리 사용량을 최적화하는 압축 잠금을 지원합니다. 버전 1.17부터는 랩 모드를 활성화하지 않고 이 기능을 사용할 수 있습니다.

개선 사항

- 데이터베이스가 유휴 상태일 때에도 싱글 코어가 100%의 CPU 사용률을 차지할 수 있는 코어 수가 적은 인스턴스에서 주로 발견되는 문제를 해결합니다.
- Aurora 클러스터에서 이진 로그 가져오기 성능이 개선되었습니다.
- 테이블 통계, 영구 스토리지 및 충돌 쓰기를 시도하는 Aurora 복제본 문제를 해결했습니다.
- Aurora 복제본에서 쿼리 캐시가 예상대로 작동하지 않는 문제를 해결했습니다.
- 엔진 재시작으로 이어지는 잠금 관리자의 교착 상태를 해결했습니다.
- 엔진 재시작으로 이어지는 읽기 전용, 자동 커밋 트랜잭션으로 인한 잠금 문제를 해결했습니다.
- 일부 쿼리가 감사 로그에 쓰여지지 않는 문제를 해결했습니다.

- 장애 조치 시 특정 파티션 유지 관리 작업의 복구 문제를 해결했습니다.

MySQL 버그 수정 통합

- 복제 필터가 사용되는 경우 LAST_INSERT_ID가 올바르지 않게 복제됩니다(버그 #69861).
- 쿼리가 INDEX_MERGE 설정에 따라 다른 결과를 반환합니다(버그 #16862316).
- 저장된 루틴의 쿼리 처리 재실행, 비효율적인 쿼리 계획(버그 #16346367)
- INNODB FTS: FTS_CACHE_APPEND_DELETED_DOC_IDS에 어설션합니다(버그 #18079671).
- RBT_EMPTY(INDEX_CACHE->WORDS)를 ALTER TABLE 변경 열에 어설션합니다(버그 #17536995).
- 저장점이 연관되었을 경우 INNODB Fulltext 검색으로 레코드를 찾지 못합니다(버그 #70333, 버그 #17458835).

Aurora MySQL 데이터베이스 엔진 업데이트 2017년 12월 11일(버전 1.16)(사용되지 않음)

버전: 1.16

Aurora MySQL 1.16이 정식 버전입니다. 스냅샷에서 복원되는 클러스터를 포함하여 새로운 데이터베이스 클러스터는 모두 Aurora 1.16에서 생성됩니다. 기존 데이터베이스 클러스터를 Aurora 1.16으로 업그레이드할 수 있지만 필수는 아닙니다. Aurora 1.14.1 또는 Aurora 1.15.1에서 새 DB 클러스터를 생성할 수 있습니다. AWS CLI 또는 Amazon RDS API를 사용하여 엔진 버전을 지정할 수 있습니다.

Aurora 버전 1.16에서는 Aurora DB 클러스터의 모든 노드가 동시에 패치되는 클러스터 패치 적용 모델을 사용합니다. 패치 적용 중에 최선을 다해 클라이언트 연결을 유지하도록 노력하는 제로 가동 중지 패치 적용 기능이 활성화됩니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다.

제로 가동 중지 패치 적용

제로 가동 중지 패치 적용(ZDP) 기능은 최선을 다해 엔진 패치 도중 클라이언트 연결을 유지하기 위해 노력합니다. ZDP에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 제로 가동 중지 패치 적용 기능 사용을 참조하세요.

새로운 기능

- Aurora MySQL은 이제 네이티브 함수를 통한 동기 AWS Lambda 호출을 지원합니다. `lambda_sync()`. 또한 비동기식 Lambda 호출의 기존 저장 프로시저의 대안으로 사용할 수 있는 네이티브 함수 `lambda_async()`도 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터에서 Lambda 함수 호출을 참조하세요.
- Aurora MySQL은 이제 해시 조인을 지원하여 동등 조인 쿼리의 속도를 높입니다. Aurora의 비용 기반 최적화로 해시 조인을 사용할 시점을 자동으로 결정할 수 있고, 쿼리 계획에 강제로 사용할 수도 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 해시 조인을 사용하여 대규모 Aurora MySQL 조인 쿼리 최적화를 참조하세요.
- Aurora MySQL은 이제 스캔 배치화를 지원하여 인 메모리 스캔 지향 쿼리의 속도를 크게 높입니다. 이 기능은 일괄 처리로 테이블 전체 스캔, 인덱스 전체 스캔 및 인덱스 범위 스캔의 성능을 향상시킵니다.

개선 사항

- 마스터에서 방금 삭제된 테이블에서 쿼리를 실행할 때 읽기 전용 복제본이 충돌하는 문제를 수정했습니다.
- `FULLTEXT` 인덱스 수가 아주 많은 데이터베이스 클러스터에서 라이터를 다시 시작할 때 예상보다 복구가 길어지는 문제를 수정했습니다.
- 이진 로그 플러시가 binlog 이벤트에서 `LOST_EVENTS` 인시던트를 초래하는 문제를 수정했습니다.
- 성능 스키마가 활성화될 때 스케줄러의 안정성 문제를 수정했습니다.
- 임시 테이블을 사용하는 하위 쿼리가 부분적 결과를 반환할 수 있는 문제를 수정했습니다.

MySQL 버그 수정 통합

없음

Aurora MySQL 데이터베이스 엔진 업데이트 2017년 11월 20일(버전 1.15.1)(사용되지 않음)

버전: 1.15.1

Aurora MySQL 1.15.1이 정식 버전입니다. 스냅샷에서 복원되는 클러스터를 포함하여 새로운 데이터베이스 클러스터는 모두 Aurora 1.15.1에서 생성됩니다. 기존 DB 클러스터를 Aurora 1.15.1로 업그레

이드할 수 있지만 필수는 아닙니다. Aurora 1.14.1에서 새 DB 클러스터를 생성할 수 있습니다. AWS CLI 또는 Amazon RDS API를 사용하여 엔진 버전을 지정할 수 있습니다.

Aurora 버전 1.15.1에서는 Aurora DB 클러스터의 모든 노드가 동시에 패치되는 클러스터 패치 적용 모델을 사용합니다. 패치 적용 중에 최선을 다해 클라이언트 연결을 유지하도록 노력하는 제로 가동 중지 패치 적용 기능이 활성화됩니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

제로 가동 중지 패치 적용

제로 가동 중지 패치 적용(ZDP) 기능은 최선을 다해 엔진 패치 도중 클라이언트 연결을 유지하기 위해 노력합니다. ZDP에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 제로 가동 중지 패치 적용 기능 사용을 참조하세요.

개선 사항

- 적응 세그먼트 선택기에서 같은 세그먼트를 두 번 선택해 특정 조건에서 읽기 지연 시간이 폭증하는 읽기 요청에 대한 문제가 수정되었습니다.
- 스레드 스케줄러에 대한 Aurora MySQL의 최적화로 인해 발생하는 문제를 수정했습니다. 이 문제는 느린 로그에 쓰는 동안 허위 오류를 검토하며 이때 연결된 쿼리 자체는 정상적으로 작동합니다.
- 큰 볼륨(> 5TB)에서 발생하는 읽기 전용 복제본의 안정성 문제를 수정했습니다.
- 가짜 대기 연결 수로 인해 작업자 스레드 수가 계속 증가하는 문제를 수정했습니다.
- 삽입 워크로드 중 긴 세마포어 대기를 유발하는 테이블 록 문제를 수정했습니다.
- Aurora MySQL 1.15에 포함된 다음 MySQL 버그 수정을 되돌렸습니다.
  - MySQL 인스턴스에서 "SYNC 인덱스 실행"이 지연됩니다(버그 #73816).
  - RBT_EMPTY(INDEX_CACHE->WORDS)를 ALTER TABLE 변경 열에 어설션합니다(버그 #17536995).
  - 저장점이 연관되었을 경우 InnoDB Fulltext 검색으로 레코드를 찾지 못합니다(버그 #70333).

MySQL 버그 수정 통합

없음

Aurora MySQL 데이터베이스 엔진 업데이트 2017년 10월 24일(버전 1.15)(사용되지 않음)

버전: 1.15

Aurora MySQL 1.15가 정식 버전입니다. 스냅샷에서 복원되는 클러스터를 포함하여 새로운 데이터베이스 클러스터는 모두 Aurora 1.15에서 생성됩니다. 기존 DB 클러스터를 Aurora 1.15로 업그레이드할 수 있지만 필수는 아닙니다. Aurora 1.14.1에서 새 DB 클러스터를 생성할 수 있습니다. AWS CLI 또는 Amazon RDS API를 사용하여 엔진 버전을 지정할 수 있습니다.

Aurora 버전 1.15에서는 Aurora DB 클러스터의 모든 노드가 동시에 패치되는 클러스터 패치 적용 모델을 사용합니다. 업데이트 후에는 데이터베이스를 다시 시작해야 하므로 다운타임이 20-30초간 발생할 수 있습니다. 다운타임 후에 DB 클러스터를 다시 사용할 수 있습니다. DB 클러스터가 현재 Aurora 1.14 또는 Aurora 1.14.1을 실행하고 있는 경우에는 워크로드에 따라 Aurora MySQL의 제로 가동 중지 패치 적용 기능을 통해 업그레이드 도중에도 클라이언트와 Aurora MySQL 기본 인스턴스의 연결을 유지할 수 있습니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

제로 가동 중지 패치 적용

제로 가동 중지 패치 적용(ZDP) 기능은 최선을 다해 엔진 패치 도중 클라이언트 연결을 유지하기 위해 노력합니다. ZDP에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 제로 가동 중지 패치 적용 기능 사용을 참조하세요.

새로운 기능

- 비동기식 키 미리 가져오기 – 비동기식 키 미리 가져오기(AKP)는 필요하기 전에 메모리 키를 미리 가져와서 캐싱되지 않은 인덱스 조인 성능을 향상시킬 수 있는 기능입니다. AKP가 주로 사용되는 사례로는 테이블 용량이 커질수록 인덱스 선택의 폭이 매우 제한적일 때 작은 용량의 외부 테이블과 큰 용량의 내부 테이블 사이의 인덱스 조인이 있습니다. 또한 Multi-Range Read(MRR) 인터페이스가 활성화되어 있을 때 보조-기본 인덱스를 조회하는 데도 AKP가 사용됩니다. 크기가 작아지면서 메모리 제약이 따르는 인스턴스는 경우에 따라 올바른 키 카디널리티를 지정하여 AKP를 사용할 수도 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 비동기식 키 프리페치를 사용하여 Aurora MySQL 인덱싱된 조인 쿼리 최적화를 참조하세요.

- 빠른 DDL–Aurora 1.13에서 릴리스된 기능을 기본값이 포함된 작업으로 확장했습니다. 이번 확장으로 기본값 유무에 상관없이 테이블 끝에 null 값이 허용되는 열을 추가하는 작업에도 빠른 DDL 기능이 적용됩니다. 이 기능은 여전히 Aurora 랩 모드에 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 빠른 DDL을 사용하여 Amazon Aurora에서 테이블 변경을 참조하세요.

개선 사항

- 이전에 WITHIN/CONTAINS 공간 쿼리를 최적화하는 도중 비어있는 결과 집합의 원인이 되었던 계산 오류가 수정되었습니다.
- 파라미터 그룹에서 `SHOW VARIABLE` 파라미터 값을 변경할 때마다 업데이트된 값이 표시되도록 `innodb_buffer_pool_size` 명령이 수정되었습니다.
- 적응형 해시 인덱싱이 비활성화되어 있고, 삽입할 레코드가 페이지의 첫 레코드일 때 빠른 DDL을 사용하여 변경된 테이블에 대한 대량 삽입 과정에서 기본 인스턴스의 안정성이 향상되었습니다.
- 사용자가 server_audit_events DB 클러스터 파라미터 값을 **`default`**로 설정하려고 할 때 Aurora의 안정성을 개선했습니다.
- Aurora 기본 인스턴스에서 ALTER TABLE 문을 실행한 후에도 다시 시작할 때까지 데이터베이스 문자 세트 변경 사항이 Aurora 복제본에 복제되지 않았던 문제를 수정했습니다.
- 이전에는 기본 인스턴스가 볼륨을 닫았더라도 Aurora 복제본을 등록할 수 있었던 기본 인스턴스의 경합 조건을 수정하여 안정성을 높였습니다.
- 대용량 테이블에서 인덱스를 생성하는 과정에서 인덱스 빌드 도중 동시 데이터 조작 언어(DML) 설명문을 활성화하도록 잠금 프로토콜을 변경하여 기본 인스턴스의 성능이 향상되었습니다.
- ALTER TABLE RENAME 쿼리 도중 InnoDB 메타데이터 불일치 문제가 수정되어 안정성이 향상되었습니다. 예: 테이블 t1의 열(c1, c2) 이름이 동일한 ALTER 문 내에서 주기적으로 t1(c2, c3)으로 변경되는 경우
- Aurora 복제본에 활성 워크로드가 없어서 기본 인스턴스가 응답하지 않는 시나리오에서 Aurora 복제본의 안정성이 향상되었습니다.
- Aurora 복제본이 테이블을 명시적으로 잠금 처리하여 복제 스레드가 기본 인스턴스에서 수신되는 DDL 변경 사항을 적용하지 못하도록 차단하는 시나리오에서 Aurora 복제본의 가용성이 향상되었습니다.
- 별도의 세션 2개에서 외부 키와 열을 동시에 테이블에 추가하면서 빠른 DDL이 활성화되어 있을 때 기본 인스턴스의 안정성이 향상되었습니다.
- 쓰기 작업이 지나치게 많은 워크로드에서 제거될 때까지 실행 취소 레코드 자르기를 차단함으로써 기본 인스턴스에서 제거 스레드의 안정성이 향상되었습니다.

- 테이블을 삭제하는 트랜잭션의 커밋 프로세스에서 잠금 해제 순서를 수정하여 안정성이 향상되었습니다.
- Aurora 복제본에서 DB 인스턴스가 스타트업을 완료하지 못하고 포트 3306이 이미 사용 중이라고 메시지를 표시하던 결함 문제가 수정되었습니다.
- 일부 information_schema 테이블(innodb_trx, innodb_lock, innodb_lock_waits)에 대해 SELECT 쿼리가 실행되면서 클러스터 안정성을 떨어뜨렸던 경합 조건 문제가 수정되었습니다.

MySQL 버그 수정 통합

- CREATE USER가 플러그인 및 암호 해시를 허용하지만 암호 해시는 무시합니다(버그 #78033).
- 파티션 분할 엔진은 분할된 인덱스에서 정렬된 항목을 반환할 수 있도록 여러 필드를 읽기 비트 집합에 추가합니다. 이는 불필요한 필드까지 읽으려고 하면서 조인 버퍼의 원인이 됩니다. 분할 필드를 모두 read_set에 추가하지 않는 대신 read_set에서 이미 설정된 접두사 필드를 기준으로 정렬하여 버퍼 문제를 수정하였습니다. key_cmp를 실행하는 경우 첫 번째 필드를 읽어야 하도록 DBUG_ASSERT가 추가되었습니다(버그 #16367691).
- MySQL 인스턴스에서 "SYNC 인덱스 실행"이 지연됩니다(버그 #73816).
- RBT_EMPTY(INDEX_CACHE->WORDS)를 ALTER TABLE 변경 열에 어설션합니다(버그 #17536995).
- 저장점이 연관되었을 경우 InnoDB Fulltext 검색으로 레코드를 찾지 못합니다(버그 #70333).

Aurora MySQL 데이터베이스 엔진 업데이트: 2018년 3월 13일(버전 1.14.4)(사용되지 않음)

버전: 1.14.4

Aurora MySQL 1.14.4가 정식 버전입니다. AWS CLI 또는 Amazon RDS API를 사용하고 엔진 버전을 지정하여 Aurora 1.14.4에서 새 DB 클러스터를 생성할 수 있습니다. 기존 1.14.x DB 클러스터를 Aurora 1.14.4로 업그레이드할 수 있지만 필수는 아닙니다.

Aurora 버전 1.14.4에서는 Aurora DB 클러스터의 모든 노드가 동시에 패치되는 클러스터 패치 적용 모델을 사용합니다. 패치 적용 중에 최선을 다해 클라이언트 연결을 유지하도록 노력하는 제로 가동 중지 패치 적용 기능이 지원됩니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

제로 가동 중지 패치 적용

제로 가동 중지 패치 적용(ZDP) 기능은 최선을 다해 엔진 패치 도중 클라이언트 연결을 유지하기 위해 노력합니다. ZDP에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 제로 가동 중지 패치 적용 기능 사용을 참조하세요.

새로운 기능

- Aurora MySQL이 이제 db.r4 인스턴스 클래스를 지원합니다.

개선 사항

- 대용량 binlog 이벤트를 쓸 때 `LOST_EVENTS`가 생성되는 문제를 해결했습니다.

MySQL 버그 수정 통합

- 무시할 수 있는 이벤트는 유효하지 않기 때문에 테스트 대상이 아닙니다(버그 #74683).
- NEW->OLD ASSERT FAILURE 'GTID_MODE > 0'(버그 #20436436)

Aurora MySQL 데이터베이스 엔진 업데이트: 2017년 9월 22일(버전 1.14.1)(사용되지 않음)

버전: 1.14.1

Aurora MySQL 1.14.1이 정식 버전입니다. 스냅샷에서 복원되는 클러스터를 포함하여 새로운 데이터베이스 클러스터는 모두 Aurora MySQL 1.14.1에서 생성됩니다. Aurora MySQL 1.14.1은 기존 Aurora MySQL DB 클러스터의 필수 업그레이드 버전이기도 합니다. 자세한 내용은 AWS 개발자 포럼 웹 사이트의 공지: Amazon Aurora의 필수 업그레이드 일정 확장을 참조하세요.

Aurora MySQL 버전 1.14.1에서는 Aurora MySQL DB 클러스터의 모든 노드가 동시에 패치되는 클러스터 패치 적용 모델을 사용합니다. 업데이트 후에는 데이터베이스를 다시 시작해야 하므로 다운타임이 20-30초간 발생할 수 있습니다. 다운타임 후에 DB 클러스터를 다시 사용할 수 있습니다. DB 클러스

터가 현재 버전 1.13 이상을 사용하고 있는 경우에는 워크로드에 따라 Aurora MySQL의 제로 가동 중지 패치 기능을 통해 업그레이드 도중에도 클라이언트와 Aurora MySQL 기본 인스턴스의 연결을 유지할 수 있습니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼과 Support를 통해 AWS Support를 AWS 사용할 수 있습니다.

개선 사항

- Aurora MySQL 랩 모드에 유지된 Fast DDL 기능의 안정성을 개선하기 위해 삽입 및 제거와 관련된 교착 상태를 수정했습니다.

Aurora MySQL 데이터베이스 엔진 업데이트: 2017년 8월 7일(버전 1.14)(사용되지 않음)

버전: 1.14

Aurora MySQL 1.14가 정식 버전입니다. 스냅샷에서 복원되는 클러스터를 포함하여 새로운 데이터베이스 클러스터는 모두 Aurora MySQL 1.14에서 생성됩니다. Aurora MySQL 1.14는 기존 Aurora MySQL DB 클러스터의 필수 업그레이드 버전이기도 합니다. 이전 버전의 Aurora MySQL 사용 중단 일정과 함께 별도의 공지 이메일을 발송해 드리겠습니다.

Aurora MySQL 버전 1.14에서는 Aurora DB 클러스터의 모든 노드가 동시에 패치되는 클러스터 패치 적용 모델을 사용합니다. 업데이트 후에는 데이터베이스를 다시 시작해야 하므로 다운타임이 20-30초간 발생할 수 있습니다. 다운타임 후에 DB 클러스터를 다시 사용할 수 있습니다. DB 클러스터가 현재 버전 1.13을 사용하고 있는 경우에는 워크로드에 따라 Aurora의 제로 가동 중지 패치 기능을 통해 업그레이드 도중에도 클라이언트와 Aurora 기본 인스턴스의 연결을 유지할 수 있습니다.

질문이나 우려 사항이 있는 경우 커뮤니티 포럼 및 Support를 통해 AWS Support를 AWS 사용할 수 있습니다.

제로 가동 중지 패치 적용

제로 가동 중지 패치 적용(ZDP) 기능은 최선을 다해 엔진 패치 도중 클라이언트 연결을 유지하기 위해 노력합니다. ZDP에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 제로 가동 중지 패치 적용 기능 사용을 참조하세요.

개선 사항

- 기본 인덱스가 아닌 보조 인덱스에서 레코드가 발견되었을 때 발생하는 잘못된 "레코드를 찾을 수 없음" 오류가 수정되었습니다.
- 개별 쓰기 작업의 범위가 32개 페이지를 넘는 경우 너무 강력한 디펜시브 어설션 기능(1.12에서 추가됨)으로 인해 발생할 수 있는 안정성 문제가 수정되었습니다. 이러한 상황은 예를 들어 BLOB 값이 클 때 발생할 수 있습니다.
- 테이블 스페이스 캐시와 딕셔너리 캐시의 불일치로 인한 안정성 문제가 수정되었습니다.
- 기본 인스턴스에 연결할 수 있는 최대 시도 수를 초과한 경우 Aurora 복제본이 응답하지 않는 문제가 수정되었습니다. 이제 아무런 작업도 없는 시간이 기본 인스턴스의 상태 확인에 사용되는 하트비트 시간보다 클 경우에는 Aurora 복제본이 다시 시작됩니다.
- `ALTER TABLE` 같은 명령을 실행할 때 하나의 연결 세션에서 배타적 메타데이터 잠금(MDL)을 설정하려고 하면 동시성이 매우 높아져 발생할 수 있는 라이브록(livelock) 문제가 수정되었습니다.
- 논리적/병렬 미리 읽기가 발생할 경우 Aurora 읽기 전용 복제본의 안정성 문제가 수정되었습니다.
- `LOAD FROM S3`의 두 가지 개선 사항
  1. 기존 재시도 외에 SDK 재시도를 추가로 사용함으로써 Amazon S3 제한 시간 오류를 개선하였습니다.
  2. 용량이 매우 큰 파일이나 다수의 파일을 로드할 때 캐싱을 통해 클라이언트 상태를 재사용함으로써 성능을 최적화하였습니다.
- `ALTER TABLE` 작업 시 빠른 DDL을 사용하여 다음 안정성 문제가 수정되었습니다.
  1. `ALTER TABLE` 문에 다수의 `ADD COLUMN` 명령이 있고, 열 이름이 오름차순을 따르지 않는 경우
  2. 업데이트할 열의 이름 문자열과 내부 시스템 테이블에서 가져오는 해당 이름 문자열이 서로 널 종료 문자(/0)가 다른 경우
  3. B-트리 분할 작업일 때
  4. 테이블의 기본 키가 가변 길이일 때
- 전체 텍스트 검색(FTS) 인덱스 캐시를 기본 인스턴스의 인덱스 캐시와 일치시키는 데 시간이 너무 오래 걸릴 때 Aurora 복제본의 안정성 문제를 수정했습니다. 이러한 문제는 기본 인스턴스에서 새롭게 생성된 FTS 인덱스 항목 중 대다수가 아직 디스크로 내려쓰기되지 않은 경우에 발생할 수 있습니다.
- 인덱스 생성 중 발생할 수 있는 안정성 문제가 수정되었습니다.
- 연결 세션당 메모리 사용량을 추적하는 새로운 인프라와 메모리 부족(OOM) 회피 전략을 수립하는 데 사용되는 원격 측정이 추가되었습니다.

- Aurora 복제본에서 ANALYZE TABLE이 잘못 허용되었던 문제가 수정되었습니다. 현재 이 코드는 차단되었습니다.
- 논리적 미리 읽기와 제거 사이의 경쟁 상태로 인해 드물게 데드록(deadlock)이 발생하면서 생기는 안정성 문제가 수정되었습니다.

MySQL 버그 수정 통합

- 이전에는 전체 텍스트 검색이 파생 테이블(FROM 절의 하위 쿼리)과 결합되면서 서버 종료의 원인이 되었습니다. 하지만 이제는 전체 텍스트 작업이 파생 테이블에 따라 결정되는 경우 구체화된 테이블에서 전체 텍스트 검색을 실행할 수 없다는 오류가 서버에서 발생합니다. (버그 #68751, 버그 #16539903)

Aurora MySQL 데이터베이스 엔진 업데이트: 2017년 5월 15일(버전 1.13)(사용되지 않음)

버전: 1.13

> **Note**
>
> 최초 출시 이후 Aurora MySQL 버전 1.13부터 SELECT INTO OUTFILE S3이라는 새로운 기능이 활성화되었으며, 이러한 변경을 반영하여 출시 정보가 업데이트되었습니다.

Aurora MySQL 1.13이 정식 버전입니다. 스냅샷에서 복원되는 클러스터를 포함하여 새로운 데이터베이스 클러스터는 모두 Aurora MySQL 1.13에서 생성됩니다. 기존 데이터베이스 클러스터를 Aurora MySQL 1.13으로 업그레이드할 수 있지만 필수는 아닙니다. Aurora 버전 1.13에서는 Aurora DB 클러스터의 모든 노드가 동시에 패치되는 클러스터 패치 적용 모델을 사용합니다. 패치 적용 중에 최선을 다해 클라이언트 연결을 유지하도록 노력하는 제로 가동 중지 패치 적용 기능이 활성화됩니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

제로 가동 중지 패치 적용

제로 가동 중지 패치 적용(ZDP) 기능은 최선을 다해 엔진 패치 도중 클라이언트 연결을 유지하기 위해 노력합니다. ZDP에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 제로 가동 중지 패치 적용 기능 사용을 참조하세요.

새로운 기능:

- SELECT INTO OUTFILE S3 – 이제부터 Aurora MySQL는 쿼리 결과를 Amazon S3 버킷의 파일 1개 이상에 업로드할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터에서 Amazon S3 버킷의 텍스트 파일로 데이터 저장을 참조하세요.

