Note di rilascio di Aurora MySQL

Amazon Aurora

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

Amazon Aurora: Note di rilascio di Aurora MySQL

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

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà dei rispettivi proprietari, che possono o meno essere affiliati, collegati o sponsorizzati da Amazon.

Table of Contents

Note di rilascio di Aurora MySQL 1
Calendari di rilascio di Aurora MySQL 2
Calendario dei rilasci per le versioni principali di Aurora MySQL 2
Calendario dei rilasci per le versioni secondarie di Aurora MySQL 4
Aurora MySQL versione 3 6
Aggiornamenti Aurora MySQL: 2025-11-13 (versione 3.11.0, compatibile con MySQL 8.0.43) 7
Miglioramenti 8
Integrazione delle correzioni di bug di MySQL Community Edition 12
Aggiornamenti Aurora MySQL: 2025-11-20 (versione 3.10.2, compatibile con MySQL 8.0.42) 13
Miglioramenti 14
Aggiornamenti Aurora MySQL: 2025-09-30 (versione 3.10.1, compatibile con MySQL 8.0.42) 14
Miglioramenti 15
Integrazione delle correzioni di bug di MySQL Community Edition 17
Aggiornamenti Aurora MySQL: 2025-07-31 (versione 3.10.0, compatibile con MySQL 8.0.42) 17
Nuove funzionalità 19
Miglioramenti 19
Integrazione delle correzioni di bug di MySQL Community Edition 22
Aggiornamenti Aurora MySQL: 2025-05-14 (versione 3.09.0, compatibile con MySQL 8.0.40) 22
Aspetti salienti 23
Miglioramenti 23
Integrazione delle correzioni di bug di MySQL Community Edition 27
Aggiornamenti Aurora MySQL: 2025-04-07 (versione 3.08.2, compatibile con MySQL 8.0.39) 27
Miglioramenti 28
Integrazione delle correzioni di bug di MySQL Community Edition 29
Aggiornamenti Aurora MySQL: 2025-02-17 (versione 3.08.1, compatibile con MySQL 8.0.39) 29
Miglioramenti 28
Integrazione delle correzioni di bug di MySQL Community Edition 29
Aggiornamenti Aurora MySQL: 2024-11-18 (versione 3.08.0, compatibile con MySQL 8.0.39) 30
Nuove funzionalità 31
Miglioramenti 33
Integrazione delle correzioni di bug di MySQL Community Edition 38
Aggiornamenti Aurora MySQL: 2024-07-23 (versione 3.07.1, compatibile con MySQL 8.0.36) 38
Miglioramenti 39
Integrazione delle correzioni di bug di MySQL Community Edition 40

Aggiornamenti Aurora MySQL: 2024-06-04 (versione 3.07.0, compatibile con MySQL 8.0.36) 40
Miglioramenti 40
Integrazione delle correzioni di bug di MySQL Community Edition 44
Aggiornamenti Aurora MySQL: 2024-06-26 (versione 3.06.1, compatibile con MySQL 8.0.34) 45
Miglioramenti 46
Integrazione delle correzioni di bug di MySQL Community Edition 48
Aggiornamenti Aurora MySQL: 2024-03-07 (versione 3.06.0, compatibile con MySQL 8.0.34) 48
Nuove funzionalità 49
Miglioramenti 46
Integrazione delle correzioni di bug di MySQL Community Edition 48
Aggiornamenti Aurora MySQL: 2024-01-31 (versione 3.05.2, compatibile con MySQL 8.0.32) 54
Miglioramenti 55
Integrazione delle correzioni di bug di MySQL Community Edition 56
Aggiornamenti Aurora MySQL: 2023-11-21 (versione 3.05.1, compatibile con MySQL 8.0.32) 57
Miglioramenti 57
Integrazione delle correzioni di bug di MySQL Community Edition 57
Aggiornamenti di Aurora MySQL: 30/10/2023 (versione 3.05.0.1, compatibile con MySQL 8.0.32) beta 58
Miglioramenti 58
Aggiornamenti di Aurora MySQL: 25/10/2023 (versione 3.05.0, compatibile con MySQL 8.0.32) 59
Miglioramenti 59
Integrazione delle correzioni di bug di MySQL Community Edition 64
Aggiornamenti Aurora MySQL: 2025-05-05 (versione 3.04.4, compatibile con MySQL 8.0.28) 64
Miglioramenti 65
Integrazione delle correzioni di bug di MySQL Community Edition 67
Aggiornamenti Aurora MySQL: 2024-06-26 (versione 3.04.3, compatibile con MySQL 8.0.28) 67
Miglioramenti 69
Integrazione delle correzioni di bug di MySQL Community Edition 70
Aggiornamenti Aurora MySQL: 2024-03-15 (versione 3.04.2, compatibile con MySQL 8.0.28) 71
Miglioramenti 72
Integrazione delle correzioni di bug di MySQL Community Edition 74
Aggiornamenti Aurora MySQL: 2023-11-13 (versione 3.04.1, compatibile con MySQL 8.0.28) 74
Miglioramenti 75
Integrazione delle correzioni di bug di MySQL Community Edition 77
Aggiornamenti Aurora MySQL: 2023-07-31 (versione 3.04.0, compatibile con MySQL 8.0.28) 77

Miglioramenti 79
Integrazione delle correzioni di bug di MySQL Community Edition 85
Aggiornamenti Aurora MySQL: 2023-12-08 (versione 3.03.3,) (obsoleta) 85
Miglioramenti 86
Integrazione delle correzioni di bug di MySQL Community Edition 88
Aggiornamenti Aurora MySQL: 2023-08-29 (versione 3.03.2) (obsoleta) 88
Miglioramenti 89
Integrazione delle correzioni di bug di MySQL Community Edition 90
Aggiornamenti Aurora MySQL: 2023-05-11 (versione 3.03.1) (obsoleta) 91
Miglioramenti 92
Integrazione delle correzioni di bug di MySQL Community Edition 94
Aggiornamenti Aurora MySQL: 3-01 (versione 3.03.0) (obsoleta) 94
Miglioramenti 95
Integrazione delle correzioni di bug di MySQL Community Edition 98
Aggiornamenti Aurora MySQL: 2023-04-17 (versione 3.02.3) (obsoleta) 99
Miglioramenti 99
Aggiornamenti Aurora MySQL: 2022-11-18 (versione 3.02.2) (obsoleto) 100
Miglioramenti 101
Integrazione delle correzioni di bug di MySQL Community Edition 98
Aggiornamenti Aurora MySQL: 2022-09-07 (versione 3.02.1) (obsoleta) 104
Miglioramenti 105
Aggiornamenti Aurora MySQL: 2022-04-20 (versione 3.02.0) (obsoleta) 106
Miglioramenti 107
Integrazione delle correzioni di bug della community di MySQL 109
Aggiornamenti Aurora MySQL: 15/04/2022 (versione 3.01.1) (obsoleta) 109
Miglioramenti 110
Integrazione delle correzioni di bug della community di MySQL 112
Aggiornamenti Aurora MySQL: 2021-11-18 (versione 3.01.0) (obsoleta) 112
Miglioramenti 113
Aurora MySQL versione 2 114
Aggiornamenti Aurora MySQL: 2025-04-09 (versione 2.12.5, compatibile con MySQL 5.7.44) - Versione RDS Extended Support 116
Miglioramenti 117
Aggiornamenti Aurora MySQL: 2024-10-23 (versione 2.12.4, compatibile con MySQL 5.7.44) - Versione RDS Extended Support 118
Miglioramenti 117

Integrazione delle correzioni di bug di MySQL Community Edition 119
Funzionalità non supportate in Aurora MySQL versione 2 119
Compatibilità MySQL 5.7 119
Aggiornamenti Aurora MySQL: 2024-07-09 (versione 2.12.3, compatibile con MySQL 5.7.44) - Versione RDS Extended Support 120
Miglioramenti 121
Integrazione delle correzioni di bug di MySQL Community Edition 123
Funzionalità non supportate in Aurora MySQL versione 2 124
Compatibilità MySQL 5.7 124
Aggiornamenti Aurora MySQL: 2024-03-19 (versione 2.12.2, compatibile con MySQL 5.7.44) - Versione RDS Extended Support 125
Miglioramenti 125
Integrazione delle correzioni di bug di MySQL Community Edition 126
Funzionalità non supportate in Aurora MySQL versione 2 127
Compatibilità MySQL 5.7 127
Aggiornamenti Aurora MySQL: 2023-12-28 (versione 2.12.1, compatibile con MySQL 5.7.40) - Versione RDS Extended Support 127
Miglioramenti 128
Integrazione delle correzioni di bug di MySQL Community Edition 131
Funzionalità non supportate in Aurora MySQL versione 2 131
Compatibilità MySQL 5.7 131
Aggiornamenti Aurora MySQL: 2023-10-25 (versione 2.12.0.1, compatibile con MySQL 5.7.40) - Versione RDS Extended Support (Beta) 132
Miglioramenti 133
Aggiornamenti Aurora MySQL: 2023-07-25 (versione 2.12.0, compatibile con MySQL 5.7.40) - Versione RDS Extended Support 133
Miglioramenti 134
Integrazione delle correzioni di bug di MySQL Community Edition 137
Funzionalità non supportate in Aurora MySQL versione 2 137
Compatibilità MySQL 5.7 137
Aggiornamenti Aurora MySQL: 2024-07-19 (versione 2.11.6, compatibile con MySQL 5.7.12) - Versione RDS Extended Support 138
Miglioramenti 139
Integrazione delle correzioni di bug di MySQL Community Edition 141
Funzionalità non supportate in Aurora MySQL versione 2 141
Compatibilità MySQL 5.7 141

Aggiornamenti Aurora MySQL: 2024-03-26 (versione 2.11.5, compatibile con MySQL 5.7.12) - Versione RDS Extended Support (predefinita) 142
Miglioramenti 143
Integrazione delle correzioni di bug di MySQL Community Edition 145
Funzionalità non supportate in Aurora MySQL versione 2 145
Compatibilità MySQL 5.7 145
Aggiornamenti Aurora MySQL: 2023-10-17 (versione 2.11.4, compatibile con MySQL 5.7.12) - Versione RDS Extended Support 146
Miglioramenti 147
Integrazione delle correzioni di bug di MySQL Community Edition 149
Funzionalità non supportate in Aurora MySQL versione 2 149
Compatibilità MySQL 5.7 149
Aggiornamenti Aurora MySQL: 2023-06-09 (versione 2.11.3, compatibile con MySQL 5.7.12) - Versione RDS Extended Support 150
Miglioramenti 151
Funzionalità non supportate in Aurora MySQL versione 2 152
Compatibilità MySQL 5.7 153
Aggiornamenti Aurora MySQL: 2023-03-24 (versione 2.11.2, compatibile con MySQL 5.7.12) - Versione RDS Extended Support 153
Miglioramenti 154
Funzionalità non supportate in Aurora MySQL versione 2 155
Compatibilità MySQL 5.7 155
Aggiornamenti Aurora MySQL: 2023-02-14 (versione 2.11.1, compatibile con MySQL 5.7.12) - Versione RDS Extended Support 156
Miglioramenti 157
Confronto con Aurora MySQL Versione 1 158
Compatibilità MySQL 5.7 159
Aggiornamenti Aurora MySQL: 2022-10-25 (versione 2.11.0, compatibile con MySQL 5.7.12) - Versione RDS Extended Support 159
Miglioramenti 160
Integrazione delle correzioni di bug di MySQL Community Edition 164
Confronto con Aurora MySQL Versione 1 166
Compatibilità MySQL 5.7 166
Aggiornamenti di Aurora MySQL 01/11/2022 (versione 2.10.3) (obsoleta) 167
Miglioramenti 168
Integrazione delle correzioni di bug di MySQL Community Edition 169

Confronto con Aurora MySQL Versione 1 170
Compatibilità MySQL 5.7 170
Aggiornamenti di Aurora MySQL 26/01/2022 (versione 2.10.2) (obsoleta) 171
Miglioramenti 171
Integrazione delle correzioni di bug di MySQL Community Edition 98
Confronto con Aurora MySQL Versione 1 175
Compatibilità MySQL 5.7 176
Aggiornamenti di Aurora MySQL 21/10/2021 (versione 2.10.1) (obsoleta) 176
Miglioramenti 177
Integrazione delle correzioni di bug della community di MySQL 178
Confronto con Aurora MySQL Versione 1 178
Compatibilità MySQL 5.7 179
Aggiornamenti di Aurora MySQL 25/05/2021 (versione 2.10.0) (obsoleta) 179
Miglioramenti 180
Integrazione delle correzioni di bug della community di MySQL 183
Confronto con Aurora MySQL Versione 1 187
Compatibilità MySQL 5.7 187
Aggiornamenti di Aurora MySQL 12/11/2021 (versione 2.09.3) (obsoleta) 188
Miglioramenti 189
Integrazione delle correzioni di bug della community di MySQL 191
Confronto con Aurora MySQL Versione 1 192
Compatibilità MySQL 5.7 192
Aggiornamenti di Aurora MySQL 26/02/2021 (versione 2.09.2) (obsoleta) 193
Miglioramenti 193
Confronto con Aurora MySQL Versione 1 195
Compatibilità MySQL 5.7 195
Aggiornamenti di Aurora MySQL 11/12/2020 (versione 2.09.1) (obsoleta) 196
Miglioramenti 197
Integrazione delle correzioni di bug della community di MySQL 198
Confronto con Aurora MySQL Versione 1 198
Compatibilità MySQL 5.7 199
Aggiornamenti di Aurora MySQL 17/09/2020 (versione 2.09.0) (obsoleta) 200
Miglioramenti 201
Integrazione delle correzioni di bug della community di MySQL 205
Confronto con Aurora MySQL Versione 1 207
Compatibilità MySQL 5.7 207

Aggiornamenti di Aurora MySQL 06/01/2022 (versione 2.08.4) (obsoleta) 208
Miglioramenti 208
Confronto con Aurora MySQL Versione 1 209
Compatibilità MySQL 5.7 210
Aggiornamenti di Aurora MySQL 12/11/2020 (versione 2.08.3) (obsoleta) 210
Miglioramenti 211
Integrazione delle correzioni di bug della community di MySQL 212
Confronto con Aurora MySQL Versione 1 212
Compatibilità MySQL 5.7 213
Aggiornamenti di Aurora MySQL 28/08/2020 (versione 2.08.2) (obsoleta) 213
Miglioramenti 214
Confronto con Aurora MySQL Versione 1 214
Compatibilità MySQL 5.7 215
Aggiornamenti di Aurora MySQL 02/06/2020 (versione 2.08.1) (obsoleta) 216
Miglioramenti 216
Confronto con Aurora MySQL Versione 1 217
Compatibilità MySQL 5.7 217
Aggiornamenti di Aurora MySQL 02/06/2020 (versione 2.08.0) (obsoleta) 218
Miglioramenti 219
Integrazione delle correzioni di bug della community di MySQL 221
Confronto con Aurora MySQL Versione 1 222
Compatibilità MySQL 5.7 223
Aggiornamenti Aurora MySQL: 2023-08-15 (versione 2.07.10) (obsoleta) 223
Miglioramenti 224
Funzionalità non supportate in Aurora MySQL versione 2 225
Compatibilità MySQL 5.7 225
Aggiornamenti Aurora MySQL: 2023-05-04 (versione 2.07.9) (obsoleta) 226
Miglioramenti 227
Funzionalità non supportate in Aurora MySQL versione 2 228
Compatibilità MySQL 5.7 228
Aggiornamenti di Aurora MySQL 16/06/2022 (versione 2.07.8) (obsoleta) 228
Miglioramenti 229
Integrazione delle correzioni di bug della community di MySQL 230
Confronto con Aurora MySQL Versione 1 230
Compatibilità MySQL 5.7 231
Aggiornamenti di Aurora MySQL 24/11/2021 (versione 2.07.7) (obsoleta) 231

Miglioramenti 232
Confronto con Aurora MySQL Versione 1 233
Compatibilità MySQL 5.7 234
Aggiornamenti di Aurora MySQL 02/09/2021 (versione 2.07.6) (obsoleta) 234
Integrazione delle correzioni di bug della community di MySQL 235
Confronto con Aurora MySQL Versione 1 235
Compatibilità MySQL 5.7 235
Aggiornamenti di Aurora MySQL 06/07/2021 (versione 2.07.5) (obsoleta) 236
Miglioramenti 237
Confronto con Aurora MySQL Versione 1 237
Compatibilità MySQL 5.7 238
Aggiornamenti di Aurora MySQL 03/04/2021 (versione 2.07.4) (obsoleta) 238
Miglioramenti 239
Integrazione delle correzioni di bug della community di MySQL 240
Confronto con Aurora MySQL Versione 1 240
Compatibilità MySQL 5.7 241
Aggiornamenti di Aurora MySQL 10/11/2020 (versione 2.07.3) (obsoleta) 241
Miglioramenti 242
Integrazione delle correzioni di bug della community di MySQL 243
Confronto con Aurora MySQL Versione 1 245
Compatibilità MySQL 5.7 245
Aggiornamenti di Aurora MySQL 17/04/2020 (versione 2.07.2) (obsoleta) 246
Miglioramenti 247
Integrazione delle correzioni di bug della community di MySQL 248
Confronto con Aurora MySQL Versione 1 248
Compatibilità MySQL 5.7 249
Aggiornamenti di Aurora MySQL 23/12/2019 (versione 2.07.1) (obsoleta) 249
Miglioramenti 250
Confronto con Aurora MySQL versione 1 251
Compatibilità MySQL 5.7 251
Aggiornamenti di Aurora MySQL 25/11/2019 (versione 2.07.0) (obsoleta) 252
Miglioramenti 253
Integrazione delle correzioni di bug della community di MySQL 254
Confronto con Aurora MySQL Versione 1 254
Compatibilità MySQL 5.7 255
Aggiornamenti di Aurora MySQL 22/11/2019 (versione 2.06.0) (obsoleta) 255

Miglioramenti 256
Confronto con Aurora MySQL Versione 1 259
Compatibilità MySQL 5.7 259
Aggiornamenti di Aurora MySQL 11/11/2019 (versione 2.05.0) (obsoleta) 260
Miglioramenti 261
Integrazione delle correzioni di bug di MySQL. 262
Confronto con Aurora MySQL Versione 1 262
Compatibilità MySQL 5.7 262
Aggiornamenti di Aurora MySQL 14/08/2020 (versione 2.04.9) (obsoleta) 263
Miglioramenti 264
Integrazione delle correzioni di bug di MySQL. 266
Confronto con Aurora MySQL Versione 1 267
Compatibilità MySQL 5.7 268
Aggiornamenti di Aurora MySQL 20/11/2019 (versione 2.04.8) (obsoleta) 268
Miglioramenti 269
Confronto con Aurora MySQL Versione 1 270
Compatibilità MySQL 5.7 271
Aggiornamenti di Aurora MySQL 14/11/2019 (versione 2.04.7) (obsoleta) 271
Miglioramenti 272
Confronto con Aurora MySQL Versione 1 273
Compatibilità MySQL 5.7 273
Aggiornamenti di Aurora MySQL 19/09/2019 (versione 2.04.6) (obsoleta) 274
Miglioramenti 275
Integrazione delle correzioni di bug di MySQL. 275
Confronto con Aurora MySQL Versione 1 276
Compatibilità MySQL 5.7 276
Aggiornamenti di Aurora MySQL 08/07/2019 (versione 2.04.5) (obsoleta) 277
Miglioramenti 277
Confronto con Aurora MySQL Versione 1 278
Compatibilità MySQL 5.7 279
Aggiornamenti di Aurora MySQL 29/05/2019 (versione 2.04.4) (obsoleta) 280
Miglioramenti 281
Confronto con Aurora MySQL Versione 1 281
Compatibilità MySQL 5.7 282
Aggiornamenti di Aurora MySQL 09/05/2019 (versione 2.04.3) (obsoleta) 282
Miglioramenti 283

Confronto con Aurora MySQL Versione 1 283
Compatibilità MySQL 5.7 284
Aggiornamenti di Aurora MySQL 02/05/2019 (versione 2.04.2) (obsoleta) 285
Miglioramenti 286
Integrazione delle correzioni di bug di MySQL. 286
Confronto con Aurora MySQL Versione 1 286
Compatibilità MySQL 5.7 287
Aggiornamenti di Aurora MySQL 25/03/2019 (versione 2.04.1) (obsoleta) 287
Miglioramenti 288
Confronto con Aurora MySQL Versione 1 288
Compatibilità MySQL 5.7 289
Aggiornamenti di Aurora MySQL 25/03/2019 (versione 2.04.0) (obsoleta) 290
Miglioramenti 291
Integrazione delle correzioni di bug di MySQL. 291
Confronto con Aurora MySQL Versione 1 291
Compatibilità MySQL 5.7 292
Aggiornamenti di Aurora MySQL 07/02/2019 (versione 2.03.4) (obsoleta) 292
Miglioramenti 293
Confronto con Aurora MySQL Versione 1 293
Compatibilità MySQL 5.7 294
Aggiornamenti di Aurora MySQL 18/01/2019 (versione 2.03.3) (obsoleta) 294
Miglioramenti 295
Integrazione delle correzioni di bug di MySQL. 296
Confronto con Aurora MySQL Versione 1 297
Compatibilità MySQL 5.7 297
Aggiornamenti di Aurora MySQL 09/01/2019 (versione 2.03.2) (obsoleta) 298
Miglioramenti 299
Confronto con Aurora MySQL Versione 1 299
Compatibilità MySQL 5.7 300
Aggiornamenti di Aurora MySQL 24/10/2018 (versione 2.03.1) (obsoleta) 300
Miglioramenti 301
Confronto con Aurora MySQL Versione 1 301
Compatibilità MySQL 5.7 302
Aggiornamenti di Aurora MySQL 11/10/2018 (versione 2.03) (obsoleta) 303
Miglioramenti 303
Integrazione delle correzioni di bug della community di MySQL 304