개선 사항:

- 복원 시간을 단축하기 위해 엔진 시작 시 CSV 형식 로그 파일 자르기를 구현했습니다. 이제 `general_log_backup`, `general_log`, `slow_log_backup` 및 `slow_log` 테이블은 데이터베이스 재시작 시 유지되지 않습니다.
- **`test`**라는 이름의 데이터베이스의 마이그레이션이 실패하는 문제를 해결했습니다.
- 올바른 잠금 세그먼트를 재사용함으로써 잠금 관리자의 가비지 수집기의 안정성을 개선했습니다.
- 고착 감지 알고리즘 동안 잘못된 어설션을 제거하여 잠금 관리자의 안정성을 개선했습니다.
- 비동기식 복제를 재활성화하고 무부하 또는 일기 전용 워크로드에서 잘못된 복제 지연이 보고되는 관련 문제를 해결했습니다. 버전 1.10에서 복제 파이프라인이 향상되었습니다. 이러한 개선 사항은 Aurora 복제본의 버퍼 캐시에 로그 스트림 업데이트를 적용하기 위한 것입니다. 이는 Aurora 복제본의 읽기 성능 및 안정성을 개선합니다.
- autocommit=OFF로 인해 서버 재부팅 시까지 예약된 이벤트가 차단되고 장기간 트랜잭션이 열린 상태로 유지되는 문제를 해결했습니다.
- 일반, 감사 및 느린 쿼리 로그가 비동기식 커밋에 의해 처리되는 쿼리를 로깅하지 못하는 문제를 해결했습니다.
- 논리적 미리 읽기(LRA) 기능의 성능을 최대 2.5배 개선했습니다. 이를 위해 B-트리의 중간 페이지 간에 미리 가져오기가 계속됩니다.
- 감사 변수가 불필요한 공간을 트리밍하는 파라미터 확인을 추가했습니다.
- SQL_CALC_FOUND_ROWS 옵션을 사용하고 FOUND_ROWS() 함수를 호출할 때 쿼리가 잘못된 결과를 반환할 수 있는, Aurora MySQL 버전 1.11에서 발생한 회귀 문제를 해결했습니다.
- 메타데이터 잠금 목록이 잘못 형성된 경우 안정성 문제를 해결했습니다.
- sql_mode가 PAD_CHAR_TO_FULL_LENGTH로 설정되고 명령 `SHOW FUNCTION STATUS WHERE Db='string'`이 실행될 때 안정성을 개선했습니다.
- Aurora 버전 업그레이드 후 잘못된 볼륨 일관성 검사로 인해 드물게 인스턴스가 나타나지 않는 경우를 해결했습니다.
- 사용자가 다수의 테이블을 보유하는 경우 Aurora Writer 성능이 저하되는, Aurora MySQL 버전 1.12에서 발생한 성능 문제를 해결했습니다.

- Aurora 라이터가 binlog 작업자로 구성되고 연결 수가 16,000에 근접할 경우의 안정성 문제를 개선했습니다.
- Aurora 마스터에서 DDL을 실행 중일 때 메타데이터 잠금을 대기하는 도중 연결이 차단될 경우 드물게 Aurora 복제본이 다시 시작하는 문제를 해결했습니다.

MySQL 버그 수정 통합

- 빈 InnoDB 테이블의 경우, 테이블이 비어 있더라도 ALTER TABLE 문을 사용하여 auto_increment 값을 낮출 수 없습니다. (버그 #69882)
- MATCH() ... 긴 문자열을 AGAINST()의 인수로 사용하는 AGAINST 쿼리가 전체 텍스트 검색 인덱스를 사용하는 InnoDB 테이블에서 실행될 경우 오류가 발생할 수 있습니다. (버그 #17640261)
- ORDER BY 및 LIMIT과 조합으로 SQL_CALC_FOUND_ROWS를 처리할 경우 FOUND_ROWS()에 잘못된 결과가 발생할 수 있습니다. (버그 #68458, 버그 #16383173)
- 외부 키가 존재할 경우 ALTER TABLE이 열의 Null 허용 여부를 변경하도록 허용하지 않습니다. (버그 #77591)

Aurora MySQL 데이터베이스 엔진 업데이트: 2017년 4월 5일(버전 1.12)(사용되지 않음)

버전: 1.12

Aurora MySQL 1.12는 현재 스냅샷으로부터의 복구를 포함하는 새 DB 클러스터 생성의 기본 설정 버전입니다.

이것은 기존 클러스터에 대한 필수 업그레이드가 아닙니다. 집합 규모의 패치를 1.11에 적용 완료한 후에 기존 클러스터를 버전 1.12로 업그레이드하는 옵션이 제공됩니다(Aurora 1.11 출시 정보 및 해당 포럼 공지사항 참조). Aurora 버전 1.12에서는 Aurora DB 클러스터의 모든 노드가 동시에 패치되는 클러스터 패치 적용 모델을 사용합니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

새로운 기능

- 빠른 DDL – Aurora MySQL을 사용하여 이제 ALTER TABLE tbl_name ADD COLUMN col_name column_definition 작업을 거의 동시에 실행할 수 있습니다. 이 작업은 테이블을 복사하거나 다른 DML 명령문에 영향을 거의 주지 않고 완료됩니다. 테이블 복사를 위해 임시 스토리지를 사용하지

않으므로 스몰 인스턴스 클래스의 라지 테이블에 대해서도 DDL 문을 유용하게 만듭니다. 현재 빠른 DDL은 테이블 끝에서 기본값 없이 null이 허용된 열에 대해서만 지원됩니다. 이 기능은 현재 Aurora lab 모드에서 사용할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 빠른 DDL을 사용하여 Amazon Aurora에서 테이블 변경을 참조하세요.

- 볼륨 상태 표시 – 새 모니터링 명령 SHOW VOLUME STATUS를 추가하여 볼륨의 노드 및 디스크 수를 표시합니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora DB 클러스터에서 볼륨 상태 표시를 참조하세요.

개선 사항

- 객체 잠금에 대해 할당된 메모리를 더욱 줄이는 압축 잠금에 대한 변경 실행. 이 개선은 lab 모드에서 사용할 수 있습니다.
- 데이터베이스가 유휴 상태이지만 `trx_active_transactions` 지표가 급속하게 감소하는 문제를 해결했습니다.
- 디스크 및 노드의 오류 시뮬레이션 시 오류 삽입 쿼리 구문과 관련된 잘못된 오류 메시지를 해결했습니다.
- 잠금 관리자의 경합 조건 및 데드 래치와 관련된 여러 문제를 해결했습니다.
- 쿼리 옵티마이저의 버퍼 오버플로우를 유발하는 문제를 해결했습니다.
- 기본 스토리지 노드의 가용 메모리가 부족한 경우 Aurora 읽기 전용 복제본의 안정성 문제를 해결했습니다.
- `wait_timeout` 파라미터 설정 후 유휴 상태의 연결이 지속되는 문제를 해결했습니다.
- 인스턴스 재부팅 후 `query_cache_size`가 예상하지 못한 값을 반환하는 문제를 해결했습니다.
- 쓰기가 스토리지로 진행되지 않는 이벤트에서 네트워크를 너무 자주 시험하는 진단 스레드의 결과로 발생하는 성능 문제를 해결했습니다.

MySQL 버그 수정 통합

- 비어 있는 상태로 인해 AUTO_INCREMENT 값이 재설정되는 동안 제거된 테이블을 재로드합니다. (버그 #21454472, 버그 #77743)
- purge_node_t 구조의 불일치로 인해 인덱스 기록이 롤백에서 발견되지 않았습니다. 이러한 불일치로 "보조 인덱스 입력 업데이트 오류", "기록을 제거할 수 없음", "삭제 표시되지 않은 보조 인덱스 입력 제거를 시도함" 등의 경고 및 오류 메시지가 표시되었습니다. (버그 #19138298, 버그 #70214, 버그 #21126772, 버그 #21065746)

- qsort 작업에 대한 스택 크기 계산을 잘못하면 스택 오버플로우가 발생합니다. (버그 #73979)
- 롤백 시 인덱스에서 기록이 발견되지 않았습니다. (버그 #70214, 버그 #72419)
- 업데이트 CURRENT_TIMESTAMP의 ALTER TABLE 추가 열 TIMESTAMP가 ZERO-datas 삽입(버그 #17392)

Aurora MySQL 데이터베이스 엔진 업데이트: 2017년 2월 23일(버전 1.11)(사용되지 않음)

버전: 1.11

출시 후 빠른 시일 안에 모든 Aurora MySQL DB 클러스터를 최신 버전으로 패치할 예정입니다. DB 클러스터는 레거시 프로시저를 이용해 패치되며 다운타임은 약 5~30초입니다.

패치 작업은 사용자가 각 데이터베이스 인스턴스에 지정한 시스템 유지 관리 기간에 진행됩니다. 이 기간은 AWS Management Console을 사용하여 확인하거나 변경할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

또는 DB 클러스터를 AWS Management Console 선택하고 클러스터 작업을 선택한 다음 지금 업그레이드를 선택하여에서 패치를 즉시 적용할 수 있습니다.

Aurora MySQL 버전 1.11에서는 Aurora DB 클러스터의 모든 노드가 동시에 패치되는 클러스터 패치 적용 모델을 사용합니다.

새로운 기능

- LOAD DATA FROM S3에 대한 MANIFEST 옵션 – 버전 1.8에서 LOAD DATA FROM S3이 제공되었습니다. 이 명령에 대한 옵션이 확장되어, Amazon S3에서 Aurora DB 클러스터에 로드할 때 매니페스트 파일을 사용하여 파일 목록을 지정할 수 있습니다. 따라서 FILE 옵션을 이용해 단일 파일에서 데이터를 로드하거나 PREFIX 옵션을 이용해 위치와 접두사가 같은 다수 파일에서 데이터를 로드하는 방식과는 달리, 하나 이상의 위치에 있는 특정 파일에서 데이터를 쉽게 불러올 수 있습니다. 메니페스트 파일 양식은 Amazon Redshift에서 사용하는 양식과 같습니다. LOAD DATA FROM S3를 MANIFEST 옵션과 함께 사용하는 방법에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 매니페스트 파일을 이용해 로드할 데이터 파일 지정을 참조하세요.
- 공간 인덱싱 기본 사용 – 이 기능은 버전 1.10의 랩 모드에서 공개되었으며, 이제 기본적으로 활성화됩니다. 공간 인덱싱은 공간 데이터를 사용하는 쿼리를 위한 대규모 데이터 세트에서의 쿼리 성능을 향상시킵니다. 공간 인덱싱 사용에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL 및 지형 정보 데이터를 참조하세요.

- 고급 감사 타이밍 변경 – 이 기능은 감사 데이터베이스 활동을 위한 고성능 시설을 제공하고자 버전 1.10.1에서 공개되었습니다. 이 버전에서는 감사 로그 타임스탬프의 정확성이 1초에서 1마이크로초로 변경되었습니다. 더욱 정확한 타임스탬프는 감사 이벤트 발생 시 상황을 파악할 때 도움이 됩니다. 감사에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터에서 고급 감사 사용을 참조하세요.

개선 사항

- `thread_handling` 파라미터를 수정해 Aurora의 스레드 풀이 지원하는 유일한 모델인 **`multiple-connections-per-thread`**에만 설정할 수 있게 했습니다.
- `buffer_pool_size` 또는 `query_cache_size` 파라미터를 DB 클러스터의 총 메모리보다 크게 설정하면 발생하는 문제를 수정했습니다. 이런 상황이 되면, Aurora은 수정된 파라미터를 기본값으로 설정하기 때문에 DB 클러스터를 재시작하면 충돌하지 않게 됩니다.
- 테이블이 다른 거래에서 무효가 되면 쿼리 캐시에서 오래된 읽기 결과가 나오는 문제를 수정했습니다.
- 삭제 표시한 binlog 파일이 바로 삭제되도록 조정되었습니다.
- **`tmp`**라는 이름으로 생성된 데이터베이스가 단기 스토리지에 저장된 시스템 데이터베이스로 처리되며 Aurora 휘발성 스토리지에서 유지되지 않는 문제가 수정되었습니다.
- SHOW TABLES가 특정 내부 시스템 테이블을 배제하도록 수정했습니다. 이러한 변경은 mysqldump가 SHOW TABLES에 있는 모든 파일을 잠그고, 그 결과 내부 시스템 테이블에 대한 쓰기가 금지되어 불필요한 페일오버가 발생하는 일을 막는 데 도움이 됩니다.
- 인수가 InnoDB 테이블의 열인 함수를 불러오는 쿼리에서 임시 테이블을 생성하면 Aurora 복제본이 재시작되는 문제가 수정되었습니다.
- Aurora 복제본 노드에서 메타데이터 잠금 충돌이 발생해 Aurora 복제본이 기본 DB 클러스터 뒤로 밀려 재시작하게 되는 문제가 수정되었습니다.
- 리더 노드의 복제 파이프라인에 있는 데드 래치 때문에 Aurora 복제본이 뒤로 밀려 재시작하게 되는 문제가 수정되었습니다.
- 1테라바이트(TB)가 넘는 암호화 볼륨에서 Aurora 복제본 랙이 과도해지는 문제가 수정되었습니다.
- 시스템 시간 읽는 방식을 개선해 Aurora 복제본 데드 래치 감지를 개선했습니다.
- 저자가 등록을 해제하면 Aurora 복제본이 한 번이 아닌 두 번 재시작하는 문제가 수정되었습니다.
- 임시 통계 때문에 고유하지 않은 인덱스 열에서 통계상의 불일치가 발생하면 Aurora 복제본에서 쿼리 성능이 느려지는 문제가 수정되었습니다.
- Aurora 복제가 관련 쿼리를 처리하고 있을 때 Aurora 복제본에서 DDL 인수를 복제하면 Aurora 복제본이 충돌하는 문제가 수정되었습니다.

- 버전 1.10에서 도입한 복제 파이프라인 개선 기본 설정을 활성화에서 비활성화로 변경했습니다. 이 개선 사항은 로그 스트림 업데이트를 Aurora 복제본의 버퍼 캐시에 적용하고자 도입되었습니다. Aurora 복제본의 읽기 성능과 안정성 개선에는 도움이 되지만, 특정 워크로드의 복제본 지연 시간을 높입니다.
- 진행 중인 DDL 인수와 대기 중인 Parallel Read Ahead가 같은 테이블에서 동시에 발생하면 DDL 거래 실행 단계에서 어설션 오류가 발생하는 문제가 수정되었습니다.
- DB 클러스터 재시작 후에도 존재할 수 있도록 일반 로그와 느린 쿼리 로그를 개선했습니다.
- ACL 모듈의 메모리 소비를 줄여 특정 장기 실행 쿼리의 메모리 부족 문제를 수정했습니다.
- 테이블에 비공간 인덱스가 있고, 쿼리에 공간 술어가 있으며, 플래너가 공간 조건을 인덱스에 부정확하게 밀어 넣으면 발생하는 재시작 문제를 수정했습니다.
- (LOB 같은) 외부에 저장된 초대형 지형 공간 객체의 삭제, 업데이트 또는 소거가 있으면 DB 클러스터가 재시작하는 문제가 수정되었습니다.
- ALTER SYSTEM SIMULATE … FOR INTERVAL을 이용한 고장 시뮬레이션이 제대로 작동하지 않는 문제가 수정되었습니다.
- 잠금 관리자에서 잘못된 불변량에 유효하지 않은 어설션이 적용되어 발생하는 안정성 문제가 수정되었습니다.
- 버전 1.10에서 도입한 다음과 같은 두 가지 InnoDB 전체 텍스트 검색 개선이 비활성화되었습니다. 일부 까다로운 워크로드에서 안정성 문제가 발생하기 때문입니다.
  - 전체 텍스트 검색 인덱스 캐시 복제 속도 개선을 위해 Aurora 복제본에 대한 읽기 요청 후에만 캐시를 업데이트합니다.
  - FTS 캐시가 디스크에 동기화될 때 MySQL 쿼리가 너무 오래 중지되지 않도록, 캐시 크기가 전체 크기의 10%를 넘어서자마자 별도 스레드에 캐시 동기화 작업을 오프로드합니다. (버그 #22516559, #73816)

MySQL 버그 수정 통합

- ALTER 테이블 DROP 외래 키를 다른 DROP 연산과 동시에 실행하면 테이블이 사라집니다. (버그 #16095573)
- ORDER BY를 사용한 일부 INFORMATION_SCHEMA 쿼리가 예전처럼 파일 정렬 최적화를 사용하지 않습니다. (버그 #16423536)
- FOUND_ROWS ()가 잘못된 테이블 행 수를 반환합니다 (버그 #68458)
- 임시 테이블을 너무 많이 열면 오류가 발생하는 대신 서버가 고장납니다. (버그 #18948649)

Aurora MySQL 데이터베이스 엔진 업데이트: 2017년 1월 12일(버전 1.10.1)(사용되지 않음)

버전: 1.10.1

Aurora MySQL의 버전 1.10.1은 옵트인 버전이며 데이터베이스 인스턴스를 패치하는 데 사용되지 않습니다. 새 Aurora 인스턴스를 만들고 기존 인스턴스를 업그레이드하는 데 사용할 수 있습니다. Amazon RDS 콘솔에서 클러스터를 선택하고 클러스터 작업을 선택한 후 지금 업데이트를 선택하여 패치를 적용할 수 있습니다. 패치 적용 시 데이터베이스 재시작이 필요하여, 일반적으로 5~30초간 가동 중지 후 DB 클러스터 사용을 재개할 수 있습니다. 이 패치는 Aurora 클러스터의 모든 노드가 동시에 패치되는 클러스터 패치 적용 모델을 사용합니다.

새로운 기능

- 고급 감사 – Aurora MySQL은 데이터베이스 활동을 감사하는 데 사용할 수 있는 고성능 고급 감사 기능을 제공합니다. 고급 감사 활성화 및 사용에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL DB 클러스터에서 고급 감사 사용을 참조하세요.

개선 사항

- 동일한 문에서 열을 만들고 해당 열에 인덱스를 추가할 때 공간 인덱싱 문제를 수정했습니다.
- DB 클러스터 재시작 시 공간 통계가 지속되지 않는 문제를 수정했습니다.

Aurora MySQL 데이터베이스 엔진 업데이트: 2016년 12월 14일(버전 1.10)(사용되지 않음)

버전: 1.10

새로운 기능

- 가동 중지 없는 패치 – 이 기능을 사용하면 중단 시간 없이 DB 인스턴스에 패치를 적용할 수 있습니다. 즉 클라이언트 애플리케이션의 접속을 끊거나 데이터베이스를 다시 시작하는 일 없이 애플리케이션 업그레이드가 수행됩니다. 이러한 접근 방식은 유지 관리 기간 동안 Aurora DB 클러스터의 가용성을 향상시킵니다. 성능 스키마에서 그와 같은 임시 데이터는 업그레이드 과정 중에 재설정된다는 점에 유의하십시오. 이 기능은 사용자 시작 패치뿐 아니라 유지 관리 기간 중 서비스 제공 패치에도 적용됩니다.

패치가 시작되면 해당 서비스에서는 열린 잠금, 트랜잭션 또는 임시 테이블이 없는지 확인한 다음, 적당한 기간 동안 대기하여 데이터베이스에 패치가 적용되어 다시 시작할 수 있도록 합니다. 패치 작업이 진행되는 동안(약 5초) 처리량이 떨어지기는 하지만 애플리케이션 세션은 보존됩니다. 적당한 기간을 찾을 수 없는 경우, 패치 작업은 표준 패치 동작으로 디폴트됩니다.

가동 중지 없는 패치는 다음 설명과 같은 일정한 한계 내에서 최대 한도로 이루어집니다.

- 이 기능은 현재 단일 노드 DB 클러스터 또는 다중 노드 DB 클러스터의 라이터 인스턴스에 대한 패치 작업에 사용할 수 있습니다.
- SSL 연결을 이 기능과 함께 사용하는 것은 지원하지 않습니다. 활성화된 SSL 연결이 있는 경우, Amazon Aurora MySQL은 가동 중지 없는 패치를 수행하는 대신에 SSL 연결이 종료되었는지 주기적으로 확인합니다. 연결이 종료되었다면 가동 중지 없는 패치가 수행됩니다. SSL 연결이 2초 이상 지속되는 경우, 가동 중지가 있는 표준 패치 작업이 수행됩니다.
- 이 기능은 Aurora 릴리스 1.10 이상에서 사용할 수 있습니다. 앞으로 우리는 가동 중지 없는 패치를 사용하여 적용할 수 없는 릴리스나 패치에 대해 알아볼 것입니다.
- 이 기능은 바이너리 로깅에 기반을 둔 복제가 활성화되어 있는 경우에는 해당되지 않습니다.

- 공간 인덱싱 – 공간 인덱싱은 공간 데이터를 사용하는 쿼리를 위한 대규모 데이터 세트에서의 쿼리 성능을 향상시킵니다. 공간 인덱싱 사용에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL 및 지형 정보 데이터를 참조하세요.

  이 기능은 기본적으로 비활성화되며 Aurora 랩 모드를 설정하여 활성화할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL 랩 모드를 참조하세요.
- 복제 파이프라인 개선 – Aurora MySQL은 이제 개선된 메커니즘을 사용하여 Aurora Replica의 버퍼 캐시에 로그 스트림 업데이트를 적용합니다. 이 기능은 복제본에 상당한 읽기 로드가 있을 뿐 아니라 마스터에도 쓰기 로드가 심할 때 Aurora 복제본의 읽기 성능 및 안정성을 향상시킵니다. 이 기능은 기본적으로 활성화되어 있습니다.
- 캐시된 읽기의 워크로드에 대한 처리량 향상 – Aurora MySQL는 이제 래치 프리 동시 알고리즘을 사용하여 읽기 뷰를 실행합니다. 이에 따라 버퍼 캐시가 처리하는 읽기 쿼리에 대한 처리량이 향상됩니다. 이를 포함한 여러 개선점 덕분에 Amazon Aurora MySQL는 SysBench 선택 전용 워크로드에 대한 MySQL 5.7의 초당 164K 읽기에 비해 초당 최대 625K 읽기의 처리량을 달성할 수 있습니다.
- 핫 행 경합이 있는 워크로드의 처리량 향상 – Aurora MySQL는 특히 핫 페이지 경합이 있는 경우(즉 같은 페이지에서 행에 대해 여러 트랜잭션이 경합하는 경우) 성능을 향상시키는 새로운 잠금 해제 알고리즘을 사용합니다. 이는 TPC-C 벤치마크를 사용한 테스트에서 MySQL 5.7에 비해 분당 트랜잭션 처리량이 최대 16배까지 향상되는 결과를 나타날 수 있습니다. 이 기능은 기본적으로 비활성화되며 Aurora 랩 모드를 설정하여 활성화할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL 랩 모드를 참조하세요.

개선 사항

- 전체 텍스트 검색 인덱스 캐시 복제 속도는 Aurora 복제본에 대한 읽기 요청 후에만 캐시를 업데이트하여 향상시킬 수 있습니다. 이러한 방식을 통해 복제 스레드가 디스크에서 읽지 못하도록 방지합니다.
- 데이터베이스 이름 또는 테이블 이름에 특수 문자가 있는 테이블에 대한 Aurora 복제본에서 사전 캐시 무효화가 작동하지 않는 문제를 해결하였습니다.
- 스토리지 열 관리가 활성화된 상태에서 분산 스토리지 노드에 대한 데이터 마이그레이션 도중 발생하는 `STUCK IO` 문제를 해결하였습니다.
- 트랜잭션을 롤백 또는 커밋할 준비를 할 때 트랜잭션 잠금 대기 스레드에 대한 어설션 점검이 실패하는 경우 잠금 관리자에서 발생하는 문제를 해결하였습니다.
- 참조 개수를 사전 테이블 항목에 올바르게 업데이트하여 손상된 사전 테이블을 열 때 발생하는 문제를 해결하였습니다.
- DB 클러스터 최소 읽기 포인트를 느린 Aurora 복제본이 보류할 수 있는 버그를 수정하였습니다.
- 쿼리 캐시의 잠재적 메모리 누수 문제를 해결하였습니다.
- 저장된 프로시저의 `IF` 문에서 쿼리가 사용될 때 Aurora 복제본이 테이블에 행 수준 잠금을 배치하는 버그를 수정하였습니다.

MySQL 버그 수정 통합

- 파생된 테이블의 UNION이 '1=0/false' 절이 있는 잘못된 결과를 반환합니다. (버그 #69471)
- 저장된 프로시저 2차 실행 시 ITEM_FUNC_GROUP_CONCAT::FIX_FIELDS에서 서버가 충돌합니다. (버그 #20755389)
- 캐시 크기가 전체 크기의 10%를 넘어서자마자 별도 스레드에 캐시 동기화 작업을 오프로드하여 FTS 캐시 동기화 중에 MySQL 쿼리가 너무 오래 중지되는 일을 방지합니다. (버그 #22516559, #73816)

Aurora MySQL 데이터베이스 엔진 업데이트: 2016년 11월 10일(버전 1.9.0, 1.9.1)(사용되지 않음)

버전: 1.9.0, 1.9.1

새로운 기능

- 향상된 인덱스 빌드 – 이제 상향식으로 인덱스를 빌드하여 보조 인덱스를 빌드하므로 불필요한 페이지 분할이 방지됩니다. 따라서 인덱스를 만들거나 테이블을 다시 빌드하는 데 필요한 시간을 75%까지 줄일 수 있습니다.(`db.r3.8xlarge` DB 인스턴스 클래스 기준). 이 기능은 Aurora MySQL 버전 1.7의 랩 모드에 있었으며 Aurora 버전 1.9 이상에서는 기본값으로 활성화되어 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL 랩 모드를 참조하세요.
- 압축 잠금(랩 모드) – 실행 시 잠금 관리자가 소모하는 메모리 양이 66%까지 대폭 감소합니다. 메모리 부족 예외가 발생하는 일 없이 더 많은 행 잠금을 얻을 수 있습니다. 이 기능은 기본적으로 비활성화되며 Aurora 랩 모드를 설정하여 활성화할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL 랩 모드를 참조하세요.
- 성능 스키마 – Aurora MySQL에 이제 성능 영향을 최소화한 성능 스키마 지원이 포함됩니다. SysBench를 이용한 자사 테스트에서 성능 스키마 이용 시 MySQL 성능이 60%까지 저하될 수 있습니다.

  Aurora DB 클러스터의 SysBench 테스트에서는 MySQL보다 4배 적은 성능 영향을 보여주었습니다. `db.r3.8xlarge` DB 인스턴스 클래스를 실행한 결과 성능 스키마를 활성화했음에도 100K SQL 쓰기/초, 550K 이상의 SQL 읽기/초였습니다.
- 핫 행 경합 개선 – 이 기능은 다량의 연결에 의해 핫 행이 소량 액세스할 때 CPU 활용을 떨어뜨리고 처리량을 늘립니다. 또한 핫 행 경합이 있을 때 `error 188`을 제거합니다.
- 메모리 부족 취급 개선 – 필수적이지 않은 잠금 SQL 명령문을 실행하고 예약된 메모리 풀을 넘어갈 때, Aurora가 해당 SQL 명령문을 강제로 롤백합니다. 이 기능은 메모리를 비우고 메모리 부족 예외로 인한 엔진 충돌을 예방합니다.
- 스마트 읽기 선택기 – 실행 시 모든 읽기 작업에 대하여 각기 다른 세그먼트 중에서 최적의 스토리지 세그먼트를 선택함으로써 읽기 지연을 개선하며, 이로써 읽기 처리량이 개선됩니다. SysBench 테스트 결과 쓰기 워크로드 성능이 27%까지 올라갔습니다.

개선 사항

- 엔진 시동 시 Aurora 복제본이 잠금이 공유된 곳을 발견한 경우의 문제를 수정했습니다.
- 제거 시스템에서 뷰 포인터 읽기가 NULL인 경우 Aurora 복제본에서 잠재적 충돌을 수정했습니다.

Aurora MySQL 데이터베이스 엔진 업데이트: 2016년 10월 26일(버전 1.8.1)(사용되지 않음)

버전: 1.8.1

개선 사항

- 트리거를 이용해 AWS Lambda 를 호출하는 BULK INSERT 절차가 실패하는 문제를 수정했습니다.
- 자동 커밋이 전역적으로 꺼진 경우 카탈로그 마이그레이션이 실패하는 문제를 수정했습니다.
- SSL을 이용해 Aurora로 연결이 실패하는 문제를 해결하였고 LogJam 공격에 대처하도록 Diffie-Hellman 그룹을 개선했습니다.

MySQL 버그 수정 통합

- LogJam 문제 때문에 OpenSSL이 Diffie-Hellman 키 길이 파라미터를 변경했습니다. (버그 #18367167)

Aurora MySQL 데이터베이스 엔진 업데이트: 2016년 10월 18일(버전 1.8)(사용되지 않음)

버전: 1.8

새로운 기능

- AWS Lambda 통합 - 이제 `mysql.lambda_async` 프로시저를 사용하여 Aurora DB 클러스터에서 AWS Lambda 함수를 비동기식으로 호출할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터에서 Lambda 함수 호출을 참조하세요.
- Amazon S3에서 데이터 로드 – 이제 `LOAD DATA FROM S3` 또는 `LOAD XML FROM S3` 명령을 사용하여 Amazon S3 버킷의 텍스트 또는 XML 파일을 Aurora DB 클러스터로 로드할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷의 텍스트 파일에서 Amazon Aurora MySQL DB 클러스터로 데이터 로드를 참조하세요.
- 카탈로그 마이그레이션 – 이제 Aurora에서 버전 관리를 지원하기 위해 클러스터 볼륨에 카탈로그 메타데이터를 유지합니다. 따라서 버전 간이나 복원 시에 카탈로그를 원활하게 마이그레이션할 수 있습니다.

- 클러스터 수준 유지 관리 및 패치 적용 – 이제 Aurora에서 전체 DB 클러스터에 대한 유지 관리 업데이트를 관리합니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

개선 사항

- 인플라이트 DDL 테이블에 대한 메타데이터 잠금을 허용하지 않을 경우 Aurora 복제본이 충돌하는 문제를 해결했습니다.
- `log_output=TABLE`일 때 느린 로그 및 일반 로그 CSV 파일의 로테이션을 간소화하기 위해 Aurora 복제본에 비 InnoDB 테이블 수정을 허용했습니다.
- 기본 인스턴스에서 Aurora 복제본로 통계로 업데이트할 때 시간 지연을 해결했습니다. 이 수정을 적용하지 않을 경우 Aurora 복제본의 통계가 기본 인스턴스의 통계와 동기화될 수 없으므로 Aurora 복제본에 미달된 다른 쿼리 계획이 생성될 수 있습니다.
- Aurora 복제본에서 잠금을 획득하지 못하는 경합 조건을 해결했습니다.
- 기본 인스턴스를 등록 또는 등록 취소하는 Aurora 복제본에서 장애가 발생할 수 있는 경합 시나리오를 해결했습니다.
- 볼륨을 열거나 닫을 때 `db.r3.large` 인스턴스에서 교착 상태가 발생할 수 있는 경합 조건을 해결했습니다.
- Aurora 분산 스토리지 서비스에서 대량 쓰기 워크로드와 장애가 결합되어 발생할 수 있는 메모리 부족 문제를 해결했습니다.
- 장기 실행 트랜잭션이 존재할 때 제거 스레드 회전으로 인한 높은 CPU 사용 문제를 해결했습니다.
- 과도한 로드 조건에서 잠금에 대한 정보를 가져오기 위해 정보 스키마 쿼리를 실행할 때 발생하는 문제를 해결했습니다.
- 경합으로 인해 스토리지 노드에 대한 Aurora 쓰기가 중지되었다가 재시작/장애 조치되는 진단 프로세스 문제를 해결했습니다.
- `CREATE TABLE [if not exists]` 문을 처리하는 동안 충돌이 발생할 경우 충돌 복구 중에 성공적으로 생성된 테이블이 삭제될 수 있는 조건을 해결했습니다.
- 카탈로그 완화를 사용하여 일반 로그와 느린 로그를 디스크에 저장하지 않을 경우 로그 순환 절차가 중단되는 문제를 해결했습니다.
- 사용자가 사용자 정의 함수 내에서 임시 테이블을 만든 다음 선택한 쿼리 목록에서 사용자 정의 함수를 사용할 때 충돌이 발생하는 문제를 해결했습니다.
- GTID 이벤트를 재생할 때 발생하는 충돌을 해결했습니다. GTID는 Aurora MySQL에서 지원되지 않습니다.

MySQL 버그 수정 통합:

- 여러 인덱스를 포함하는 하나의 열에 모든 인덱스를 끌어서 놓은 경우 외래 키 제약 조건에서 인덱스를 요구할 때 InnoDB에서 DROP INDEX 작업을 차단하지 못합니다. (버그 #16896810)
- 외래 키 제약 조건 추가 충돌을 해결합니다. (버그 #16413976)
- 저장 프로시저에서 커서를 가져오는 동시에 테이블을 분석하거나 플러시할 때 발생하는 충돌을 해결했습니다. (버그 #18158639)
- 사용자가 테이블에서 AUTO_INCREMENT 값을 최대 자동 증분 열 값보다 작게 변경할 때 발생하는 자동 증분 버그를 해결했습니다. (버그 #16310273)

Aurora MySQL 데이터베이스 엔진 업데이트: 2016년 9월 20일(버전 1.7.1)(사용되지 않음)

버전: 1.7.1

개선 사항

- InnoDB 전체 텍스트 검색 캐시가 가득 차 있을 때 Aurora 복제본이 중단되는 문제를 해결합니다.
- 스레드 풀의 작업자 스레드가 스스로를 기다릴 때 데이터베이스 엔진이 중단되는 문제를 해결합니다.
- 테이블의 메타데이터 잠금으로 인해 교착 상태가 발생할 때 Aurora 복제본이 중단되는 문제를 해결합니다.
- 스레드 풀의 두 작업자 스레드 간 경합 조건으로 인해 데이터베이스 엔진이 중단되는 문제를 해결합니다.
- 과도한 로드 조건에서 분산 스토리지 하위 시스템에 대한 쓰기 작업의 진전 상태를 모니터링 에이전트가 감지하지 못하는 경우 발생하는 불필요한 장애 조치의 문제를 해결합니다.

Aurora MySQL 데이터베이스 엔진 업데이트: 2016년 8월 30일(버전 1.7.0)(사용되지 않음)

버전: 1.7.0

새로운 기능

- NUMA 인식 스케줄러 – Aurora MySQL 엔진의 작업 스케줄러가 이제 NUMA(Non-Uniform Memory Access)를 인식합니다. 이로 인해 `db.r3.8xlarge` DB 인스턴스 클래스에 대한 처리량 성능이 향상되어 CPU 간 소켓 경합이 최소화됩니다.
- 병렬 미리읽기가 백그라운드에서 비동기적으로 작동 – 전용 스레드를 사용하여 스레드 경합을 줄임으로써 성능을 개선하도록 병렬 미리읽기 기능을 수정했습니다.
- 향상된 인덱스 빌드(랩 모드) – 이제 상향식으로 인덱스를 빌드하여 보조 인덱스를 빌드하므로 불필요한 페이지 분할이 방지됩니다. 따라서 인덱스를 만들거나 테이블을 다시 빌드하는 데 필요한 시간을 줄일 수 있습니다. 이 기능은 기본적으로 비활성화되며 Aurora 랩 모드를 설정하여 활성화할 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL 랩 모드를 참조하세요.

개선 사항

- 인스턴스에 대해 요청되는 연결 수가 급증할 경우 연결을 설정하는 데 오래 걸리던 문제를 해결했습니다.
- ALTER TABLE을 InnoDB를 사용하지 않는 분할된 테이블에서 실행할 경우 충돌이 발생하던 문제를 해결했습니다.
- 높은 쓰기 워크로드로 인해 장애 조치가 발생할 수 있던 문제를 해결했습니다.
- RENAME TABLE을 분할된 테이블에서 실행할 경우 오류가 발생하던 잘못된 어설션을 해결했습니다.
- 삽입 중심 워크로드 중에 트랜잭션을 롤백할 때 안정성이 향상되었습니다.
- 전체 텍스트 검색 인덱스가 Aurora 복제본에서 실행되지 않던 문제를 해결했습니다.

MySQL 버그 수정 통합

- LOCK_grant 잠금을 분할하여 확장성을 향상했습니다. (포트 WL #8355)
- 저장 프로시저의 SELECT에 커서를 두면 segfault가 발생합니다. (포트 버그 #16499751)
- MySQL에서는 일부 특수한 경우에 잘못된 결과를 제공합니다. (버그 #11751794)
- #11751794 버그에 대한 패치에 의해 GET_SEL_ARG_FOR_KEYPART에서– 충돌이 발생합니다. (버그 #16208709)
- GROUP BY를 통한 간단한 쿼리에 대해 잘못된 결과가 표시됩니다. (버그 #17909656)

- 범위 조건자를 통한 semijoin 쿼리에서 추가 행이 표시됩니다. (버그 #16221623)
- IN 하위 쿼리 뒤에 ORDER BY 절을 추가하면 중복 행이 반환될 수 있습니다. (버그 #16308085)
- 쿼리에 대한 설명이 GROUP BY, MyISAM에 대한 간략 스캔과 충돌합니다. (버그 #16222245)
- 인용된 int 조건자를 사용하여 느슨한 인덱스 스캔을 수행하면 임의의 데이터가 반환됩니다. (버그 #16394084)
- 최적화 프로그램에서 느슨한 인덱스 스캔을 사용 중인 경우 임시 테이블을 생성하려고 하면 서버가 종료될 수 있습니다. (버그 #16436567)
- COUNT(DISTINCT)는 NULL 값을 계산하지 않지만 최적화 프로그램에서 느슨한 인덱스 스캔을 사용하는 경우에는 계산됩니다. (버그 #17222452)
- 쿼리에 MIN()/MAX() 및 aggregate_function(DISTINCT)이 모두 포함되어 있고(예: SUM(DISTINCT)) 느슨한 인덱스 스캔을 사용하여 쿼리를 실행한 경우 MIN()/MAX()의 결과 값이 잘못 설정되었습니다. (버그 #17217128)

Aurora MySQL 데이터베이스 엔진 업데이트: 2016년 6월 1일(버전 1.6.5)(사용되지 않음)

버전: 1.6.5

새로운 기능

- 효율적인 이진 로그 저장 – 효율적인 이진 로그 저장 기능은 이제 모든 Aurora MySQL DB 클러스터에서 기본적으로 활성화되며 구성할 수 없습니다. 효율적인 이진 로그 저장은 2016년 4월 업데이트에 소개되었습니다. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트: 2016년 4월 6일(버전 1.6)(사용되지 않음) 단원을 참조하십시오.

개선 사항

- 기본 인스턴스에 과도한 워크로드가 발생하는 경우 Aurora 복제본의 안전성이 개선되었습니다.
- 분할된 테이블 및 테이블 이름에 특수 문자가 있는 테이블에서 쿼리를 실행할 때 Aurora 복제본의 안전성이 개선되었습니다.
- 보안 연결을 사용할 때 연결 문제가 수정되었습니다.

MySQL 버그 수정 통합

- SLAVE CAN'T CONTINUE REPLICATION AFTER MASTER'S CRASH RECOVERY (Port Bug #17632285)

Aurora MySQL 데이터베이스 엔진 업데이트: 2016년 4월 6일(버전 1.6)(사용되지 않음)

버전: 1.6

이 업데이트에는 다음의 기능 향상이 포함되어 있습니다.

새로운 기능

- 병렬 미리읽기 – 병렬 미리읽기는 이제 모든 Aurora MySQL DB 클러스터에서 기본적으로 활성화되며 구성할 수 없습니다. 병렬 미리읽기는 2015년 12월 업데이트에서 소개되었습니다. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트: 2015년 12월 3일(버전 1.4)(사용되지 않음) 섹션을 참조하세요.

  이 릴리스에는 병렬 미리읽기가 기본적으로 활성화되어 있을 뿐 아니라 병렬 미리읽기에 대한 다음과 같은 개선 사항이 포함되어 있습니다.
  - 로직을 개선하여 병렬 미리읽기가 덜 적극적인 상태가 되도록 합니다. 이렇게 하면 DB 클러스터에 많은 병렬 워크로드가 발생할 경우 도움이 됩니다.
  - 더 작은 테이블에서 안전성이 개선되었습니다.
- 효율적인 이진 로그 저장(랩 모드) – 이제 Aurora MySQL에서 MySQL 이진 로그 파일이 더 효율적으로 저장됩니다. 새로운 저장을 구현함으로써 이진 로그 파일을 이전보다 미리 삭제할 수 있으며, 이진 로그 복제 마스터인 Aurora MySQL DB 클러스터의 인스턴스에 대한 시스템 성능이 개선됩니다.

  효율적인 이진 로그 저장을 활성화하려면 기본 인스턴스나 Aurora 복제본의 파라미터 그룹에서 `aurora_lab_mode` 파라미터를 1로 설정합니다. `aurora_lab_mode` 파라미터는 기본적으로 `default.aurora5.6` 파라미터 그룹에 속하는 인스턴스 수준 파라미터입니다. DB 파라미터 그룹 수정에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 DB 파라미터 그룹의 파라미터 수정을 참조하세요. 파라미터 그룹 및 Aurora MySQL에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora MySQL 구성 파라미터를 참조하세요.

  MySQL 이진 로그 복제 마스터 인스턴스인 Aurora MySQL DB 클러스터에서 인스턴스에 대한 효율적인 이진 로그 저장 기능만 설정합니다.

- AURORA_VERSION 시스템 변수 – 이제 AURORA_VERSION 시스템 변수를 쿼리하여 해당 Aurora MySQL DB 클러스터의 Aurora 버전을 확인할 수 있습니다.

  Aurora 버전을 확인하려면 다음 쿼리 중 하나를 사용하십시오.

  ```
  select AURORA_VERSION();
  select @@aurora_version;
  show variables like '%version';
  ```

  DB 클러스터를 수정할 AWS Management Console 때 또는 describe-db-engine-versions AWS CLI 명령 또는 DescribeDBEngineVersions API 작업을 호출하여에서 Aurora 버전을 볼 수도 있습니다.
- 잠금 관리자 메모리 사용량 측정치 – 잠금 관리자 메모리 사용량에 대한 정보가 이제 측정치로 제공됩니다.

  잠금 관리자 메모리 사용량 지표를 가져오려면 다음 쿼리 중 하나를 사용합니다.

  ```
  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');
  ```

개선 사항

- binlog 및 XA 트랜잭션 복구 중 안전성이 개선되었습니다.
- 많은 연결로 발생하는 메모리 문제가 수정되었습니다.
- `Read Throughput`, `Read IOPS`, `Read Latency`, `Write Throughput`, `Write IOPS`, `Write Latency` 및 `Disk Queue Depth` 지표의 정확성이 개선되었습니다.
- 충돌 후 대량 인스턴스가 느리게 시작되는 안전성 문제가 수정되었습니다.
- 동기화 메커니즘 및 캐시 제거에 관련된 데이터 사전의 동시성이 개선되었습니다.
- Aurora 복제본의 안정성 및 성능 개선 사항:
  - 기본 인스턴스에 대한 과도하거나 폭증하는 쓰기 워크로드 중 Aurora 복제본의 안정성 문제가 수정되었습니다.
  - db.r3.4xlarge 및 db.r3.8xlarge 인스턴스의 복제 지연이 개선되었습니다.
  - 애플리케이션 로그 레코드와 Aurora 복제본 동시 읽기 사이의 경합이 감소되어 성능이 개선되었습니다.

- 새로 생성되거나 업데이트된 통계로 Aurora 복제본의 통계를 새로 고치는 문제가 수정되었습니다.
- 기본 인스턴스에 트랜잭션이 많고 동일한 데이터의 Aurora 복제본에 대한 동시 읽기가 많은 경우 Aurora 복제본의 안정성이 개선되었습니다.
- `UPDATE` 및 `DELETE` 문을 `JOIN` 문과 함께 실행할 때 Aurora 복제본의 안정성이 개선되었습니다.
- `INSERT ... SELECT` 문을 실행하는 경우 Aurora Replicas의 안정성이 개선되었습니다.

MySQL 버그 수정 통합

- 백포트 버그 #18694052 5.6에서 `!M_ORDERED_REC_BUFFER' 어설션 실패 수정(포트 버그 #18305270)
- MEMCPY(), HA_PARTITION::POSITION의 SEGV 오류(포트 버그 # 18383840)
- 파티셔닝, INDEX_MERGE 및 NO PK에서 잘못된 결과 도출(포트 버그 # 18167648)
- FLUSH TABLES FOR EXPORT: HA_PARTITION::EXTRA의 어설션 오류(포트 버그 # 16943907)
- 가상 HA_ROWS HANDLER::MULTI_RANGE_READ_INFO_CONST에서 서버 충돌(포트 버그 # 16164031)
- SEL_ARG::RB_INSERT()에서 범위 최적화 프로그램 충돌(포트 버그 # 16241773)

Aurora MySQL 데이터베이스 엔진 업데이트: 2016년 1월 11일(버전 1.5)(사용되지 않음)

버전: 1.5

이 업데이트에는 다음의 기능 향상이 포함되어 있습니다.

개선 사항

- Aurora 스토리지 배포 중 유휴 인스턴스에 발생하는 10초 간의 쓰기 작업 중지 오류를 수정했습니다.
- 이제 `innodb_file_per_table`이 `No`로 설정된 경우 논리적 미리읽기가 작동합니다. 논리적 미리읽기에 대한 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트: 2015년 12월 3일(버전 1.4)(사용되지 않음) 단원을 참조하십시오.
- Aurora 복제본이 기본 인스턴스에 다시 연결되는 문제를 해결했습니다. 이에 따라 오류 삽입 쿼리를 사용하여 Aurora 복제본을 테스트할 때 `quantity` 파라미터에 큰 값을 지정하는 경우 발생하는 문

제도 해결되었습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Aurora 복제본 실패 테스트를 참조하세요.

- 속도가 느려 다시 시작되는 Aurora 복제본을 모니터링하는 기능이 개선되었습니다.
- Aurora 복제본의 속도를 느리게 하고 등록이 취소되게 한 다음, 다시 시작되도록 하는 문제를 해결했습니다.
- 교착 상태 중에 `show innodb status` 명령을 실행하는 경우 발생하는 문제를 해결했습니다.
- 쓰기 처리량이 높을 때 크기가 큰 인스턴스에 대한 장애 조치 문제를 해결했습니다.

MySQL 버그 수정 통합

- 데이터베이스 이름이 숫자로 시작하는 테이블에 영향을 주는 MySQL 전체 텍스트 검색 기능이 완전히 해결되지 않았던 문제를 해결했습니다. (포트 버그 #17607956)

Aurora MySQL 데이터베이스 엔진 업데이트: 2015년 12월 3일(버전 1.4)(사용되지 않음)

버전: 1.4

이 업데이트에는 다음의 기능 향상이 포함되어 있습니다.

새로운 기능

- 빠른 입력 기능 – 기본 키에 의해 정렬되는 병렬 입력을 빠르게 처리해 줍니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL 성능 개선 사항을 참조하세요.
- 대용량 데이터 세트 읽기 성능 – Aurora MySQL는 입출력(IO)이 많은 워크로드를 자동으로 탐지하여 추가 스레드를 시작함으로써 DB 클러스터의 성능을 높여 줍니다. Aurora 스케줄러가 입출력 작업을 확인하고 시스템의 최적 스레드 수를 자동으로 조정할지 결정하여 낮은 오버헤드로 입출력 중심 워크로드와 CPU 중심 워크로드 간을 빠르게 조정합니다.
- 병렬 미리 읽기 – 기본 인스턴스나 Aurora 복제본(범위 쿼리 포함)에서 사용할 수 있는 메모리에 비해 너무 큰 B-트리 스캔의 성능을 개선합니다. 병렬 미리읽기는 페이지 읽기 패턴을 자동으로 탐지하여 페이지가 필요하기 전에 버퍼 캐시에 페이지를 미리 가져옵니다. 병렬 미리읽기는 동일한 트랜잭션 내에서 동시에 여러 테이블에 작동합니다.

개선 사항:

- Aurora 스토리지 배포 중 발생하는 간단한 Aurora 데이터베이스 가용성 문제를 해결했습니다.
- `max_connection` 제한을 올바르게 적용했습니다.
- Aurora가 binlog 마스터이고 대용량 데이터 로드 이후 데이터베이스가 다시 시작되는 경우 binlog 제거 문제를 개선했습니다.
- 테이블 캐시에 발생하는 메모리 관리 문제를 해결했습니다.
- 빠른 복구를 위해 공유된 메모리 버퍼 캐시에 방대한 페이지에 대한 지원을 추가했습니다.
- 스레드 로컬 스토리지가 초기화되지 않는 문제를 해결했습니다.
- 기본적으로 16K 연결이 허용됩니다.
- 입출력 중심 워크로드에 대해 동적 스레드 풀이 지원됩니다.
- 쿼리 캐시에 UNION을 포함하는 뷰를 제대로 무효화할 때 발생하는 문제를 해결했습니다.
- 사전 통계 스레드에서 발생하는 안정성 문제를 해결했습니다.
- 캐시 제거와 관련하여 사전 하위 시스템에서 발생하는 메모리 누수 문제를 해결했습니다.
- 마스터에 쓰기 로드가 매우 낮은 경우 Aurora 복제본에서 읽기 지연 시간이 크게 발생하는 문제를 해결했습니다.
- ALTER TABLE과 같은 DDL 분할된 테이블에서 작업을 수행할 때 Aurora 복제본의 안정성 문제를 해결했습니다... REORGANIZE PARTITION을 수행하는 것처럼 DDL 분할된 테이블에서 작업을 수행할 때 Aurora Replicas에서 발생하는 안정성 문제를 해결했습니다.
- 볼륨이 커지는 동안 Aurora 복제본에서 발생하는 안정성 문제를 해결했습니다.
- Aurora 복제본의 비클러스터 인덱스에서 스캔 수행 시 발생하는 성능 문제를 해결했습니다.
- Aurora 복제본의 속도를 느리게 하고 결국 등록이 취소되게 한 다음, 다시 시작되도록 하는 안정성 문제를 해결했습니다.

MySQL 버그 수정 통합

- FTSPARSE()의 SEGV 오류 (버그 #16446108)
- 열의 이름을 바꾸는 동안 InnoDB 데이터 사전이 업데이트되지 않습니다. (버그 #19465984)
- 테이블 이름을 다른 데이터베이스로 변경하면 FTS 충돌이 발생합니다. (버그 #16834860)
- 잘린 테이블에서 트리거를 준비하지 못하면 오류 1054가 발생합니다. (버그 #18596756)
- 메타데이터를 변경하면 트리거 실행에 문제가 발생할 수 있습니다. (버그 #18684393)
- 길이가 긴 UTF8 VARCHAR 필드에 구제화가 선택되지 않습니다. (버그 #17566396)

- 제한 사항 X가 있는 ORDER BY 실행 시 실행 계획의 성능 저하. (Bug #16697792)
- 백포트(Backport) 버그#11765744 - 5.1, 5.5 및 5.6. (버그 #17083851)
- SQL/SQL_SHOW.CC에서의 뮤텍스 문제로 SIG6가 발생합니다. 소스는 FILL_VARIABLES와 유사합니다. (버그 #20788853)
- 백포트(Backport) 버그 #18008907 - 5.5 이상 버전 (버그 #18903155)
- MySQL 5.7에서 스택 오버플로우 오류에 대해 수정 적용 (버그 #19678930)

Aurora MySQL 데이터베이스 엔진 업데이트: 2015년 10월 16일(버전 1.2, 1.3)(사용되지 않음)

(버전 1.2, 1.3)

이 업데이트에는 다음의 기능 향상이 포함되어 있습니다.