Confronto con Aurora MySQL Versione 1 304
Compatibilità MySQL 5.7 305
Aggiornamenti di Aurora MySQL 08/10/2018 (versione 2.02.5) (obsoleta) 305
Miglioramenti 306
Confronto con Aurora MySQL Versione 1 306
Compatibilità MySQL 5.7 307
Aggiornamenti di Aurora MySQL 21/09/2018 (versione 2.02.54) (obsoleta) 308
Miglioramenti 308
Integrazione delle correzioni di bug della community di MySQL 308
Confronto con Aurora MySQL Versione 1 309
Compatibilità MySQL 5.7 310
Aggiornamenti di Aurora MySQL 23/08/2018 (versione 2.02.3) (obsoleta) 311
Confronto con Aurora MySQL Versione 1 311
Compatibilità MySQL 5.7 312
Differenze di CLI tra Aurora MySQL 2.x e Aurora MySQL 1.x 313
Miglioramenti 313
Aggiornamenti di Aurora MySQL 04/06/2018 (versione 2.02.2) (obsoleta) 314
Miglioramenti 314
Confronto con Aurora MySQL 5.6 315
Compatibilità MySQL 5.7 315
Differenze di CLI tra Aurora MySQL 2.x e Aurora MySQL 1.x 313
Miglioramenti 314
Aggiornamenti di Aurora MySQL 03/05/2018 (versione 2.02) (obsoleta) 317
Confronto con Aurora MySQL 5.6 317
Compatibilità MySQL 5.7 318
Differenze di CLI tra Aurora MySQL 2.x e Aurora MySQL 1.x 313
Miglioramenti 319
Integrazione delle correzioni di bug di MySQL. 320
Aggiornamenti di Aurora MySQL 13/03/2018 (versione 2.01.1) (obsoleta) 320
Confronto con Aurora MySQL 5.6 320
Compatibilità MySQL 5.7 321
Differenze di CLI tra Aurora MySQL 2.x e Aurora MySQL 1.x 322
Miglioramenti 322
Aggiornamenti di Aurora MySQL 06/02/2018 (versione 2.01) (obsoleta) 322
Confronto con Aurora MySQL 5.6 323
Compatibilità MySQL 5.7 323

Differenze di CLI tra Aurora MySQL 2.x e Aurora MySQL 1.x 313
Aurora MySQL versione 1 (obsoleta) 325
Aggiornamenti di Aurora MySQL 30/09/2021 (versione 1.23.4) (obsoleta) 327
Miglioramenti 327
Aggiornamenti di Aurora MySQL 28/06/2021 (versione 1.23.3) (obsoleta) 328
Miglioramenti 328
Aggiornamenti di Aurora MySQL 18/03/2021 (versione 1.23.2) (obsoleta) 328
Miglioramenti 329
Integrazione delle correzioni di bug della community di MySQL 331
Aggiornamenti di Aurora MySQL 24/11/2020 (versione 1.23.1) (obsoleta) 331
Miglioramenti 331
Aggiornamenti di Aurora MySQL 02/09/2020 (versione 1.23.0) (obsoleta) 332
Miglioramenti 333
Integrazione delle correzioni di bug della community di MySQL 336
Aggiornamenti di Aurora MySQL 03/06/2021 (versione 1.22.5) (obsoleta) 338
Miglioramenti 338
Aggiornamenti di Aurora MySQL 03/04/2021 (versione 1.22.4) (obsoleta) 339
Miglioramenti 340
Aggiornamenti di Aurora MySQL 11/09/2020 (versione 1.22.3) (obsoleta) 340
Miglioramenti 341
Integrazione delle correzioni di bug della community di MySQL 342
Aggiornamenti di Aurora MySQL 05/03/2021 (versione 1.22.2) (obsoleta) 343
Miglioramenti 344
Aggiornamenti di Aurora MySQL 23/12/2019 (versione 1.22.1) (obsoleta) 344
Miglioramenti 345
Aggiornamenti di Aurora MySQL 25/11/2019 (versione 1.22.0) (obsoleta) 345
Miglioramenti 346
Integrazione delle correzioni di bug della community di MySQL 350
Aggiornamenti di Aurora MySQL 25/11/2019 (versione 1.21.0) (obsoleta) 350
Miglioramenti 351
Integrazione delle correzioni di bug della community di MySQL 352
Aggiornamenti di Aurora MySQL 05/03/2020 (versione 1.20.1) (obsoleta) 353
Miglioramenti 353
Aggiornamenti di Aurora MySQL 11/11/2019 (versione 1.20.0) (obsoleta) 354
Miglioramenti 354
Integrazione delle correzioni di bug della community di MySQL 356

Aggiornamenti di Aurora MySQL 05/03/2020 (versione 1.19.6) (obsoleta) 356
Miglioramenti 357
Aggiornamenti di Aurora MySQL 19/09/2019 (versione 1.19.5) (obsoleta) 357
Miglioramenti 358
Integrazione delle correzioni di bug della community di MySQL 358
Aggiornamenti di Aurora MySQL 05/06/2019 (versione 1.19.2) (obsoleta) 359
Miglioramenti 360
Aggiornamenti di Aurora MySQL 09/05/2019 (versione 1.19.1) (obsoleta) 360
Miglioramenti 361
Aggiornamenti di Aurora MySQL 07/02/2019 (versione 1.19.0) (obsoleta) 361
Funzionalità 362
Miglioramenti 362
Integrazione delle correzioni di bug della community di MySQL 363
Aggiornamenti di Aurora MySQL 02/09/2018 (versione 1.18.0) (obsoleta) 364
Funzionalità 364
Aggiornamenti di Aurora MySQL 05/03/2020 (versione 1.17.9) (obsoleta) 366
Miglioramenti 366
Aggiornamenti di Aurora MySQL 17/01/2019 (versione 1.17.8) (obsoleta) 366
Miglioramenti 367
Integrazione delle correzioni di bug della community di MySQL 367
Aggiornamenti di Aurora MySQL 08/10/2018 (versione 1.17.7) (obsoleta) 367
Miglioramenti 368
Integrazione delle correzioni di bug della community di MySQL 369
Aggiornamenti di Aurora MySQL 06/09/2018 (versione 1.17.6) (obsoleta) 369
Miglioramenti 370
Integrazione delle correzioni di bug della community di MySQL 370
Aggiornamenti di Aurora MySQL 14/08/2018 (versione 1.17.5) (obsoleta) 370
Miglioramenti 371
Aggiornamenti di Aurora MySQL 07/08/2018 (versione 1.17.4) (obsoleta) 371
Miglioramenti 372
Aggiornamenti di Aurora MySQL 05/06/2018 (versione 1.17.3) (obsoleta) 372
Miglioramenti 373
Aggiornamenti di Aurora MySQL 27/04/2018 (versione 1.17.2) (obsoleta) 374
Miglioramenti 374
Aggiornamenti di Aurora MySQL 23/03/2018 (versione 1.17.1) (obsoleta) 374
Miglioramenti 375

Aggiornamenti di Aurora MySQL 13/03/2018 (versione 1.17) (obsoleta) 375
Applicazione di patch senza tempi di inattività 376
Nuove funzionalità 376
Miglioramenti 376
Integrazione delle correzioni di bug di MySQL. 377
Aggiornamenti di Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta) 377
Applicazione di patch senza tempi di inattività 378
Nuove funzionalità 378
Miglioramenti 378
Integrazione delle correzioni di bug di MySQL. 379
Aggiornamenti di Aurora MySQL 20/11/2017 (versione 1.15.1) (obsoleta) 379
Applicazione di patch senza tempi di inattività 379
Miglioramenti 380
Integrazione delle correzioni di bug di MySQL. 380
Aggiornamenti di Aurora MySQL 24/10/2017 (versione 1.15) (obsoleta) 380
Applicazione di patch senza tempi di inattività 381
Nuove funzionalità 381
Miglioramenti 382
Integrazione delle correzioni di bug di MySQL. 380
Aggiornamenti di Aurora MySQL 13/03/2018 (versione 1.14.4) (obsoleta) 383
Applicazione di patch senza tempi di inattività 384
Nuove funzionalità 384
Miglioramenti 384
Integrazione delle correzioni di bug di MySQL. 384
Aggiornamenti di Aurora MySQL 22/09/2017 (versione 1.14.1) (obsoleta) 385
Miglioramenti 385
Aggiornamenti di Aurora MySQL 07/08/2017 (versione 1.14) (obsoleta) 385
Applicazione di patch senza tempi di inattività 386
Miglioramenti 386
Integrazione delle correzioni di bug di MySQL. 387
Aggiornamenti di Aurora MySQL 15/05/2017 (versione 1.13) (obsoleta) 388
Applicazione di patch senza tempi di inattività 388
Nuove caratteristiche 388
Miglioramenti 389
Integrazione delle correzioni di bug di MySQL. 390
Aggiornamenti di Aurora MySQL 04/05/2017 (versione 1.12) (obsoleta) 390

Nuove funzionalità 391
Miglioramenti 391
Integrazione delle correzioni di bug di MySQL. 392
Aggiornamenti di Aurora MySQL 23/02/2017 (versione 1.11) (obsoleta) 392
Nuove caratteristiche 393
Miglioramenti 393
Integrazione delle correzioni di bug di MySQL. 396
Aggiornamenti di Aurora MySQL 12/01/2017 (versione 1.10.1) (obsoleta) 396
Nuove caratteristiche 396
Miglioramenti 397
Aggiornamenti di Aurora MySQL 14/12/2016 (versione 1.10) (obsoleta) 397
Nuove caratteristiche 397
Miglioramenti 399
Integrazione delle correzioni di bug di MySQL. 399
Aggiornamenti di Aurora MySQL: 10/11/2016 (versioni 1.9.0, 1.9.1) (obsolete) 400
Nuove caratteristiche 400
Miglioramenti 401
Aggiornamenti di Aurora MySQL 26/10/2016 (versione 1.8.1) (obsoleta) 401
Miglioramenti 401
Integrazione delle correzioni di bug di MySQL. 401
Aggiornamenti di Aurora MySQL 18/10/2016 (versione 1.8) (obsoleta) 402
Nuove funzionalità 402
Miglioramenti 402
Integrazione delle correzioni di bug di MySQL. 403
Aggiornamenti di Aurora MySQL 20/09/2016 (versione 1.7.1) (obsoleta) 404
Miglioramenti 404
Aggiornamenti di Aurora MySQL 30/08/2016 (versione 1.7.0) (obsoleta) 404
Nuove caratteristiche 404
Miglioramenti 405
Integrazione delle correzioni di bug di MySQL. 405
Aggiornamenti di Aurora MySQL 01/06/2016 (versione 1.6.5) (obsoleta) 406
Nuove caratteristiche 406
Miglioramenti 406
Integrazione delle correzioni di bug di MySQL. 407
Aggiornamenti di Aurora MySQL 06/04/2016 (versione 1.6) (obsoleta) 407
Nuove caratteristiche 407

Miglioramenti 408
Integrazione delle correzioni di bug di MySQL. 409
Aggiornamenti di Aurora MySQL 11/01/2016 (versione 1.5) (obsoleta) 410
Miglioramenti 410
Integrazione delle correzioni di bug di MySQL. 410
Aggiornamenti di Aurora MySQL 03/12/2015 (versione 1.4) (obsoleta) 411
Nuove caratteristiche 411
Miglioramenti 411
Integrazione delle correzioni di bug di MySQL. 412
Aggiornamenti di Aurora MySQL 16/10/2015 (versioni 1.2, 1.3) (obsolete) 413
Correzioni 413
Miglioramenti 413
Integrazione delle correzioni di bug di MySQL. 414
Aggiornamenti di Aurora MySQL 24/08/2015 (versione 1.1) (obsoleta) 416
Correzione dei bug di MySQL attraverso gli aggiornamenti di Aurora MySQL 418
Correzione dei bug di MySQL attraverso gli aggiornamenti di Aurora MySQL 3.x 418
Correzione dei bug di MySQL attraverso gli aggiornamenti di Aurora MySQL 2.x 437
Correzione dei bug di MySQL attraverso gli aggiornamenti di Aurora MySQL 1.x 459
Vulnerabilità di sicurezza risolte in Aurora MySQL 480
Cronologia dei documenti 488
..... dvii

Note di rilascio di Amazon Aurora edizione compatibile con MySQL

Le versioni di Amazon Aurora compatibili con MySQL vengono aggiornate regolarmente. Gli aggiornamenti vengono applicati ai cluster database di Aurora MySQL durante le finestre di manutenzione del sistema. La tempistica di applicazione degli aggiornamenti dipende dall' Regione AWS impostazione della finestra di manutenzione per il cluster DB e anche dal tipo di aggiornamento.

Le versioni di Amazon Aurora MySQL vengono rese disponibili a tutti Regioni AWS nel corso di più giorni. Alcune Regioni possono mostrare temporaneamente una versione del motore che non è ancora disponibile in una Regione diversa. Le note di rilascio vengono pubblicate dopo che le nuove versioni sono disponibili in tutte le regioni.

Gli aggiornamenti vengono applicati a tutte le istanze in un cluster di database contemporaneamente. Un aggiornamento richiede il riavvio del database su tutte le istanze in un cluster di database.
Di conseguenza potrebbero verificarsi 20 - 30 secondi di inattività, al termine dei quali si potrà ricominciare a usare i cluster database o i cluster. Puoi visualizzare o modificare le impostazioni della finestra di manutenzione dalla Console di gestione AWS.

Argomenti

- Calendari di rilascio per Amazon Aurora MySQL
- Aggiornamenti del motore del database per Amazon Aurora MySQL versione 3
- Aggiornamenti del motore del database Amazon Aurora MySQL versione 2
- Aggiornamenti del motore del database per Amazon Aurora MySQL versione 1 (obsoleta)
- Correzione dei bug di MySQL attraverso gli aggiornamenti del motore di database Aurora MySQL
- Vulnerabilità di sicurezza risolte in Aurora MySQL

Calendari di rilascio per Amazon Aurora MySQL

I calendari di rilascio in questa pagina possono aiutarti a pianificare gli aggiornamenti delle versioni principali e secondarie. Per ulteriori informazioni sugli upgrade, il controllo delle versioni e il ciclo di vita di Amazon Aurora, consulta le versioni di Amazon Aurora.

Argomenti

- Calendario dei rilasci per le versioni principali di Aurora MySQL
- Calendario dei rilasci per le versioni secondarie di Aurora MySQL

Calendario dei rilasci per le versioni principali di Aurora MySQL

Le versioni principali di Aurora MySQL sono disponibili con il supporto standard almeno fino alla fine del ciclo di vita della community corrispondente. Puoi continuare a eseguire una versione principale a pagamento dopo la data di fine del supporto standard per Aurora. Per ulteriori informazioni, consulta Using Amazon RDS Extended Support e i prezzi di Amazon Aurora.

Aurora MySQL attualmente supporta le seguenti versioni principali.

> **Note**
>
> È inoltre possibile visualizzare le informazioni sulle date di supporto per le principali versioni del motore utilizzando l'API AWS CLI o RDS. Per ulteriori informazioni, consulta Visualizzazione delle date di supporto per le versioni del motore con supporto esteso per Amazon RDS.

Versione principale della community	Versione principale di Aurora	Data di fine vita nella community	Data di fine del supporto standard per Aurora	Data di inizio validità dei prezzi per l'estensione del supporto RDS (1 anno)	Data di inizio validità dei prezzi per l'estensione del supporto RDS (3 anni)	Data di fine dell'estensione del supporto RDS	Versioni secondarie idonee per l'estensione del supporto per Amazon RDS
MySQL 8.0	Aurora MySQL versione 3	Aprile 2026	30 aprile 2028	1 maggio 2028	Non applicabile	31 luglio 2029	Da definire
MySQL 5.7	Aurora MySQL versione 2	Ottobre 2023	31 ottobre 2024	1 dicembre 2024	Non applicabile	28 febbraio 2027	Aurora MySQL 2.11 e 2.12
MySQL 5.6 (obsoleto)	Aurora MySQL versione 1 (obsoleta)	5 febbraio 2021	28 febbraio 2023	Non applicabile	Non applicabile	Non applicabile	Non applicabile

Note

Il supporto esteso per Amazon RDS per Aurora MySQL versione 2 inizia il 1° novembre 2024, ma non verrà addebitato alcun costo fino al 1° dicembre 2024. Tra il 1° novembre e il 30 novembre 2024, tutti i cluster di database Aurora MySQL versione 2 sono coperti dal supporto esteso per Amazon RDS.

Calendario dei rilasci per le versioni secondarie di Aurora MySQL

Aurora MySQL attualmente supporta le seguenti versioni secondarie.

In generale, le versioni secondarie di Aurora vengono rilasciate trimestralmente. La pianificazione del rilascio potrebbe variare in base a funzionalità aggiuntive o correzioni.

Le versioni secondarie possono raggiungere la fine del supporto standard prima delle versioni principali corrispondenti. Ad esempio, la versione 3.07 raggiungerà la fine del supporto standard ad agosto 2025, mentre la versione principale 3 raggiungerà la fine del supporto standard il 30 aprile 2028. Amazon RDS supporterà versioni secondarie 3.* aggiuntive rilasciate tra queste date.

I costi di Amazon RDS Extended Support si applicano solo a determinate versioni secondarie dopo che una versione principale è idonea per l'Extended Support. Per ulteriori informazioni sulle versioni principali idonee per Extended Support, vedereCalendario dei rilasci per le versioni principali di Aurora MySQL.

Versione Aurora MySQL	Data di rilascio di Aurora MySQL	Data di fine del supporto standard per Aurora MySQL
3.10 (Compatibile con Community MySQL 8.0.42) (LTS)	31 luglio 2025	30 aprile 2028
3.09 (Compatibile con Community MySQL 8.0.40)	14 maggio 2025	14 maggio 2026
3.08 (Compatibile con Community MySQL 8.0.39)	18 novembre 2024	31 marzo 2026
3.04 (Compatibile con Community MySQL 8.0.28) (LTS)	31 luglio 2023	31 ottobre 2026
2.12 [1] (Compatibile con Community MySQL 5.7.40 o 5.7.44 [2])	25 luglio 2023	31 ottobre 2024
2.11 [1] (Compatibile con Community MySQL 5.7.12)	25 ottobre 2022	31 ottobre 2024

LTS — Versioni con supporto a lungo termine (LTS) di Aurora MySQL. Per ulteriori informazioni, consulta Supporto a lungo termine (LTS) e versioni beta per Amazon Aurora MySQL.

[1] Questa versione secondaria continuerà a essere disponibile quando la versione principale sarà disponibile in Amazon RDS Extended Support.

[2] Le versioni di Aurora MySQL da 2.12 a 2.12.1 sono compatibili con la versione 5.7.40 di MySQL e le versioni 2.12.2 e successive sono compatibili con la versione 5.7.44 di MySQL.

Aggiornamenti del motore del database per Amazon Aurora MySQL versione 3

Di seguito sono riportati alcuni aggiornamenti del motore del database per Amazon Aurora MySQL versione 3:

Argomenti

- Aggiornamenti del motore di database Aurora MySQL 2025-11-13 (versione 3.11.0, compatibile con MySQL 8.0.43)
- Aggiornamenti del motore di database Aurora MySQL 2025-11-20 (versione 3.10.2, compatibile con MySQL 8.0.42)
- Aggiornamenti del motore di database Aurora MySQL 2025-09-30 (versione 3.10.1, compatibile con MySQL 8.0.42)
- Aggiornamenti del motore di database Aurora MySQL 2025-07-31 (versione 3.10.0, compatibile con MySQL 8.0.42)
- Aggiornamenti del motore di database Aurora MySQL 2025-05-14 (versione 3.09.0, compatibile con MySQL 8.0.40)
- Aggiornamenti del motore di database Aurora MySQL 2025-04-07 (versione 3.08.2, compatibile con MySQL 8.0.39)
- Aggiornamenti del motore di database Aurora MySQL 2025-02-17 (versione 3.08.1, compatibile con MySQL 8.0.39)
- Aggiornamenti del motore di database Aurora MySQL 2024-11-18 (versione 3.08.0, compatibile con MySQL 8.0.39)
- Aggiornamenti del motore di database Aurora MySQL 2024-07-23 (versione 3.07.1, compatibile con MySQL 8.0.36)
- Aggiornamenti del motore di database Aurora MySQL 2024-06-04 (versione 3.07.0, compatibile con MySQL 8.0.36)
- Aggiornamenti del motore di database Aurora MySQL 2024-06-26 (versione 3.06.1, compatibile con MySQL 8.0.34)
- Aggiornamenti del motore di database Aurora MySQL 2024-03-07 (versione 3.06.0, compatibile con MySQL 8.0.34)
- Aggiornamenti del motore di database Aurora MySQL 2024-01-31 (versione 3.05.2, compatibile con MySQL 8.0.32)

- Aggiornamenti del motore di database Aurora MySQL 2023-11-21 (versione 3.05.1, compatibile con MySQL 8.0.32)
- Aggiornamenti del motore del database di Aurora MySQL 30/10/2023 (versione 3.05.0.1, compatibile con MySQL 8.0.32) beta
- Aggiornamenti del motore del database di Aurora MySQL 25/10/2023 (versione 3.05.0, compatibile con MySQL 8.0.32)
- Aggiornamenti del motore di database Aurora MySQL 2025-05-05 (versione 3.04.4, compatibile con MySQL 8.0.28)
- Aggiornamenti del motore di database Aurora MySQL 2024-06-26 (versione 3.04.3, compatibile con MySQL 8.0.28)
- Aggiornamenti del motore di database Aurora MySQL 2024-03-15 (versione 3.04.2, compatibile con MySQL 8.0.28)
- Aggiornamenti del motore di database Aurora MySQL 2023-11-13 (versione 3.04.1, compatibile con MySQL 8.0.28)
- Aggiornamenti del motore di database Aurora MySQL 2023-07-31 (versione 3.04.0, compatibile con MySQL 8.0.28)
- Aggiornamenti del motore di database Aurora MySQL 2023-12-08 (versione 3.03.3) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 2023-08-29 (versione 3.03.2) (obsoleto)
- Aggiornamenti del motore di database Aurora MySQL 2023-05-11 (versione 3.03.1) (obsoleto)
- Aggiornamenti del motore di database Aurora MySQL (versione 3.03.0) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 2023-04-17 (versione 3.02.3) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 2022-11-18 (versione 3.02.2) (obsoleto)
- Aggiornamenti del motore di database Aurora MySQL 2022-09-07 (versione 3.02.1) (obsoleto)
- Aggiornamenti del motore di database Aurora MySQL 2022-04-20 (versione 3.02.0) (obsoleto)
- Aggiornamenti del motore di database Aurora MySQL 2022-04-15 (versione 3.01.1) (obsoleto)
- Aggiornamenti del motore di database Aurora MySQL 2021-11-18 (versione 3.01.0) (obsoleto)

Aggiornamenti del motore di database Aurora MySQL 2025-11-13 (versione 3.11.0, compatibile con MySQL 8.0.43)

Versione: 3.11.0

Aurora MySQL 3.11.0 è disponibile a livello generale. Le versioni di Aurora MySQL 3.11 sono compatibili con MySQL 8.0.43. Per ulteriori informazioni sulle modifiche apportate alla community consulta MySQL 8.0 Release Notes.

Per informazioni sulle nuove funzionalità di Aurora MySQL versione 3, consulta Aurora MySQL versione 3 compatibile con MySQL 8.0. Per le differenze tra Aurora MySQL versione 3 e Aurora MySQL versione 2, consulta Confronto tra Aurora MySQL versione 2 e Aurora MySQL versione 3. Per un confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition, consulta Confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition nella Amazon Aurora User Guide.

Puoi eseguire un aggiornamento sul posto sfruttando l'applicazione di patch ZDP (zero-downtime patching), ripristinare uno snapshot o avviare un aggiornamento gestito utilizzando blue/green Amazon RDS Deployments da qualsiasi cluster Aurora MySQL versione 2 Blue/Green attualmente supportato in un cluster Aurora MySQL versione 3.11.0.

Per informazioni sulla pianificazione di un aggiornamento a Aurora MySQL versione 3, vedere Pianificazione di un aggiornamento della versione principale per un cluster Aurora MySQL. Per informazioni generali sugli upgrade Aurora MySQL, consulta Upgrading Aurora MySQL DB Clusters nella Amazon Aurora User Guide.

Per informazioni sulla risoluzione dei problemi, consulta Risoluzione dei problemi per l'aggiornamento in loco di Aurora MySQL nella Guida per l'utente di Amazon Aurora.

In caso di domande o dubbi,Supportoè disponibile nei forum della community e tramite. Supporto Per ulteriori informazioni, consulta la sezione Manutenzione di un cluster Aurora DB nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Problemi di sicurezza risolti e CVEs

Medio 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

Basso: CVEs

- CVE-2025-50098
- CVE-2025-50100
- CVE-2025-50104

Miglioramenti della disponibilità

- È stato risolto un problema relativo al ripristino logico `aurora_enhanced_binlog` abilitato che poteva impedire il riavvio del database in caso di transazioni interrotte.
- È stato risolto un problema che poteva causare il riavvio dell'istanza di writer quando l'inoltro globale di scrittura o l'inoltro di scrittura locale venivano disabilitati.
- È stato risolto un problema che poteva causare la mancata risposta di writer quando l'inoltro di scrittura veniva disabilitato mentre due o più repliche inoltravano le richieste.

- È stato risolto un problema che poteva causare il riavvio del motore durante `KILL <query-id>` l'esecuzione dopo l'esecuzione `EXPLAIN FOR CONNECTION <query-id>` di una query parallela in esecuzione.
- È stato risolto un problema a causa del quale la funzione di prevenzione out-of-memory (OOM) non manteneva in modo persistente il valore del parametro `aurora_oom_response` DB configurato dopo il riavvio del database.
- È stato risolto un problema che impediva agli utenti con `CONNECTION_ADMIN` o `SUPER` privilegi di eseguire una connessione aggiuntiva oltre il `max_connections` limite, come supportato in MySQL Community Edition.
- È stato risolto un problema con la scalabilità di Aurora Serverless V2 che provocava il riavvio delle istanze DB impedendo lo scambio di pagine di memoria critiche.
- È stato risolto un problema che poteva causare il riavvio delle istanze Aurora MySQL Serverless v2 quando il `innodb_purge_threads` parametro veniva configurato manualmente su un valore diverso da quello predefinito. Il `innodb_purge_threads` parametro è ora gestito automaticamente per le istanze Aurora Serverless v2 e non può essere modificato.
- È stato risolto un problema che poteva causare il fallimento del riavvio di un'istanza di lettura quando il writer eseguiva un numero elevato di operazioni DDL.
- È stato risolto un problema che poteva causare la mancata risposta dell'istanza writer se le istanze del lettore si riavviavano durante l'utilizzo di Global Write Forwarding o Local Write Forwarding.
- È stato risolto un problema che poteva causare il riavvio del motore durante il patching senza interruzioni (ZDP) /il riavvio senza interruzioni (ZDR), durante la conservazione delle connessioni con transazioni attive. SSL/TLS
- È stato risolto un problema che, in rare condizioni, poteva causare l'invio delle modifiche sull'istanza writer all'istanza del lettore in un ordine incompatibile con i thread di elaborazione delle query. Ciò può portare a un punto morto tra il thread di elaborazione delle query e il thread di applicazione della replica, che a sua volta può causare il riavvio dell'istanza del lettore.
- È stato risolto un problema che impediva riavvii non necessari del server di database dovuti alla valutazione errata dello stato dell'agente di monitoraggio.

Miglioramenti generali

- È stato risolto un problema a causa del quale le scritture sul database potevano bloccarsi durante l'esecuzione di una transazione di lunga durata che causava il riavvio del database o il fallimento dell'aggiornamento di una versione principale.

- È stato risolto un problema a causa del quale le repliche venivano riavviate erroneamente quando si accedeva al writer.
- Prestazioni di database ottimizzate per istanze di database con grandi volumi di archiviazione Aurora.
- È stato risolto un problema nell'ottimizzatore di intervallo a causa del quale le espressioni costanti non venivano riconosciute durante il calcolo dei limiti dell'intervallo. (Bug della community #112737)
- È stato risolto un problema a causa del quale l'esportazione di uno snapshot DB su Amazon S3 risultava più lenta durante l'esportazione di tabelle contenenti colonne JSON con valori nulli.
- È stato risolto un problema a causa del quale i file di log binari temporanei non venivano puliti correttamente dopo il rollback delle transazioni quando si utilizzava la registrazione binaria. Questa correzione impedisce il consumo di spazio di archiviazione dovuto ai file temporanei conservati e, in alcuni casi, può anche prevenire anomalie nei file di registro binari.
- Prestazioni IOPS di scrittura migliorate quando la variabile `innodb_flush_log_at_trx_commit` di sistema è impostata su. `0`
- È stato risolto un problema a causa del quale zero downtime patching (ZDP)/zero downtime restart (ZDR) potevano causare il riavvio dell'istanza DB durante il ripristino degli avvisi con codici di errore non validi.
- Disattiva automaticamente `aurora_oom_response` le azioni (tranne la stampa, se configurata) quando `aurora_oom_response` non riesce a risolvere la pressione della memoria dopo un periodo di soglia (nell'ordine di pochi minuti).
- È stato risolto un problema a causa del quale una colonna con aggiornamenti JSON parziali falliva nell'esportazione parallela, causando un fallback interno verso un'esportazione Amazon RDS molto più lenta.
- È stato risolto un problema che poteva causare problemi di gestione della memoria durante le operazioni di query parallele sulla tabella con i campi blob.
- È stato risolto un problema che poteva causare l'`Previous_gtids`esclusione di alcuni eventi GTIDs con Enhanced Binlog abilitato e `gtid_mode` impostato su o. `ON` `ON_PERMISSIVE`
- È stato risolto un problema che causava un imprevisto «errore interno di inoltro di scrittura» sulle istanze del lettore quando l'inoltro di scrittura era abilitato.
- È stata risolta una condizione di gara che poteva causare letture errate delle pagine dal buffer pool durante le operazioni di scale-down di Aurora Serverless o durante l'eliminazione delle pagine dal pool di buffer. Ref community Bug #116305.
- È stato risolto un problema per cui l'esportazione di uno snapshot DB su Amazon S3 era più lenta durante l'esportazione di tabelle contenenti indici secondari con espressioni generate.

- È stato risolto un problema che causava parametri imprecisi quando più connessioni si interrompevano `AbortedClients` in modo imprevisto.
- È stata aggiunta una nuova variabile globale `aurora_lambda_request_timeout` per consentire agli utenti di configurare i timeout delle AWS Lambda richieste (impostazione predefinita: 10 secondi). Per ulteriori informazioni sull'invocazione di una funzione Lambda da un cluster Aurora MySQL DB, vedere Richiamo di una funzione Lambda da un cluster Aurora MySQL DB.
- È stato risolto un problema a causa del quale l'inoltro locale delle scritture smetteva di funzionare dopo il riavvio dell'istanza del database con un riavvio senza tempi di inattività.
- È stato risolto un problema che poteva causare il fallimento di un'operazione di riavvio dell'istanza di database se `max_user_connections` impostata su un valore basso.
- Prestazioni di esportazione parallela migliorate ottimizzando il processo di bootstrap per grandi volumi (>64 TB) e riducendo il tempo complessivo delle operazioni di esportazione.
- È stato risolto un problema che poteva causare il riavvio dell'istanza writer durante l'esecuzione `ALTER TABLE` in parallelo con le query di lettura.
- È stato risolto un problema che poteva far sì che le esportazioni dei cluster DB richiedessero molto più tempo del previsto in presenza di tabelle di dimensioni superiori a 14 TB.
- È stato risolto un problema che causava un tracciamento impreciso delle richieste di query parallele durante l'esecuzione di `EXPLAIN ANALYZE` istruzioni in cui il `Aurora_pq_request_in_progress` contatore non veniva aggiornato con precisione.
- È stato risolto un problema che, in condizioni non comuni, può causare il riavvio dell'istanza del database quando il volume del database si avvicina alla dimensione massima consentita.
- È stato risolto un problema che poteva causare il riavvio di un'istanza di database durante il ripristino delle connessioni durante il riavvio senza tempi di inattività e l'applicazione di patch senza tempi di inattività.

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla 8.0.43 inclusa. Per ulteriori informazioni, consulta Correzione dei bug di MySQL attraverso gli aggiornamenti del motore di database Aurora MySQL 3.x.

- È stato risolto un problema a causa del quale una query del modulo `SELECT 1 FROM t WHERE CAST(a AS UNSIGNED INTEGER) = 1 AND a = (SELECT 1 FROM t)` provocava un errore di asserzione. `item_func.cc` (Correzione di bug #36128964 della community)

- Una richiesta del modulo `SELECT 1 FROM t WHERE CAST(a AS UNSIGNED INTEGER) = 1 AND a = (SELECT 1 FROM t)` ha portato a un'affermazione `initem_func.cc`. (Bug #36128964)

Per ulteriori informazioni sulle modifiche apportate alla community, consulta le Note di rilascio di MySQL 8.0.43.

Aggiornamenti del motore di database Aurora MySQL 2025-11-20 (versione 3.10.2, compatibile con MySQL 8.0.42)

Versione: 3.10.2

Aurora MySQL 3.10.2 è disponibile a livello generale. Le versioni di Aurora MySQL 3.10 sono compatibili con MySQL 8.0.42. Per ulteriori informazioni sulle modifiche apportate alla community consulta MySQL 8.0 Release Notes.

Per informazioni sulle nuove funzionalità di Aurora MySQL versione 3, consulta Aurora MySQL versione 3 compatibile con MySQL 8.0. Per le differenze tra Aurora MySQL versione 3 e Aurora MySQL versione 2, consulta Confronto tra Aurora MySQL versione 2 e Aurora MySQL versione 3. Per un confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition, consulta Confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition nella Amazon Aurora User Guide.

Puoi eseguire un aggiornamento sul posto sfruttando Zero Downtime Patching (ZDP), ripristinare uno snapshot o avviare un aggiornamento gestito blue/green utilizzando Amazon RDS Deployments da Blue/Green qualsiasi cluster Aurora MySQL versione 2 attualmente supportato in un cluster Aurora MySQL versione 3.10.2.

Per informazioni sulla pianificazione di un aggiornamento a Aurora MySQL versione 3, vedere Pianificazione di un aggiornamento della versione principale per un cluster Aurora MySQL. Per informazioni generali sugli upgrade Aurora MySQL, consulta Upgrading Aurora MySQL DB Clusters nella Amazon Aurora User Guide.

Per informazioni sulla risoluzione dei problemi, consulta Risoluzione dei problemi per l'aggiornamento in loco di Aurora MySQL nella Guida per l'utente di Amazon Aurora.

In caso di domande o dubbi,Supportoè disponibile nei forum della community e tramite. Supporto Per ulteriori informazioni, consulta la sezione Manutenzione di un cluster Aurora DB nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Miglioramenti della disponibilità:

- È stato risolto un problema che poteva causare il riavvio del motore durante Zero-downtime patching (ZDP) /Zero-downtime restart (ZDR), quando si preservavano le connessioni con transazioni attive. SSL/TLS
- È stato risolto un problema che poteva causare il riavvio del motore durante l'esecuzione dopo l'esecuzione di una Parallel Query in esecuzione. `KILL <query-id>` `EXPLAIN FOR CONNECTION <query-id>`
- È stato risolto un problema che poteva causare la mancata risposta di writer quando l'inoltro di scrittura veniva disabilitato mentre due o più repliche inoltravano le richieste.
- È stato risolto un problema che poteva causare il riavvio dell'istanza di writer quando l'inoltro globale di scrittura o l'inoltro di scrittura locale venivano disabilitati.

Miglioramenti generali:

- È stato risolto un problema che causava un tracciamento impreciso delle richieste di Parallel Query durante l'esecuzione di `EXPLAIN ANALYZE` istruzioni in cui il `Aurora_pq_request_in_progress` contatore non veniva aggiornato con precisione.
- Disattiva automaticamente `aurora_oom_response` le azioni (tranne la stampa, se configurata) quando `aurora_oom_response` non riesce a risolvere la pressione della memoria dopo un periodo di soglia (dell'ordine di pochi minuti).
- Prestazioni IOPS di scrittura migliorate quando la variabile di sistema `innodb_flush_log_at_trx_commit` è impostata su. `0`
- È stato risolto un problema a causa del quale le repliche venivano riavviate erroneamente quando si accedeva al writer.
- È stato risolto un problema a causa del quale le scritture sul database potevano bloccarsi durante l'esecuzione di una transazione di lunga durata con conseguente riavvio del database o mancata esecuzione dell'aggiornamento della versione principale.

Aggiornamenti del motore di database Aurora MySQL 2025-09-30 (versione 3.10.1, compatibile con MySQL 8.0.42)

Versione: 3.10.1

Aurora MySQL 3.10.1 è disponibile a livello generale. Le versioni di Aurora MySQL 3.10 sono compatibili con MySQL 8.0.42. Per ulteriori informazioni sulle modifiche apportate alla community consulta MySQL 8.0 Release Notes.

Per informazioni sulle nuove funzionalità di Aurora MySQL versione 3, consulta Aurora MySQL versione 3 compatibile con MySQL 8.0. Per le differenze tra Aurora MySQL versione 3 e Aurora MySQL versione 2, consulta Confronto tra Aurora MySQL versione 2 e Aurora MySQL versione 3. Per un confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition, consulta Confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition nella Amazon Aurora User Guide.

Puoi eseguire un aggiornamento sul posto sfruttando Zero Downtime Patching (ZDP), ripristinare uno snapshot o avviare un aggiornamento gestito blue/green utilizzando Amazon RDS Deployments da Blue/Green qualsiasi cluster Aurora MySQL versione 2 attualmente supportato in un cluster Aurora MySQL versione 3.10.1.

Per informazioni sulla pianificazione di un aggiornamento ad Aurora MySQL versione 3, vedere Pianificazione di un aggiornamento della versione principale per un cluster Aurora MySQL. Per informazioni generali sugli aggiornamenti di Aurora MySQL, consulta Aggiornamento dei cluster database Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Per informazioni sulla risoluzione dei problemi, consulta Risoluzione dei problemi per l'aggiornamento in loco di Aurora MySQL nella Guida per l'utente di Amazon Aurora.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta la sezione Manutenzione di un cluster Aurora DB nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Miglioramenti della disponibilità:

- È stato risolto un problema relativo all'ordine di blocco delle pagine inviate alle istanze di Reader. Può causare il riavvio dell'istanza del lettore a causa di un deadlatch durante l'esecuzione di query SELECT su tabelle aggiornate dal relativo autore.
- È stato risolto un problema che poteva causare il fallimento del riavvio di un'istanza di lettura quando il writer eseguiva un numero elevato di operazioni DDL.
- È stato risolto un problema che poteva causare il riavvio delle istanze Aurora MySQL Serverless v2 quando il parametro innodb_purge_threads veniva configurato manualmente su un valore diverso

da quello predefinito. Il parametro innodb_purge_threads è ora gestito automaticamente per le istanze Aurora Serverless v2 e non può essere modificato.

- È stato risolto un problema con la scalabilità di Aurora Serverless V2 che provocava il riavvio delle istanze DB impedendo lo scambio di pagine di memoria critiche.
- È stato risolto un problema per cui l'eliminazione dell'esaurimento della memoria (OOM) da Aurora non manteneva il valore del parametro DB aurora_oom_response configurato dopo il riavvio del database.
- È stata risolta una condizione di gara che poteva causare problemi di disponibilità dopo un failover nei cluster Multi-AZ.
- È stato risolto un problema nel ripristino logico con aurora_enhanced_binlog abilitato che poteva impedire il riavvio del database in caso di transazioni interrotte.
- È stato risolto un problema a causa del quale le istanze del database potevano non rispondere durante i carichi di lavoro di Parallel Query (PQ).
- È stato risolto un problema che impediva agli utenti con privilegi CONNECTION_ADMIN o SUPER di eseguire una connessione aggiuntiva oltre il limite max_connections, come supportato in MySQL Community Edition.

Miglioramenti generali:

- È stato risolto un problema che poteva causare il riavvio continuo del server di database quando veniva consentito al volume di raggiungere la dimensione massima a causa dell'errata autorizzazione di determinati tipi di query dopo che il volume superava una determinata soglia.
- È stato risolto un problema che poteva causare la mancata risposta dell'istanza writer se le istanze Reader si riavviavano durante l'utilizzo di Global Write Forwarding o Local Write Forwarding.
- È stato risolto un problema che poteva causare il riavvio dell'istanza di writer durante l'esecuzione di ALTER TABLE in parallelo con le query di lettura.
- È stato risolto un problema per migliorare la disponibilità con l'avvio di grandi volumi (> 64 TB) durante l'operazione di esportazione parallela.
- È stato risolto un problema che poteva causare il fallimento di un'operazione di riavvio dell'istanza di database se max_user_connections era impostato su un valore basso.
- È stato risolto un problema che causava AbortedClients metriche imprecise quando più connessioni si interrompevano in modo imprevisto.