수정 사항

- 오랜 시간 실행되는 트랜잭션이 있는 새로운 잠금 관리자에서 메모리 부족 문제 해결
- 비-RDS for MySQL 데이터베이스를 복제할 때 발생하는 보안 취약성 해결
- 쿼롬 쓰기가 스토리지 오류와 함께 올바르게 재시도되도록 업데이트됨
- 복제 지연을 보다 정확하게 보고하도록 업데이트됨
- 많은 수의 동시 트랜잭션이 동일한 행을 수정하려고 하는 경우 경합을 줄임으로써 성능 개선
- 두 개의 테이블을 조인함으로써 생성된 뷰의 쿼리 캐시 무효화 해결
- `UNCOMMITTED_READ`가 격리된 트랜잭션의 쿼리 캐시 비활성화

개선 사항

- 웜 캐시에 있는 느린 카탈로그 쿼리의 성능 개선
- 사전 통계의 동시성 개선
- 새로운 쿼리 캐시 리소스 관리자, 익스텐트 관리, Amazon Aurora 스마트 스토리지에 저장되어 있는 파일, 로그 레코드의 배치 쓰기에 대한 안정성 개선

MySQL 버그 수정 통합

- innodb 내의 쿼리 삭제 시 결과적으로 어설션과의 충돌 발생 (버그 #1608883)
- 이벤트 스케줄러, 이벤트 실행 또는 새로운 연결에 대해 새로운 스레드를 생성하지 못한 경우 오류 로그에 아무 메시지도 작성되지 않습니다. (버그 #16865959)
- 한 연결이 기본 데이터베이스를 변경하고 동시에 다른 연결이 SHOW PROCESSLIST를 수행하는 경우, 첫 번째 연결의 기본 데이터베이스 메모리를 표시하려고 할 때 두 번째 연결에서 잘못된 메모리에 액세스할 수 있습니다. (버그 #11765252)
- 설계 상 PURGE BINARY LOGS는 사용 중이거나 활성화 상태의 이진 로그 파일을 제거하지 않지만, 이러한 경우에도 아무런 메시지가 표시되지 않습니다. (버그 #13727933)
- 일부 문에서는 최적화 프로그램이 필요 없는 하위 쿼리 절을 제거하는 경우 메모리 누수가 발생할 수 있습니다. (버그 #15875919)
- 종료 중에 서버가 초기화되지 않은 뮤텍스를 잠그려고 시도할 수 있습니다. (버그 #16016493)
- GROUP_CONCAT() 및 ORDER BY 절을 사용하여 여러 열의 이름을 지정하는 준비된 문을 사용하면 서버가 종료될 수 있습니다. (버그 #16075310)
- 복제본 작업자 스레드에 성능 스키마 장비가 누락되었습니다. (버그 #16083949)
- `STOP SLAVE`는 하나 이상의 상태 변수 `Slave_retried_transactions`, `Slave_heartbeat_period`, `Slave_received_heartbeats`, `Slave_last_heartbeat` 또는 `Slave_running`에 대한 값을 검색한 SHOW SATUS와 같은 문과 동시에 실행될 때 교착 상태가 발생할 수 있습니다. (버그 #16088188)
- 검색 용어에 따옴표가 있는 경우 부울 모드를 사용한 전체 텍스트 쿼리에서 0이 반환될 수 있습니다. (버그 #16206253)
- 하위 쿼리에서 조인의 ON 절에 하위 쿼리가 있는 준비된 문을 실행하는 경우 최적화 프로그램이 중복되는 하위 쿼리 절을 제거하려고 시도하면 어설션이 발생합니다. (버그 #16318585)
- GROUP_CONCAT 불안정성, ITEM_SUM::CLEAN_UP_AFTER_REMOVAL에서 충돌 발생 (버그 #16347450)
- INFORMATION_SCHEMA.INNODB_FT_DEFAULT_STOPWORD와 동일한 구조의 InnoDB 테이블을 생성하여 기본 InnoDB 전체 텍스트 검색(FTS) 불용어 목록을 변경하려고 시도하면 오류가 발생합니다. (버그 #16373868)
- 작업자의 클라이언트 스레드에서 FLUSH TABLES WITH READ LOCK을 수행한 후 마스터에서 일부 업데이트를 수행하면 `SHOW SLAVE STATUS`를 실행할 때 작업자가 멈춥니다. (버그 #16387720)
- 전체 텍스트 검색에서 "abc-def"와 같은 구분된 검색 문자열을 파싱할 때 이제 InnoDB에서 MyISAM과 동일한 단어 구분자를 사용합니다. (버그 #16419661)

- FTS_AST_TERM_SET_WILDCARD에서 충돌이 발생합니다. (버그 #16429306)
- FTS RQG 테스트 시 FTS_AST_VISIT()에서 SEGFAULT가 발생합니다. (버그 #16435855)
- 빌드 디버그 시 최적화 프로그램이 하위 쿼리를 가리키는 Item_ref를 제거하면 서버가 종료됩니다. (버그 #16509874)
- InnoDB 테이블에서 전체 텍스트 검색으로 + 또는 - 연산자와 결합된 리터럴 문구를 검색하면 오류가 발생합니다. (버그 #16516193)
- `START SLAVE`서버가 --master-info-repository=TABLE relay-log-info-repository=TABLE 옵션을 사용하고 자동 커밋을 0으로 설정하고 를 함께 사용하여 시작되면 에서 오류가 발생합니다.`--skip-slave-start` (버그 #16533802)
- InnoDB 전체 텍스트 검색(FTS)의 크기가 매우 크면 과도한 메모리 양을 소비할 수 있습니다. (버그 #16625973)
- 검색 문자열에 이진을 직접 사용하면 이진에 NULL 바이트 및 다른 의미 없는 문자가 포함될 수 있으므로 빌드 디버그 시 OPT_CHECK_ORDER_BY에서 어설션이 발생할 수 있습니다. (버그 #16766016)
- 일부 문에서는 최적화 프로그램이 필요 없는 하위 쿼리 절을 제거하는 경우 메모리 누수가 발생할 수 있습니다. (버그 #16807641)
- 작업자에 대한 새로운 연결에서 `STOP SLAVE`를 실행한 다음, 원래 연결을 사용하여 `SHOW SLAVE STATUS`를 실행하여 FLUSH TABLES WITH READ LOCK을 실행하면 교착 상태가 발생할 가능성이 있습니다. (버그 #16856735)
- 잘못된 구분 기호와 함께 GROUP_CONCAT()를 실행하면 서버가 종료될 수 있습니다. (버그 #16870783)
- 패턴이 해당 뮤텍스(`Slave_heartbeat_period`, `Slave_last_heartbeat`, `Slave_received_heartbeats`, `Slave_retried_transactions`, `Slave_running`)를 사용하는 상태 변수와 일치하지 않는 경우에도 서버가 SHOW STATUS LIKE '패턴' 문에 대해 LOCK_active_mi 및 active_mi->rli->data_lock 뮤텍스에서 과도한 잠금을 설정했습니다. (버그 #16904035)
- IN BOOLEAN MODE 수정자를 사용하여 전체 텍스트 검색을 실행하면 어설션 오류가 발생합니다. (버그 #16927092)
- InnoDB 테이블에서 전체 텍스트 검색으로 + 부울 연산자를 사용하여 검색하면 오류가 발생합니다. (버그 #17280122)
- 4웨이 교착 상태: 좀비, binlog 삭제, 프로세스 목록 표시, binlog 표시 (버그 #17283409)
- 커밋 잠금을 기다리고 있는 SQL 스레드가 중단되었다가 다시 시작되는 경우 트랜잭션이 작업자에서 건너 뛰게 됩니다. (버그 #17450876)

- "종료되지 않는" 토큰으로 인해 InnoDB 전체 텍스트 검색 오류가 발생합니다. 문자열 및 문자열 길이가 문자열 비교를 위해 전달되어야 합니다. (버그 #17659310)
- 많은 수의 분할된 InnoDB 테이블이 MySQL 5.6 또는 5.7에서 사용될 때 이전 릴리스의 MySQL Server에서 사용될 때보다 동일한 테이블에서 사용하는 메모리 양이 많이 증가될 수 있습니다. (버그 #17780517)
- 전체 텍스트 쿼리의 경우 num_token이 max_proximity_item보다 적은지 확인하지 못하여 어설션이 발생할 수 있습니다. (버그 #18233051)
- 비어 있는 InnoDB 테이블이 많이 있는 경우 INFORMATION_SCHEMA 테이블 및 COLUMNS 테이블에 대한 특정 쿼리가 과도한 메모리 사용을 초래할 수 있습니다. (버그 #18592390)
- 트랜잭션을 커밋할 때 이제 플래그가 사용되어 스레드 자체를 확인하지 않고 스레드 생성 여부를 확인합니다. 이에 따라 특히 master_info_repository=TABLE과 함께 서버를 실행할 때 더 많은 리소스가 사용됩니다. (버그 #18684222)
- 마스터가 DML을 실행하는 동안 작업자에 있는 클라이언트 스레드가 FLUSH TABLES WITH READ LOCK을 실행하는 경우, 동일한 클라이언트에 있는 `SHOW SLAVE STATUS`를 실행하면 차단되어 교착 상태가 발생합니다. (버그 #19843808)
- GROUP_CONCAT()로 주문하면 서버가 종료될 수 있습니다. (버그 #19880368)

Aurora MySQL 데이터베이스 엔진 업데이트: 2015년 8월 24일(버전 1.1)(사용되지 않음)

버전: 1.1

이 업데이트에는 다음의 기능 향상이 포함되어 있습니다.

- MySQL 데이터베이스(binlog 복제)를 사용하여 복제하는 경우 복제 안정성이 개선되었습니다. MySQL을 사용한 Aurora MySQL 복제에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora를 사용한 복제를 참조하세요.
- 릴레이 로그의 크기에 대한 1기가바이트(GB) 제한이 복제 작업자인 Aurora MySQL DB 클러스터에 대해 수집됩니다. 이에 따라 Aurora DB 클러스터에 대한 파일 관리가 개선되었습니다.
- 미리읽기, 리커시브 외래 키 관계 및 Aurora 복제의 영역에서 안정성이 개선되었습니다.
- MySQL 버그 수정 통합.
  - 이름이 숫자로 시작하는 InnoDB 데이터베이스가 전체 텍스트 검색(FTS) 구문 분석기 오류를 일으킵니다. (버그 #17607956)

  - 이름이 숫자로 시작하는 데이터베이스에서 InnoDB 전체 텍스트 검색이 실패합니다. (버그 #17161372)
  - Windows에서 실행되는 InnoDB 데이터베이스의 경우, 전체 텍스트 검색(FTS) 객체 ID가 예상되는 16진수 형식이 아닙니다. (버그 #16559254)
  - MySQL 5.6에 도입된 코드 회귀가 DROP TABLE 및 ALTER TABLE 성능에 악영향을 미쳤습니다. 이로 인해 MySQL Server 5.5.x와 5.6.x 사이에서 성능 저하가 발생했을 수 있습니다. (버그 #16864741)
- 로깅을 단순화하여 필요한 스토리지 양 및 로그 파일 크기가 줄어들었습니다.

Aurora MySQL 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그

다음 섹션은 Aurora MySQL 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그를 설명한 것입니다.

주제

- Aurora MySQL 3.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그
- Aurora MySQL 2.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그
- Aurora MySQL 1.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그

Aurora MySQL 3.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그

MySQL 8.0과 호환되는 버전의 Aurora에는 해당 MySQL 호환성 버전을 통한 모든 MySQL 버그 수정 사항이 포함되어 있습니다. 다음 표에는 Aurora MySQL 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그와 이 버그들이 어느 업데이트에서 수정되었는지가 나와 있습니다.

데이터베이스 엔진 업데이트	MySQL 호환 버전	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트 2024년 11월 18일(버전 3.08.0, MySQL 8.0.39와 호환)	8.0.39	3.08.0	• JOIN 및 UNION 작업이 모두 있는 특정 쿼리의 결과 세트에서 NULL 값이 잘못 생략되는 문제를 수정했습니다. (커뮤니티 버그 수정 #114301)

데이터베이스 엔진 업데이트	MySQL 호환 버전	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트 2024-06-04(버전 3.07.0) (사용되지 않음)	8.0.36	3.07.0	• 캐시 라인 값을 잘못 계산하여 Graviton 기반 인스턴스에서 데이터베이스를 다시 시작하는 동안 오류가 발생하는 문제를 수정했습니다. (커뮤니티 버그 수정 #35479763) • 저장된 루틴 내의 일부 하위 쿼리 인스턴스가 항상 올바르게 처리되지 않는 문제를 수정했습니다. (커뮤니티 버그 수정 #35377192) • 백그라운드 TLS 인증서 교체로 인해 CPU 사용이 높아질 수 있는 문제를 해결했습니다(커뮤니티 버그 수정 #34284186). • InnoDB가 3.05 이하의 Aurora MySQL 버전에서 MySQL 시스템 스키마의 테이블에 `INSTANT` 열을 추가하도록 허용하여 Aurora MySQL 버전 3.05.0으로 업그레이드한 후 서버가 예기치 않게 종료(데이터베이스 인스턴스 다시 시작)될 수 있는 문제를 수정했습니다. (커뮤니티 버그 수정 #35625510)

데이터베이스 엔진 업데이트	MySQL 호환 버전	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트 2024-03-07(버전 3.06.0) (사용되지 않음)	8.0.34	3.06.0	• 캐시 라인 값을 잘못 계산하여 Graviton 인스턴스에서 데이터베이스를 다시 시작하는 동안 오류가 발생하는 문제를 수정했습니다. (커뮤니티 버그 수정 #35479763) • 저장된 루틴 내의 일부 하위 쿼리 인스턴스가 항상 올바르게 처리되지 않는 문제를 수정했습니다. (커뮤니티 버그 수정 #35377192) • 백그라운드 TLS 인증서 교체로 인해 CPU 사용이 높아질 수 있는 문제를 해결했습니다. (커뮤니티 버그 수정 #34284186) • InnoDB가 3.05 이하의 Aurora MySQL 버전에서 MySQL 시스템 스키마의 테이블에 `INSTANT` 열을 추가하도록 허용하여 Aurora MySQL 버전 3.05.0으로 업그레이드한 후 서버가 예기치 않게 종료(데이터베이스 인스턴스 다시 시작)될 수 있는 문제를 수정했습니다. (커뮤니티 버그 수정 #35625510)

데이터베이스 엔진 업데이트	MySQL 호환 버전	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트 2024-01-31(버전 3.05.2) (사용되지 않음)	8.0.32	3.05.2	• `records_in_range` 가 `INSERT` 작업에 대해 과도한 수의 디스크 읽기를 수행하여 성능이 점진적으로 저하되는 문제를 수정했습니다. (커뮤니티 버그 수정 #34976138)
Aurora MySQL 데이터베이스 엔진 업데이트 2023-11-21(버전 3.05.1) (사용되지 않음)	8.0.32	3.05.1	• InnoDB에서 시스템 스키마의 MySQL 테이블에 Aurora MySQL 버전 3.01부터 Aurora MySQL 버전 3.04까지 `INSTANT ADD` 열이 추가된 경우 Aurora MySQL이 버전 3.05.0으로 업그레이드된 후 이러한 테이블의 DML로 인해 서버가 예기치 않게 종료되는 문제를 해결했습니다. (커뮤니티 버그 수정 #35625510)
Aurora MySQL 데이터베이스 엔진 업데이트 2023-10-25(버전 3.05.0) (사용되지 않음)	8.0.32	3.05.0	• 백그라운드 TLS 인증서 교체로 인해 CPU 사용률이 높아질 수 있는 문제를 해결했습니다. (커뮤니티 버그 수정 #34284186)

데이터베이스 엔진 업데이트	MySQL 호환 버전	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트 2024년 3월 15일(버전 3.04.2, MySQL 8.0.28과 호환)	8.0.28	3.04.2	• 캐시 라인 값을 잘못 계산하여 Graviton 기반 인스턴스에서 데이터베이스를 다시 시작하는 동안 오류가 발생하는 문제를 수정했습니다. (커뮤니티 버그 수정 #35479763) • 여러 AND, OR 또는 XOR 조건이 포함된 SELECT 문을 하위 쿼리로 사용하여 저장된 루틴을 반복적으로 실행하면 과도한 소비와 최종 가상 메모리 소진이 발생할 수 있습니다. (커뮤니티 버그 수정 #33852530)
Aurora MySQL 데이터베이스 엔진 업데이트 2023년 11월 13일(버전 3.04.1, MySQL 8.0.28과 호환)	8.0.28	3.04.1	• 백그라운드 TLS 인증서 교체로 인해 CPU 사용률이 높아질 수 있는 문제를 해결했습니다(커뮤니티 버그 수정 #34284186).

데이터베이스 엔진 업데이트	MySQL 호환 버전	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트 2023-07-31(버전 3.04.0) , MySQL 8.0.28과 호환)	8.0.28	3.04.0	• 고유 임시 테이블 페이지를 포함하는 버퍼 블록이 페이지 순회 중에 재배치되어 어설션 오류가 발생하는 문제를 해결했습니다. (버그 #33715694) • InnoDB: 온라인 DDL 작업이 범위를 벗어난 메모리에 액세스하는 것을 방지합니다. (버그 #34750489, 버그 #108925) • 여러 개의 중첩된 공통 테이블 표현식(CTE)으로 구성된 복잡한 SQL 문을 처리할 때 때때로 잘못된 쿼리 결과가 생성될 수 있는 문제를 해결했습니다. (버그 #34572040, 버그 #34634469, 버그 #33856374)
Aurora MySQL 데이터베이스 엔진 업데이트 2023년 12월 8일(버전 3.03.3) (사용되지 않음)	8.0.26	3.03.3	• 백그라운드 TLS 인증서 교체로 인해 CPU 사용률이 높아질 수 있는 문제를 해결했습니다(커뮤니티 버그 수정 #34284186).

데이터베이스 엔진 업데이트	MySQL 호환 버전	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트 2023년 8월 29일(버전 3.03.2) (사용되지 않음)	8.0.26	3.03.2	• 여러 개의 중첩된 공통 테이블 표현식(CTE)으로 구성된 복잡한 SQL 문을 처리할 때 때때로 잘못된 쿼리 결과가 생성될 수 있는 문제를 해결했습니다. (버그 #34572040, 버그 #34634469, 버그 #33856374) • InnoDB: 동일한 테이블에 대한 통계를 초기화 해제하고 초기화하려는 스레드 간의 경합 상태로 인해 어설션 오류가 발생했습니다. (버그 #33135425) • InnoDB: 온라인 DDL 작업이 범위를 벗어난 메모리에 액세스하는 것을 방지합니다. (버그 #34750489, 버그 #108925)
Aurora MySQL 데이터베이스 엔진 업데이트 2023년 5월 11일(버전 3.03.1) (사용되지 않음)	8.0.26	3.03.1	• 고유 임시 테이블 페이지를 포함하는 버퍼 블록이 페이지 순회 중에 재배치되어 어설션 오류가 발생하는 문제를 해결했습니다. (버그 #33715694)

데이터베이스 엔진 업데이트	MySQL 호환 버전	버전	수정된 MySQL 버그

Aurora MySQL 데이터베이스 엔진 업데이트 2023년 3월 1일(버전 3.03.0) (사용되지 않음)	8.0.26	3.03.0	• JSON 및 TEXT를 비롯한 일부 열 유형의 크기가 정렬에서 가장 큰 행의 크기보다 15배 이상 크지 않은 경우 때때로 정렬 버퍼가 소진되는 문제를 해결했습니다. 이제 정렬 버퍼의 크기가 가장 큰 정렬 키의 15배이기만 하면 됩니다. (버그 #103325, 버그 #105532, 버그 #32738705, 버그 #33501541)
			• InnoDB가 경우에 따라 테이블 파티션의 일부 정식 이름을 올바르게 처리하지 않는 문제를 해결했습니다. (버그 #32208630)
			• OR 조건을 사용한 쿼리를 실행할 때 null 허용 여부 속성의 부정확한 계산으로 인해 특정 상황에서 잘못된 결과가 반환될 수 있는 문제를 해결했습니다. (버그 #34060289)
			• 다음 두 조건이 충족될 때 특정 상황에서 잘못된 결과가 반환될 수 있는 문제를 해결했습니다.
			  • 파생 테이블이 외부 쿼리 블록에 병합되어 있습니다.
			  • 쿼리에 왼쪽 조인과 IN 하위 쿼리가 포함되어 있습니다.
			  (버그 #34060289)
			• 최대 정수 열 값을 초과할 때 잘못된 AUTO_INCREMENT 값이 생성되는 문제를 해결했습니

데이터베이스 엔진 업데이트	MySQL 호환 버전	버전	수정된 MySQL 버그
			다. 이 오류는 최대 열 값을 고려하지 않았기 때문에 발생했습니다. 이 경우 이전의 유효한 AUTO_INCREMENT 값이 반환되어 중복 키 오류가 발생합니다. (버그 #87926, 버그 #26906787) • 성능 스키마에 대한 DROP 권한을 취소할 수 없는 문제를 해결했습니다. (버그 #33578113) • 실행 사이에 삭제되었다가 다시 생성된 하나 이상의 테이블에 대해 작동하며 EXISTS를 사용하는 IF 문이 포함된 저장 프로시저가 첫 번째 간접 호출 이후 후속 간접 호출에서 올바르게 실행되지 않는 문제를 해결했습니다. (버그 #32855634) • 하위 쿼리와 외부 쿼리 블록의 뷰를 참조하는 쿼리가 예기치 않게 다시 시작되는 문제를 해결했습니다. (버그 #32324234)

데이터베이스 엔진 업데이트	MySQL 호환 버전	버전	수정된 MySQL 버그

Aurora MySQL 데이터베이스 엔진 업데이트 2022년 11월 18일(버전 3.02.2) (사용되지 않음)	8.0.23	3.02.2	• OR 조건을 사용한 쿼리를 실행할 때 null 허용 여부 속성의 부정확한 계산으로 인해 특정 상황에서 잘못된 결과가 반환될 수 있는 문제를 해결했습니다. (버그 #34060289)
			• 다음 두 조건이 충족될 때 특정 상황에서 잘못된 결과가 반환될 수 있는 문제를 해결했습니다.
			  • 파생 테이블이 외부 쿼리 블록에 병합되어 있습니다.
			  • 쿼리에 왼쪽 조인과 IN 하위 쿼리가 포함되어 있습니다. (버그 #34060289)
			• 성능 스키마에 대한 DROP 권한을 취소할 수 없는 문제를 해결했습니다. (버그 #33578113)
			• 실행 사이에 삭제되었다가 다시 생성된 하나 이상의 테이블에 대해 작동하며 EXISTS를 사용하는 IF 문이 포함된 저장 프로시저가 첫 번째 간접 호출 이후 후속 간접 호출에서 올바르게 실행되지 않는 문제를 해결했습니다. (MySQL 버그 #32855634)
			• 최대 정수 열 값을 초과할 때 잘못된 AUTO_INCREMENT 값이 생성되었습니다. 이 오류는 최대 열 값을 고려하지 않았기 때문에 발생했습니다. 이 경우 이전의 유효한 AUTO_INCREMENT

데이터베이스 엔진 업데이트	MySQL 호환 버전	버전	수정된 MySQL 버그
			값이 반환되어 중복 키 오류가 발생합니다. (버그 #87926, 버그 #26906787) • 특정 테이블 ID로 사용자가 생성한 테이블이 포함된 Aurora MySQL 버전 1(MySQL 5.6과 호환) 데이터베이스 클러스터를 업그레이드하는 동안 오류가 발생할 수 있는 문제를 해결했습니다. 이러한 테이블 ID를 할당하면 Aurora MySQL 버전 2(MySQL 5.7과 호환)에서 Aurora MySQL 버전 3(MySQL 8.0과 호환)으로 업그레이드하는 동안 데이터 딕셔너리 테이블 ID가 충돌할 수 있습니다. (버그 #33919635)
Aurora MySQL 데이터베이스 엔진 업데이트 2022년 4월 20일(버전 3.02.0) (사용되지 않음)	8.0.23	3.02.0	저장 프로시저 내에서 커서에 사용되는 임시 테이블을 잘못 처리하여 예기치 않은 서버 동작이 발생할 수 있는 문제를 해결했습니다. (버그 #32416811)

데이터베이스 엔진 업데이트	MySQL 호환 버전	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트 2022년 4월 15일(버전 3.01.1) (사용되지 않음)	8.0.23	3.01.1	저장 프로시저 내에서 커서에 사용되는 임시 테이블을 잘못 처리하여 예기치 않은 서버 동작이 발생할 수 있는 문제를 해결했습니다. (버그 #32416811)

Aurora MySQL 2.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그

MySQL 5.7 호환 버전 Aurora에는 MySQL 5.7.44을 통한 모든 MySQL 버그 수정이 포함되어 있습니다. 다음 표에는 Aurora MySQL 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그와 이 버그들이 어느 업데이트에서 수정되었는지가 나와 있습니다.

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트 2024년 7월 9일(버전 2.12.3, MySQL 5.7.44와 호환) - RDS 추가 지원 버전	2.12.3	• 문을 실행하는 동안 임시 테이블이 트리거되어 예기치 않은 DB 엔진 재시작이 발생할 수 있는 문제를 수정했습니다. • 인덱싱된 표현식을 사용하는 단일 테이블 UPDATE 및 DELETE 문이 준비된 문으로 실행될 때 서버가 종료될 수 있는 결함을 수정했습니다. (버그 #29257254)

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트 2023년 12월 28일(버전 2.12.1, MySQL 5.7.40과 호환) - RDS 추가 지원 버전	2.12.1	• `SHOW PROCESSLIST` 문과 동시에 실행될 때 기존 및 새 원격 연결이 중단될 수 있는 문제를 해결했습니다. (커뮤니티 버그 #34857411) • 복제: 일부 바이너리 로그 이벤트가 경우에 따라 올바르게 처리되지 않았습니다. (버그 #34617506) • 전체 텍스트 검색(FTS) 구문 분석기 플러그인에 의한 단일 문자 토큰 처리를 수정했습니다. (버그 #35432973)
Aurora MySQL 데이터베이스 엔진 업데이트 2023년 7월 25일(버전 2.12.0, MySQL 5.7.40과 호환) - RDS 추가 지원 버전	2.12.0	• 백그라운드 TLS 인증서 교체로 인해 CPU 사용률이 높아질 수 있는 문제를 해결했습니다. (커뮤니티 버그 수정 #34284186)
Aurora MySQL 데이터베이스 엔진 업데이트 2023년 10월 17일(버전 2.11.4, MySQL 5.7.12와 호환) - RDS 추가 지원 버전	2.11.4	• 복제: 일부 바이너리 로그 이벤트가 경우에 따라 올바르게 처리되지 않았습니다. (버그 #34617506) • 백그라운드 TLS 인증서 교체로 인해 CPU 사용률이 높아질 수 있는 문제를 해결했습니다. (커뮤니티 버그 수정 #34284186) • 준비된 문에서 일부 유형의 하위 쿼리로 인해 서버가 종료될 수 있습니다. (버그 #33100586)

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트 2022년 10월 25일(버전 2.11.0, MySQL 5.7.12와 호환) - RDS 추가 지원 버전	2.11.0	• 성능 스키마 문 이벤트 테이블(예: events_statements_current)에서 문자 집합 정보를 읽는 코드가 해당 문자 집합 정보에 대한 동시 쓰기를 방지하지 못하는 문제를 해결했습니다. 결과적으로 SQL 쿼리 텍스트 문자 집합이 유효하지 않아 서버가 종료될 수 있었습니다. 이 수정을 사용하면 잘못된 문자 집합으로 인해 SQL_TEXT 열이 잘리며 서버가 종료되지 않습니다. (버그 #23540008) • InnoDB: 커뮤니티 버그 #25189192, 버그 #84038에 대한 수정 백포트. 테이블을 다른 스키마로 이동한 RENAME TABLE 작업 이후 InnoDB가 INNODB_SYS_DATAFILES 데이터 딕셔너리 테이블을 업데이트하지 못하는 문제를 해결했습니다. 이로 인해 다시 시작 시 테이블스페이스 데이터 파일을 찾을 수 없다는 오류가 발생했습니다. • InnoDB: 새 인덱스를 추가할 때 서버가 내부적으로 정의된 외래 키 인덱스를 삭제하고 가상 생성 열에 정의된 보조 인덱스를 외래 키 인덱스로 사용하려고 시도하여 서버가 종료되는 문제를 해결했습니다. InnoDB는 이제 외래 키 제약 조건이 가상 생성 열에 정의된 보조 인덱스를 참조하도록 허용합니다. (버그 #23533396) • INSERT ... ON DUPLICATE KEY UPDATE 작업을 동시에 실행하는 두 세션에서 교착 상태가 발생하는 문제를 해결했습니다. 튜플을 부분적으로 롤백하는 동안 다른 세션에서 튜플을 업데이트할 수 있습니다. 이 버그에 대한 수정은 버그 #11758237, 버그 #17604730 및 버그 #20040791에 대한 수정을 되돌립니다. (버그 #25966845) • automatic_sp_privileges가 활성화된 경우에도 루틴 생성자에게 EXECUTE 및 ALTER ROUTINE 권한이 올바르게 부여되지 않는 문제를 해결했습니다. (버그 #27407480) • 커뮤니티 버그#24671968에 대한 수정 백포트: WHERE 절에 종속 하위 쿼리가 포함되어 있고, 테이블에서 선택 목록의 열과 뒤따르는 하위 쿼리의 열에 보조 인덱스가 있고, GROUP BY 또는 DISTINCT가 쿼리에서 느슨한 인덱스 스캔을 사용하도록

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
		허용하는 경우 쿼리가 잘못된 결과를 생성할 수 있는 문제를 해결했습니다. • 외래 키가 있는 여러 테이블에 대해 다중 테이블 삭제 문을 실행하면 복제가 중단되는 문제를 해결했습니다. (버그 #80821) • slave_skip_errors가 활성화된 상태에서도 특수한 경우에 특정 슬레이브 오류가 무시되지 않는 문제를 해결했습니다. 테이블 열기 및 잠금이 실패하거나 행 기반 복제를 실행하는 서버에서 필드 변환이 실패할 경우 오류가 심각한 것으로 간주되어 slave_skip_errors 상태가 무시됩니다. 이 수정을 통해 slave_skip_errors가 활성화된 상태에서 트랜잭션을 적용하는 동안 보고된 모든 오류가 올바르게 처리됩니다. (버그 #70640, 버그 #17653275) • SET PASSWORD 문이 MySQL 5.6 마스터에서 MySQL 5.7 슬레이브로 복제되거나 log_builtin_as_identified_by_password 시스템 변수가 ON으로 설정된 MySQL 5.7 마스터에서 MySQL 5.7 슬레이브로 복제되며 암호 해시 자체도 슬레이브에 저장되기 전에 해시되는 문제를 해결했습니다. 이제 이 문제가 해결되었으며 복제된 암호 해시가 원래 슬레이브에 전달된 대로 저장됩니다. (버그 #24687073) • 여러 수준의 JSON 배열, 객체 또는 둘 다로 래핑되어 있는 대규모 하위 문서로 구성된 JSON 값의 직렬화를 완료하는 데 때때로 과도한 시간이 소요되는 문제를 해결했습니다. (버그 #23031146) • 구문 오류 등으로 인해 구문 분석할 수 없는 문은 더 이상 느린 쿼리 로그에 기록되지 않습니다. (버그 #33732907)

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트 2022년 11월 1일(버전 2.10.3) (사용되지 않음)	2.10.3	• 성능 스키마 문 이벤트 테이블(예: events_statements_current)에서 문자 집합 정보를 읽는 코드가 해당 문자 집합 정보에 대한 동시 쓰기를 방지하지 못하는 문제를 해결했습니다. 결과적으로 SQL 쿼리 텍스트 문자 집합이 유효하지 않아 서버가 종료될 수 있었습니다. 이 수정을 사용하면 잘못된 문자 집합으로 인해 SQL_TEXT 열이 잘리며 서버가 종료되지 않습니다. (버그 #23540008) • UPDATE에 1,024바이트보다 큰 프라이머리 키가 있는 임시 테이블이 필요하고 해당 테이블이 InnoDB를 사용하여 생성된 경우 서버가 종료될 수 있는 문제를 해결했습니다. (버그 #25153670)
Aurora MySQL 데이터베이스 엔진 업데이트 2022년 1월 26일(버전 2.10.2) (사용되지 않음)	2.10.2	• 테이블 통계와 관련된 코드의 오류로 인해 dict0stats.cc(http://dict0stats.cc/) 소스 파일에서 어설션이 발생하는 InnoDB의 문제를 수정했습니다. (버그 #24585978) • 인덱스가 온라인으로 빌드될 때 가상 열에 대한 세컨더리 인덱스가 손상되었습니다. UPDATE(https://dev.mysql.com/doc/refman/5.7/en/update.html) 문의 경우 이를 다음과 같이 수정합니다. 인덱스 레코드의 가상 열 값이 NULL로 설정되면 클러스터 인덱스 레코드에서 이 값을 생성합니다. (버그 #30556595) • ASSERTION "!OTHER_LOCK" IN LOCK_REC_ADD_TO_QUEUE(버그 #29195848) • HANDLE_FATAL_SIGNAL (SIG=11) IN __STRCHR_SSE2(버그 #28653104) • 잠금 대기 중 쿼리 중단으로 인해 InnoDB에서 오류가 발생할 수 있는 문제를 수정했습니다. (버그 #28068293) • 트랜잭션 격리 수준이 REPEATABLE READ로 설정된 경우 인터리브 트랜잭션에서 슬레이브 애플리케이션이 교착 상태에 빠질 수 있습니다. (버그 #25040331) • 잠금 대기 시간 초과로 인해 binlog 복제본이 중단될 수 있는 문제를 수정했습니다(버그 #27189701).

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트 2021년 10월 21일(버전 2.10.1)(사용되지 않음)	2.10.1	CURRENT_TIMESTAMP가 트리거에서 0을 생성합니다. (버그 #25209512)

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트 2021년 5월 25일(버전 2.10.0) (사용되지 않음)	2.10.0	• 트랜잭션 격리 수준이 REPEATABLE READ로 설정된 경우 인터리브 트랜잭션에서 슬레이브 애플리케이션이 교착 상태에 빠질 수 있습니다. (버그 #25040331) • 저장 프로시저에 다른 보기를 참조하는 보기를 참조하는 문이 포함되어 있으면 프로시저를 두 번 이상 성공적으로 호출할 수 없습니다. (버그 #87858, 버그 #26864199) • OR 조건이 많은 쿼리에서 이제 옵티마이저의 메모리 효율성이 향상되어 range_optimizer_max_mem_size 시스템 변수에 의해 적용되는 메모리 제한을 초과할 가능성이 적어집니다. 또한 해당 변수의 기본값이 1,536,000에서 8,388,608로 늘어났습니다. (버그 #79450, 버그 #22283790) • 복제: 복제본의 SQL 스레드에서 호출되어 릴레이 로그의 다음 이벤트를 읽는 next_event() 함수에서 SQL 스레드가 오류(예: 닫힌 릴레이 로그)로 인해 가져온 relaylog.log_lock 을 해제하지 않아 릴레이 로그의 잠금을 가져오기 위해 대기 중인 다른 모든 스레드가 중단되는 문제가 있었습니다. 이 수정을 적용하면 이 상황에서 SQL 스레드가 함수에서 나가기 전에 잠금이 해제됩니다. (버그 #21697821) • 가상 열이 있는 ALTER TABLE에 대한 메모리 손상이 해결되었습니다. (버그 #24961167, 버그 #24960450) • 복제: 다중 스레드 복제본에서 slave_pending_jobs_size_max 보다 큰 트랜잭션을 처리해야 하는 경우 이 크기를 사용하여 작은 대기열 크기로 다중 스레드 복제본을 구성할 수 없었습니다. slave_pending_jobs_size_max보다 큰 모든 패킷은 패킷이 slave_max_allowed_packet으로 설정된 제한보다 작은 경우에도 ER_MTS_EVENT_BIGGER_PENDING_JOBS_SIZE_MAX 오류와 함께 거부되었습니다. 이 수정을 적용하면 slave_pending_jobs_size_max가 하드 제한이 아닌 소프트 제한이 됩니다. 패킷 크기가 slave_pending_jobs_size_max를 초과하지만 slave_max_allowed_packet보다 작은 경우 모든 복제본 작업자의 대기열이 빌 때까지 트랜잭션이 보류된 후 처리됩니다. 모든 후속 트랜잭션은 대규모 트랜잭션이 완료될 때까

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
		지 보류됩니다. 따라서 복제본 작업자의 대기열 크기는 제한되지만 경우에 따라 더 큰 트랜잭션을 허용할 수 있습니다. (버그 #21280753, 버그 #77406) • 복제: 다중 스레드 복제본을 사용할 때 적용자 오류에 표시되는 작업자 ID 데이터가 성능 스키마 복제 테이블에 표시된 데이터와 일치하지 않았습니다. (버그 #25231367) • 복제: -gtid-mode=ON, -log-bin=OFF이고 -slave-skip-errors를 사용하여 실행되는 GTID 기반 복제 복제본에서 Exec_Master_Log_Pos 가 올바르게 업데이트되지 않는 무시할 수 있는 오류가 발생하는 경우 Exec_Master_Log_Pos 가 Read_master_log_pos 와 느슨하게 동기화되었습니다. GTID_NEXT 가 지정되지 않은 경우 복제본은 단일 문 트랜잭션에서 롤백할 때 GTID 상태를 업데이트하지 않습니다. 트랜잭션이 완료되었지만 GTID 상태가 다르게 표시되므로 Exec_Master_Log_Pos 가 업데이트되지 않습니다. 이 수정은 GTID_NEXT 가 지정된 경우에만 트랜잭션 롤백 시 GTID 상태 업데이트 제한을 제거합니다. (버그 #22268777) • 복제: 바이너리 로깅이 사용 중지된 경우 부분적으로 실패한 문에 자동 생성되거나 지정된 GTID가 올바르게 사용되지 않았습니다. 이 수정을 적용하면 이진 로깅이 사용 중지된 경우 부분적으로 실패한 DROP TABLE, 부분적으로 실패한 DROP USER 또는 부분적으로 실패한 DROP VIEW에 관련 GTID가 각각 사용되고 @@GLOBAL.GTID_EXECUTED 및 mysql.gtid_executed 테이블에 저장됩니다. (버그 #21686749) • 복제: MySQL 5.7을 실행하는 복제본에서 MySQL 5.5의 일부가 아닌 server_uuid 검색 중에 발생하는 오류로 인해 MySQL 5.5 소스에 연결할 수 없었습니다. 이 문제는 server_uuid 검색 방법의 변경으로 인해 발생했습니다. (버그 #22748612) • Binlog 복제: 이 수정 전에는 GTID 트랜잭션 건너뛰기 메커니즘이 XA 트랜잭션에 대해 올바르게 작동하지 않았습니다. 서버에는 과거에 특정 트랜잭션이 이미 실행된 경우 GTID 트랜잭션을 자동으로 건너뛰는 메커니즘이 있습니다. (버그#25041920)

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
		• 잘못된 트랜잭션 ID가 지정되어 실패한 XA ROLLBACK 문이 올바른 트랜잭션 ID로 바이너리 로그에 기록되었기 때문에 복제본의 작업에 사용될 수 있었습니다. 이제 바이너리 로깅이 수행되기 전에 오류 상황이 확인되고 실패한 XA ROLLBACK 문이 로깅되지 않습니다. (버그 #26618925) • 복제: 소스 로그 파일 이름 및 소스 로그 위치를 지정하지 않은 CHANGE MASTER TO 문을 사용하여 설정된 복제본이 START SLAVE 실행 전에 중단되고 -relay-log-recovery 설정 옵션을 사용하여 다시 시작되는 경우 복제가 시작되지 않았습니다. 이 문제는 릴레이 로그 복구가 시도되기 전에 수신기 스레드가 시작되지 않아 릴레이 로그에서 소스 로그 파일 이름과 소스 로그 위치를 제공하는 데 필요한 로그 회전 이벤트를 사용할 수 없기 때문에 발생했습니다. 이제 이 상황에서 복제본은 릴레이 로그 복구를 건너뛰고 경로를 로깅한 다음 복제를 시작합니다. (버그 #28996606, 버그 #93397) • 복제: 행 기반 복제에서 `utf8mb3` 열이 있는 테이블을 `utf8mb4` 문자 집합으로 정의된 열이 있는 동일한 정의의 테이블로 복제할 때 필드 길이가 잘못 표시된 메시지가 반환되었습니다. (버그 #25135304, 버그 #83918) • 복제: GTID를 사용 중인 복제 복제본에서 RESET SLAVE 문이 실행되는 경우 기존 릴레이 로그 파일이 제거되지만 채널에 대해 수신된 GTID 세트가 지워지기 전에 새로운 대체 릴레이 로그 파일이 생성되었습니다. 따라서 이전 GTID 세트가 새 릴레이 로그 파일에 `PREVIOUS_GTIDS` 이벤트로 기록되고 이로 인해 두 서버에 설정된 gtid_executed가 모두 비어 있음에도 불구하고 복제본에 소스보다 많은 GTID가 있다는 내용의 치명적 오류가 복제에서 발생합니다. 이제 `RESET SLAVE`가 실행되면 새 릴레이 로그 파일이 생성되기 전에 수신된 GTID 세트가 지워지므로 이러한 상황이 발생하지 않습니다. (버그 #27411175) • 복제: GTID를 복제에 사용하는 경우 동일한 GTID의 빈 트랜잭션 또는 대체 트랜잭션을 삽입하는 권장 방법으로 구문 분석 오류(ER_PARSE_ERROR)를 야기한 문이 포함된 트랜잭션을 수동으로 건너뛸 수 없습니다. 이 작업을 수행하면 복제본에서

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그

		GTID를 이미 사용된 것으로 식별하여 해당 GTID를 공유하는 원치 않는 트랜잭션을 건너뛰어야 합니다. 그러나 구문 분석 오류의 경우 GTID를 건너뛰어야 하는지 여부를 확인하기 전에 문이 구문 분석되므로 의도가 트랜잭션을 건너뛰는 것이라 할지라도 구문 분석 오류로 인해 복제 적용자 스레드가 중지되었습니다. 이 수정에서는 GTID가 이미 사용되었으므로 해당 트랜잭션을 건너뛰어야 하는 경우 복제 적용자 스레드가 구문 분석 오류를 무시합니다. 이 동작 변경은 `mysqlbinlog` 에 의해 생성된 바이너리 로그 출력으로 구성되는 워크로드에는 적용되지 않습니다. 이 상황에서는 건너뛴 트랜잭션 바로 뒤에 오는 트랜잭션에 구문 분석 오류가 있는 경우 오류가 생성되어야 하지만 자동으로 건너뛸 위험이 있습니다. (버그 #27638268)
		• 복제: GTID에 대한 SQL 스레드에서 부분 트랜잭션을 건너뛸 수 있습니다. (버그 #25800025)
		• 복제: `WAIT_UNTIL_SQL_THREAD_AFTER_GTIDS()` 에 제공된 제한 시간 파라미터가 음수 또는 분수인 경우 서버가 예기치 않은 방식으로 동작합니다. 이 수정에서는 다음과 같이 수정됩니다.
		  • 분수 제한 시간 값을 반올림 없이 있는 그대로 읽습니다.
		  • 서버가 엄격한 SQL 모드에 있는 경우 음수 제한 시간 값이 오류와 함께 거부됩니다. 서버가 엄격한 SQL 모드에 있지 않은 경우 함수가 대기 없이 즉시 `NULL`을 반환한 다음 경고를 생성합니다. (버그 #24976304, 버그 #83537)
		• 복제: 분수 값이 포함된 제한 시간(예: 1.5)에 `WAIT_FOR_EXECUTED_GTID_SET()` 함수가 사용된 경우 제한 시간이 근사한 정수(초)로 반내림되고 값이 1초 미만(예: 0.1)인 경우 0으로 반내림되었음을 의미하는 캐스팅 로직 오류가 발생했습니다. 이제 캐스팅 로직이 수정되어 제한 시간 값이 원래 지정된 대로 반올림/반내림 없이 적용됩니다. Dirkjan Bussink 씨의 기여에 감사드립니다. (버그 #29324564, 버그 #94247)
		• GTID를 사용하는 경우 다중 문 트랜잭션 내의 연결이 끊긴 XA 트랜잭션의 XA COMMIT에서 어설션이 발생했습니다. (버그 #22173903)

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
		• 복제: gtid_next 값이 수동으로 설정된 경우 알 수 없는 트랜잭션 식별자에 대해 XA ROLLBACK 문이 실행되면 디버그 빌드에서 어설션이 발생했습니다. 이제 XA ROLLBACK 문이 오류와 함께 실패하는 경우 서버에서 GTID 상태 업데이트를 시도하지 않습니다. (버그 #27928837, 버그 #90640) • 여러 CASE 함수가 ORDER BY 절에 사용되는 경우 잘못된 정렬 순서 문제가 해결됩니다. (버그 #22810883) • 정렬을 사용한 일부 쿼리는 최적화 중에 초기화되지 않은 열에 액세스하여 서버가 종료될 수 있습니다. (버그 #27389294)
Aurora MySQL 데이터베이스 엔진 업데이트 2021년 11월 12일(버전 2.09.3)(사용되지 않음)	2.09.3	• ASSERTION !M_PREBUILT->TRX->CHECK_FOREIGNS. (버그 #23533396) • 복제: * WAIT_FOR_EXECUTED_GTID_SET() 함수의 잠금 문제로 인해 특정 상황에서 서버가 정지될 수 있습니다. 이제 문제가 수정되었습니다. (버그 #29550513)
Aurora MySQL 데이터베이스 엔진 업데이트 2020년 12월 11일(버전 2.09.1)(사용되지 않음)	2.09.1	• 복제: 트랜잭션 격리 수준이 반복 읽기로 설정된 경우 인터리브 트랜잭션이 슬레이브 응용 프로그램이 교착 상태에 빠질 수 있습니다. (버그 #25040331) • 기본값 CURRENT_TIMESTAMP가 있는 TIMESTAMP 또는 DATETIME 열을 포함하는 테이블의 경우 테이블에 BEFORE INSERT 트리거가 있으면 열을 0000-00-00 00:00:00 으로 초기화할 수 있습니다. (버그 #25209512, 버그 #84077) • VALUES 목록에서 조인을 포함하는 하위 쿼리를 사용하여 두 번째 또는 이후 행에 대한 값을 생성하는 삽입문의 경우 필요한 권한을 확인하지 못한 후 서버가 종료될 수 있습니다. (버그 #23762382)

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트 2020년 11월 12일(버전 2.08.3)(사용되지 않음)	2.08.3	• 버그 #23762382 - 선택 항목에 조인된 값 쿼리를 입력하면 잘못된 동작이 발생합니다. • 버그 #25209512 - 현재_타임 스탬프가 트리거에서 0을 생성합니다.
Aurora MySQL 데이터베이스 엔진 업데이트 2020년 6월 2일(버전 2.08.0)(사용되지 않음)	2.08.0	• 버그 #25289359: 전체 텍스트 캐시 크기가 전체 텍스트 캐시 크기 제한을 초과한 경우에 데이터가 동기화될 때 취해진 전체 텍스트 캐시 잠금이 해제되지 않았습니다. • 버그 #29138644: MySQL 서버가 실행되는 동안 시스템 시간을 수동으로 변경하면 페이지 클리너 스레드가 지연됩니다. • 버그 #25222337: 가상 인덱스의 NULL 가상 열 필드 이름으로 인해 외래 키 제약 조건의 영향을 받는 가상 열을 채우면서 필드 이름을 비교하는 동안 서버가 종료되었습니다. • 버그 #25053286: 보기에 액세스한 쿼리가 포함된 저장 프로시저를 실행하면 세션이 종료될 때까지 비워지지 않은 메모리를 할당할 수 있습니다. • 버그 #25586773: 특정 SELECT 문의 내용에서 테이블을 만든 문을 포함하는 저장 프로시저를 실행하면 메모리 누수가 발생할 수 있습니다. • 버그 #28834208: 로그 애플리케이션 중에 OPTIMIZE TABLE 작업 후 InnoDB가 가상 열 인덱스 업데이트를 확인하기 전에 가상 열을 채우지 않았습니다. • 버그 #26666274: 32비트 부호 없는 정수 오버플로우로 인해 성능 스키마 버퍼 컨테이너의 무한 루프가 발생합니다.
Aurora MySQL 데이터베이스 엔진 업데이트 2022년 6월 16일(버전 2.07.8)(사용되지 않음)	2.07.8	UPDATE에 1,024바이트보다 큰 프라이머리 키가 있는 임시 테이블이 필요하고 해당 테이블이 InnoDB를 사용하여 생성된 경우 서버가 종료될 수 있습니다. (버그 #25153670)

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트 2021년 9월 2일(버전 2.07.6)(사용되지 않음)	2.07.6	• 64K 크기의 레코드를 삽입하는 데 너무 많은 시간이 걸립니다. (버그#23031146)
Aurora MySQL 데이터베이스 엔진 업데이트 2021년 3월 4일(버전 2.07.4)(사용되지 않음)	2.07.4	• ' '(공백), '%' 또는 ', '가 포함된 토큰을 처리할 때 전체 텍스트 ngram 파서에서 발생하던 문제가 수정되었습니다. ngram 파서를 사용하는 경우 고객은 FTS 인덱스를 재구축해야 합니다. (버그 #25873310) • 중첩된 SQL 뷰를 사용하여 쿼리를 실행하는 동안 엔진이 재시작될 수 있는 문제가 해결되었습니다. (버그 #27214153, 버그 #26864199)

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트 2020년 11월 10일(버전 2.07.3)(사용되지 않음)	2.07.3	• InnoDB: 마스터의 XA 준비 단계에 성공적으로 실행된 동시 XA 트랜잭션이 슬레이브에서 재생될 때 충돌하여 애플리케이션 스레드에서 잠금 대기 시간 초과가 발생합니다. 충돌은 트랜잭션이 슬레이브에서 순차적으로 재생될 때 다른 GAP 잠금 범위 때문에 발생합니다. 이러한 유형의 충돌을 방지하기 위해 XA 트랜잭션이 준비 단계에 도달하면 READ COMMITTED 격리 수준에서 XA 트랜잭션에 의해 수행된 GAP 잠금이 해제되고 더 이상 상속되지 않습니다. (버그 #27189701, 버그 #25866046) • InnoDB: READ COMMITED 격리 수준을 사용하는 동안 외래 키 유효성 검사 중 불필요하게 간격 잠금이 발생했습니다. (버그 #25082593) • 복제: XA 트랜잭션을 사용할 때 복제 슬레이브의 적용자 (SQL) 스레드에 대한 잠금 대기 시간 초과 또는 교착 상태가 발생한 경우 자동 재시도가 작동하지 않습니다. 원인은 SQL 스레드가 롤백을 수행하는 동안 XA 트랜잭션을 롤백하지 않는다는 것입니다. 즉 트랜잭션을 다시 시도할 때 발생하는 첫 번째 이벤트는 XA 트랜잭션이 이미 진행 중일 때 유효하지 않은 XA START 이며 XAER_RMFAIL 오류가 발생합니다. (버그 #24764800) • 복제: 트랜잭션 격리 수준이 반복 읽기로 설정된 경우 인터리브 트랜잭션이 슬레이브 응용 프로그램이 교착 상태에 빠질 수 있습니다. (버그 #25040331) • 복제: 기존의 모든 릴레이 로그 파일 (Relay_Log_Space) 의 총 합산 크기에 대해 SHOW 슬레이브 상태문에서 반환된 값이 릴레이 로그 파일에 사용된 실제 디스크 공간보다 훨씬 커질 수 있습니다. I/O 스레드가 값을 업데이트하는 동안 변수를 잠그지 않았으므로 SQL 스레드가 릴레이 로그 파일을 자동으로 삭제하고 I/O 스레드가 값 업데이트를 완료하기 전에 감소된 값을 쓸 수 있습니다. 그런 다음 I/O 스레드는 SQL 스레드의 업데이트를 무시하고 삭제된 파일의 공간을 다시 추가하여 원래 크기 계산을 작성했습니다. 이제 Relay_Log_Space 값이 업데이트 중에 잠겨 동시 업데이트를 방지하고 정확한 계산을 보장합니다. (버그 #26997096, 버그 #87832)

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
		• VALUES 목록에서 조인을 포함하는 하위 쿼리를 사용하여 두 번째 또는 이후 행에 대한 값을 생성하는 INSERT 문의 경우 필요한 권한을 확인하지 못한 후 서버가 종료될 수 있습니다. (버그 #23762382) • 기본값 CURRENT_TIMESTAMP가 있는 TIMESTAMP 또는 DATETIME 열을 포함하는 테이블의 경우 테이블에 BEFORE INSERT 트리거가 있으면 열을 `0000-00-00 00:00:00` 으로 초기화할 수 있습니다. (버그 #25209512, 버그 #84077) • 다중 스레드가 메타데이터 성능 스키마 개체를 등록 및 등록 취소하려고 동시에 시도하면 서버 종료가 발생할 수 있습니다. (버그 #26502135) • 특정 SELECT문의 내용에서 테이블을 만든 문을 포함하는 저장 프로시저를 실행하면 메모리 누수가 발생할 수 있습니다. (버그 #25586773) • 보기에 액세스한 쿼리가 포함된 저장 프로시저를 실행하면 세션이 종료될 때까지 비워지지 않은 메모리를 할당할 수 있습니다. (버그 #25053286) • 특정 하위 쿼리 구체화 사례로 인해 서버가 종료될 수 있습니다. 이러한 쿼리는 이제 구체화가 비활성화되었음을 나타내는 오류를 생성합니다. (버그 #26402045) • 버그 #18898433: 조인 버퍼링을 사용하는 경우(예: 블록 중첩 루프 알고리즘 사용) 왼쪽 조인이 많은 쿼리가 느려졌습니다. (버그 #18898433, 버그 #72854) • 최적화 프로그램은 두 번째 열에 대해 LIKE 절이 있는 내부 조인을 실행할 때 복합 인덱스의 두 번째 열을 건너 뜁니다. (버그 #28086754)

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트 2020년 4월 17일(버전 2.07.2)(사용되지 않음)	2.07.2	• 버그 #23104498: 사용된 총 메모리를 보고할 때 성능 스키마의 문제를 수정했습니다(https://github.com/mysql/mysql-server/commit/20b6840df5452f47313c6f9a6ca075bfbc00a96b). • 버그 #22551677: 데이터베이스를 오프라인으로 전환하려고 할 때 데이터베이스 엔진 충돌을 유발할 수 있는 성능 스키마의 문제를 수정했습니다(https://github.com/mysql/mysql-server/commit/05e2386eccd32b6b444b900c9f8a87a1d8d531e9). • 버그 #23550835, 버그 #23298025, 버그 #81464: 내부 버퍼 용량 초과로 인해 데이터베이스 엔진 충돌이 발생하는 성능 스키마의 문제를 수정했습니다(https://github.com/mysql/mysql-server/commit/b4287f93857bf2f99b18fd06f555bbe5b12debfc, https://github.com/mysql/mysql-server/commit/b4287f93857bf2f99b18fd06f555bbe5b12debfc).
Aurora MySQL 데이터베이스 엔진 업데이트 2019년 11월 25일(버전 2.07.0)(사용되지 않음)	2.07.0	• 버그 #26251621: 트리거 및 GCOL로 인한 잘못된 동작 • 버그 #22574695: ASSERTION `!TABLE || (!TABLE->READ_SET || BITMAP_IS_SET(TABLE->READ_SET, FIEL • 버그 #25966845: 복제 키의 INSERT로 인해 교착 상태 발생 • 버그 #23070734: 동시 TRUNCATE 테이블로 인해 중단 발생 • 버그 #26191879: 외래 키 CASCADE에서 과도한 메모리 사용 • 버그 #20989615: INNODB AUTO_INCREMENT에서 동일한 값을 두 번 산출
Aurora MySQL 데이터베이스 엔진 업데이트 2019년 11월 11일(버전 2.05.0)(사용되지 않음)	2.05.0	• 버그#23054591: PURGE BINARY LOGS TO가 전체 binlog 파일을 읽고 있어 MySql이 지연되고 있습니다.