- È stata risolta una condizione di gara che poteva causare letture errate delle pagine dal buffer pool durante le operazioni di scale-down di Aurora Serverless o durante l'eliminazione delle pagine dal pool di buffer. Ref community Bug #116305.
- È stato risolto un problema che causava un «errore interno di inoltro di scrittura» imprevisto sulle istanze del lettore quando l'inoltro di scrittura è abilitato.
- È stato risolto un problema che poteva causare l'esclusione dell'evento binlog Previous_GTIDS GTIDs con Enhanced Binlog abilitato e gtid_mode impostato su ON o ON_PERMISSIVE.
- È stato risolto un problema che può causare problemi di gestione della memoria durante le operazioni di query parallele sulla tabella con i campi blob.
- È stato risolto il problema per cui una colonna con aggiornamenti JSON parziali non riusciva a eseguire l'esportazione parallela, causando il fallback interno verso un'esportazione RDS molto più lenta.
- È stato risolto un problema che poteva causare riavvii imprevisti delle istanze in caso di elevato carico di lavoro di inoltro simultaneo delle scritture.
- È stato risolto un problema per cui Zero Downtime Patching (ZDP)/Zero Downtime Restart (ZDR) potevano causare il riavvio dell'istanza DB durante il ripristino degli avvisi con codici di errore non validi.
- È stato risolto un problema a causa del quale i file di registro binari temporanei non venivano puliti correttamente dopo il rollback delle transazioni quando si utilizzava la registrazione binaria. Questa correzione impedisce un inutile consumo di storage da parte dei file temporanei conservati e, in alcuni casi, potrebbe anche prevenire anomalie nei file di registro binari.

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla 8.0.42 inclusa. Per ulteriori informazioni, consulta Correzione dei bug di MySQL attraverso gli aggiornamenti del motore di database Aurora MySQL 3.x.

- Una query nel formato SELECT 1 FROM t WHERE CAST (a AS UNSIGNED INTEGER) = 1 AND a = (SELECT 1 FROM t) ha portato a un'asserzione in item_func.cc. (Bug #36128964)

Aggiornamenti del motore di database Aurora MySQL 2025-07-31 (versione 3.10.0, compatibile con MySQL 8.0.42)

Versione: 3.10.0

Aurora MySQL 3.10.0 è disponibile a livello generale. Le versioni di Aurora MySQL 3.10 sono compatibili con MySQL 8.0.42. Per ulteriori informazioni sulle modifiche apportate alla community consulta MySQL 8.0 Release Notes.

> **Note**
>
> Questa versione è designata come versione di supporto a lungo termine (LTS). Per ulteriori informazioni, consulta Versioni con supporto a lungo termine (Long-Term Support, LTS) di Aurora MySQL nella Guida per l'utente di Amazon Aurora.
> Si consiglia di non impostare il `AutoMinorVersionUpgrade` parametro su `true` (o abilitare l'aggiornamento automatico della versione secondaria in) per le versioni LTS. Console di gestione AWS Ciò potrebbe comportare l'aggiornamento del cluster DB alla versione di destinazione successiva per la campagna Automatic Minor Version Upgrade, che potrebbe non essere una versione LTS.

Per informazioni sulle nuove funzionalità di Aurora MySQL versione 3, consulta Aurora MySQL versione 3 compatibile con MySQL 8.0. Per le differenze tra Aurora MySQL versione 3 e Aurora MySQL versione 2, consulta Confronto tra Aurora MySQL versione 2 e Aurora MySQL versione 3. Per un confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition, consulta Confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition nella Amazon Aurora User Guide.

Puoi eseguire un aggiornamento sul posto sfruttando Zero Downtime Patching (ZDP), ripristinare uno snapshot o avviare un aggiornamento gestito blue/green utilizzando Amazon RDS Deployments da Blue/Green qualsiasi cluster Aurora MySQL versione 2 attualmente supportato in un cluster Aurora MySQL versione 3.10.0.

Per informazioni sulla pianificazione di un aggiornamento a Aurora MySQL versione 3, vedere Pianificazione di un aggiornamento della versione principale per un cluster Aurora MySQL. Per informazioni generali sugli aggiornamenti di Aurora MySQL, consulta Aggiornamento dei cluster database Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Per informazioni sulla risoluzione dei problemi, consulta Risoluzione dei problemi per l'aggiornamento in loco di Aurora MySQL nella Guida per l'utente di Amazon Aurora.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster Aurora DB nella Guida per l'utente di Amazon Aurora.

Nuove funzionalità

- Aurora MySQL versione 3.10 estende il supporto della cache dei relay log in memoria per le repliche di log binarie. Questa funzionalità, introdotta per la prima volta nella versione 3.05, può migliorare la velocità di replica dei log binari fino al 40%. La cache del relay log in memoria è abilitata per impostazione predefinita per la replica di log binario a thread singolo, la replica multithread con posizionamento automatico GTID abilitato e, a partire dalla versione 3.10, è abilitata anche per la replica multithread con (anche senza). `replica_preserve_commit_order = ON` GTIDs È possibile controllare `aurora_in_memory_relaylog`questa funzionalità utilizzando un nuovo parametro in 3.10,. Per ulteriori informazioni, consulta Ottimizzazioni dei log binari in Aurora MySQL.
- Amazon Aurora ha raddoppiato la sua capacità di storage massima da 128 TiB a 256 TiB, abilitando carichi di lavoro più grandi in un singolo cluster di database. Per utilizzare il limite di archiviazione aumentato con Aurora MySQL, aggiorna il cluster alla versione 3.10 (compatibile con MySQL 8.0.42) o successiva. Dopo l'aggiornamento, lo storage Aurora si ridimensiona automaticamente fino a 256 TiB in base alla quantità di dati nel volume del cluster.

Miglioramenti

Correzioni di sicurezza

Medio: CVEs

- 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

Basso: CVEs

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

Miglioramenti della disponibilità:

- È stato risolto un problema che causava riavvii non necessari del server di database che si verificavano a causa di una valutazione errata dell'avanzamento del ripristino.

Miglioramenti generali:

- Latenza di commit ridotta quando l'opzione I/O Optimized è abilitata.
- È stato risolto un problema che impediva alle istanze del lettore di generare log di errore quando l'inoltro di scrittura è abilitato e il parametro viene modificato. `aurora_replica_read_consistency`
- È stato risolto un problema che poteva causare il riavvio di un'istanza del lettore a causa dell'interazione tra il thread di replica e una query che accedeva a tabelle non presenti nella cache del buffer.
- È stato risolto un problema a causa del quale l'inoltro locale della scrittura smette di funzionare dopo il riavvio dell'istanza del database con un riavvio senza tempi di inattività.
- È stato risolto un problema nell'inoltro di scrittura a causa del quale le query inoltrate potevano comportarsi in modo errato per le istruzioni contenenti opzioni impostate.
- È stato risolto un problema di stabilità a causa del quale l'inserimento di metadati per un tablespace di annullamento provocava un riavvio imprevisto del database.
- È stato risolto un problema che poteva far sì che le esportazioni dei cluster DB richiedessero molto più tempo del previsto in presenza di tabelle di dimensioni superiori a 14 TB.
- È stato risolto un problema che poteva causare una segnalazione errata della variabile di `Innodb_buffer_pool_pages_misc` stato.

- È stato aggiunto il supporto per la conservazione `LAST_INSERT_ID` durante l'applicazione di patch senza interruzioni (ZDP) o il riavvio senza interruzioni (ZDR).
- Aurora MySQL utilizza valori a 8 bit per l'indice virtuale per IDs prevenire problemi di formattazione di annullamento di MySQL, poiché il superamento di questo limite potrebbe causare l'indisponibilità del cluster. Quando si avvicina questo limite, il sistema ora scrive messaggi di avviso nel registro degli errori di MySQL. Se viene raggiunto il limite, i tentativi di aggiungere un nuovo indice restituiscono un errore. Per ulteriori informazioni sulle best practice relative agli indici virtuali, consulta Errori di overflow dell'ID dell'indice virtuale nella documentazione di Aurora MySQL.

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla 8.0.42 inclusa. Per ulteriori informazioni, consulta Correzione dei bug di MySQL attraverso gli aggiornamenti del motore di database Aurora MySQL 3.x.

- L'uscita dal server potrebbe derivare da tentativi simultanei da parte di più thread di registrare e annullare la registrazione degli oggetti Performance Schema dei metadati o di acquisire e rilasciare blocchi di metadati. (Bug 26502135)

Aggiornamenti del motore di database Aurora MySQL 2025-05-14 (versione 3.09.0, compatibile con MySQL 8.0.40)

Versione: 3.09.0

Aurora MySQL 3.09.0 è disponibile a livello generale. Le versioni di Aurora MySQL 3.09 sono compatibili con MySQL 8.0.40. Per ulteriori informazioni sulle modifiche apportate alla community consulta MySQL 8.0 Release Notes.

Per informazioni sulle nuove funzionalità di Aurora MySQL versione 3, consulta Aurora MySQL versione 3 compatibile con MySQL 8.0. Per le differenze tra Aurora MySQL versione 3 e Aurora MySQL versione 2, consulta Confronto tra Aurora MySQL versione 2 e Aurora MySQL versione 3. Per un confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition, consulta Confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition nella Amazon Aurora User Guide.

Puoi eseguire un aggiornamento sul posto che sfrutta zero-downtime-patch, ripristinare uno snapshot o avviare un aggiornamento gestito blue/green utilizzando Amazon RDS Blue/Green Deployments

da qualsiasi cluster Aurora MySQL versione 2 attualmente supportato in un cluster Aurora MySQL versione 3.09.0.

Per informazioni sulla pianificazione di un aggiornamento a Aurora MySQL versione 3, vedere Pianificazione di un aggiornamento della versione principale per un cluster Aurora MySQL. Per informazioni generali sugli aggiornamenti di Aurora MySQL, consulta Aggiornamento dei cluster database Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Per informazioni sulla risoluzione dei problemi, consulta Risoluzione dei problemi per l'aggiornamento in loco di Aurora MySQL nella Guida per l'utente di Amazon Aurora.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta la sezione Manutenzione di un cluster DB nella Guida per l'utente di Amazon Aurora.

Aspetti salienti

- Database globali Aurora MySQL migliorati per consentire alle istanze di lettura secondarie di completare l'avvio e soddisfare le richieste di lettura durante eventi non pianificati (guasti hardware, interruzioni della rete). In precedenza, le istanze del lettore secondario non potevano riavviarsi durante tali eventi. Per ulteriori informazioni, consulta la resilienza interregionale per i cluster secondari di Global Database nella Guida per l'utente di Amazon Aurora.
- Riduzione dei tempi di inattività della scrittura durante gli switchover interregionali di Aurora MySQL Database a meno di un minuto, riducendo al minimo i tempi di inattività durante gli switch regionali pianificati.

Miglioramenti

Correzioni di sicurezza

CVEsCritico:

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

Medio: CVEs

- 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

Miglioramenti della disponibilità:

- È stato risolto un problema per cui le query a più istruzioni inoltrate dalle istanze Reader a Writer potevano bloccarsi quando `innodb_flush_log_at_trx_commit` era impostata su on the writer e diversa da zero `0` sul reader, prevenendo potenziali errori di inoltro della scrittura.
- È stato risolto un problema di deadlock con Enhanced Binlog abilitato che poteva causare il riavvio del database durante l'esecuzione `SHOW BINARY LOGS` simultanea del commit delle transazioni

sui motori BLACKHOLE o dell'esecuzione di istruzioni, evitando potenziali scritture bloccate e problemi di disponibilità delle istanze. `XA PREPARE`

- Sono state corrette le condizioni di gara nell'inoltro di scrittura che potevano causare il riavvio dell'istanza di Aurora writer impedendo l'accettazione di nuove richieste prima del completamento completo delle richieste precedenti, migliorando la stabilità delle operazioni di inoltro di scrittura.
- È stato risolto un problema sulla replica a causa del quale un'interruzione di rete poteva non ristabilire correttamente la connessione con il writer, causando il blocco della replica e il potenziale riavvio dell'istanza.
- La risposta Aurora MySQL Out of Memory (OOM) ora implementa il ridimensionamento a fasi del pool di buffer che riduce gradualmente l'utilizzo della memoria in base allo stato della memoria di sistema (LOW/RESERVED) quando abilitato tramite il parametro DB, offrendo una migliore gestione della memoria durante le situazioni di pressione della memoria. `aurora_oom_response`
- È stato migliorato il tempo di ripristino dei file Binlog durante il riavvio del database ottimizzando il processo di ripristino in modo che richieda un tempo costante indipendentemente dalla dimensione del file Binlog. In precedenza, in alcuni casi, il tempo di ripristino era proporzionale alla dimensione dell'ultimo file Binlog.
- È stato risolto un problema che poteva causare riavvii imprevisti del server MySQL durante l'esecuzione simultanea di operazioni di troncamento delle tabelle InnoDB durante l'interrogazione. `performance_schema.data_lock_waits`
- È stato risolto un problema che poteva causare il riavvio di un'istanza di database quando si eseguivano eventi binlog di grandi dimensioni in condizioni di scarsa archiviazione.
- È stato risolto un problema a causa del quale le operazioni di ridimensionamento del pool di buffer attivate durante la prevenzione dell'esaurimento della memoria (OOM) potevano non rispondere durante scenari con carichi di lavoro elevati, con conseguente possibile riavvio del database.
- È stato risolto un problema che poteva causare un ciclo di riavvio del database durante la creazione di un trigger. Il problema può verificarsi anche quando viene aggiunto un nuovo file Binlog o Relaylog o quando questi file ruotano.
- È stato risolto un problema che poteva causare il riavvio dell'istanza del lettore Aurora quando si utilizzava l'inoltro di scrittura con query di commit implicite o a più istruzioni.

Miglioramenti generali:

- È stato risolto un problema per cui `ALTER TABLE ... REBUILD / OPTIMIZE TABLE` le operazioni potevano consumare una quantità eccessiva di memoria allocando

`innodb_ddl_buffer_size` byte per thread DDL anziché dividere la dimensione del buffer tra i thread, impedendo il potenziale utilizzo eccessivo della memoria durante le operazioni DDL.

- È stato modificato il valore predefinito per`aurora_oom_response`, su tutte le classi di istanze DB che hanno più di 4 GiB di memoria, da print a print, decline, kill_connect. Per ulteriori informazioni, consulta i problemi relativi ad Amazon Aurora MySQL nella Guida per out-of-memory l'utente di Amazon Aurora.
- I seguenti privilegi sono stati aggiunti a:,,,`rds_superuser_role`. `FLUSH_OPTIMIZER_COSTS` `FLUSH_STATUS` `FLUSH_TABLES` `FLUSH_USER_RESOURCES` Per informazioni `surds_superuser_role`, consulta la documentazione relativa agli account utente Amazon Master con Amazon Aurora. Per ulteriori informazioni su questi privilegi dinamici, consulta la documentazione di MySQL.
- A partire da questa versione di Aurora MySQL, l'ottimizzazione rapida degli inserti non è più abilitata. Per informazioni Aurora, consulta Miglioramenti alle prestazioni di Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- È stato risolto un problema relativo a una violazione errata della `max_user_connections` soglia, che causava errori di connessione per alcuni utenti. Ciò si verifica in alcuni casi limite, ad esempio quando le connessioni vengono create e interrotte quasi immediatamente.
- È stato risolto un problema di registrazione di controllo che causava un elevato utilizzo della CPU, con conseguente mancata risposta dell'istanza del server di database.
- È stato risolto un problema di gestione della memoria durante l'utilizzo di transazioni XA, che impediva il possibile riavvio dell'istanza quando Enhanced Binlog era abilitato.
- È stato risolto un problema a causa del quale le prestazioni delle query peggioravano quando l'ottimizzatore effettuava stime dei costi errate a causa dell'aggiornamento errato delle statistiche dell'indice di Bufferpool dopo il riavvio del server del database.
- È stato risolto un problema che impediva ai clienti di disattivare la funzionalità di inoltro di scrittura locale a causa di un thread di lavoro bloccato.
- È stato risolto un problema a causa del quale l'esecuzione del `SHOW BINARY LOGS` comando richiedeva più tempo in un cluster in cui Enhanced Binlog era abilitato o era abilitato in precedenza. Questo problema potrebbe anche causare un aumento della latenza di commit se più `SHOW BINARY LOGS` comandi venivano eseguiti contemporaneamente.

Aggiornamenti e migrazioni:

- È stato risolto un problema per cui Zero Downtime Patching (ZDP) poteva avere esito negativo durante il tentativo di preservare la connessione, appartenente a un utente che era stato interrotto.

Ulteriori informazioni sul `DROP USER` comando e sul suo effetto sulle connessioni attive sono disponibili nella documentazione di MySQL.

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla 8.0.40 inclusa. Per ulteriori informazioni, consulta Correzione dei bug di MySQL attraverso gli aggiornamenti del motore di database Aurora MySQL 3.x.

- Mentre venivano ricevute e applicate transazioni di grandi dimensioni e veniva effettuata una richiesta di interruzione del canale di replica utilizzando`STOP REPLICA`, MySQL non lo faceva correttamente e successivamente non elaborava alcun comando di canale. Inoltre, il processo di spegnimento del server non è stato completato correttamente e ha richiesto l'interruzione del processo MySQL o il riavvio del sistema host. (Bug #115966, Bug #37008345)

Aggiornamenti del motore di database Aurora MySQL 2025-04-07 (versione 3.08.2, compatibile con MySQL 8.0.39)

Versione: 3.08.2

Aurora MySQL 3.08.2 è disponibile a livello generale. Le versioni di Aurora MySQL 3.08 sono compatibili con MySQL 8.0.39. Per ulteriori informazioni sulle modifiche apportate alla community consulta MySQL 8.0 Release Notes.

Per informazioni sulle nuove funzionalità di Aurora MySQL versione 3, consulta Aurora MySQL versione 3 compatibile con MySQL 8.0. Per le differenze tra Aurora MySQL versione 3 e Aurora MySQL versione 2, consulta Confronto tra Aurora MySQL versione 2 e Aurora MySQL versione 3. Per un confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition, consulta Confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition nella Amazon Aurora User Guide.

Le versioni di Aurora MySQL attualmente supportate sono 2.11.*, 2.12.*, 3.04.*, 3.05.*, 3.06.*, 3.07.* e 3.08.*.

Puoi eseguire un aggiornamento sul posto, ripristinare uno snapshot o avviare un aggiornamento gestito blue/green utilizzando Amazon RDS Blue/Green Deployments da qualsiasi cluster Aurora MySQL versione 2 attualmente supportato in un cluster Aurora MySQL versione 3.08.2.

Per informazioni sulla pianificazione di un aggiornamento a Aurora MySQL versione 3, vedere Pianificazione di un aggiornamento della versione principale per un cluster Aurora MySQL. Per informazioni generali sugli aggiornamenti di Aurora MySQL, consulta Aggiornamento dei cluster database Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Per informazioni sulla risoluzione dei problemi, consulta Troubleshooting for Aurora MySQL in-place upgrade nella Amazon Aurora User Guide.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Correzioni di sicurezza:

- CVE-2024-11053

Miglioramenti della disponibilità:

- È stato risolto un problema sulla replica a causa del quale un'interruzione della rete poteva non ristabilire correttamente la connessione con il writer.
- È stato risolto un problema per cui una query inoltrata con più istruzioni poteva fallire se `innodb_flush_log_at_trx_commit` impostata su at writer. `0`
- È stato risolto un problema che poteva causare il riavvio di un'istanza DB di Aurora writer quando l'inoltro di scrittura è abilitato.
- È stato risolto un problema che poteva causare il riavvio di un'istanza DB di Aurora reader quando si utilizzava l'inoltro di scrittura con query di commit implicite o a più istruzioni.

Miglioramenti generali

- A partire da questa versione di Aurora MySQL, l'ottimizzazione rapida degli inserti non è più abilitata. Per informazioni Aurora, consulta Miglioramenti alle prestazioni di Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- È stato risolto un problema relativo a una violazione errata della `max_user_connections` soglia, che causava errori di connessione per alcuni utenti. Ciò si verifica in alcuni casi limite, ad esempio quando le connessioni vengono create e interrotte quasi immediatamente.

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla 8.0.39 inclusa, oltre a quanto segue. Per ulteriori informazioni, consulta MySQL bugs fixed by Aurora MySQL 3.x database engine updates.

- Mentre venivano ricevute e applicate transazioni di grandi dimensioni e veniva effettuata una richiesta di interruzione del canale di replica utilizzando`STOP REPLICA`, MySQL non lo faceva correttamente e successivamente non elaborava alcun comando di canale. Inoltre, il processo di spegnimento del server non è stato completato correttamente e ha richiesto l'interruzione del processo MySQL o il riavvio del sistema host. (Community Bug Fix #115966 e #37008345)

Aggiornamenti del motore di database Aurora MySQL 2025-02-17 (versione 3.08.1, compatibile con MySQL 8.0.39)

Versione: 3.08.1

Aurora MySQL 3.08.1 è disponibile a livello generale. Le versioni di Aurora MySQL 3.08 sono compatibili con MySQL 8.0.39. Per ulteriori informazioni sulle modifiche apportate alla community consulta MySQL 8.0 Release Notes.

Per informazioni sulle nuove funzionalità di Aurora MySQL versione 3, consulta Aurora MySQL versione 3 compatibile con MySQL 8.0. Per le differenze tra Aurora MySQL versione 3 e Aurora MySQL versione 2, consulta Confronto tra Aurora MySQL versione 2 e Aurora MySQL versione 3. Per un confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition, consulta Confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition nella Amazon Aurora User Guide.

Le versioni di Aurora MySQL attualmente supportate sono 2.11.*, 2.12.*, 3.04.*, 3.05.*, 3.06.*, 3.07.* e 3.08.*.