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트 2020년 8월 14일(버전 2.04.9) (사용되지 않음)	2.04.9	• 버그 #23070734, 버그 #80060: 동시 TRUNCATE 테이블로 인해 중단이 발생합니다. • 버그 #23103937: PS_TRUNCATE_ALL_TABLES()이 SUPER_READ_ONLY MODE 모드에서 작동하지 않습니다. • 버그 #22551677: 서버를 오프라인으로 전환할 때 성능 스키마 내의 경합 상태로 인해 서버가 종료될 수 있습니다. • 버그 #27082268: 잘못된 FTS 동기화. • BUG #12589870: 쿼리 캐시가 활성화된 경우 다중 쿼리 문으로 다시 시작되는 문제를 해결했습니다. • 버그 #26402045: 특정 하위 쿼리 구체화 사례로 인해 서버가 종료될 수 있습니다. 이러한 쿼리는 이제 구체화가 비활성화되었음을 나타내는 오류를 생성합니다. • 버그 #18898433: 조인 버퍼링을 사용하는 경우(예: 블록 중첩 루프 알고리즘 사용) 왼쪽 조인이 많은 쿼리가 느려졌습니다. • 버그 #25222337: 가상 인덱스의 NULL 가상 열 필드 이름으로 인해 외래 키 제약 조건의 영향을 받는 가상 열을 채우면서 필드 이름을 비교하는 동안 서버가 종료되었습니다.(https://github.com/mysql/mysql-server/commit/273d5c9d7072c63b6c47dbef6963d7dc491d5131) • 버그 #25053286: 보기에 액세스한 쿼리가 포함된 저장 프로시저를 실행하면 세션이 종료될 때까지 비워지지 않은 메모리를 할당할 수 있습니다.(https://github.com/mysql/mysql-server/commit/d7b37d4d141a95f577916448650c429f0d6e193d) • 버그 #25586773: 특정 SELECT(https://dev.mysql.com/doc/refman/5.7/en/select.html) 문의 내용에서 테이블을 만든 문을 포함하는 저장 프로시저를 실행하면 메모리 누수가 발생할 수 있습니다.(https://github.com/mysql/mysql-server/commit/88301e5adab65f6750f66af284be410c4369d0c1) • 버그 #26666274: 성능 스키마 버퍼 컨테이너의 무한 루프가 발생합니다.

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
		• 버그 #23550835, 버그 #23298025, 버그 #81464: 내부 버퍼가 꽉 찼을 때 SELECT 성능 스키마 테이블로 인해 서버가 종료될 수 있습니다.
Aurora MySQL 데이터베이스 엔진 업데이트 2019년 9월 19일(버전 2.04.6)(사용되지 않음)	2.04.6	• 버그#23054591: PURGE BINARY LOGS TO가 전체 binlog 파일을 읽고 있어 MySql이 지연되고 있습니다.
Aurora MySQL 데이터베이스 엔진 업데이트 2019년 5월 2일(버전 2.04.2)(사용되지 않음)	2.04.2	버그 #24829050 - INDEX_MERGE_INTERSECTION 최적화로 인해 잘못된 쿼리 결과 산출
Aurora MySQL 데이터베이스 엔진 업데이트 2018년 10월 11일(버전 2.03)(사용되지 않음)	2.03	• REVERSE SCAN ON A PARTITIONED TABLE DOES ICP - ORDER BY DESC(버그 #24929748). • JSON_OBJECT CREATES INVALID JSON CODE(버그 #26867509). • INSERTING LARGE JSON DATA TAKES AN INORDINATE AMOUNT OF TIME(버그 #22843444). • PARTITIONED TABLES USE MORE MEMORY IN 5.7 THAN 5.6(버그 #25080442).

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트 2018년 9월 21일(버전 2.02.4) (사용되지 않음)	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

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
		• 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 데이터베이스 엔진 업데이트 2018년 5월 3일(버전 2.02)(사용되지 않음)	2.02.0	좌측 조인이 외부 측에 잘못된 결과를 반환합니다(버그 #22833364)

Aurora MySQL 1.x 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그

MySQL 5.6과 호환되는 버전의 Aurora에는 MySQL 5.6.10을 통한 모든 MySQL 버그 수정이 포함되어 있습니다. 다음 표에는 Aurora MySQL 데이터베이스 엔진 업데이트를 통해 수정한 MySQL 버그와 이 버그들이 어느 업데이트에서 수정되었는지가 나와 있습니다.

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트 2021년 3월 18일(버전 1.23.2)(사용되지 않음)	1.23.2	• 복제: SHOW BINLOG EVENTS 문이 실행되는 동안 모든 병렬 트랜잭션이 차단되었습니다. 이 수정 프로그램은 이제 SHOW BINLOG EVENTS 프로세스가 파일의 끝 위치를 계산하는 동안에만 잠기므로 병렬 트랜잭션이 오랜 기간 차단되지 않도록 합니다. (버그 #76618, 버그 #20928790)
Aurora MySQL 데이터베이스 엔진 업데이트 2020년 9월	1.23.0	• ALTER TABLE ADD COLUMN ALGORITHM=QUICK 이 있는 Binlog 이벤트는 커뮤니티 에디션과 호환되도록 ALGORITHM =DEFAULT 로 다시 작성됩니다.

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
2일(버전 1.23.0)(사용되지 않음)		• 버그 #22350047: 이전 STMTS가 커밋된 저장점으로 롤백한 후 클라이언트가 중단된 경우 • 버그 #29915479: COM_BINLOG_DUMP 없이 COM_REGISTER_SLAVE를 실행하면 서버가 종료될 수 있습니다. • 버그 #30441969: 버그 #29723340: 데이터 세트가 있는 SQL 쿼리 후 MySQL Server 충돌 • 버그 #30628268: 메모리 부족 충돌 • 버그 #27081349: 공간 기능을 사용하여 삭제할 때 예기치 않은 동작 • 버그 #27230859: 잘못된 폴리곤을 처리하는 동안 예기치 않은 동작 • 버그 #27081349: 공간을 사용하여 삭제할 때 예기치 않은 동작 • 버그 #26935001: ALTER TABLE AUTO_INCREMENT가 폐기된 테이블스페이스에서 인덱스를 읽으려고 시도합니다. • 버그 #29770705: 특정 WHERE 절을 사용하여 SELECT를 실행하는 동안 서버가 충돌했습니다. • 버그 #27659490: 동적 범위 및 인덱스 병합을 사용하여 선택하면 너무 많은 메모리를 사용합니다(OOM). • 버그 #24786290: 마스터에서 버그 #74145가 발생한 후 복제가 중단됩니다. • 버그 #27703912: 많은 준비 과정에서 과도한 메모리 사용 • 버그 #20527363: 임시 테이블 충돌 자르기: !DICT_TF2_FLAG_IS_SET(TABLE, DICT_TF2_TEMPORARY) • 버그 #23103937: PS_TRUNCATE_ALL_TABLES()이 SUPER_READ_ONLY 모드에서 작동하지 않습니다. • 버그 #25053286: 절차에서 조건이 있는 뷰를 사용하면 잘못된 동작이 발생합니다(5.6.36에서 수정됨). • 버그 #25586773: SP의 루프에서 테이블 선택 생성에 대한 잘못된 동작(5.6.39에서 수정됨) • 버그 #27407480: AUTOMATIC_SP_PRIVILEGES에는 MySQL 사용자 테이블에 대한 삽입 권한이 필요합니다.

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
		• 버그 #26997096: `relay_log_space` 값이 동기화된 방식으로 업데이트되지 않으므로 값이 릴레이 로그에서 사용되는 실제 디스크 공간보다 훨씬 높습니다. • 버그 #15831300: SLAVE_TYPE_CONVERSIONS=ALL_NON_LOSSY가 예상대로 작동하지 않습니다. • SSL 버그 백포트 버그 #17087862, 버그 #20551271 • 버그 #16894092: 5.6.6 이상에서 INSERT INTO ... SELECT ... FROM에 대한 성능 회귀(5.6.15에서 수정됨) • `SLAVE_TYPE_CONVERSIONS` 와 관련된 버그 수정을 포팅합니다.
Aurora MySQL 데이터베이스 엔진 업데이트 2020년 11월 9일(버전 1.22.3) (사용되지 않음)	1.22.3	• 버그 #26654685: 외래 키 검사 중에 손상된 인덱스 ID가 어설션을 일으켰습니다. • 버그 #15831300: 기본적으로 마스터의 작은 유형에서 슬레이브의 더 큰 유형으로 정수를 프로모션할 때 (예: 마스터의 SmallINT 열에서 슬레이브의 BIGINT 열로) 프로모션된 값은 서명된 것처럼 처리됩니다. 이제 이러한 경우 slave_type_conversions 서버 시스템 변수에 지정된 값 집합에 속한 `ALL_SIGNED` 과 `ALL_UNSIGNED` 를 하나 또는 두 가지 모두 사용하여 이 동작을 수정하거나 재정의할 수 있습니다. 자세한 내용은 행 기반 복제: 특성 수준 프로모션 및 수준 디모션과 변수 설명을 참조하십시오. • 버그 #17449901: `foreign_key_checks=0` 을 사용할 때 InnoDB는 외래 키 제약 조건에 필요한 인덱스를 삭제하여 테이블을 일관성 없는 상태로 배치하고 테이블 로드시 발생하는 외래 키 검사가 실패하는 원인을 제공했습니다. InnoDB는 이제 외래 키 제약 조건에 필요한 인덱스를 삭제하는 것을 방지하며 foreign_key_checks=0도 포함합니다. 외래 키 인덱스를 삭제하기 전에 외래 키 제약 조건을 제거해야 합니다. • 버그 #20768847: 테이블 변경 ... 외래 키 종속성이 있는 테이블에 대한 DROP INDEX 작업은 어설션을 일으켰습니다.

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트 2019년 11월 25일(버전 1.22.0)(사용되지 않음)	1.22.0	• 버그 #16346241 - ITEM_PARAM::QUERY_VAL_STR의 서버 충돌 • 버그 #17733850 - ITEM_NAME_CONST::ITEM_NAME_CONST()의 NAME_CONST() 충돌 • 버그 #20989615: INNODB AUTO_INCREMENT에서 동일한 값을 두 번 산출 • 버그 #20181776 - 액세스 제어에 와일드카드가 포함된 경우 가장 제한적인 호스트와 일치하지 않음 • Bug #27326796 - MYSQL이 PARS0PARS.CC 파일의 INNODB ASSERTION 실패와 충돌 • Bug #20590013 - FULLTEXT 인덱스가 있는데 이를 삭제하면 더 이상 온라인 DDL을 수행할 수 없음
Aurora MySQL 데이터베이스 엔진 업데이트 2019년 11월 25일(버전 1.21.0)(사용되지 않음)	1.21.0	• Bug #19929406: HANDLE_FATAL_SIGNAL (SIG=11) IN __MEMMOVE_SSSE3_BACK FROM STRING::COPY • Bug #17059925: UNION 문에서 행 검토 값이 잘못 계산되었습니다. 이러한 오류는 성능 스키마 문 테이블(예: events_statements_current)의 `ROWS_EXAMINED` 열 값이 너무 커지는 것으로 나타났습니다. • Bug #11827369: `SELECT ... FROM DUAL` 중첩 하위 쿼리가 포함된 일부 쿼리로 인해 어설션이 발생하였습니다. • 버그 #16311231: `IN` 절에 XOR 작업이 포함된 `WHERE` 절에 하위 쿼리가 들어 있는 쿼리의 경우 잘못된 결과가 반환되었습니다.

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트 2019년 11월 11일(버전 1.20.0)(사용되지 않음)	1.20.0	• Bug #19929406: HANDLE_FATAL_SIGNAL (SIG=11) IN __MEMMOVE_SSSE3_BACK FROM STRING::COPY • Bug #17059925: UNION 문에서 행 검토 값이 잘못 계산되었습니다. 이러한 오류는 성능 스키마 문 테이블(예: events_statements_current)의 ROWS_EXAMINED 열 값이 너무 커지는 것으로 나타났습니다. • Bug #11827369: SELECT ... FROM DUAL 중첩 하위 쿼리가 포함된 일부 쿼리로 인해 어설션이 발생하였습니다. • 버그 #16311231: IN 절에 XOR 작업이 포함된 WHERE 절에 하위 쿼리가 들어 있는 쿼리의 경우 잘못된 결과가 반환되었습니다.
Aurora MySQL 데이터베이스 엔진 업데이트 2019년 9월 19일(버전 1.19.5)(사용되지 않음)	1.19.5	• CVE-2018-2696 • CVE-2015-4737 • Bug #19929406: HANDLE_FATAL_SIGNAL (SIG=11) IN __MEMMOVE_SSSE3_BACK FROM STRING::COPY • Bug #17059925: UNION 문에서 행 검토 값이 잘못 계산되었습니다. 이러한 오류는 성능 스키마 문 테이블(예: events_statements_current)의 ROWS_EXAMINED 열 값이 너무 클 때 명백하게 발생하였습니다. • Bug #11827369: SELECT ... FROM DUAL 중첩 하위 쿼리가 포함된 일부 쿼리로 인해 어설션이 발생하였습니다. • Bug #16311231: IN 절에 XOR 작업이 포함된 WHERE 절에 하위 쿼리가 들어 있는 쿼리의 경우 잘못된 결과가 반환되었습니다.
Aurora MySQL 데이터베이스 엔진 업데이트 2019년 2월 7일(버전 1.19.0)(사용되지 않음)	1.19.0	• 버그 #32917: ORPHAN TEMP-POOL 파일을 감지하여 정상적으로 처리 • 버그 #63144: 존재하지 않는 경우 테이블 생성, 메타데이터 잠금이 너무 제한적임

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트 2019년 1월 17일(버전 1.17.8) (사용되지 않음)	1.17.8	• 버그 #13418638: 존재하지 않는 경우 테이블 생성, 메타데이터 잠금이 너무 제한적임
Aurora MySQL 데이터베이스 엔진 업데이트 2018년 10월 8일(버전 1.17.7) (사용되지 않음)	1.17.7	• 누락 테이블을 초래하는 외래 키 열에서 인덱스 삭제. (버그 #16208542) • add_derived_key()의 메모리 누수. (버그 #76349) • 파티션 분할된 테이블의 경우, 인덱스 병합 사용 여부에 따라 쿼리가 다른 결과를 반환할 수 있습니다. (버그 #16862316) • 인덱스 병합 최적화(인덱스 병합 최적화 참조)를 사용하는 쿼리를 HASH로 파티션 분할된 테이블에서 실행할 경우 잘못된 결과가 반환될 수 있습니다. (버그 #17588348)
Aurora MySQL 데이터베이스 엔진 업데이트 2018년 9월 6일(버전 1.17.6)(사용되지 않음)	1.17.6	• BINARY 열의 이름을 변경하거나 기본값을 변경한 ALTER TABLE 문의 경우, 인플레이스가 아닌 테이블 복사를 사용하여 변경이 수행되었습니다. (버그 #67141, 버그 #14735373, 버그 #69580, 버그 #17024290) • 그룹인 정규 테이블과 파생 테이블 간의 외부 조인으로 인해 서버 종료가 발생할 수 있습니다. (버그 #16177639)

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트 2018년 3월 13일(버전 1.17)(사용되지 않음)	1.17.0	• 복제 필터가 사용되는 경우 LAST_INSERT_ID가 올바르지 않게 복제됩니다(버그 #69861). • 쿼리가 INDEX_MERGE 설정에 따라 다른 결과를 반환합니다(버그 #16862316). • 저장된 루틴의 쿼리 처리 재실행, 비효율적인 쿼리 계획(버그 #16346367) • InnoDB FTS: FTS_CACHE_APPEND_DELETED_DOC_IDS에 어설션합니다(버그 #18079671). • RBT_EMPTY(INDEX_CACHE->WORDS)를 ALTER TABLE 변경 열에 어설션합니다(버그 #17536995). • 저장점이 연관되었을 경우 InnoDB Fulltext 검색으로 레코드를 찾지 못합니다(버그 #70333, 버그 #17458835).
Aurora MySQL 데이터베이스 엔진 업데이트 2017년 11월 20일(버전 1.15.1)(사용되지 않음)	1.15.1	• 되돌림 — MySQL 인스턴스에서 "SYNC 인덱스 실행"이 지연됩니다(버그 #73816). • 되돌림 — RBT_EMPTY(INDEX_CACHE->WORDS)를 ALTER TABLE 변경 열에 어설션합니다(버그 #17536995). • 되돌림 — 저장점이 연관되었을 경우 InnoDB Fulltext 검색으로 레코드를 찾지 못합니다(버그 #70333).

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트 2017년 10월 24일(버전 1.15)(사용되지 않음)	1.15.0	• CREATE USER가 플러그인 및 암호 해시를 허용하지만 암호 해시는 무시합니다(버그 #78033). • 파티션 분할 엔진은 분할된 인덱스에서 정렬된 항목을 반환할 수 있도록 여러 필드를 읽기 비트 집합에 추가합니다. 이는 불필요한 필드까지 읽으려고 하면서 조인 버퍼의 원인이 됩니다. 분할 필드를 모두 read_set에 추가하지 않는 대신 read_set에서 이미 설정된 접두사 필드를 기준으로 정렬하여 버퍼 문제를 수정하였습니다. key_cmp를 실행하는 경우 첫 번째 필드를 읽어야 하도록 DBUG_ASSERT가 추가되었습니다(버그 #16367691). • MySQL 인스턴스에서 "SYNC 인덱스 실행"이 지연됩니다(버그 #73816). • RBT_EMPTY(INDEX_CACHE->WORDS)를 ALTER TABLE 변경 열에 어설션합니다(버그 #17536995). • 저장점이 연관되었을 경우 InnoDB Fulltext 검색으로 레코드를 찾지 못합니다(버그 #70333).
Aurora MySQL 데이터베이스 엔진 업데이트: 2018년 3월 13일(버전 1.14.4)(사용되지 않음)	1.14.4	• 무시할 수 있는 이벤트는 유효하지 않기 때문에 테스트 대상이 아닙니다(버그 #74683). • NEW->OLD ASSERT FAILURE 'GTID_MODE > 0'(버그 #20436436)
Aurora MySQL 데이터베이스 엔진 업데이트: 2017년 8월 7일(버전 1.14)(사용되지 않음)	1.14.0	이전에는 전체 텍스트 검색이 파생 테이블(FROM 절의 하위 쿼리)과 결합되면서 서버 종료의 원인이 되었습니다. 하지만 이제는 전체 텍스트 작업이 파생 테이블에 따라 결정되는 경우 구체화된 테이블에서 전체 텍스트 검색을 실행할 수 없다는 오류가 서버에서 발생합니다. (버그 #68751, 버그 #16539903)

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트: 2017년 5월 15일(버전 1.13)(사용되지 않음)	1.13.0	• 비어 있는 상태로 인해 AUTO_INCREMENT 값이 재설정되는 동안 제거된 테이블을 재로드합니다. (버그 #21454472, 버그 #77743) • purge_node_t 구조의 불일치로 인해 인덱스 기록이 롤백에서 발견되지 않았습니다. 이러한 불일치로 "보조 인덱스 입력 업데이트 오류", "기록을 제거할 수 없음", "삭제 표시되지 않은 보조 인덱스 입력 제거를 시도함" 등의 경고 및 오류 메시지가 표시되었습니다. (버그 #19138298, 버그 #70214, 버그 #21126772, 버그 #21065746) • qsort 작업에 대한 스택 크기 계산을 잘못하면 스택 오버플로우가 발생합니다. (버그 #73979) • 롤백 시 인덱스에서 기록이 발견되지 않았습니다. (버그 #70214, 버그 #72419) • 업데이트 CURRENT_TIMESTAMP의 ALTER TABLE 추가 열 TIMESTAMP가 ZERO-datas 삽입(버그 #17392)
Aurora MySQL 데이터베이스 엔진 업데이트: 2017년 4월 5일(버전 1.12)(사용되지 않음)	1.12.0	• 비어 있는 상태로 인해 AUTO_INCREMENT 값이 재설정되는 동안 제거된 테이블을 재로드합니다. (버그 #21454472, 버그 #77743) • purge_node_t 구조의 불일치로 인해 인덱스 기록이 롤백에서 발견되지 않았습니다. 이러한 불일치로 "보조 인덱스 입력 업데이트 오류", "기록을 제거할 수 없음", "삭제 표시되지 않은 보조 인덱스 입력 제거를 시도함" 등의 경고 및 오류 메시지가 표시되었습니다. (버그 #19138298, 버그 #70214, 버그 #21126772, 버그 #21065746) • qsort 작업에 대한 스택 크기 계산을 잘못하면 스택 오버플로우가 발생합니다. (버그 #73979) • 롤백 시 인덱스에서 기록이 발견되지 않았습니다. (버그 #70214, 버그 #72419) • 업데이트 CURRENT_TIMESTAMP의 ALTER TABLE 추가 열 TIMESTAMP가 ZERO-datas 삽입(버그 #17392)

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트: 2017년 2월 23일(버전 1.11)(사용되지 않음)	1.11.0	• ALTER 테이블 DROP 외래 키를 다른 DROP 연산과 동시에 실행하면 테이블이 사라집니다. (버그 #16095573) • ORDER BY를 사용한 일부 INFORMATION_SCHEMA 쿼리가 예전처럼 파일 정렬 최적화를 사용하지 않습니다. (버그 #16423536) • FOUND_ROWS ()가 잘못된 테이블 행 수를 반환합니다 (버그 #68458) • 임시 테이블을 너무 많이 열면 오류가 발생하는 대신 서버가 고장납니다. (버그 #18948649)
Aurora MySQL 데이터베이스 엔진 업데이트: 2016년 12월 14일(버전 1.10)(사용되지 않음)	1.10.0	• 파생된 테이블의 UNION이 '1=0/false' 절이 있는 잘못된 결과를 반환합니다. (버그 #69471) • 저장된 프로시저 2차 실행 시 ITEM_FUNC_GROUP_CONCAT::FIX_FIELDS에서 서버가 충돌합니다. (버그 #20755389) • 캐시 크기가 전체 크기의 10%를 넘어서자마자 별도 스레드에 캐시 동기화 작업을 오프로드하여 FTS 캐시 동기화 중에 MySQL 쿼리가 너무 오래 중지되는 일을 방지합니다. (버그 #22516559, #73816)
Aurora MySQL 데이터베이스 엔진 업데이트: 2016년 10월 26일(버전 1.8.1)(사용되지 않음)	1.8.1	• LogJam 문제 때문에 OpenSSL이 Diffie-Hellman 키 길이 파라미터를 변경했습니다. (버그 #18367167)