Puoi eseguire un aggiornamento sul posto, ripristinare uno snapshot o avviare un aggiornamento gestito blue/green utilizzando Amazon RDS Blue/Green Deployments da qualsiasi cluster Aurora MySQL versione 2 attualmente supportato in un cluster Aurora MySQL versione 3.08.1.

Per informazioni sulla pianificazione di un aggiornamento a Aurora MySQL versione 3, vedere Pianificazione di un aggiornamento della versione principale per un cluster Aurora MySQL. Per informazioni generali sugli aggiornamenti di Aurora MySQL, consulta Aggiornamento dei cluster database Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Per informazioni sulla risoluzione dei problemi, consulta Risoluzione dei problemi per l'aggiornamento in loco di Aurora MySQL nella Guida per l'utente di Amazon Aurora.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Miglioramenti della disponibilità:

- Sono stati apportati miglioramenti che riducono il tempo di riavvio dell'istanza DB durante i riavvii del database, gli aggiornamenti di patch o versioni secondarie e i failover.

Aggiornamenti e migrazioni

- È stato risolto un problema che poteva causare il fallimento di Zero Downtime Patching (ZDP) durante il tentativo di ripristinare i ruoli e le autorizzazioni degli utenti abbandonati. Per ulteriori informazioni sull'`DROP USER` operazione, vedere DROP USER Statement nella documentazione di MySQL.

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla 8.0.39 inclusa, oltre a quanto segue. Per ulteriori informazioni, consulta MySQL bugs fixed by Aurora MySQL 3.x database engine updates.

Aggiornamenti del motore di database Aurora MySQL 2024-11-18 (versione 3.08.0, compatibile con MySQL 8.0.39)

Versione: 3.08.0

Aurora MySQL 3.08.0 è disponibile a livello generale. Le versioni di Aurora MySQL 3.08 sono compatibili con MySQL 8.0.39. Per ulteriori informazioni sulle modifiche apportate alla community consulta MySQL 8.0 Release Notes.

Per informazioni sulle nuove funzionalità di Aurora MySQL versione 3, consulta Aurora MySQL versione 3 compatibile con MySQL 8.0. Per le differenze tra Aurora MySQL versione 3 e Aurora

MySQL versione 2, consulta Confronto tra Aurora MySQL versione 2 e Aurora MySQL versione 3. Per un confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition, consulta Confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition nella Amazon Aurora User Guide.

Le versioni di Aurora MySQL attualmente supportate sono 2.11.*, 2.12.*, 3.04.*, 3.05.*, 3.06.*, 3.07.* e 3.08.*.

Puoi eseguire un aggiornamento sul posto, ripristinare uno snapshot o avviare un aggiornamento gestito blue/green utilizzando Amazon RDS Blue/Green Deployments da qualsiasi cluster Aurora MySQL versione 2 attualmente supportato in un cluster Aurora MySQL versione 3.08.0.

Per informazioni sulla pianificazione di un aggiornamento ad Aurora MySQL versione 3, vedere Pianificazione di un aggiornamento della versione principale per un cluster Aurora MySQL. Per informazioni generali sugli aggiornamenti di Aurora MySQL, consulta Aggiornamento dei cluster database Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Per informazioni sulla risoluzione dei problemi, consulta Risoluzione dei problemi per l'aggiornamento in loco di Aurora MySQL nella Guida per l'utente di Amazon Aurora.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Nuove funzionalità

- Sono state aggiunte tre nuove CloudWatch metriche Amazon per consentire agli utenti di monitorare il processo di eliminazione di InnoDB:
  - `PurgeBoundary`
  - `PurgeFinishedPoint`
  - `TruncateFinishedPoint`

  Per ulteriori informazioni, consulta i CloudWatchparametri di Amazon Aurora per Amazon Aurora nella Guida per l'utente di Amazon Aurora.
- È stata aggiunta una nuova CloudWatch metrica `TransactionAgeMaximum` per consentire agli utenti di identificare transazioni di lunga durata che potrebbero ostacolare il limite di eliminazione. Per ulteriori informazioni, consulta i CloudWatch parametri di Amazon Aurora per Amazon Aurora nella Guida per l'utente di Amazon Aurora.
- Sono state aggiunte tre nuove CloudWatch metriche per evitare out-of-memory (OOM):

  - `AuroraMillisecondsSpentInOomRecovery`
  - `AuroraNumOomRecoverySuccessful`
  - `AuroraNumOomRecoveryTriggered`

  Per ulteriori informazioni, consulta i CloudWatchparametri di Amazon Aurora per Amazon Aurora nella Guida per l'utente di Amazon Aurora.

- Sono state modificate tre CloudWatch metriche per evitare out-of-memory (OOM) di passare dai totali correnti ai contatori incrementali:
  - `AuroraMemoryNumDeclinedSqlTotal`
  - `AuroraMemoryNumKillConnTotal`
  - `AuroraMemoryNumKillQueryTotal`

  Per ulteriori informazioni, consulta i CloudWatchparametri di Amazon Aurora per Amazon Aurora nella Guida per l'utente di Amazon Aurora.

- Sono state aggiunte due variabili di stato globali per mostrare la quantità di memoria utilizzata dalle tabelle temporanee interne: e. `aurora_temptable_ram_allocation` `aurora_temptable_max_ram_allocation` Queste variabili di stato globali aumentano l'osservabilità e aiutano a diagnosticare i problemi relativi all'utilizzo della memoria interna delle tabelle temporanee.

  Per ulteriori informazioni, consulta le variabili di stato globali di Aurora MySQL nella Guida per l'utente di Amazon Aurora.

- È stata introdotta la nuova variabile di sistema `aurora_optimizer_trace_print_before_purge` per stampare le tracce dell'ottimizzatore nel registro degli errori prima che il server eliminasse le tracce dalla memoria. È possibile attivare un'eliminazione in base alle soglie impostate dalle variabili di sistema optimizer_trace_offset e optimizer_trace_limit.

- È stato aggiunto il supporto per le seguenti classi di istanze DB:
  - db.r7i
  - db.r8g

  Per ulteriori informazioni, consulta Motori DB supportati per classi di istanza database nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Problemi di sicurezza risolti e: CVEs

- È stato introdotto un nuovo utente per la replica dei log binari (binlog),. `rdsrepladmin_priv_checks_user` Per ulteriori informazioni, consulta Privilege checks user for binary log replication nella Amazon Aurora User Guide.
- È stato risolto un problema a causa del quale i parametri di input per le stored procedure Aurora MySQL potevano essere gestiti in modo errato.

Questa versione include tutte le correzioni CVE della community fino a MySQL 8.0.39 incluso. Sono incluse le seguenti correzioni CVE:

- 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

Miglioramenti della disponibilità:

- È stato corretto un difetto che poteva causare il riavvio imprevisto del server dopo l'esecuzione dei comandi DML (Data Manipulation Language) su una tabella in cui sono già state eliminate istantaneamente delle colonne, in modo che il conteggio delle colonne eliminate e non eliminate fosse superiore a 1017.
- È stato risolto un problema che poteva causare la mancata rotazione corretta dei file di registro del database, con conseguente aumento dell'utilizzo dello spazio di archiviazione locale su un'istanza DB.
- È stato risolto un problema che poteva causare l'interruzione della chiusura delle connessioni da parte dell'istanza DB in condizioni di memoria insufficiente quando aurora_oom_response era abilitata, con conseguente riavvio che altrimenti avrebbe potuto essere evitato. out-of-memory
- È stato risolto un problema che poteva causare il riavvio di un'istanza Reader DB durante la liberazione della memoria utilizzata per l'applicazione di registro.
- È stato risolto un problema nel calcolo delle metriche interne per gli indici di ricerca full-text (FTS) che poteva causare il riavvio del database.
- È stato risolto un problema della community che a volte poteva produrre risultati di query errati quando una `LIMIT` clausola veniva utilizzata nelle sottoquery insieme al pushdown delle condizioni dell'indice.
- È stato risolto un problema che causava una violazione dell'accesso alla memoria che portava al rilascio di un oggetto mutex non più di proprietà del thread.
- È stato risolto un problema che poteva causare il riavvio di una replica di log binario (binlog) durante l'elaborazione di un numero elevato di file di log di inoltro durante il ripristino del relay log.
- È stato risolto un problema che poteva causare il riavvio di un'istanza DB di Aurora reader quando l'inoltro di scrittura è abilitato.

- È stato risolto un problema per cui una query contenente più `UNION` clausole poteva allocare una grande quantità di memoria, con conseguente riavvio dell'istanza DB.
- È stato risolto un problema che causava il riavvio dell'istanza Writer DB durante l'esecuzione di una query parallela su un'istanza Reader DB.
- È stato risolto un problema a causa del quale la replica binlog si bloccava sulla replica a causa di un deadlock rilevato nel I/O thread della replica durante l'esecuzione del comando. `FLUSH RELAY LOGS`
- È stato risolto un problema che causava il riavvio di un'istanza DB durante la gestione di set GTID di grandi dimensioni su un cluster DB con binlog avanzato abilitato.
- È stato risolto un problema che può causare il riavvio di una replica binlog quando la cache del relay log in memoria è abilitata. La cache del relay log in memoria è abilitata sulle repliche binlog gestite da Aurora MySQL quando si utilizza la replica di log binario a thread singolo o la replica multithread con posizionamento automatico GTID abilitato.
- È stato risolto un problema che poteva causare il riavvio di un'istanza di replica binlog quando si applicavano le istruzioni DCL (Data Control Language) durante l'avvio del motore di database.
- È stato risolto un problema che, in rare condizioni, poteva causare il riavvio di un'istanza Reader DB a causa di un deadlatch durante l'esecuzione di `SELECT` query su tabelle aggiornate dalla relativa istanza DB di scrittura.
- È stato risolto un problema che poteva causare il riavvio di un'istanza del lettore Aurora Global Database con una sessione di inoltro di scrittura attiva.
- È stato risolto un problema che poteva causare il riavvio delle repliche di lettura di Aurora in caso di alcuni rari ordini di commit delle transazioni sull'istanza Writer DB.
- È stato risolto un problema che può causare il riavvio di un database quando gli eventi pianificati vengono annullati durante l'esecuzione su istanze DB con binlog avanzato abilitato.

Miglioramenti generali:

- È stato risolto un problema a causa del quale la connessione client poteva bloccarsi durante l'applicazione di patch senza interruzioni (ZDP) o il riavvio senza interruzioni (ZDR).
- È stato risolto un problema che, in rari casi, causava il riavvio dell'istanza del database a causa di un problema di gestione della memoria che poteva verificarsi durante la gestione di un errore di tabella aperta.
- È stato risolto un problema che poteva causare errori o avvisi imprevisti di violazione della chiave primaria nelle istruzioni SQL su alcune righe quando si eseguono `INSERT` istruzioni simultanee su

una tabella con una colonna chiave `AUTO_INCREMENT` primaria e una colonna chiave univoca e quando un'`INSERT`istruzione presenta violazioni di chiave univoche su righe diverse.

- Lo strumento Performance Schema `memory/sql/sp_head::main_mem_root` è ora uno strumento controllato. Di conseguenza, la memoria allocata per l'analisi e la rappresentazione dei programmi memorizzati ora contribuisce al raggiungimento dei limiti di memoria per connessione.
- È stato risolto un problema per cui una riga diventava illeggibile tramite l'indice spaziale durante un aggiornamento.
- È stato risolto un problema che impediva agli utenti di disabilitare l'inoltro di scrittura locale.
- È stato risolto un problema a causa del quale una `SELECT COUNT` query poteva restituire un risultato errato durante l'utilizzo dell'inoltro di scrittura.
- È stato risolto un problema che poteva portare a risultati di query errati quando ZDP ripristinava erroneamente le variabili di sessione impostate come suggerimenti nelle query.
- È stato risolto un problema relativo al troncamento automatico delle tablespace di annullamento quando sono più grandi della soglia innodb_max_undo_log_size negli scenari di aggiornamento.
- È stato risolto un problema per cui la latenza di commit e il throughput di commit non venivano misurati quando era impostato su. `innodb_flush_log_at_trx_commit` `0`
- È stato risolto un problema che poteva causare la segnalazione errata del ritardo della replica di Aurora.
- È stato risolto un problema che forniva un valore errato per la `threads_running` variabile quando si utilizzava Aurora Global Database.
- È stato risolto un problema a causa del quale una replica binlog Aurora MySQL con ottimizzazione parallela dell'indice secondario abilitata presentava un riavvio quando si applicavano le modifiche di replica su tabelle con chiavi esterne.
- È stato risolto un problema di riavvio causato dal ridimensionamento prolungato della tabella degli hash di blocco durante gli eventi di ridimensionamento o ridimensionamento.
- È stato risolto un problema che poteva causare il riavvio di un'istanza DB dopo aver disabilitato la registrazione binaria, quando il binlog avanzato era precedentemente abilitato.
- Sono state introdotte ottimizzazioni per ridurre l'utilizzo della memoria durante il ripristino del dizionario logico di dati in presenza di un numero elevato di tabelle.
- È stato risolto un problema a causa del quale un utente poteva riscontrare un `ERROR 1377 (HY000): Fatal error during log purge` errore durante l'esecuzione della stored procedure mysql.rds_set_external_source su una replica binlog, quando la replica dei log binari era già configurata.

- È stato corretto un difetto che impediva la persistenza dei privilegi del ruolo utente dopo ZDR.
- È stato risolto un problema di memoria associato ai ruoli predefiniti del view definer.
- È stato risolto un problema che può causare il riavvio di un'istanza DB quando `SHOW VOLUME STATUS` viene eseguita.
- È stato risolto un problema di riavvio causato dal ridimensionamento prolungato del pool di buffer durante gli eventi di scalabilità verso l'alto o verso il basso.
- È stato risolto un problema di riavvio causato dal logical read ahead (LRA) che accedeva alle pagine liberate a causa del ridimensionamento del buffer durante un evento di ridimensionamento.
- È stato risolto un problema che riguardava `SELECT` le query che restituivano risultati errati quando la query veniva utilizzata `LEFT OUTER JOIN` con la trasformazione semijoin che utilizza la materializzazione come strategia.
- È stato risolto un problema che poteva causare errori nel completamento del processo di disabilitazione dell'inoltro delle scritture.
- È stato risolto un problema a causa del quale le `BlockedTransactions` CloudWatch metriche `ActiveTransactions` e riportavano valori inferiori al previsto.
- È stato risolto un problema a causa del quale la replica binlog si interrompeva quando la replica elabora un'`DELETE`istruzione multitable che elimina in modo esplicito sia una tabella principale che una tabella secondaria.
- È stato risolto un problema che può causare il riavvio di un'istanza DB durante l'elaborazione di un trigger con funzioni definite dall'utente che restituiscono un tipo. `enum`
- È stato risolto un problema per cui la `DMLLatency` CloudWatch metrica mostrava valori errati per un'istanza di replica di log binario durante la replica utilizzando set to. `binlog_format` `row`
- È stato risolto un problema per cui `UPDATE` le query lente `INSERT` ed eseguite da MySQL Event Scheduler non venivano registrate nel registro delle query lente a meno che non fossero precedute da una query lenta. `DELETE` `SELECT`
- È stato risolto un problema che, in rari casi, causava l'indisponibilità intermittente di una replica di lettura Aurora o incongruenze nella definizione della tabella, a volte con l'errore, sulla replica. `Table does not exist` Ciò è dovuto alle query di lettura simultanee sulla replica e alle operazioni DDL (Data Definition Language) sull'istanza Writer DB.
- È stato risolto un problema a causa del quale l'esecuzione del `SHOW BINARY LOGS` comando richiedeva più tempo su un cluster DB in cui il binlog avanzato era abilitato o era abilitato in precedenza. Questo problema potrebbe anche causare un aumento della latenza di commit se più `SHOW BINARY LOGS` comandi venivano eseguiti contemporaneamente.

Aggiornamenti e migrazioni:

- Sono state migliorate le prestazioni dell'aggiornamento della versione principale da Aurora MySQL versione 2 alla versione 3 per cluster DB con un gran numero di oggetti di database (come tabelle, trigger e routine).

  Per classi di istanze DB più grandi, il processo di aggiornamento del database aggiorna i metadati degli oggetti MySQL tradizionali al nuovo dizionario di dati atomico MySQL 8.0 in parallelo utilizzando più thread.
- È stato risolto un problema che causava il fallimento degli aggiornamenti o delle migrazioni quando la versione del motore Aurora MySQL DB di destinazione è 3.04.0 o successiva. Ciò si verifica quando il parametro del cluster `lower_case_table_names` DB è impostato su e le regole di `1` confronto del database MySQL sono incompatibili con i nomi di tabella in minuscolo.

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla 8.0.39 inclusa, oltre a quanto segue. Per ulteriori informazioni, consulta MySQL bugs fixed by Aurora MySQL 3.x database engine updates.

- È stato risolto un problema a causa del quale `NULL` i valori venivano erroneamente omessi dal set di risultati per alcune query con entrambe le operazioni. `JOIN` `UNION` (Correzione di un bug della community #114301)

Aggiornamenti del motore di database Aurora MySQL 2024-07-23 (versione 3.07.1, compatibile con MySQL 8.0.36)

Versione: 3.07.1

Aurora MySQL 3.07.1 è disponibile a livello generale. Le versioni di Aurora MySQL 3.07 sono compatibili con MySQL 8.0.36. Per ulteriori informazioni sulle modifiche apportate alla community consulta MySQL 8.0 Release Notes.

Per informazioni sulle nuove funzionalità di Aurora MySQL versione 3, consulta Aurora MySQL versione 3 compatibile con MySQL 8.0. Per le differenze tra Aurora MySQL versione 3 e Aurora MySQL versione 2, vedi Confronto tra Aurora MySQL versione 2 e Aurora MySQL versione 3. Per

un confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition, consulta Confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition nella Amazon Aurora User Guide.

Le versioni di Aurora MySQL attualmente supportate sono 2.11.*, 2.12.*, 3.03.*, 3.04.*, 3.05.*, 3.06.* e 3.07.*.

Puoi eseguire un aggiornamento sul posto, ripristinare uno snapshot o avviare un aggiornamento gestito blue/green utilizzando Amazon RDS Blue/Green Deployments da qualsiasi cluster Aurora MySQL versione 2 attualmente supportato in un cluster Aurora MySQL versione 3.07.1.

Per informazioni sulla pianificazione di un aggiornamento a Aurora MySQL versione 3, vedere Pianificazione di un aggiornamento della versione principale per un cluster Aurora MySQL. Per informazioni generali sugli aggiornamenti di Aurora MySQL, consulta Aggiornamento dei cluster database Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Per informazioni sulla risoluzione dei problemi, consulta Risoluzione dei problemi per l'aggiornamento in loco di Aurora MySQL nella Guida per l'utente di Amazon Aurora.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Risolti i problemi di sicurezza e CVEs:

- È stato introdotto un nuovo utente per la replica dei log binari (binlog),. `rdsrepladmin_priv_checks_user` Per ulteriori informazioni, consulta Privilege checks user for binary log replication nella Amazon Aurora User Guide.

Questa versione include tutte le correzioni CVE della community fino a MySQL 8.0.36 incluso.

Miglioramenti della disponibilità:

- È stato risolto un problema che poteva causare il riavvio di un'istanza Reader DB durante la liberazione della memoria utilizzata per l'applicazione di registro.
- È stato risolto un problema nel calcolo delle metriche interne per gli indici di ricerca full-text (FTS) che poteva causare il riavvio del database.
- È stato risolto un problema che poteva disabilitare la registrazione binaria quando si verificava un errore durante l'esecuzione di una transazione di grandi dimensioni.

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla 8.0.36 inclusa. Per ulteriori informazioni, consulta Correzione dei bug di MySQL attraverso gli aggiornamenti del motore di database Aurora MySQL 3.x.

Aggiornamenti del motore di database Aurora MySQL 2024-06-04 (versione 3.07.0, compatibile con MySQL 8.0.36)

Versione: 3.07.0

Aurora MySQL 3.07.0 è disponibile a livello generale. Le versioni di Aurora MySQL 3.07 sono compatibili con MySQL 8.0.36. Per ulteriori informazioni sulle modifiche apportate alla community consulta MySQL 8.0 Release Notes.

Per informazioni sulle nuove funzionalità di Aurora MySQL versione 3, consulta Aurora MySQL versione 3 compatibile con MySQL 8.0. Per le differenze tra Aurora MySQL versione 3 e Aurora MySQL versione 2, vedi Confronto tra Aurora MySQL versione 2 e Aurora MySQL versione 3. Per un confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition, consulta Confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition nella Amazon Aurora User Guide.