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트: 2016년 10월 18일(버전 1.8) (사용되지 않음)	1.8.0	• 여러 인덱스를 포함하는 하나의 열에 모든 인덱스를 끌어서 놓은 경우 외래 키 제약 조건에서 인덱스를 요구할 때 InnoDB에서 DROP INDEX 작업을 차단하지 못합니다. (버그 #16896810) • 외래 키 제약 조건 추가 충돌을 해결합니다. (버그 #16413976) • 저장 프로시저에서 커서를 가져오는 동시에 테이블을 분석하거나 플러시할 때 발생하는 충돌을 해결했습니다. (버그 #18158639) • 사용자가 테이블에서 AUTO_INCREMENT 값을 최대 자동 증분 열 값보다 작게 변경할 때 발생하는 자동 증분 버그를 해결했습니다. (버그 #16310273)

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트: 2016년 8월 30일(버전 1.7.0)(사용되지 않음)	1.7.0	• LOCK_grant 잠금을 분할하여 확장성을 향상했습니다. (포트 WL #8355) • 저장 프로시저의 SELECT에 커서를 두면 segfault가 발생합니다. (포트 버그 #16499751) • MySQL에서는 일부 특수한 경우에 잘못된 결과를 제공합니다. (버그 #11751794) • #11751794 버그에 대한 패치에 의해 GET_SEL_ARG_FOR_KEYPART에서– 충돌이 발생합니다. (버그 #16208709) • GROUP BY를 통한 간단한 쿼리에 대해 잘못된 결과가 표시됩니다. (버그 #17909656) • 범위 조건자를 통한 semijoin 쿼리에서 추가 행이 표시됩니다. (버그 #16221623) • IN 하위 쿼리 뒤에 ORDER BY 절을 추가하면 중복 행이 반환될 수 있습니다. (버그 #16308085) • 쿼리에 대한 설명이 GROUP BY, MyISAM에 대한 간략 스캔과 충돌합니다. (버그 #16222245) • 인용된 int 조건자를 사용하여 느슨한 인덱스 스캔을 수행하면 임의의 데이터가 반환됩니다. (버그 #16394084) • 최적화 프로그램에서 느슨한 인덱스 스캔을 사용 중인 경우 임시 테이블을 생성하려고 하면 서버가 종료될 수 있습니다. (버그 #16436567) • COUNT(DISTINCT)는 NULL 값을 계산하지 않지만 최적화 프로그램에서 느슨한 인덱스 스캔을 사용하는 경우에는 계산됩니다. (버그 #17222452) • 쿼리에 MIN()/MAX() 및 aggregate_function(DISTINCT)이 모두 포함되어 있고(예: SUM(DISTINCT)) 느슨한 인덱스 스캔을 사용하여 쿼리를 실행한 경우 MIN()/MAX()의 결과 값이 잘못 설정되었습니다. (버그 #17217128)

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트: 2016년 6월 1일(버전 1.6.5)(사용되지 않음)	1.6.5	• SLAVE CAN'T CONTINUE REPLICATION AFTER MASTER'S CRASH RECOVERY (Port Bug #17632285)
Aurora MySQL 데이터베이스 엔진 업데이트: 2016년 4월 6일(버전 1.6)(사용되지 않음)	1.6.0	• 백포트 버그 #18694052 5.6에서 `!M_ORDERED_REC_BUFFER' 어설션 실패 수정(포트 버그 #18305270) • MEMCPY(), HA_PARTITION::POSITION의 SEGV 오류(포트 버그 # 18383840) • 파티셔닝, INDEX_MERGE 및 NO PK에서 잘못된 결과 도출(포트 버그 # 18167648) • FLUSH TABLES FOR EXPORT: HA_PARTITION::EXTRA의 어설션 오류(포트 버그 # 16943907) • 가상 HA_ROWS HANDLER::MULTI_RANGE_READ_INFO_CONST에서 서버 충돌(포트 버그 # 16164031) • SEL_ARG::RB_INSERT()에서 범위 최적화 프로그램 충돌(포트 버그 # 16241773)
Aurora MySQL 데이터베이스 엔진 업데이트: 2016년 1월 11일(버전 1.5)(사용되지 않음)	1.5.0	• 데이터베이스 이름이 숫자로 시작하는 테이블에 영향을 주는 MySQL 전체 텍스트 검색 기능이 완전히 해결되지 않았던 문제를 해결했습니다. (포트 버그 #17607956)

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트: 2015년 12월 3일(버전 1.4)(사용되지 않음)	1.4	• FTSPARSE()의 SEGV 오류 (버그 #16446108) • 열의 이름을 바꾸는 동안 InnoDB 데이터 사전이 업데이트되지 않습니다. (버그 #19465984) • 테이블 이름을 다른 데이터베이스로 변경하면 FTS 충돌이 발생합니다. (버그 #16834860) • 잘린 테이블에서 트리거를 준비하지 못하면 오류 1054가 발생합니다. (버그 #18596756) • 메타데이터를 변경하면 트리거 실행에 문제가 발생할 수 있습니다. (버그 #18684393) • 길이가 긴 UTF8 VARCHAR 필드에 구제화가 선택되지 않습니다. (버그 #17566396) • 제한 사항 X가 있는 ORDER BY 실행 시 실행 계획의 성능 저하. (Bug #16697792) • 백포트(Backport) 버그#11765744 - 5.1, 5.5 및 5.6. (버그 #17083851) • SQL/SQL_SHOW.CC에서의 뮤텍스 문제로 SIG6가 발생합니다. 소스는 FILL_VARIABLES와 유사합니다. (버그 #20788853) • 백포트(Backport) 버그 #18008907 - 5.5 이상 버전 (버그 #18903155) • MySQL 5.7에서 스택 오버플로우 오류에 대해 수정 적용 (버그 #19678930)

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트: 2015년 10월 16일(버전 1.2, 1.3)(사용되지 않음)	1.2, 1.3	• innodb 내의 쿼리 삭제 시 결과적으로 어설션과의 충돌 발생 (버그 #1608883) • 이벤트 스케줄러, 이벤트 실행 또는 새로운 연결에 대해 새로운 스레드를 생성하지 못한 경우 오류 로그에 아무 메시지도 작성되지 않습니다. (버그 #16865959) • 한 연결이 기본 데이터베이스를 변경하고 동시에 다른 연결이 SHOW PROCESSLIST를 수행하는 경우, 첫 번째 연결의 기본 데이터베이스 메모리를 표시하려고 할 때 두 번째 연결에서 잘못된 메모리에 액세스할 수 있습니다. (버그 #11765252) • 설계 상 PURGE BINARY LOGS는 사용 중이거나 활성화 상태의 이진 로그 파일을 제거하지 않지만, 이러한 경우에도 아무런 메시지가 표시되지 않습니다. (버그 #13727933) • 일부 문에서는 최적화 프로그램이 필요 없는 하위 쿼리 절을 제거하는 경우 메모리 누수가 발생할 수 있습니다. (버그 #15875919) • 종료 중에 서버가 초기화되지 않은 뮤텍스를 잠그려고 시도할 수 있습니다. (버그 #16016493) • GROUP_CONCAT() 및 ORDER BY 절을 사용하여 여러 열의 이름을 지정하는 준비된 문을 사용하면 서버가 종료될 수 있습니다. (버그 #16075310) • 복제본 작업자 스레드에 성능 스키마 장비가 누락되었습니다. (버그 #16083949) • `STOP SLAVE`는 하나 이상의 상태 변수 `Slave_retried_transactions`, `Slave_heartbeat_period`, `Slave_received_heartbeats`, `Slave_last_heartbeat` 또는 `Slave_running`에 대한 값을 검색한 SHOW SATUS와 같은 문과 동시에 실행될 때 교착 상태가 발생할 수 있습니다. (버그 #16088188) • 검색 용어에 따옴표가 있는 경우 부울 모드를 사용한 전체 텍스트 쿼리에서 0이 반환될 수 있습니다. (버그 #16206253)

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
		• 하위 쿼리에서 조인의 ON 절에 하위 쿼리가 있는 준비된 문을 실행하는 경우 최적화 프로그램이 중복되는 하위 쿼리 절을 제거하려고 시도하면 어설션이 발생합니다. (버그 #16318585) • GROUP_CONCAT 불안정성, ITEM_SUM::CLEAN_UP _AFTER_REMOVAL에서 충돌 발생 (버그 #16347450) • INFORMATION_SCHEMA.INNODB_FT_DEFAULT _STOPWORD와 동일한 구조의 InnoDB 테이블을 생성하여 기본 InnoDB 전체 텍스트 검색(FTS) 불용어 목록을 변경하려고 시도하면 오류가 발생합니다. (버그 #16373868) • 작업자의 클라이언트 스레드에서 FLUSH TABLES WITH READ LOCK을 수행한 후 마스터에서 일부 업데이트를 수행하면 `SHOW SLAVE STATUS`를 실행할 때 작업자가 멈춥니다. (버그 #16387720) • 전체 텍스트 검색에서 "abc-def"와 같은 구분된 검색 문자열을 파싱할 때 이제 InnoDB에서 MyISAM과 동일한 단어 구분자를 사용합니다. (버그 #16419661) • FTS_AST_TERM_SET_WILDCARD에서 충돌이 발생합니다. (버그 #16429306) • FTS RQG 테스트 시 FTS_AST_VISIT()에서 SEGFAULT가 발생합니다. (버그 #16435855) • 빌드 디버그 시 최적화 프로그램이 하위 쿼리를 가리키는 Item_ref를 제거하면 서버가 종료됩니다. (버그 #16509874) • InnoDB 테이블에서 전체 텍스트 검색으로 + 또는 - 연산자와 결합된 리터럴 문구를 검색하면 오류가 발생합니다. (버그 #16516193) • 서버가 `START SLAVE` relay-log-info-repository=TABLE 옵션을 사용하고 자동 커밋을 0으로 설정하고 `--master-info-repository=TABLE` 를 함께 사용하여 시작되면 `--skip-slave-start` 에서 오류가 발생합니다. (버그 #16533802) • InnoDB 전체 텍스트 검색(FTS)의 크기가 매우 크면 과도한 메모리 양을 소비할 수 있습니다. (버그 #16625973)

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
		• 검색 문자열에 이진을 직접 사용하면 이진에 NULL 바이트 및 다른 의미 없는 문자가 포함될 수 있으므로 빌드 디버그 시 OPT_CHECK_ORDER_BY에서 어설션이 발생할 수 있습니다. (버그 #16766016) • 일부 문에서는 최적화 프로그램이 필요 없는 하위 쿼리 절을 제거하는 경우 메모리 누수가 발생할 수 있습니다. (버그 #16807641) • 작업자에 대한 새로운 연결에서 `STOP SLAVE`를 실행한 다음, 원래 연결을 사용하여 `SHOW SLAVE STATUS`를 실행하여 FLUSH TABLES WITH READ LOCK을 실행하면 교착 상태가 발생할 가능성이 있습니다. (버그 #16856735) • 잘못된 구분 기호와 함께 GROUP_CONCAT()를 실행하면 서버가 종료될 수 있습니다. (버그 #16870783) • 패턴이 해당 뮤텍스(`Slave_heartbeat_period` , `Slave_last_heartbeat` , `Slave_received_heartbeats` , `Slave_retried_transactions` , `Slave_running`)를 사용하는 상태 변수와 일치하지 않는 경우에도 서버가 SHOW STATUS LIKE '패턴' 문에 대해 LOCK_active_mi 및 active_mi->rli->data_lock 뮤텍스에서 과도한 잠금을 설정했습니다. (버그 #16904035) • IN BOOLEAN MODE 수정자를 사용하여 전체 텍스트 검색을 실행하면 어설션 오류가 발생합니다. (버그 #16927092) • InnoDB 테이블에서 전체 텍스트 검색으로 + 부울 연산자를 사용하여 검색하면 오류가 발생합니다. (버그 #17280122) • 4웨이 교착 상태: 좀비, binlog 삭제, 프로세스 목록 표시, binlog 표시 (버그 #17283409) • 커밋 잠금을 기다리고 있는 SQL 스레드가 중단되었다가 다시 시작되는 경우 트랜잭션이 작업자에서 건너 뛰게 됩니다. (버그 #17450876) • "종료되지 않는" 토큰으로 인해 InnoDB 전체 텍스트 검색 오류가 발생합니다. 문자열 및 문자열 길이가 문자열 비교를 위해 전달되어야 합니다. (버그 #17659310)

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
		• 많은 수의 분할된 InnoDB 테이블이 MySQL 5.6 또는 5.7에서 사용될 때 이전 릴리스의 MySQL Server에서 사용될 때보다 동일한 테이블에서 사용하는 메모리 양이 많이 증가될 수 있습니다. (버그 #17780517) • 전체 텍스트 쿼리의 경우 num_token이 max_proximity_item보다 적은지 확인하지 못하여 어설션이 발생할 수 있습니다. (버그 #18233051) • 비어 있는 InnoDB 테이블이 많이 있는 경우 INFORMATION_SCHEMA 테이블 및 COLUMNS 테이블에 대한 특정 쿼리가 과도한 메모리 사용을 초래할 수 있습니다. (버그 #18592390) • 트랜잭션을 커밋할 때 이제 플래그가 사용되어 스레드 자체를 확인하지 않고 스레드 생성 여부를 확인합니다. 이에 따라 특히 master_info_repository=TABLE과 함께 서버를 실행할 때 더 많은 리소스가 사용됩니다. (버그 #18684222) • 마스터가 DML을 실행하는 동안 작업자에 있는 클라이언트 스레드가 FLUSH TABLES WITH READ LOCK을 실행하는 경우, 동일한 클라이언트에 있는 `SHOW SLAVE STATUS`를 실행하면 차단되어 교착 상태가 발생합니다. (버그 #19843808) • GROUP_CONCAT()로 주문하면 서버가 종료될 수 있습니다. (버그 #19880368)

데이터베이스 엔진 업데이트	버전	수정된 MySQL 버그
Aurora MySQL 데이터베이스 엔진 업데이트: 2015년 8월 24일(버전 1.1)(사용되지 않음)	1.1	• 이름이 숫자로 시작하는 InnoDB 데이터베이스가 전체 텍스트 검색(FTS) 구문 분석기 오류를 일으킵니다. (버그 #17607956) • 이름이 숫자로 시작하는 데이터베이스에서 InnoDB 전체 텍스트 검색이 실패합니다. (버그 #17161372) • Windows에서 실행되는 InnoDB 데이터베이스의 경우, 전체 텍스트 검색(FTS) 객체 ID가 예상되는 16진수 형식이 아닙니다. (버그 #16559254) • MySQL 5.6에 도입된 코드 회귀가 DROP TABLE 및 ALTER TABLE 성능에 악영향을 미쳤습니다. 이로 인해 MySQL Server 5.5.x와 5.6.x 사이에서 성능 저하가 발생했을 수 있습니다. (버그 #16864741)

Aurora MySQL에서 수정된 보안 취약성

일반적인 취약성 및 노출도(CVE)는 공개적으로 알려진 사이버 보안 취약성에 대한 항목의 목록입니다. 각 항목에는 식별 번호, 설명 및 하나 이상의 공개 참조가 포함되어 있습니다.

이 페이지에서 Aurora MySQL에서 수정된 보안 취약성 목록을 찾을 수 있습니다. 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora의 보안을 참조하세요. Aurora MySQL의 보안에 대한 자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL를 사용한 보안을 참조하세요.

알려진 취약성으로부터 보호하려면 항상 최신 Aurora 릴리스로 업그레이드하는 것이 좋습니다. 이 페이지를 사용하여 Aurora MySQL의 특정 버전에 특정 보안 취약성에 대한 수정 사항이 있는지 확인할 수 있습니다. 클러스터에 보안 수정이 없는 경우 해당 수정을 위해 업그레이드해야 하는 Aurora MySQL 버전을 확인할 수 있습니다.

Aurora MySQL 버전 1, 2, 3에서 수정된 CVE는 해당 버전의 릴리스 정보에도 나열됩니다.

- Amazon Aurora MySQL 버전 3에 대한 데이터베이스 엔진 업데이트
- Amazon Aurora MySQL 버전 2에 대한 데이터베이스 엔진 업데이트
- Amazon Aurora MySQL 버전 1에 대한 데이터베이스 엔진 업데이트(사용되지 않음)

> **Note**
>
> Aurora MySQL 버전 3의 초기 릴리스에는 커뮤니티 MySQL 8.0.23에 맞게 수정된 모든 CVE가 포함되어 있습니다. 수정된 향후 CVE의 경우 여기 및 Aurora MySQL 버전 3 릴리스 노트에 나열된 CVE를 찾아보세요.

CVE와 수정된 최소 Aurora MySQL 버전

- CVE-2024-37371: 3.08.0
- CVE-2024-25062: 3.08.0, 2.12.4
- CVE-2024-21185: 3.08.0
- CVE-2024-21179: 3.08.0
- CVE-2024-21177: 3.08.0, 2.12.4
- CVE-2024-21173: 3.08.0

- CVE-2024-21166: 3.08.0
- CVE-2024-21165: 3.08.0
- CVE-2024-21163: 3.08.0
- CVE-2024-21162: 3.08.0
- CVE-2024-21160: 3.08.0
- CVE-2024-21159: 3.08.0
- CVE-2024-21157: 3.08.0
- CVE-2024-21142: 3.08.0, 2.12.4
- CVE-2024-21137: 3.08.0
- CVE-2024-21135: 3.08.0
- CVE-2024-21134: 3.08.0
- CVE-2024-21130: 3.08.0
- CVE-2024-21129: 3.08.0
- CVE-2024-21127: 3.08.0
- CVE-2024-21125: 3.08.0
- CVE-2024-21097: 2.12.3, 2.11.6
- CVE-2024-21096: 2.12.3, 2.11.6
- CVE-2024-21069: 2.12.3, 2.11.6
- CVE-2024-21062: 2.12.3, 2.11.6
- CVE-2024-21057: 2.12.3, 2.11.6
- CVE-2024-21055: 2.12.3, 2.11.6
- CVE-2024-21054: 2.12.3, 2.11.6
- CVE-2024-21047: 2.12.3, 2.11.6
- CVE-2024-21013: 3.08.0, 2.12.3, 2.11.6
- CVE-2024-21009: 2.12.3, 2.11.6
- CVE-2024-21008: 2.12.3, 2.11.6
- CVE-2024-21000: 3.08.0
- CVE-2024-20998: 2.12.3, 2.11.6
- CVE-2024-20996: 3.08.0
- CVE-2024-20993: 2.12.3, 2.11.6

- CVE-2024-20963: 2.12.2, 2.11.6, 2.11.5
- CVE-2024-0853: 3.08.0, 3.06.1, 3.04.3, 2.12.3, 2.11.6
- CVE-2023-44487: 3.08.0, 3.06.1, 2.12.4, 2.12.3, 2.11.6
- CVE-2023-39975: 3.07.0, 3.06.0, 3.05.2, 3.04.2, 2.12.2, 2.11.6, 2.11.5
- CVE-2023-38546: 3.07.0, 3.06.0, 3.04.2, 2.11.5
- CVE-2023-38545: 3.07.0, 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, 2.11.6, 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, 2.12.3, 2.12.0, 2.11.6, 2.11.3
- CVE-2023-21840: 2.12.0
- CVE-2023-0215: 3.04.0, 3.03.2, 2.12.0, 2.11.3
- CVE-2022-43551: 3.04.0, 3.03.2, 2.12.0, 2.11.3
- CVE-2022-37434: 3.05.0, 3.04.0, 3.03.2, 2.12.0, 2.11.3
- CVE-2022-32221: 3.03.0, 2.12.0, 2.11.1, 2.07.9
- CVE-2022-24407: 2.12.1, 2.11.4
- CVE-2022-21635: 3.04.0
- CVE-2022-21556: 3.04.0
- CVE-2022-21460: 2.11.0
- CVE-2022-21451: 3.03.0, 3.02.2, 2.11.0
- CVE-2022-21444: 3.03.0, 3.02.2, 2.11.0, 2.10.3
- CVE-2022-21417 : 2.11.0
- CVE-2022-21352: 3.04.0
- CVE-2022-21344: 2.10.3
- CVE-2022-21304: 2.11.0, 2.10.3
- CVE-2022-21303: 2.11.0
- CVE-2022-21245: 2.11.0, 2.10.3, 2.07.8

- CVE-2022-0778: 3.02.1, 2.11.0
- CVE-2021-36222: 3.02.2, 3.01.1, 2.12.0, 2.11.1, 2.11.0, 2.10.3, 2.10.2, 2.07.8
- CVE-2021-35630: 3.04.0
- CVE-2021-35624: 3.04.0, 2.10.2
- CVE-2021-35604: 2.10.2
- CVE-2021-28196: 2.11.0
- CVE-2021-23841: 2.11.0, 2.10.0, 2.09.3, 1.23.3
- CVE-2021-22946: 3.02.0, 3.01.1, 2.12.0
- CVE-2021-22926: 3.02.2, 3.01.1, 2.11.1, 2.11.0, 2.10.3, 2.10.22.07.8
- CVE-2021-3712: 2.09.3
- CVE-2021-3449: 2.11.0, 2.10.0, 2.09.3, 1.23.3
- CVE-2021-2390: 2.10.2
- CVE-2021-2389: 2.10.2
- CVE-2021-2385: 2.10.2
- CVE-2021-2356: 2.10.2
- CVE-2021-2307: 2.11.0, 2.10.1, 2.09.3, 1.23.4
- CVE-2021-2226: 2.11.0, 2.10.1, 2.09.3, 1.23.4
- CVE-2021-2202: 2.11.0
- CVE-2021-2194: 2.11.0, 2.10.1
- CVE-2021-2179: 2.11.0
- CVE-2021-2178: 2.11.0
- CVE-2021-2174: 2.11.0, 2.10.1, 2.09.3
- CVE-2021-2171: 2.11.0, 2.10.1, 2.09.3
- CVE-2021-2169: 2.12.0, 2.11.1, 2.11.0, 2.10.1, 2.09.3
- CVE-2021-2166: 2.11.0, 2.10.1, 2.09.3
- CVE-2021-2160: 2.11.0, 2.10.1, 1.23.4
- CVE-2021-2154: 2.11.0, 2.10.1, 2.09.3, 1.23.4
- CVE-2021-2144: 2.07.3
- CVE-2021-2060: 2.10.1, 2.09.3, 1.23.4
- CVE-2021-2032: 2.10.1, 2.09.3, 1.23.4

- CVE-2021-2001: 2.10.1, 2.09.3, 1.23.4
- CVE-2020-28196: 2.10.0, 2.09.3, 1.23.3
- CVE-2020-14867: 1.23.2, 1.22.4
- CVE-2020-14812: 2.09.2, 2.07.4, 1.23.2, 1.22.4
- CVE-2020-14793: 2.09.2, 2.07.4, 1.23.2, 1.22.4
- CVE-2020-14790: 2.10.0, 2.09.2, 2.07.4
- CVE-2020-14776: 2.10.0
- CVE-2020-14775: 2.09.2, 2.07.4
- CVE-2020-14769: 2.09.3, 2.09.2, 2.07.4, 1.23.2, 1.22.4
- CVE-2020-14765: 2.09.2, 2.07.4, 1.23.2, 1.22.4
- CVE-2020-14760: 2.09.2, 2.07.4
- CVE-2020-14672: 2.09.2, 2.07.4, 1.23.2, 1.22.4
- CVE-2020-14567: 2.10.0, 2.09.1, 2.08.3, 2.07.3
- CVE-2020-14559: 2.10.0, 2.09.1, 2.08.3, 2.07.3, 1.23.1, 1.22.3
- CVE-2020-14553: 2.10.0, 2.09.1, 2.08.3, 2.07.3
- CVE-2020-14547: 2.10.0, 2.09.1, 2.08.3, 2.07.3
- CVE-2020-14540: 2.10.0, 2.09.1, 2.08.3, 2.07.3
- CVE-2020-14539: 2.10.0, 1.23.1, 1.22.3
- CVE-2020-11105: 3.07.0, 3.06.0, 3.05.2, 3.04.2, 2.12.1, 2.11.5
- CVE-2020-11104: 3.07.0, 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 릴리스 정보 문서 기록

다음 표에서는 Aurora MySQL 릴리스 정보에 대한 문서 릴리스를 소개합니다.

변경 사항	설명	날짜
Aurora MySQL 버전 3.12.0, MySQL 8.0.44와 호환	Aurora MySQL 버전 3.12.0을 사용할 수 있습니다. 이 버전은 MySQL 8.0.44와 호환됩니다. 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환됩니다 섹션을 참조하세요.	2026년 2월 17일
Aurora MySQL 버전 3.10.3, MySQL 8.0.42와 호환	Aurora MySQL 버전 3.10.3을 사용할 수 있습니다. 이 버전은 MySQL 8.0.42와 호환됩니다. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2026-01-02(버전 3.10.3, MySQL 8.0.42와 호환)을 참조하세요.	2026년 1월 2일
Aurora MySQL 버전 3.04.6, MySQL 8.0.28과 호환	Aurora MySQL 버전 3.04.6을 사용할 수 있습니다. 이 버전은 MySQL 8.0.28과 호환됩니다. 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환됩니다 섹션을 참조하세요.	2026년 1월 2일
Aurora MySQL 버전 3.11.1, MySQL 8.0.43과 호환	Aurora MySQL 버전 3.11.1을 사용할 수 있습니다. 이 버전은 MySQL 8.0.43과 호환됩니다. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2025-12-16(버전 3.11.1, MySQL 8.0.43과 호환)을 참조하세요.	2025년 12월 16일

Aurora MySQL 버전 3.10.2, MySQL 8.0.42와 호환	Aurora MySQL 버전 3.10.2를 사용할 수 있습니다. 이 버전은 MySQL 8.0.42와 호환됩니다. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2025-11-20(버전 3.10.2, MySQL 8.0.42와 호환)을 참조하세요.	2025년 11월 20일
Aurora MySQL 버전 3.11.0, MySQL 8.0.43과 호환	Aurora MySQL 버전 3.11.0을 사용할 수 있습니다. 이 버전은 MySQL 8.0.43과 호환됩니다. 자세한 내용은 Aurora MySQL 데이터베이스 엔진 업데이트 2025-11-13(버전 3.11.0, MySQL 8.0.43과 호환)을 참조하세요.	2025년 11월 13일
Aurora MySQL 버전 3.10.1, MySQL 8.0.42와 호환	Aurora MySQL 버전 3.10.1을 사용할 수 있습니다. 이 버전은 MySQL 8.0.42와 호환됩니다. 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환됩니다 섹션을 참조하세요.	2025년 9월 30일
Aurora MySQL 버전 3.10.0, MySQL 8.0.42와 호환	Aurora MySQL 버전 3.10.0을 사용할 수 있습니다. 이 버전은 MySQL 8.0.42와 호환됩니다. 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환됩니다 섹션을 참조하세요.	2025년 7월 31일

Aurora MySQL 버전 3.09.0, MySQL 8.0.40과 호환	Aurora MySQL 버전 3.09.0을 사용할 수 있습니다. 이 버전은 MySQL 8.0.40과 호환됩니다. 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환됩니다 섹션을 참조하세요.	2025년 5월 14일
Aurora MySQL 버전 3.04.4, MySQL 8.0.28과 호환	Aurora MySQL 버전 3.04.4를 사용할 수 있습니다. 이 버전은 MySQL 8.0.28과 호환됩니다. 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환됩니다 섹션을 참조하세요.	2025년 5월 5일
Aurora MySQL 버전 2.12.5, MySQL 5.7.44과 호환	Aurora MySQL 버전 2.12.5를 사용할 수 있습니다. 이 버전은 MySQL 5.7.44과 호환됩니다. 자세한 내용은 Aurora MySQL 버전 2는 MySQL 5.7과 호환을 참조하세요.	2025년 4월 9일
Aurora MySQL 버전 3.08.2, MySQL 8.0.39와 호환	Aurora MySQL 버전 3.08.2를 사용할 수 있습니다. 이 버전은 MySQL 8.0.39와 호환됩니다. 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환됩니다 섹션을 참조하세요.	2025년 2월 17일
Aurora MySQL 버전 3.08.1, MySQL 8.0.39와 호환	Aurora MySQL 버전 3.08.1을 사용할 수 있습니다. 이 버전은 MySQL 8.0.39와 호환됩니다. 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환됩니다 섹션을 참조하세요.	2025년 2월 17일

Aurora MySQL 버전 3.08.0, MySQL 8.0.39와 호환	Aurora MySQL 버전 3.08.0을 사용할 수 있습니다. 이 버전은 MySQL 8.0.39와 호환됩니다. 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환됩니다 섹션을 참조하세요.	2024년 11월 18일
Aurora MySQL 버전 2.12.4, MySQL 5.7.44과 호환	Aurora MySQL 버전 2.12.4를 사용할 수 있습니다. 이 버전은 MySQL 5.7.44과 호환됩니다. 자세한 내용은 Aurora MySQL 버전 2는 MySQL 5.7과 호환을 참조하세요.	2024년 10월 23일
Aurora MySQL 버전 3.07.1, MySQL 8.0.36과 호환	Aurora MySQL 버전 3.07.1을 사용할 수 있습니다. 이 버전은 MySQL 8.0.36과 호환됩니다. 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환됩니다 섹션을 참조하세요.	2024년 7월 23일
Aurora MySQL 버전 2.11.6, MySQL 5.7.12과 호환	Aurora MySQL 버전 2.11.6을 사용할 수 있습니다. 이 버전은 MySQL 5.7.12와 호환됩니다. 자세한 내용은 Aurora MySQL 버전 2는 MySQL 5.7과 호환을 참조하세요.	2024년 7월 19일
Aurora MySQL 버전 2.12.3, MySQL 5.7.44과 호환	Aurora MySQL 버전 2.12.3을 사용할 수 있습니다. 이 버전은 MySQL 5.7.44과 호환됩니다. 자세한 내용은 Aurora MySQL 버전 2는 MySQL 5.7과 호환을 참조하세요.	2024년 7월 9일

Aurora MySQL 버전 3.06.1, MySQL 8.0.34와 호환	Aurora MySQL 버전 3.06.1을 사용할 수 있습니다. 이 버전은 MySQL 8.0.34와 호환됩니다. 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환됩니다 섹션을 참조하세요.	2024년 6월 26일
Aurora MySQL 버전 3.04.3, MySQL 8.0.28과 호환	Aurora MySQL 버전 3.04.3을 사용할 수 있습니다. 이 버전은 MySQL 8.0.28과 호환됩니다. 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환됩니다 섹션을 참조하세요.	2024년 6월 26일
Aurora MySQL 버전 3.07.0, MySQL 8.0.36과 호환	Aurora MySQL 버전 3.07.0을 사용할 수 있습니다. 이 버전은 MySQL 8.0.36과 호환됩니다. 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환됩니다 섹션을 참조하세요.	2024년 6월 4일
Aurora MySQL 버전 2.11.5, MySQL 5.7.12과 호환	Aurora MySQL 버전 2.11.5를 사용할 수 있습니다. 이 버전은 MySQL 5.7.12와 호환됩니다. 자세한 내용은 Aurora MySQL 버전 2는 MySQL 5.7과 호환을 참조하세요.	2024년 3월 26일
Aurora MySQL 버전 2.12.2, MySQL 5.7.44과 호환	Aurora MySQL 버전 2.12.2를 사용할 수 있습니다. 이 버전은 MySQL 5.7.44과 호환됩니다. 자세한 내용은 Aurora MySQL 버전 2는 MySQL 5.7과 호환을 참조하세요.	2024년 3월 19일

Aurora MySQL 버전 3.04.2, MySQL 8.0.28과 호환	Aurora MySQL 버전 3.04.2를 사용할 수 있습니다. 이 버전은 MySQL 8.0.28과 호환됩니다. 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환됩니다 섹션을 참조하세요.	2024년 3월 15일
Aurora MySQL 버전 3.06.0, MySQL 8.0.34와 호환	Aurora MySQL 버전 3.06.0을 사용할 수 있습니다. 이 버전은 MySQL 8.0.34와 호환됩니다. 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환됩니다 섹션을 참조하세요.	2024년 3월 7일
Aurora MySQL 버전 3.05.2, MySQL 8.0.32와 호환	Aurora MySQL 버전 3.05.2를 사용할 수 있습니다. 이 버전은 MySQL 8.0.32와 호환됩니다. 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환됩니다 섹션을 참조하세요.	2024년 1월 31일
Aurora MySQL 버전 2.12.1, MySQL 5.7.40과 호환	Aurora MySQL 버전 2.12.1을 사용할 수 있습니다. 이 버전은 MySQL 5.7.40과 호환됩니다. 자세한 내용은 Aurora MySQL 버전 2는 MySQL 5.7과 호환을 참조하세요.	2023년 12월 28일
Aurora MySQL 버전 3.03.3, MySQL 8.0.26과 호환	Aurora MySQL 버전 3.03.3을 사용할 수 있습니다. 이 버전은 MySQL 8.0.26과 호환됩니다. 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환됩니다 섹션을 참조하세요.	2023년 12월 8일

Aurora MySQL 버전 3.05.1, MySQL 8.0.32와 호환	Aurora MySQL 버전 3.05.1을 사용할 수 있습니다. 이 버전은 MySQL 8.0.32와 호환됩니다. 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환됩니다 섹션을 참조하세요.	2023년 11월 21일
Aurora MySQL 버전 3.04.1, MySQL 8.0.28과 호환	Aurora MySQL 버전 3.04.1을 사용할 수 있습니다. 이 버전은 MySQL 8.0.28과 호환됩니다. 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환됩니다 섹션을 참조하세요.	2023년 11월 13일
Aurora MySQL 버전 3.05.0.1, MySQL 8.0.32와 호환, 베타	Aurora MySQL 버전 3.05.0.1을 사용할 수 있습니다. 이 버전은 MySQL 8.0.32와 호환됩니다. 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환됩니다 섹션을 참조하세요.	2023년 10월 30일
Aurora MySQL 버전 3.05.0, MySQL 8.0.32와 호환	Aurora MySQL 버전 3.05.0을 사용할 수 있습니다. 이 버전은 MySQL 8.0.32와 호환됩니다. 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환됩니다 섹션을 참조하세요.	2023년 10월 25일
Aurora MySQL 버전 2.12.0.1, MySQL 5.7.40과 호환, 베타	Aurora MySQL 버전 2.12.0.1 베타를 사용할 수 있습니다. 이 버전은 MySQL 5.7.40과 호환됩니다. 자세한 내용은 Aurora MySQL 버전 2는 MySQL 5.7과 호환을 참조하세요.	2023년 10월 25일

Aurora MySQL 버전 2.11.4, MySQL 5.7.12와 호환	Aurora MySQL 버전 2.11.4를 사용할 수 있습니다. 이 버전은 MySQL 5.7.12와 호환됩니다. 자세한 내용은 Aurora MySQL 버전 2는 MySQL 5.7과 호환을 참조하세요.	2023년 10월 17일
Aurora MySQL 버전 3.03.2, MySQL 8.0.26과 호환	Aurora MySQL 버전 3.03.2를 사용할 수 있습니다. 이 버전은 MySQL 8.0.26과 호환됩니다. 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환됩니다 섹션을 참조하세요.	2023년 8월 29일
Aurora MySQL 버전 2.07.10, MySQL 5.7.12와 호환	Aurora MySQL 버전 2.07.10을 사용할 수 있습니다. 이 버전은 MySQL 5.7.12와 호환됩니다. 자세한 내용은 Aurora MySQL 버전 2는 MySQL 5.7과 호환을 참조하세요.	2023년 8월 15일
Aurora MySQL 버전 3.04.0, MySQL 8.0.28과 호환	Aurora MySQL 버전 3.04.0을 사용할 수 있습니다. 이 버전은 MySQL 8.0.28과 호환됩니다. 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환됩니다 섹션을 참조하세요.	2023년 7월 31일
Aurora MySQL 버전 2.12.0, MySQL 5.7.40과 호환	Aurora MySQL 버전 2.12.0을 사용할 수 있습니다. 이 버전은 MySQL 5.7.40과 호환됩니다. 자세한 내용은 Aurora MySQL 버전 2는 MySQL 5.7과 호환을 참조하세요.	2023년 7월 25일

Aurora MySQL 버전 2.11.3, MySQL 5.7.12와 호환	Aurora MySQL 버전 2.11.3을 사용할 수 있습니다. 이 버전은 MySQL 5.7.12와 호환됩니다. 자세한 내용은 Aurora MySQL 버전 2는 MySQL 5.7과 호환을 참조하세요.	2023년 6월 9일
Aurora MySQL 버전 3.03.1, MySQL 8.0.26과 호환	Aurora MySQL 버전 3.03.1을 사용할 수 있습니다. 이 버전은 MySQL 8.0.26과 호환됩니다. 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환됩니다 섹션을 참조하세요.	2023년 5월 11일
Aurora MySQL 버전 2.07.9, MySQL 5.7.12와 호환	Aurora MySQL 버전 2.07.9를 사용할 수 있습니다. 이 버전은 MySQL 5.7.12와 호환됩니다. 자세한 내용은 Aurora MySQL 버전 2는 MySQL 5.7과 호환을 참조하세요.	2023년 5월 4일
Aurora MySQL 버전 3.02.3, MySQL 8.0.23과 호환	Aurora MySQL 버전 3.02.3을 사용할 수 있습니다. 이 버전은 MySQL 8.0.23과 호환됩니다. 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환됩니다 섹션을 참조하세요.	2023년 4월 17일
Aurora MySQL 버전 2.11.2, MySQL 5.7.12와 호환	Aurora MySQL 버전 2.11.2를 사용할 수 있습니다. 이 버전은 MySQL 5.7.12와 호환됩니다. 자세한 내용은 Aurora MySQL 버전 2는 MySQL 5.7과 호환을 참조하세요.	2023년 3월 24일

Aurora MySQL 버전 3.03.0, MySQL 8.0.26과 호환	Aurora MySQL 버전 3.03.0을 사용할 수 있습니다. 이 버전은 MySQL 8.0.26과 호환됩니다. 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환됩니다 섹션을 참조하세요.	2023년 3월 1일
Aurora MySQL 버전 2.11.1, MySQL 5.7.12와 호환	Aurora MySQL 버전 2.11.1을 사용할 수 있습니다. 이 버전은 MySQL 5.7.12와 호환됩니다. 자세한 내용은 Aurora MySQL 버전 2는 MySQL 5.7과 호환을 참조하세요.	2023년 2월 14일
Aurora MySQL 버전 3.02.2, MySQL 8.0.23과 호환	Aurora MySQL 버전 3.02.2를 사용할 수 있습니다. 이 버전은 MySQL 8.0.23과 호환됩니다. 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환됩니다 섹션을 참조하세요.	2022년 11월 18일
Aurora MySQL 버전 2.10.3, MySQL 5.7과 호환	Aurora MySQL 버전 2.10.3을 사용할 수 있습니다. 이 버전은 MySQL 5.7과 호환됩니다. 자세한 내용은 Aurora MySQL 버전 2는 MySQL 5.7과 호환을 참조하세요.	2022년 11월 1일
Aurora MySQL 버전 2.11.0, MySQL 5.7.12와 호환	Aurora MySQL 버전 2.11.0을 사용할 수 있습니다. 이 버전은 MySQL 5.7.12와 호환됩니다. 자세한 내용은 Aurora MySQL 버전 2는 MySQL 5.7과 호환을 참조하세요.	2022년 10월 25일

Aurora MySQL 버전 3.02.1, MySQL 8.0.23과 호환	Aurora MySQL 버전 3.02.1을 사용할 수 있습니다. 이 버전은 MySQL 8.0.23과 호환됩니다. 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환됩니다 섹션을 참조하세요.	2022년 9월 7일
MySQL 5.6 호환 Aurora Serverless v1용 인플레이스 업그레이드	MySQL 5.6 호환 Aurora Serverless v1 클러스터에 대해 인플레이스 업그레이드를 수행하여 기존 클러스터를 MySQL 5.7 호환 Aurora Serverless v1 클러스터로 변경할 수 있습니다. 자세한 내용은 Aurora MySQL Serverless 5.7 엔진 업데이트 2020년 6월 18일(버전 2.07.1)을 참조하세요.	2022년 6월 16일
MySQL 5.6 호환 Aurora Serverless v1용 인플레이스 업그레이드	MySQL 5.6 호환 Aurora Serverless v1 클러스터에 대해 인플레이스 업그레이드를 수행하여 기존 클러스터를 MySQL 5.7 호환 Aurora Serverless v1 클러스터로 변경할 수 있습니다. 자세한 내용은 Aurora MySQL Serverless 5.7 엔진 업데이트 2020년 6월 18일(버전 2.07.1)을 참조하세요.	2022년 6월 16일
Aurora MySQL 데이터베이스 엔진 업데이트 2022년 6월 16일(버전 2.07.8)을 사용할 수 있습니다.	Aurora MySQL 버전 2.07.8을 사용할 수 있습니다.	2022년 6월 16일

Aurora MySQL 버전 3.02.0, MySQL 8.0.23과 호환	Aurora MySQL 버전 3.02.0을 사용할 수 있습니다. 이 버전은 MySQL 8.0.23과 호환됩니다. 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환됩니다 섹션을 참조하세요.	2022년 4월 20일
Aurora MySQL 버전 3.01.1, MySQL 8.0.23과 호환	Aurora MySQL 버전 3.01.1을 사용할 수 있습니다. 이 버전은 MySQL 8.0.23과 호환됩니다. 자세한 내용은 Aurora MySQL 버전 3은 MySQL 8.0과 호환됩니다 섹션을 참조하세요.	2022년 4월 15일
최초 릴리스	Aurora MySQL 릴리스 정보의 초기 릴리스입니다.	2022년 3월 22일
Aurora MySQL 버전 2.10.2	Aurora MySQL 버전 2.10.2를 사용할 수 있습니다.	2022년 1월 26일
Aurora MySQL 버전 2.08.4	Aurora MySQL 버전 2.08.4를 사용할 수 있습니다.	2022년 1월 6일
Aurora MySQL 버전 2.07.7	Aurora MySQL 버전 2.07.7을 사용할 수 있습니다.	2021년 11월 24일
MySQL 8.0.23과 호환되는 Aurora MySQL 버전 3.01.0	Aurora MySQL 버전 3.01.0을 사용할 수 있습니다. 이 버전은 MySQL 8.0.23과 호환됩니다. 자세한 내용은 MySQL 8.0과 호환되는 Aurora MySQL 버전 3을 참조하세요.	2021년 11월 18일
Aurora MySQL 버전 2.09.3	Aurora MySQL 버전 2.09.3을 사용할 수 있습니다.	2021년 11월 12일
Aurora MySQL 버전 2.10.1	Aurora MySQL 버전 2.10.1을 사용할 수 있습니다.	2021년 10월 21일

Aurora MySQL 버전 1.23.4	Aurora MySQL 버전 1.23.4를 사용할 수 있습니다.	2021년 9월 30일
Aurora MySQL 버전 2.07.6	Aurora MySQL 버전 2.07.6을 사용할 수 있습니다.	2021년 9월 2일
Aurora MySQL 버전 2.07.5	Aurora MySQL 버전 2.07.5를 사용할 수 있습니다.	2021년 7월 6일
Aurora MySQL 버전 1.23.3	Aurora MySQL 버전 1.23.3을 사용할 수 있습니다.	2021년 6월 28일
Aurora MySQL 버전 1.22.5	Aurora MySQL 버전 1.22.5를 사용할 수 있습니다.	2021년 6월 3일
Aurora MySQL 버전 2.10.0	Aurora MySQL 버전 2.10.0을 사용할 수 있습니다. 주요 사항으로는 라이터 다시 시작 중 리더 인스턴스의 가용성 개선, 제로 다운타임 패치 적용(ZDP) 개선, 제로 다운타임 다시 시작(ZDR) 개선 및 binlog I/O 캐시 최적화가 있습니다.	2021년 5월 25일
Aurora MySQL 버전 1.23.2	Aurora MySQL 버전 1.23.2를 사용할 수 있습니다.	2021년 3월 18일
Aurora MySQL 버전 2.07.4	Aurora MySQL 버전 2.07.4를 사용할 수 있습니다.	2021년 3월 4일
Aurora MySQL 버전 1.22.4	Aurora MySQL 버전 1.22.4를 사용할 수 있습니다.	2021년 3월 4일
Aurora MySQL 버전 2.09.2	Aurora MySQL 버전 2.09.2를 사용할 수 있습니다.	2021년 2월 26일
Aurora MySQL 버전 2.09.1	Aurora MySQL 버전 2.09.1을 사용할 수 있습니다.	2020년 12월 11일

Aurora MySQL 버전 1.23.1	Aurora MySQL 버전 1.23.1를 사용할 수 있습니다.	2020년 11월 24일
Aurora MySQL 버전 2.08.3	Aurora MySQL 버전 2.08.3을 사용할 수 있습니다.	2020년 11월 12일
Aurora MySQL 버전 2.07.3	Aurora MySQL 버전 2.07.3을 사용할 수 있습니다.	2020년 11월 10일
Aurora MySQL 버전 1.22.3	Aurora MySQL 버전 1.22.3을 사용할 수 있습니다.	2020년 11월 9일
Aurora MySQL 버전 2.09.0	Aurora MySQL 버전 2.09.0을 사용할 수 있습니다.	2020년 9월 17일
Aurora MySQL 버전 1.23.0	Aurora MySQL 버전 1.23.0을 사용할 수 있습니다.	2020년 9월 2일
Aurora MySQL 버전 2.08.2	Aurora MySQL 버전 2.08.2를 사용할 수 있습니다.	2020년 8월 28일
Aurora MySQL 버전 2.04.9	Aurora MySQL 버전 2.04.9를 사용할 수 있습니다.	2020년 8월 14일
Aurora MySQL 버전 2.08.1	Aurora MySQL 버전 2.08.1을 사용할 수 있습니다.	2020년 6월 18일
Aurora MySQL 버전 1.22.2(병렬 쿼리 클러스터용)	Aurora MySQL 버전 1.22.2는 병렬 쿼리 클러스터를 생성 때 사용할 수 있습니다.	2020년 6월 18일
Aurora MySQL 버전 1.20.1(병렬 쿼리 클러스터용)	Aurora MySQL 버전 1.20.1은 병렬 쿼리 클러스터를 생성할 때 사용할 수 있습니다.	2020년 6월 11일
Aurora MySQL 버전 2.08.0	Aurora MySQL 버전 2.08.0을 사용할 수 있습니다.	2020년 6월 2일

Aurora MySQL 버전 1.19.6(병렬 쿼리 클러스터용)	Aurora MySQL 버전 1.19.6은 병렬 쿼리 클러스터를 생성할 때 사용할 수 있습니다.	2020년 6월 2일
Aurora MySQL 버전 2.07.2	Aurora MySQL 버전 2.07.2를 사용할 수 있습니다.	2020년 4월 17일
Aurora MySQL 버전 1.22.2	Aurora MySQL 버전 1.22.2를 사용할 수 있습니다.	2020년 3월 5일
Aurora MySQL 버전 1.20.1	Aurora MySQL 버전 1.20.1을 사용할 수 있습니다.	2020년 3월 5일
Aurora MySQL 버전 1.19.6	Aurora MySQL 버전 1.19.6을 사용할 수 있습니다.	2020년 3월 5일
Aurora MySQL 버전 1.17.9	Aurora MySQL 버전 1.17.9를 사용할 수 있습니다.	2020년 3월 5일
Aurora MySQL 버전 2.07.1	Aurora MySQL 버전 2.07.1을 사용할 수 있습니다.	2019년 12월 23일
Aurora MySQL 버전 1.22.1	Aurora MySQL 버전 1.22.1을 사용할 수 있습니다.	2019년 12월 23일
Aurora MySQL 버전 2.07.0	Aurora MySQL 버전 2.07.0을 사용할 수 있습니다.	2019년 11월 25일
Aurora MySQL 버전 1.22.0	Aurora MySQL 버전 1.22.0을 사용할 수 있습니다.	2019년 11월 25일
Aurora MySQL 버전 1.21.0	Aurora MySQL 버전 1.21.0을 사용할 수 있습니다.	2019년 11월 25일
Aurora MySQL 버전 2.06.0	Aurora MySQL 버전 2.06.0을 사용할 수 있습니다.	2019년 11월 22일
Aurora MySQL 버전 2.04.8	Aurora MySQL 버전 2.04.8을 사용할 수 있습니다.	2019년 11월 20일

Aurora MySQL 버전 2.04.7	Aurora MySQL 버전 2.04.7을 사용할 수 있습니다.	2019년 11월 14일
Aurora MySQL 버전 2.05.0	Aurora MySQL 버전 2.05.0을 사용할 수 있습니다.	2019년 11월 11일
Aurora MySQL 버전 1.20.0	Aurora MySQL 버전 1.20.0을 사용할 수 있습니다.	2019년 11월 11일
Aurora MySQL 버전 2.04.6	Aurora MySQL 버전 2.04.6을 사용할 수 있습니다.	2019년 9월 19일
Aurora MySQL 버전 1.19.5	Aurora MySQL 버전 1.19.5를 사용할 수 있습니다.	2019년 9월 19일
Aurora MySQL 버전 2.04.5	Aurora MySQL 버전 2.04.5를 사용할 수 있습니다.	2019년 7월 8일
Aurora MySQL 버전 1.19.2	Aurora MySQL 버전 1.19.2를 사용할 수 있습니다.	2019년 6월 5일
Aurora MySQL 버전 2.04.4	Aurora MySQL 버전 2.04.4를 사용할 수 있습니다.	2019년 5월 29일
Aurora MySQL 버전 2.04.3	Aurora MySQL 버전 2.04.3을 사용할 수 있습니다.	2019년 5월 9일
Aurora MySQL 버전 1.19.1	Aurora MySQL 버전 1.19.1을 사용할 수 있습니다.	2019년 5월 9일
Aurora MySQL 버전 2.04.2	Aurora MySQL 버전 2.04.2를 사용할 수 있습니다.	2019년 5월 2일
Aurora MySQL 버전 2.04.1	Aurora MySQL 버전 2.04.1을 사용할 수 있습니다.	2019년 3월 25일
Aurora MySQL 버전 2.04	Aurora MySQL 버전 2.04를 사용할 수 있습니다.	2019년 3월 25일

Aurora MySQL 버전 2.03.4	Aurora MySQL 버전 2.03.4를 사용할 수 있습니다.	2019년 2월 7일
Aurora MySQL 버전 1.19.0	Aurora MySQL 버전 1.19.0을 사용할 수 있습니다.	2019년 2월 7일
Aurora MySQL 버전 2.03.3	Aurora MySQL 버전 2.03.3을 사용할 수 있습니다.	2019년 1월 18일
Aurora MySQL 버전 1.17.8	Aurora MySQL 버전 1.17.8을 사용할 수 있습니다.	2019년 1월 17일
Aurora MySQL 버전 2.03.2	Aurora MySQL 버전 2.03.2를 사용할 수 있습니다.	2019년 1월 9일
Aurora MySQL 버전 2.03.1	Aurora MySQL 버전 2.03.1을 사용할 수 있습니다.	2018년 10월 24일
Aurora MySQL 버전 2.03	Aurora MySQL 버전 2.03을 사용할 수 있습니다.	2018년 10월 11일
Aurora MySQL 버전 2.02.5	Aurora MySQL 버전 2.02.5를 사용할 수 있습니다.	2012년 10월 8일
Aurora MySQL 버전 1.17.7	Aurora MySQL 버전 1.17.7을 사용할 수 있습니다.	2012년 10월 8일
Aurora MySQL 버전 2.02.4	Aurora MySQL 버전 2.02.4를 사용할 수 있습니다.	2018년 9월 21일
Aurora MySQL 버전 1.18.0	Aurora MySQL 버전 1.18.0을 사용할 수 있습니다.	2018년 9월 20일
Aurora MySQL 버전 1.17.6	Aurora MySQL 버전 1.17.6을 사용할 수 있습니다.	2018년 9월 6일

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.