Le versioni di Aurora MySQL attualmente supportate sono 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.03.*, 3.04.*, 3.05.*, 3.06.* e 3.07.*.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Risolti i problemi di sicurezza e CVEs:

- Supporto abilitato per la crittografia convalidata FIPS, un'implementazione completamente proprietaria. AWS Per ulteriori informazioni, vedi AWS-LC È ora certificato FIPS 140-3 sul Security Blog.AWS

Questa versione include tutte le correzioni CVE della community fino a MySQL 8.0.36 incluso. Sono incluse le seguenti correzioni CVE:

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

Miglioramenti della disponibilità:

- È stato risolto un problema che poteva causare il riavvio di un'istanza Reader DB durante la lettura di una tabella che veniva alterata o eliminata sull'istanza DB di Writer.
- È stato risolto un problema che poteva causare il riavvio di un'istanza DB di Aurora MySQL Writer quando una sessione di inoltro della scrittura veniva chiusa durante l'esecuzione di una query inoltrata.
- È stato risolto un problema che causava il riavvio di un'istanza DB durante la gestione di set GTID di grandi dimensioni su un'istanza abilitata per i log binari.
- È stato risolto un problema durante l'elaborazione di `INSERT` query su tabelle partizionate InnoDB che poteva causare un calo graduale della memoria libera nell'istanza.
- È stato risolto un problema che, in rare condizioni, poteva causare il riavvio delle istanze Reader DB.
- È stato risolto un problema che poteva causare il riavvio di un'istanza di database quando si eseguivano contemporaneamente le istruzioni SHOW STATUS e PURGE BINARY LOGS. `PURGE BINARY LOGS`è un'istruzione gestita che viene eseguita per rispettare il periodo di conservazione binlog configurato dall'utente.
- È stato risolto un problema che può causare la chiusura imprevista del server dopo l'esecuzione di istruzioni DML (Data Manipulation Language) su una tabella le cui colonne non virtuali sono state riordinate con un'istruzione or. `MODIFY COLUMN` `CHANGE COLUMN`
- È stato risolto un problema che, durante il riavvio di un'istanza di database, può causare un ulteriore riavvio.
- È stato risolto un problema che poteva causare il riavvio di un'istanza Reader DB che utilizza l'inoltro di scrittura quando un'istruzione di commit implicita inoltrata rileva un errore.
- È stato risolto un problema che, in rare condizioni, può causare il riavvio di un'istanza del lettore quando si eseguono `SELECT` interrogazioni su tabelle con un vincolo di chiave esterna.

- È stato risolto un problema per cui le istanze DB che utilizzavano volumi cluster Aurora da più TB potevano subire un aumento dei tempi di inattività durante il riavvio a causa di errori di convalida del pool di buffer di InnoDB.
- È stato risolto un problema che può causare il riavvio di un database quando viene definito un vincolo a cascata `UPDATE` o di chiave `DELETE` esterna su una tabella in cui una colonna virtuale è coinvolta come colonna nel vincolo di chiave esterna o come membro della tabella di riferimento.
- È stato risolto un problema che poteva interrompere il ripristino del database durante l'avvio se il riavvio si verificava durante l'esecuzione di operazioni di inserimento complesse che coinvolgevano le colonne. `AUTO_INCREMENT`
- È stato risolto un problema Aurora Serverless v2 che poteva causare il riavvio del database durante il ridimensionamento.

Miglioramenti generali:

- I/O Utilizzo ridotto e prestazioni migliorate per un sottoinsieme di query di scansione con intervallo di chiavi primarie che utilizzano query parallele.
- Aurora MySQL versione 3.06.0 ha aggiunto il supporto per l'integrazione con Amazon Bedrock. Come parte di ciò, sono state aggiunte nuove parole chiave riservate (`accept`,, `aws_bedrock_invoke_modelaws_sagemaker_invoke_endpoint`, `content_type` e). `timeout_ms` Nella versione 3.07.0 di Aurora MySQL, queste parole chiave sono state modificate in parole chiave non riservate, consentite come identificatori senza virgolette. Per ulteriori informazioni su come MySQL gestisce le parole chiave riservate e non riservate, consulta Parole chiave e parole riservate nella documentazione di MySQL.
- È stato risolto un problema che non restituiva chiaramente un messaggio di errore al client quando richiamava il servizio Amazon Bedrock da un cluster Aurora MySQL DB in un ambiente in Regione AWS cui Amazon Bedrock non è ancora disponibile.
- È stato risolto un problema che poteva causare un consumo eccessivo di memoria durante l'interrogazione `BLOB` delle colonne utilizzando la query parallela Aurora.
- È stato aggiunto il supporto per `connection_memory_chunk_size` i parametri `connection_memory_limit` e da impostare a livello di sessione in modo che si comportino come in MySQL Community Edition. `connection_memory_limit`Viene utilizzato per impostare la quantità massima di memoria che può essere utilizzata da una singola connessione utente. Il `connection_memory_chunk_size` parametro può essere utilizzato per impostare la dimensione dei blocchi per gli aggiornamenti del contatore globale dell'utilizzo della memoria.

- È stato risolto un problema per cui l'utente non era in grado di interrompere alcuna query o impostare i timeout di sessione per le query. `performance_schema`
- È stato risolto un problema a causa del quale la replica del log binario (binlog) configurata per l'utilizzo di certificati SSL personalizzati (mysql.rds_import_binlog_ssl_material) poteva fallire quando l'istanza di replica era in fase di sostituzione dell'host.
- È `Aurora_fts_cache_memory_used` stata aggiunta la variabile di stato globale per tenere traccia dell'utilizzo della memoria per il sistema di ricerca full-text in tutte le tabelle. Per ulteriori informazioni, consulta le variabili di stato globali di Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- È stato risolto un problema per cui un cluster Amazon Redshift configurato come destinazione zero-ETL poteva subire un aumento temporaneo quando un cluster Amazon IntegrationLagAurora MySQL DB era configurato come replica di log binario, con l'integrazione Enhanced Binlog e Zero-ETL abilitata.
- È stato risolto un problema relativo alla gestione dei file di log di controllo che poteva rendere i file di registro inaccessibili per il download o la rotazione e, in alcuni casi, aumentare l'utilizzo della CPU.
- Ripristino delle `AUTO_INCREMENT` chiavi ottimizzato per ridurre i tempi di completamento per il ripristino delle istantanee, l'esecuzione del point-in-time ripristino e la clonazione di cluster DB con un numero elevato di tabelle nel database.
- È stato risolto un problema per cui l'evento wait/io/redo_log_flush non veniva visualizzato nelle tabelle di riepilogo degli eventi di attesa di Performance Schema.
- È stato risolto un problema che poteva causare errori chiave duplicati per le `AUTO_INCREMENT` colonne che utilizzavano indici decrescenti dopo un'operazione di ripristino di un'istantanea, un backtrack o un'operazione di clonazione del database.
- È stato risolto un problema che poteva causare il riavvio di un'istanza Writer DB quando un'istanza Reader DB che utilizza l'inoltro di scrittura esegue un'istruzione DML (Data Manipulation Language) che contiene un valore di timestamp e il parametro del database è impostato su. `time_zone` `UTC`
- È stato risolto un problema per cui una `SELECT` query su un'istanza di Aurora reader poteva fallire con la tabella degli errori che non esisteva quando la tabella ha almeno un indice di ricerca full-text (FTS) e viene eseguita un'`TRUNCATE`istruzione sull'istanza DB di Aurora writer.
- È stato risolto un problema che, in rari casi, causava il fallimento dello zero-downtime patching (ZDP).

- È stato risolto un problema che può causare un set di risultati incompleto durante l'esecuzione di query che coinvolgono `LEFT JOIN` o `RIGHT JOIN` utilizzano l'algoritmo hash join con query parallela.

Aggiornamenti e migrazioni:

- È stato risolto un problema che poteva causare errori di aggiornamento da Aurora MySQL versione 2 a Aurora MySQL versione 3 quando nello schema della tabella è presente una colonna definita dall'utente. `FTS_DOC_ID`
- È stato risolto un problema che poteva causare errori di aggiornamento da Aurora MySQL versione 2 a Aurora MySQL versione 3 a causa di un problema di sincronizzazione durante l'elaborazione dei tablespace InnoDB.
- È stato risolto un problema che poteva causare il fallimento degli aggiornamenti delle versioni principali di Aurora MySQL versione 3 a causa della presenza di voci orfane per i tablespace già eliminati nelle tabelle di sistema InnoDB in Aurora MySQL versione 2.
- È stato risolto un problema per cui il valore SERVER_ID non veniva aggiornato dopo uno switchover di distribuzione Amazon RDS. Blue/Green Ciò ha portato a problemi in cui i driver intelligenti come il driver JDBC di Amazon Web Services (AWS) non erano in grado di scoprire la topologia del cluster DB dopo uno switchover. blue/green Con questa correzione, i cluster Aurora DB rinominati come parte di una Blue/Green distribuzione RDS, in esecuzione su Aurora MySQL versione 3.07 e successive, avranno il valore aggiornato come parte dello switchover. `SERVER_ID` Per le versioni precedenti, le istanze DB nei cluster blu e verdi possono essere riavviate per aggiornare il valore. `SERVER_ID`

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla 8.0.36 inclusa, oltre a quanto segue. Per ulteriori informazioni, consulta MySQL bugs fixed by Aurora MySQL 3.x database engine updates.

- È stato risolto un problema per cui il valore della riga della cache poteva essere calcolato in modo errato, causando un errore durante il riavvio del database su istanze basate su Graviton. (Correzione di bug della community #35479763)
- È stato risolto un problema per cui alcune istanze di sottoquery all'interno delle routine archiviate non venivano gestite correttamente. (Correzione di un bug della community #35377192)

- È stato risolto un problema che poteva causare un maggiore utilizzo della CPU a causa della rotazione in background del certificato TLS (Community Bug Fix #34284186).
- È stato risolto un problema in cui InnoDB consentiva l'aggiunta di `INSTANT` colonne alle tabelle nello schema del sistema MySQL nelle versioni di Aurora MySQL precedenti alla 3.05, il che poteva portare alla chiusura inaspettata del server (riavvio dell'istanza del database) dopo l'aggiornamento alla versione 3.05.0 di Aurora MySQL. (Correzione di bug della community #35625510).

Aggiornamenti del motore di database Aurora MySQL 2024-06-26 (versione 3.06.1, compatibile con MySQL 8.0.34)

Versione: 3.06.1

Aurora MySQL 3.06.1 è disponibile a livello generale. Le versioni di Aurora MySQL 3.06 sono compatibili con MySQL 8.0.34. Per ulteriori informazioni sulle modifiche apportate alla community consulta MySQL 8.0 Release Notes.

Per informazioni sulle nuove funzionalità di Aurora MySQL versione 3, consulta Aurora MySQL versione 3 compatibile con MySQL 8.0. Per le differenze tra Aurora MySQL versione 3 e Aurora MySQL versione 2, vedi Confronto tra Aurora MySQL versione 2 e Aurora MySQL versione 3. Per un confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition, consulta Confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition nella Amazon Aurora User Guide.

Le versioni di Aurora MySQL attualmente supportate sono 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.03.*, 3.04.*, 3.05.*, 3.06.* e 3.07.*.

Puoi eseguire un aggiornamento sul posto, ripristinare uno snapshot o avviare un aggiornamento gestito blue/green utilizzando Amazon RDS Blue/Green Deployments da qualsiasi cluster Aurora MySQL versione 2 attualmente supportato in un cluster Aurora MySQL versione 3.06.1.

Per informazioni sulla pianificazione di un aggiornamento ad Aurora MySQL versione 3, vedere Pianificazione di un aggiornamento della versione principale per un cluster Aurora MySQL DB. Per informazioni generali sugli aggiornamenti di Aurora MySQL, consulta Aggiornamento dei cluster database Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Per informazioni sulla risoluzione dei problemi, consulta Risoluzione dei problemi per l'aggiornamento in loco di Aurora MySQL nella Guida per l'utente di Amazon Aurora.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Risolti i problemi di sicurezza e CVEs:

Questa versione include tutte le correzioni CVE della community fino a MySQL 8.0.34 incluso. Sono incluse le seguenti correzioni CVE:

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

Miglioramenti della disponibilità:

- È stato risolto un problema che causava il riavvio di un'istanza DB Aurora MySQL durante l'esecuzione di una query parallela.
- È stato risolto un problema che poteva causare il riavvio di un'istanza Reader DB durante la lettura di una tabella che veniva alterata o eliminata sull'istanza DB di Writer.
- È stato risolto un problema che causava una violazione dell'accesso alla memoria che portava al rilascio di un oggetto mutex non più di proprietà del thread.
- È stato risolto un problema che poteva causare il riavvio di un'istanza DB di Aurora MySQL Writer quando una sessione di inoltro della scrittura veniva chiusa durante l'esecuzione di una query inoltrata.
- È stato risolto un problema che causava il riavvio di un'istanza DB durante la gestione di set GTID di grandi dimensioni su un'istanza abilitata per i log binari.
- È stato risolto un problema che, in rare condizioni, può causare il riavvio di un'istanza di Reader quando si eseguono `SELECT` interrogazioni su tabelle con un vincolo di chiave esterna.
- È stato risolto un problema che causava il riavvio di un'istanza DB durante il tentativo di ripristinare il dizionario dei dati InnoDB durante il ripristino del database.
- È stato risolto un problema Aurora Serverless v2 che poteva causare il riavvio del database durante il ridimensionamento.

Miglioramenti generali:

- È stato risolto un problema nel codice di pubblicazione delle metriche a causa del quale la memoria poteva essere utilizzata dopo essere stata liberata.
- È stato risolto un problema che causava ripetuti riavvii del motore DB a causa di un oggetto undo tablespace inesistente.
- È stato risolto un problema relativo al troncamento automatico dei tablespace di annullamento quando sono più grandi della soglia innodb_max_undo_log_size negli scenari di aggiornamento.
- È stato risolto un problema che forniva un valore errato per la variabile `threads_running` status quando si utilizzava Aurora Global Database.
- È stato risolto un problema a causa del quale una replica di lettura del log binario Aurora MySQL (binlog) con ottimizzazione parallela dell'indice secondario abilitata presentava un riavvio quando si applicavano le modifiche di replica su tabelle con chiavi esterne.
- Aurora MySQL versione 3.06.0 ha aggiunto il supporto per l'integrazione con Amazon Bedrock. Come parte di ciò, sono state aggiunte nuove parole chiave riservate (`accept`,, `aws_bedrock_invoke_modelaws_sagemaker_invoke_endpoint`, `content_type` e). `timeout_ms` Nella versione 3.06.1 di Aurora MySQL, queste parole chiave sono state modificate in parole chiave non riservate, consentite come identificatori senza virgolette. Per ulteriori informazioni su come MySQL gestisce le parole chiave riservate e non riservate, consulta Parole chiave e parole riservate nella documentazione di MySQL.
- È stato risolto un problema che non restituiva chiaramente un messaggio di errore al client quando richiamava il servizio Amazon Bedrock da un cluster Aurora MySQL DB in un ambiente in Regione AWS cui Amazon Bedrock non è ancora disponibile.
- È stato risolto un problema che causava il riavvio di un'istanza DB a causa di informazioni imprecise sul lock-holder `rw_lock` quando si utilizzavano letture parallele.
- È stato risolto un problema che poteva causare il riavvio di un'istanza DB quando `SHOW VOLUME STATUS` viene eseguita.
- È stato risolto un problema di gestione della memoria che comportava una diminuzione della memoria liberabile nel tempo durante l'esecuzione delle `SELECT ... INTO OUTFILE ...` query.
- È stato aggiunto il supporto per `connection_memory_chunk_size` i parametri `connection_memory_limit` e da impostare a livello di sessione in modo che si comportino in modo simile alle funzionalità corrispondenti in MySQL Community Edition. Il `connection_memory_limit` parametro imposta la quantità massima di memoria che può essere utilizzata da una singola connessione utente. Il `connection_memory_chunk_size` parametro

imposta la dimensione dei blocchi per gli aggiornamenti al contatore globale dell'utilizzo della memoria.

- È stato risolto un problema che poteva causare il riavvio di un'istanza DB quando la memoria locale sull'istanza DB raggiunge la piena capacità.
- È stato risolto un problema per cui lo schema delle prestazioni non era abilitato quando la gestione automatizzata di Performance Insights era attivata per le istanze DB db.t4g.medium e db.t4g.large.
- È stato risolto un problema che poteva causare il riavvio di un'istanza Writer DB quando un'istanza DB reader che utilizza l'inoltro di scrittura esegue un'istruzione DML (Data Manipulation Language) che contiene un valore di timestamp e il parametro del database è impostato su. `time_zone` `UTC`
- È stato risolto un problema durante il patching ZDP (Zero-Downtime Patching) che impediva a un'istanza DB di chiudere le connessioni client al raggiungimento del valore minimo configurato dal cliente di uno dei due. `wait_timeout` `interactive_timeout`

Aggiornamenti e migrazioni:

- È stato risolto un problema che causava il fallimento degli aggiornamenti o delle migrazioni quando la versione del motore Aurora MySQL DB di destinazione è 3.04.0 o successiva. Ciò si verifica quando il parametro del cluster `lower_case_table_names` DB è impostato su e le regole di `1` confronto del database MySQL sono incompatibili con i nomi di tabella in minuscolo.

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla 8.0.34 inclusa. Per ulteriori informazioni, consulta MySQL bugs fixed by Aurora MySQL 3.x database engine updates.

Aggiornamenti del motore di database Aurora MySQL 2024-03-07 (versione 3.06.0, compatibile con MySQL 8.0.34)

Versione: 3.06.0

Aurora MySQL 3.06.0 è disponibile a livello generale. Le versioni di Aurora MySQL 3.06 sono compatibili con MySQL 8.0.34. Per ulteriori informazioni sulle modifiche apportate alla community consulta MySQL 8.0 Release Notes.

Per informazioni sulle nuove funzionalità di Aurora MySQL versione 3, consulta Aurora MySQL versione 3 compatibile con MySQL 8.0. Per le differenze tra Aurora MySQL versione 3 e Aurora

MySQL versione 2, vedi Confronto tra Aurora MySQL versione 2 e Aurora MySQL versione 3. Per un confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition, consulta Confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition nella Amazon Aurora User Guide.

Le versioni di Aurora MySQL attualmente supportate sono 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.03.*, 3.04.*, 3.05.* e 3.06.*.

Puoi eseguire un aggiornamento sul posto, ripristinare uno snapshot o avviare un aggiornamento gestito blue/green utilizzando Amazon RDS Blue/Green Deployments da qualsiasi cluster Aurora MySQL versione 2 attualmente supportato in un cluster Aurora MySQL versione 3.06.0.

Per informazioni sulla pianificazione di un aggiornamento ad Aurora MySQL versione 3, vedere Pianificazione di un aggiornamento della versione principale per un cluster Aurora MySQL DB. Per informazioni generali sugli aggiornamenti di Aurora MySQL, consulta Aggiornamento dei cluster database Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Per informazioni sulla risoluzione dei problemi, consulta Risoluzione dei problemi per l'aggiornamento in loco di Aurora MySQL nella Guida per l'utente di Amazon Aurora.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Nuove funzionalità

- La versione 3.06.0 di Aurora MySQL supporta l'integrazione con Amazon Bedrock e introduce le nuove parole chiave riservate,, , e. `accept` `aws_bedrock_invoke_model` `aws_sagemaker_invoke_endpoint` `content_type` `timeout_ms` Controlla le definizioni degli oggetti per l'utilizzo delle nuove parole chiave riservate prima di eseguire l'aggiornamento alla versione 3.06.0. Per mitigare il conflitto con le nuove parole chiave riservate, citate le parole chiave riservate utilizzate nelle definizioni degli oggetti. Per ulteriori informazioni sull'integrazione con Amazon Bedrock e sulla gestione delle parole chiave riservate, consulta Cos'è Amazon Bedrock? nella Guida per l'utente di Amazon Aurora. Per ulteriori informazioni, vedere Parole chiave e parole riservate, Tabella INFORMATION_SCHEMA KEYWORDS e Schema Object Names nella documentazione MySQL.
- Prestazioni migliorate per le repliche di log binari durante la replica di transazioni per tabelle di grandi dimensioni con più di un indice secondario. Questa funzionalità introduce un pool di thread per applicare le modifiche all'indice secondario in

parallelo su una replica binlog. La funzionalità è controllata dal parametro del cluster `aurora_binlog_replication_sec_index_parallel_workers` DB, che controlla il numero totale di thread paralleli disponibili per applicare le modifiche all'indice secondario. Per ulteriori informazioni, consulta Ottimizzazione della replica dei log binari nella Guida per l'utente di Amazon Aurora.

- È stata aggiunta una nuova procedura memorizzata `mysql.rds_set_read_only` che consente di modificare il valore della variabile di sistema globale `read_only` sulle istanze di database nel cluster Aurora MySQL. Per ulteriori informazioni, consulta Replicating in the Amazon Aurora User Guide.
- È stata aggiunta una nuova procedura memorizzata `mysql.rds_set_binlog_source_ssl` che consente di impostare la crittografia su una replica di log binario specificando un valore per. `SOURCE_SSL` Per ulteriori informazioni, consulta Replicating in the Amazon Aurora User Guide.
- Amazon Aurora Machine Learning è un'integrazione ottimizzata tra il database Aurora MySQL e i servizi di machine learning (ML). AWS Amazon Bedrock è ora supportato e consente di richiamare modelli di machine learning in Amazon Bedrock direttamente dal cluster Aurora MySQL DB tramite SQL. Per ulteriori informazioni sull'uso di Amazon Bedrock con il cluster Aurora MySQL DB, consulta Using Amazon Aurora machine learning with Aurora MySQL nella Amazon Aurora User Guide.
- La versione 3.06 di Aurora MySQL aggiunge il supporto per il troncamento automatico dei tablespace undo. Questa ottimizzazione consente di recuperare lo spazio inutilizzato nelle tablespace di annullamento dopo che i log di annullamento sono stati eliminati.

Miglioramenti

CVEsRisolti i problemi di sicurezza e:

Le seguenti correzioni CVE sono incluse in questa versione:

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

Miglioramenti della disponibilità:

- È stato risolto un problema a causa del quale un'istanza DB di replica in lettura non può essere avviata correttamente in presenza di un carico di lavoro elevato nell'istanza DB di Writer.
- È stato risolto un problema a causa del quale un'istanza DB di Aurora MySQL Writer poteva fallire a causa di un difetto nella comunicazione con lo storage Aurora. Il difetto si verifica a seguito di un'interruzione della comunicazione tra l'istanza DB e lo storage sottostante a seguito di un aggiornamento software dell'istanza di archiviazione Aurora.
- È stato risolto un problema durante l'elaborazione di `INSERT` query su tabelle partizionate InnoDB che poteva causare un calo graduale della memoria libera nell'istanza.
- È stato risolto un problema che poteva causare il riavvio o il failover di un'istanza DB Aurora MySQL a causa di una diminuzione della memoria liberabile quando si utilizzava l'hash join durante l'esecuzione delle query.
- È stato risolto un problema che poteva causare il riavvio di un'istanza di database quando si eseguivano contemporaneamente le istruzioni SHOW STATUS e PURGE BINARY LOGS. `PURGE BINARY LOGS` è un'istruzione gestita che viene eseguita per rispettare il periodo di conservazione dei binlog configurato dall'utente.
- È stato risolto un problema che può causare la chiusura imprevista del server dopo l'esecuzione di istruzioni DML (Data Manipulation Language) su una tabella le cui colonne non virtuali sono state riordinate con un'istruzione or. `MODIFY COLUMN` `CHANGE COLUMN`
- È stato risolto un problema che, durante il riavvio di un'istanza di database, può causare un ulteriore riavvio.
- È stato risolto un problema che può causare il riavvio del database quando viene definito un vincolo a cascata `UPDATE` o di chiave `DELETE` esterna su una tabella in cui una colonna virtuale è coinvolta come colonna nel vincolo di chiave esterna o come membro della tabella di riferimento.
- In Aurora MySQL 2.10, abbiamo aggiunto il supporto per il riavvio di un cluster Aurora DB con disponibilità di lettura. Questa funzionalità consente alle istanze DB Reader di rimanere online durante il riavvio di un'istanza DB Writer. Questa funzionalità è ora supportata sui database globali secondari Regioni AWS di Aurora MySQL, assicurando che sia ancora possibile soddisfare le richieste di lettura durante il riavvio di un'istanza di writer sul cluster primario. In precedenza, al riavvio di un'istanza di writer, venivano riavviate anche tutte le istanze di lettura in un cluster secondario Aurora MySQL. Con questa versione, le istanze di lettura del cluster secondario continuano a fornire richieste di lettura durante il riavvio di un'istanza di writer, migliorando la disponibilità di lettura nel cluster. Per ulteriori informazioni, consulta Riavvio di un cluster Aurora con disponibilità in lettura.

- È stato risolto un problema che poteva interrompere il ripristino del database durante l'avvio se il riavvio si verificava durante l'esecuzione di operazioni di inserimento complesse che coinvolgevano le colonne. `AUTO_INCREMENT`

Miglioramenti generali:

- È stato risolto un problema che poteva causare il fallimento di una query parallela a causa di problemi transitori di rete durante la lettura dei dati dal volume del cluster Aurora.
- È stato risolto un problema per cui l'utente non era in grado di interrompere alcuna query o impostare i timeout di sessione per le query. `performance_schema`
- È stato risolto un problema a causa del quale la replica del log binario (binlog) configurata per l'utilizzo di certificati SSL personalizzati (mysql.rds_import_binlog_ssl_material) poteva fallire quando l'istanza di replica era in fase di sostituzione dell'host.
- Le istanze DB di piccole dimensioni con meno o uguale a 4 GiB di memoria ora chiudono le connessioni che consumano più memoria quando l'istanza DB è sotto pressione. È inoltre possibile ottimizzare il pool di buffer per ridurne le dimensioni. Per ulteriori informazioni, consulta i problemi relativi ad Amazon Aurora MySQL nella Guida per out-of-memory l'utente di Amazon Aurora.
- È stata modificata la risposta predefinita per`aurora_oom_response`, su tutte le classi di istanze DB che hanno più di 4 GiB di memoria, da vuota a. `print` Per ulteriori informazioni, consulta i problemi relativi ad Amazon Aurora MySQL nella Guida per out-of-memory l'utente di Amazon Aurora.
- È stato risolto un problema relativo alla gestione dei file di registro di controllo che poteva rendere i file di registro inaccessibili per il download o la rotazione e, in alcuni casi, aumentare l'utilizzo della CPU.
- Ripristino delle `AUTO_INCREMENT` chiavi ottimizzato per ridurre i tempi di completamento per il ripristino delle istantanee, l'esecuzione del point-in-time ripristino e la clonazione di cluster DB con un numero elevato di tabelle nel database.
- È stato risolto un problema per cui l'evento wait/io/redo_log_flush non veniva visualizzato nelle tabelle di riepilogo degli eventi di attesa di Performance Schema.
- Sono state aggiunte le `Aurora_lockmgr_buffer_pool_memory_used` metriche `Aurora_lockmgr_memory_used` and per tenere traccia dell'utilizzo della memoria del gestore dei blocchi. Per ulteriori informazioni, consulta le variabili di stato globali di Aurora MySQL nella Guida per l'utente di Amazon Aurora.

- È stato risolto un problema a causa del quale le istanze di replica in lettura di piccole dimensioni possono subire un maggiore ritardo di replica dopo l'aggiornamento da versioni di Aurora MySQL precedenti alla 2.11.*.
- È stato risolto un problema che poteva causare errori chiave duplicati per le `AUTO_INCREMENT` colonne che utilizzavano indici decrescenti dopo un'operazione di ripristino di un'istantanea, un backtrack o un'operazione di clonazione del database.
- È stato risolto un problema per cui una `SELECT` query su un'istanza di Aurora reader poteva fallire con la tabella degli errori che non esisteva quando la tabella ha almeno un indice di ricerca full-text (FTS) e viene eseguita un'`TRUNCATE`istruzione sull'istanza DB di Aurora writer.
- È stato risolto un problema che può causare un set di risultati incompleto durante l'esecuzione di query che coinvolgono `LEFT JOIN` o `RIGHT JOIN` utilizzano l'algoritmo hash join con query parallela.

Aggiornamenti e migrazioni:

- È stato risolto un problema che poteva causare il fallimento degli aggiornamenti delle versioni principali se nello schema della tabella è presente una `FTS_DOC_ID` colonna definita dall'utente.
- È stato risolto un problema che poteva causare errori di aggiornamento da Aurora SQLversion My 2 a Aurora MySQL versione 3 a causa di un problema di sincronizzazione durante l'elaborazione dei tablespace InnoDB.
- È stato risolto un problema che poteva causare il fallimento degli aggiornamenti delle versioni principali di Aurora MySQL versione 3 a causa della presenza di voci orfane per i tablespace già eliminati nelle tabelle di sistema InnoDB in Aurora MySQL versione 2.

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla 8.0.34 inclusa, oltre a quanto segue. Per ulteriori informazioni, consulta MySQL bugs fixed by Aurora MySQL 3.x database engine updates.

- È stato risolto un problema per cui il valore della riga della cache poteva essere calcolato in modo errato, causando un errore durante il riavvio del database su istanze basate su Graviton. (Correzione di bug della community #35479763)
- È stato risolto un problema per cui alcune istanze di sottoquery all'interno delle routine archiviate non venivano sempre gestite correttamente. (Correzione di un bug della community #35377192)

- È stato risolto un problema che poteva causare un maggiore utilizzo della CPU a causa della rotazione in background del certificato TLS (Community Bug Fix #34284186).
- È stato risolto un problema in cui InnoDB consentiva l'aggiunta di `INSTANT` colonne alle tabelle nello schema del sistema MySQL nelle versioni di Aurora MySQL precedenti alla 3.05, il che poteva portare alla chiusura inaspettata del server (riavvio dell'istanza del database) dopo l'aggiornamento alla versione 3.05.0 di Aurora MySQL. (Correzione di bug della community #35625510).

Aggiornamenti del motore di database Aurora MySQL 2024-01-31 (versione 3.05.2, compatibile con MySQL 8.0.32)

Versione: 3.05.2

Aurora MySQL 3.05.2 è disponibile a livello generale. Le versioni 3.05 di Aurora MySQL sono compatibili con MySQL 8.0.32. Per ulteriori informazioni sulle modifiche apportate alla community consulta MySQL 8.0 Release Notes.

Per informazioni sulle nuove funzionalità di Aurora MySQL versione 3, consulta Aurora MySQL versione 3 compatibile con MySQL 8.0. Per le differenze tra Aurora MySQL versione 3 e Aurora MySQL versione 2, vedi Confronto tra Aurora MySQL versione 2 e Aurora MySQL versione 3. Per un confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition, consulta Confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition nella Amazon Aurora User Guide.

Le versioni Aurora MySQL attualmente supportate sono 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.03.*, 3.04.* e 3.05.*.

Puoi eseguire un aggiornamento sul posto, ripristinare uno snapshot o avviare un aggiornamento gestito blue/green utilizzando Amazon RDS Blue/Green Deployments da qualsiasi cluster Aurora MySQL versione 2 attualmente supportato in un cluster Aurora MySQL versione 3.05.2.

Per informazioni sulla pianificazione di un aggiornamento ad Aurora MySQL versione 3, consulta Pianificazione dell'aggiornamento per Aurora MySQL versione 3. Per informazioni generali sugli aggiornamenti di Aurora MySQL, consulta Aggiornamento dei cluster database Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Per informazioni sulla risoluzione dei problemi, consulta Risoluzione dei problemi di upgrade con Aurora MySQL versione 3 nella Amazon Aurora User Guide.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Risolti i problemi di sicurezza e CVEs:

Le seguenti correzioni CVE sono incluse in questa versione:

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

Miglioramenti della disponibilità:

- È stato risolto un problema per cui l'elaborazione di `INSERT` query su tabelle partizionate InnoDB può causare un calo graduale della memoria libera nell'istanza.
- È stato risolto un problema che può causare il riavvio di un'istanza di database quando si eseguono contemporaneamente le istruzioni SHOW STATUS e PURGE BINARY LOGS. `PURGE BINARY LOGS` è un'istruzione gestita che viene eseguita per rispettare il periodo di conservazione dei binlog configurato dall'utente.
- È stato risolto un problema che può causare la chiusura imprevista del server dopo l'esecuzione di istruzioni DML (Data Manipulation Language) su una tabella le cui colonne non virtuali sono state riordinate con un'istruzione or. `MODIFY COLUMN` `CHANGE COLUMN`
- È stato risolto un problema che, durante il riavvio di un'istanza di database, può causare un ulteriore riavvio.

Miglioramenti generali:

- È stato risolto un problema per cui l'utente non era in grado di interrompere alcuna query o impostare i timeout di sessione per le query. `performance_schema`
- È stato risolto un problema a causa del quale la configurazione della replica del registro binario (binlog) utilizzando certificati SSL personalizzati (mysql.rds_import_binlog_ssl_material) poteva fallire quando l'istanza di replica era in fase di sostituzione dell'host.

- È stato risolto un problema relativo alla gestione dei file di registro di controllo che poteva rendere i file di registro inaccessibili per il download o la rotazione e, in alcuni casi, aumentare l'utilizzo della CPU.
- Nelle versioni di Aurora MySQL precedenti alla 3.05.2, gli utenti non sono in grado di recuperare l'output delle istanze DB del lettore Aurora MySQL. `SHOW ENGINE INNODB STATUS` Ciò è dovuto al comportamento predefinito di InnoDB quando `innodb_read_only` è abilitato.

  In Aurora MySQL versione 3.05.2 e successive, quando `SHOW ENGINE INNODB STATUS` viene eseguito su un'istanza di lettura, l'output viene scritto nel log degli errori MySQL, semplificando la risoluzione dei problemi.

  Per ulteriori informazioni sull'utilizzo dei log degli errori MySQL, consulta Log degli errori di Aurora MySQL. Per ulteriori informazioni`SHOW ENGINE INNODB STATUS`, vedere l'istruzione SHOW ENGINE nella documentazione di MySQL.

Aggiornamenti e migrazioni:

- È stato risolto un problema che poteva causare errori di aggiornamento da Aurora MySQL versione 2 a Aurora MySQL versione 3 quando nello schema della tabella è presente una colonna definita dall'utente. `FTS_DOC_ID`
- È stato risolto un problema che poteva causare errori di aggiornamento da Aurora MySQL versione 2 a Aurora MySQL versione 3 a causa di un problema di sincronizzazione durante l'elaborazione dei tablespace InnoDB.
- È stato risolto un problema che poteva causare il fallimento degli aggiornamenti delle versioni principali di Aurora MySQL versione 3 a causa della presenza di voci orfane per i tablespace già eliminati nelle tabelle di sistema InnoDB in Aurora MySQL versione 2.

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla 8.0.32 inclusa, oltre a quanto segue. Per ulteriori informazioni, consulta MySQL bugs fixed by Aurora MySQL 3.x database engine updates.

- È stato risolto un problema a causa del quale veniva `records_in_range` eseguito un numero eccessivo di letture del disco per `INSERT` le operazioni, con conseguente graduale calo delle prestazioni. (Correzione di bug #34976138 della community)

Aggiornamenti del motore di database Aurora MySQL 2023-11-21 (versione 3.05.1, compatibile con MySQL 8.0.32)

Versione: 3.05.1

Aurora MySQL 3.05.1 è disponibile a livello generale. Le versioni 3.05 di Aurora MySQL sono compatibili con MySQL 8.0.32. Per ulteriori informazioni, consulta le Note di rilascio di MySQL 8.0.

Le versioni di Aurora MySQL attualmente supportate sono 2.07.*, 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.*, 3.04.* e 3.05.*.

È possibile aggiornare un cluster di database Aurora MySQL 3.* esistente ad Aurora MySQL 3.05.1. È inoltre possibile ripristinare un'istantanea da qualsiasi versione di Aurora MySQL attualmente supportata in Aurora MySQL 3.05.1.

Se aggiorni un database globale Aurora MySQL alla versione 3.05.*, devi aggiornare i tuoi cluster di database primari e secondari alla stessa identica versione, incluso il livello di patch. Per ulteriori informazioni sull'aggiornamento della versione secondaria di un database globale Aurora, consulta Aggiornamenti delle versioni secondarie.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Risolti i problemi di sicurezza ed CVEs elencati di seguito:

Questa versione include tutte le CVEs correzioni della community fino a MySQL 8.0.32 incluso.

- CVE-2023-38545

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla 8.0.32 inclusa, oltre a quanto riportato di seguito. Per ulteriori informazioni, consulta MySQL bugs fixed by Aurora MySQL 3.x database engine updates.

- È stato risolto un problema in InnoDB che, se a una tabella MySQL in uno schema di sistema veniva aggiunta una colonna `INSTANT ADD` tra Aurora MySQL versioni 3.01 e Aurora MySQL

versioni 3.04 e dopo l'aggiornamento di Aurora MySQL alla versione 3.05.0, su queste tabelle il server si chiudeva inaspettatamente. DMLs (Correzione di un bug della community 35625510)

Aggiornamenti del motore del database di Aurora MySQL 30/10/2023 (versione 3.05.0.1, compatibile con MySQL 8.0.32) beta

Versione: 3,05.0.1

Aurora MySQL 3.05.0.1 è generalmente disponibile nelle seguenti regioni: Stati Uniti orientali (Virginia settentrionale), Stati Uniti orientali (Ohio), Stati Uniti occidentali (California settentrionale), Stati Uniti occidentali (Oregon), (Stati Uniti orientali) e (Stati Uniti occidentali). AWS GovCloud AWS GovCloud Si tratta di una versione iniziale, che prevede solo correzioni di sicurezza. Queste correzioni verranno implementate in modo più ampio in tutte le regioni con la prossima versione della patch, 3.05.1. Le versioni 3.05 di Aurora MySQL sono compatibili con MySQL 8.0.32. Per ulteriori informazioni sulle modifiche apportate alla community consulta MySQL 8.0 Release Notes.

Le versioni di Aurora MySQL attualmente supportate sono 2.07.*, 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.*, 3.04.* e 3.05.*.

È possibile aggiornare un cluster di database di Aurora MySQL 3.* esistente ad Aurora MySQL 3.05.0.1. È anche possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 3.05.0.1.

Se aggiorni un database globale Aurora MySQL alla versione 3.05.*, devi aggiornare i tuoi cluster di database primari e secondari alla stessa identica versione, incluso il livello di patch. Per ulteriori informazioni sull'aggiornamento della versione secondaria di un database globale Aurora, consulta Aggiornamenti delle versioni secondarie.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Problemi di sicurezza risolti ed CVEs elencati di seguito:

Questa versione include tutte le CVEs correzioni della community fino a MySQL 8.0.32 incluso.

- CVE-2023-38545

Aggiornamenti del motore del database di Aurora MySQL 25/10/2023 (versione 3.05.0, compatibile con MySQL 8.0.32)

Versione: 3.05.0

Aurora MySQL 3.05.0 è disponibile a livello generale. Le versioni 3.05 di Aurora MySQL sono compatibili con MySQL 8.0.32. Per ulteriori informazioni sulle modifiche apportate alla community consulta MySQL 8.0 Release Notes.

Per informazioni sulle nuove funzionalità di Aurora MySQL versione 3, consulta Aurora MySQL versione 3 compatibile con MySQL 8.0. Per le differenze tra Aurora MySQL versione 3 e Aurora MySQL versione 2, vedi Confronto tra Aurora MySQL versione 2 e Aurora MySQL versione 3. Per un confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition, vedi Confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition.

Le versioni Aurora MySQL attualmente supportate sono 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.03.*, 3.04.* e 3.05.*.

Puoi eseguire un aggiornamento sul posto, ripristinare uno snapshot o avviare un aggiornamento gestito blue/green utilizzando Amazon RDS Blue/Green Deployments da qualsiasi cluster Aurora MySQL versione 2 attualmente supportato in un cluster Aurora MySQL versione 3.05.0.

Per informazioni sulla pianificazione di un aggiornamento ad Aurora MySQL versione 3, consulta Pianificazione dell'aggiornamento per Aurora MySQL versione 3 nella Guida per l'utente di Amazon Aurora. Per informazioni generali sugli aggiornamenti di Aurora MySQL, consulta Aggiornamento dei cluster database Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Per informazioni sulla risoluzione dei problemi, consulta Risoluzione dei problemi di aggiornamento con Aurora MySQL versione 3.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Nuove caratteristiche:

- È stato aggiunto il supporto per il salvataggio dei dati da un cluster di database Aurora MySQL nei file di testo in un bucket Amazon S3 crittografato con una chiave KMS (SSE-KMS). Per ulteriori

informazioni, consulta Salvataggio dei dati da un cluster DB Amazon Aurora MySQL nei file di testo in un bucket Amazon S3.

- È stata introdotta una nuova variabile di stato globale `aurora_tmz_version` per indicare la versione corrente delle informazioni sul fuso orario (TZ) utilizzate dal motore. I valori seguono la versione del database dei fusi orari IANA e sono formattati come "YYYYsuffix«, ad esempio 2022a e 2023c. Per ulteriori informazioni, consulta Variabili di stato globali di Aurora MySQL.

Risolti i problemi di sicurezza ed elencati di seguito: CVEs

Correzioni e altri miglioramenti per ottimizzare la gestione in un ambiente gestito. Ulteriori correzioni CVE sono riportate di seguito:

- CVE-2022-37434

Miglioramenti della disponibilità:

- È stato risolto un problema per cui le istanze database di Aurora MySQL che utilizzavano query parallele potevano subire un riavvio del database durante l'esecuzione di un numero elevato di query parallele simultanee.
- È stato risolto un problema di conflitto di blocchi causato da un thread di log di audit che alla fine portava a un utilizzo elevato della CPU e al timeout delle applicazioni client.
- È stato risolto un problema che poteva causare il ripristino errato del set GTID eseguito su un cluster di replica di log binario (binlog) con binlog avanzato abilitato quando qualsiasi origine binlog disponeva di `gtid_mode` impostata su `ON` o `ON_PERMISSIVE`. Questo problema può causare il riavvio dell'istanza di scrittura del cluster di replica un'ulteriore volta durante il ripristino o portare a risultati errati durante l'esecuzione di query sul set GTID eseguito.
- È stato risolto un problema di gestione della memoria che può causare il riavvio di un'istanza database Aurora MySQL o un failover a causa di una diminuzione della memoria liberabile quando è abilitato il log binario avanzato.
- È stato risolto un problema che può causare il riavvio di un'istanza database quando si tenta di leggere una pagina del database che appartiene a una tabella eliminata.
- È stato risolto un problema che può causare il riavvio dell'istanza di lettura quando l'istanza di scrittura aumenta il volume del database fino a un multiplo di 160 GB.
- È stato risolto un problema a causa del quale un'istanza database Aurora MySQL con la funzionalità di registro binario avanzata abilitata poteva rimanere bloccata durante l'avvio dell'istanza database durante l'esecuzione del processo di ripristino del registro binario.

- È stato risolto un problema per cui un'istanza database Aurora MySQL poteva subire più riavvii durante l'avvio dell'istanza mentre venivano inizializzati segmenti di rollback di grandi dimensioni.
- È stato risolto un problema durante l'applicazione di patch senza tempi di inattività che causava il riavvio dell'istanza con conseguente chiusura imprevista delle connessioni al database.
- È stato risolto un problema che poteva causare il riavvio di un'istanza database a causa di un deadlatch quando si eseguivano contemporaneamente le istruzioni SHOW STATUS e PURGE BINARY LOGS. L'eliminazione dei log binari è un'istruzione gestita che viene eseguita per rispettare il periodo di conservazione di binlog configurato dall'utente.
- È stato risolto un problema che poteva causare la non disponibilità del cluster di database al riavvio dell'istanza di scrittura mentre il database stava creando o eliminando i trigger sulle tabelle di sistema interne.
- È stato risolto un problema che può causare il riavvio di un'istanza database a causa di una lunga attesa del semaforo quando si utilizza la funzionalità binlog avanzata su un cluster con una replica Aurora.
- È stato risolto un problema che può causare il riavvio di un'istanza database durante l'esecuzione di una query che fa riferimento a una funzione di aggregazione.
- È stato risolto un problema che, in rare condizioni, può causare il riavvio dell'istanza database quando Aurora Serverless v2 tenta erroneamente di aggiornare la cache della tabella durante il dimensionamento.
- È stato risolto un problema per cui i metodi di accesso alla scansione degli indici non supportati venivano presi in considerazione per le espressioni di tabella comuni (CTE) durante la materializzazione di tabelle temporanee intermedie, il che poteva portare a comportamenti indesiderati, tra cui riavvii del database o risultati di query errati. Abbiamo risolto questo problema evitando l'uso di tali metodi di accesso alla scansione degli indici non supportati sulle tabelle che utilizzano il motore TempTable di archiviazione.

Miglioramenti generali:

- È stato risolto un problema che può causare l'indisponibilità del database quando il binlog avanzato è abilitato su un cluster di database Aurora Serverless v2 in esecuzione su Aurora MySQL 3.04.0.
- Sono stati rimossi i metadati di archiviazione non utilizzati prima di scrivere sull'archiviazione di Aurora quando è abilitata la funzionalità binlog avanzato. Ciò evita determinati scenari in cui potrebbe verificarsi un riavvio o un failover del database a causa della maggiore latenza di scrittura dovuta all'aumento dei byte trasmessi sulla rete.

- Con l'aggiunta delle tabelle `malloc_stats` e `malloc_stats_totals` in `performance_schema`, sono state aggiunte tre variabili di sistema avanzate per controllare il comportamento di Jemalloc, un allocatore di memoria interno:
  - `aurora_jemalloc_background_thread`.
  - `aurora_jemalloc_dirty_decay_ms`.
  - `aurora_jemalloc_tcache_enabled`.
- È stato risolto un problema per cui le tabelle degli schemi delle prestazioni specifiche di Aurora non venivano create durante un aggiornamento o una migrazione.
- È stata aggiunta una nuova variabile di sistema, `aurora_use_vector_instructions`. Quando questo parametro è abilitato, Aurora MySQL utilizza istruzioni di elaborazione vettoriale ottimizzate per migliorare le prestazioni su carichi di lavoro pesanti. I/O Questa impostazione è `ON` per impostazione predefinita in Aurora MySQL 3.05 e versioni successive. Per ulteriori dettagli, consulta Parametri di configurazione Aurora MySQL.
- È stato risolto un problema che poteva causare la visualizzazione di risultati errati nelle `NumBinaryLogFiles` metriche attivate quando il CloudWatch binlog avanzato era abilitato.
- Il timeout delle richieste per le operazioni di machine learning di Aurora MySQL su Amazon Sagemaker è stato aumentato da 3 a 30 secondi. Questo aiuta a risolvere un problema per cui i clienti possono riscontrare un numero maggiore di tentativi o errori per le richieste ad Amazon Sagemaker dal machine learning di Aurora MySQL quando utilizzano batch di dimensioni maggiori.
- È stato aggiunto il supporto per le tabelle `malloc_stats` e `malloc_stats_totals` nel database performance_schema.
- La parola chiave `FROM` nel comando `LOAD DATA FROM S3` è stata aggiornata in modo che sia facoltativa. Per ulteriori informazioni, consulta Caricamento dei dati in un cluster DB Amazon Aurora MySQL da file di testo in un bucket Amazon S3.
- È stato aggiunto il supporto per il parametro `innodb_aurora_instant_alter_column_allowed`, che controlla se è possibile utilizzare l'algoritmo `INSTANT` per le operazioni `ALTER COLUMN`. Per ulteriori informazioni, consulta Parametri a livello di cluster.
- È stato risolto un problema che può impedire la creazione di nuove connessioni client al database quando è abilitato l'inoltro di scrittura.
- È stato risolto un problema che poteva far sì che la modifica del parametro del database `table_open_cache` non avesse effetto fino al riavvio dell'istanza database.

- È stato risolto un problema che poteva causare errori di chiave duplicata per le colonne `AUTO_INCREMENT` che utilizzavano indici decrescenti dopo un'operazione di ripristino di uno snapshot, un backtrack o un'operazione di clonazione del database.
- È stato risolto un problema relativo alle scansioni degli indici in cui poteva essere restituito un risultato impreciso durante l'esecuzione di una query `SELECT` con la clausola `GROUP BY` e il parametro `aurora_parallel_query` impostati su `ON`.
- È stato risolto un problema che poteva causare l'esaurimento della memoria disponibile durante l'esecuzione di query sulla tabella `INFORMATION_SCHEMA INNODB_TABLESPACES`.
- È stato risolto un problema per cui l'istanza di lettura non era in grado di aprire una tabella, con ERROR 1146. Questo problema si verifica quando si eseguono determinati tipi di Data Definition Language (DDL) online mentre l'algoritmo `INPLACE`viene utilizzato sull'istanza di scrittura.
- È stato risolto un problema per impedire il riavvio dell'istanza durante il dimensionamento di Aurora Serverless v2 quando il processo di monitoraggio interno inviava erroneamente richieste di dimensionamento duplicate.
- È stato risolto un problema che poteva causare il riavvio del database quando gli utenti di log binario (binlog) connessi utilizzano un server di replica binlog duplicato. IDs
- È stata introdotta una cache di log di inoltro in memoria per le repliche di log binari gestite da Aurora MySQL. Questo miglioramento può contribuire a ottenere un aumento fino al 40% della velocità di trasmissione effettiva di replica dei log binari. Questo miglioramento viene abilitato automaticamente quando si utilizza la replica di log binario a thread singolo o quando si utilizza la replica multi-thread con il posizionamento automatico GTID abilitato.

Aggiornamenti e migrazioni:

- L'aggiornamento da MySQL 5.7 a MySQL 8.0 con un numero molto elevato di tabelle in un unico database ha causato un consumo eccessivo di memoria nel server. È stato riscontrato che, durante il processo di verifica della possibilità di aggiornare le tabelle, abbiamo recuperato in anticipo tutti gli oggetti `Table` del dizionario di dati, elaborandoli ciascuno e recuperandone il nome, quindi abbiamo eseguito la verifica della compatibilità delle versioni nell'elenco. In questo caso non era necessario recuperare tutti gli oggetti in anticipo e l'operazione ha contribuito notevolmente al consumo di memoria. Per correggere questo problema, in questi casi recuperiamo un oggetto `Table` alla volta, eseguiamo tutti i controlli necessari, ne recuperiamo il nome e rilasciamo l'oggetto, prima di procedere con quello successivo. (Bug 34526001)

- Sono state migliorate le prestazioni degli aggiornamenti delle versioni principali da Aurora MySQL versione 2 alla versione 3 eseguendo controlli del tablespace in parallelo utilizzando tutte le v disponibili sull'istanza del database. CPUs

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla 8.0.32 inclusa, oltre a quanto riportato di seguito. Per ulteriori informazioni, consulta MySQL bugs fixed by Aurora MySQL 3.x database engine updates.

- È stato risolto un problema che può causare un maggiore utilizzo della CPU a causa della rotazione dei certificati TLS in background. (Correzione di un bug della community 34284186)

Aggiornamenti del motore di database Aurora MySQL 2025-05-05 (versione 3.04.4, compatibile con MySQL 8.0.28)

Versione: 3.04.4

Aurora MySQL 3.04.4 è disponibile a livello generale. Le versioni di Aurora MySQL 3.04 sono compatibili con MySQL 8.0.28. Per ulteriori informazioni sulle modifiche apportate alla community consulta MySQL 8.0 Release Notes.

Per informazioni sulle nuove funzionalità di Aurora MySQL versione 3, consulta Aurora MySQL versione 3 compatibile con MySQL 8.0. Per le differenze tra Aurora MySQL versione 3 e Aurora MySQL versione 2, consulta Confronto tra Aurora MySQL versione 2 e Aurora MySQL versione 3. Per un confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition, consulta Confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition nella Amazon Aurora User Guide.

> **Note**
>
> Questa versione è designata come versione di supporto a lungo termine (LTS). Per ulteriori informazioni, consulta Versioni con supporto a lungo termine (Long-Term Support, LTS) di Aurora MySQL nella Guida per l'utente di Amazon Aurora.
> Ti consigliamo di non impostare il **`AutoMinorVersionUpgrade`** parametro su (o abilitare l'aggiornamento automatico della versione secondaria in**`true`**) per le versioni LTS. Console di gestione AWS Ciò potrebbe comportare l'aggiornamento del cluster DB alla versione di

destinazione successiva per la campagna Automatic Minor Version Upgrade, che potrebbe non essere una versione LTS.

Puoi eseguire un aggiornamento sul posto che sfrutta zero-downtime-patch, ripristinare uno snapshot o avviare un aggiornamento gestito blue/green utilizzando Amazon RDS Blue/Green Deployments da qualsiasi cluster Aurora MySQL versione 2 attualmente supportato in un cluster Aurora MySQL versione 3.04.4.

Per informazioni sulla pianificazione di un aggiornamento a Aurora MySQL versione 3, vedere Pianificazione di un aggiornamento della versione principale per un cluster Aurora MySQL. Per informazioni generali sugli aggiornamenti di Aurora MySQL, consulta Aggiornamento dei cluster database Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Per informazioni sulla risoluzione dei problemi, consulta Troubleshooting for Aurora MySQL in-place upgrade nella Amazon Aurora User Guide.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Correzioni di sicurezza

Critico CVEs:

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

Miglioramenti della disponibilità

- È stato risolto un problema sulla replica a causa del quale un'interruzione di rete poteva non ristabilire correttamente la connessione con il writer.
- È stato risolto un problema che poteva causare il riavvio di una replica di log binario (binlog) durante l'elaborazione di un numero elevato di file di log di inoltro durante il ripristino del relay log.
- È stato risolto un problema che causava il riavvio di un'istanza del lettore di database durante l'esecuzione di una query utilizzando la funzionalità Parallel Query.

- È stato risolto un problema che, in rare condizioni, poteva disabilitare la registrazione binaria quando si verificava un errore durante il commit di una transazione di grandi dimensioni.
- È stato risolto un problema che poteva causare il riavvio delle repliche di lettura di Aurora in caso di alcuni rari ordini di commit delle transazioni sull'istanza Writer DB.
- È stato risolto un problema che poteva causare il riavvio del database quando gli eventi pianificati venivano interrotti durante l'esecuzione su istanze con Enhanced Binlog abilitato.
- È stato risolto un problema per cui le istanze di database che utilizzavano volumi cluster Aurora da più TB potevano subire un aumento dei tempi di inattività durante il riavvio a causa di errori di convalida del bufferpool di InnoDB.

Miglioramenti generali

- I seguenti privilegi sono stati aggiunti `ards_superuser_role`:`FLUSH_OPTIMIZER_COSTS`, `FLUSH_STATUSFLUSH_TABLES`, e`FLUSH_USER_RESOURCES`. Per informazioni `surds_superuser_role`, consulta Amazon Master User Accounts with Amazon Aurora nella Amazon Aurora User Guide. Per ulteriori informazioni su questi privilegi dinamici, consulta la documentazione MySQL.
- È stato risolto un problema a causa del quale una connessione preservata veniva gestita in modo errato durante Zero-downtime patching (ZDP) /Zero-downtime restart (ZDR), che poteva portare il client ad attendere indefinitamente il completamento di una query.
- È stato risolto un problema per cui la riga diventava illeggibile tramite l'indice spaziale durante un aggiornamento.
- È stato risolto un problema a causa del quale una query contenente un suggerimento sull'ottimizzatore che veniva interrotta durante un riavvio senza tempi di inattività o un'operazione di patching senza tempi di inattività poteva essere gestita in modo errato.
- È stato risolto un problema per cui la latenza di commit non veniva misurata quando era impostata su. `innodb_flush_log_at_trx_commit` `0`
- Sono state introdotte ottimizzazioni per ridurre l'utilizzo della memoria durante il ripristino del dizionario logico dei dati in presenza di un numero elevato di tabelle.
- È stato risolto un problema a causa del quale l'esecuzione del `SHOW BINARY LOGS` comando richiedeva più tempo in un cluster in cui Enhanced Binlog era abilitato o era abilitato in precedenza. Questo problema potrebbe anche causare un aumento della latenza di commit se più `SHOW BINARY LOGS` comandi venivano eseguiti contemporaneamente.
- È stato risolto un problema di memoria associato ai ruoli predefiniti del view definer.

- È stato risolto un problema che poteva causare un errore nel completamento del processo di disabilitazione della funzionalità di inoltro della scrittura.
- È stato risolto un problema che può impedire la creazione di nuove connessioni client al database quando l'inoltro di scrittura è abilitato.
- È stato risolto un problema che può causare il riavvio di un'istanza del database writer quando un'istanza di lettura che utilizza l'inoltro di scrittura esegue un'istruzione DML che contiene un `timestamp` valore e il parametro del `time_zone` database è impostato su «UTC».
- È stato risolto un problema che causava l'indisponibilità intermittente di una replica di lettura Aurora o le incongruenze nella definizione della tabella con errori `'Table does not exist'` sulla replica dovuti a query di lettura simultanee sulla replica e operazioni DDL sul writer.
- È stato risolto un problema che poteva causare un set di risultati incompleto durante l'esecuzione di query che coinvolgono o utilizzano l'algoritmo hash-join con Parallel Query. `LEFT- RIGHT-JOIN`
- A partire da questa versione di Aurora MySQL, l'ottimizzazione rapida degli inserti non è più abilitata. Per informazioni Aurora, consulta Miglioramenti alle prestazioni di Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Aggiornamenti e migrazioni

- Sono stati rimossi i ruoli predefiniti creati inutilmente durante l'aggiornamento da a. AMS2 AMS3

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla 8.0.28 inclusa. Per ulteriori informazioni, consulta Correzione dei bug di MySQL attraverso gli aggiornamenti del motore di database Aurora MySQL 3.x.

- È stato risolto un problema che risolveva il deadlock quando`FLUSH STATUS`, `COM_CHANGE_USER` e veniva eseguito contemporaneamente. `SHOW PROCESS LIST` (Bug #35218030)

Aggiornamenti del motore di database Aurora MySQL 2024-06-26 (versione 3.04.3, compatibile con MySQL 8.0.28)

Versione: 3.04.3

Aurora MySQL 3.04.3 è disponibile a livello generale. Le versioni di Aurora MySQL 3.04 sono compatibili con MySQL 8.0.28. Per ulteriori informazioni sulle modifiche apportate alla community, consulta le Note di rilascio di MySQL 8.0.

Per informazioni sulle nuove funzionalità di Aurora MySQL versione 3, consulta Aurora MySQL versione 3 compatibile con MySQL 8.0. Per le differenze tra Aurora MySQL versione 3 e Aurora MySQL versione 2, vedi Confronto tra Aurora MySQL versione 2 e Aurora MySQL versione 3. Per un confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition, vedi Confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition.

> **Note**
>
> Questa versione è designata come versione di supporto a lungo termine (LTS). Per ulteriori informazioni, consulta Versioni con supporto a lungo termine (Long-Term Support, LTS) di Aurora MySQL nella Guida per l'utente di Amazon Aurora.
> Si consiglia di non impostare il **`AutoMinorVersionUpgrade`** parametro su **`true`** (o abilitare l'aggiornamento automatico della versione secondaria in) per le versioni LTS. Console di gestione AWS Ciò potrebbe comportare l'aggiornamento del cluster DB alla versione di destinazione successiva per la campagna Automatic Minor Version Upgrade, che potrebbe non essere una versione LTS.

Le versioni di Aurora MySQL attualmente supportate sono 2.07.9, 2.7.10, 2.11.*, 2.12.*, 3.03.*, 3.04.*, 3.05.*, 3.06.* e 3.07.*.

Puoi eseguire un aggiornamento sul posto, ripristinare uno snapshot o avviare un aggiornamento gestito blue/green utilizzando Amazon RDS Blue/Green Deployments da qualsiasi cluster Aurora MySQL versione 2 attualmente disponibile in un cluster Aurora MySQL versione 3.04.3.

Per informazioni sulla pianificazione di un aggiornamento ad Aurora MySQL versione 3, vedere Pianificazione di un aggiornamento della versione principale per un cluster Aurora MySQL DB. Per informazioni generali sugli aggiornamenti di Aurora MySQL, consulta Aggiornamento dei cluster database Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Per informazioni sulla risoluzione dei problemi, consulta Troubleshooting for Aurora MySQL in-place upgrade nella Amazon Aurora User Guide.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Risolti i problemi di sicurezza e CVEs:

Questa versione include tutte le correzioni CVE della community fino a MySQL 8.0.28 incluso. Sono incluse le seguenti correzioni CVE:

- CVE-2024-0853

Miglioramenti della disponibilità:

- È stato risolto un problema che causava il riavvio di un'istanza DB Aurora MySQL durante l'esecuzione di una query parallela.
- È stato risolto un problema che poteva causare il riavvio di un'istanza Reader DB durante la lettura di una tabella che veniva alterata o eliminata sull'istanza DB di Writer.
- È stato risolto un problema che causava una violazione dell'accesso alla memoria che portava al rilascio di un oggetto mutex non più di proprietà del thread.
- È stato risolto un problema che poteva causare il riavvio di un'istanza DB di Aurora MySQL Writer quando una sessione di inoltro della scrittura veniva chiusa durante l'esecuzione di una query inoltrata.
- È stato risolto un problema che causava il riavvio di un'istanza DB durante la gestione di set GTID di grandi dimensioni su un'istanza abilitata per i log binari.
- È stato risolto un problema durante l'elaborazione di `INSERT` query su tabelle partizionate InnoDB che può causare un calo graduale della memoria libera nell'istanza DB.
- È stato risolto un problema che, in rare condizioni, può causare il riavvio di un'istanza Reader durante l'esecuzione di `SELECT` query su tabelle con un vincolo di chiave esterna.
- È stato risolto un problema che può causare il riavvio di un database quando il ripristino del dizionario dati InnoDB impiega molto tempo durante il ripristino del database.
- È stato risolto un problema che può causare il riavvio di un database quando viene definito un vincolo a cascata `UPDATE` o di chiave `DELETE` esterna su una tabella in cui una colonna virtuale è coinvolta come colonna nel vincolo di chiave esterna o come membro della tabella di riferimento.

- È stato risolto un problema Aurora Serverless v2 che poteva causare il riavvio del database durante il ridimensionamento.

Miglioramenti generali:

- È stato risolto un problema che forniva un valore errato per la variabile `threads_running` status quando si utilizzava Aurora Global Database.
- È stato risolto un problema che causava il riavvio di un'istanza DB a causa di informazioni imprecise sul lock-holder `rw_lock` quando si utilizzavano letture parallele.
- È stato risolto un problema di gestione della memoria che comportava una diminuzione della memoria liberabile nel tempo durante l'esecuzione delle query. `SELECT ... INTO OUTFILE ...`
- È stato risolto un problema che poteva causare il riavvio di un'istanza DB quando la memoria locale sull'istanza DB raggiunge la piena capacità.
- È stato risolto un problema per cui lo schema delle prestazioni non era abilitato quando la gestione automatizzata di Performance Insights era attivata per le istanze DB db.t4g.medium e db.t4g.large.
- È stato risolto un problema durante l'applicazione di patch ZDP (zero-downtime patching) che impediva a un'istanza DB di chiudere le connessioni client quando raggiungeva l'istanza configurata dal cliente di una delle due. `wait_timeout` `interactive_timeout`
- È stato risolto un problema per cui una `SELECT` query su un'istanza del lettore Aurora poteva fallire con la tabella degli errori che non esisteva quando la tabella ha almeno un indice di ricerca completo (FTS) e viene eseguita un'`TRUNCATE`istruzione sull'istanza DB di Aurora writer.

Aggiornamenti e migrazioni:

- È stato risolto un problema che causava il fallimento degli aggiornamenti o delle migrazioni quando la versione del motore Aurora MySQL DB di destinazione è 3.04.0 o successiva. Ciò si verifica quando il parametro del cluster `lower_case_table_names` DB è impostato su e le regole di `1` confronto del database MySQL sono incompatibili con i nomi di tabella in minuscolo.

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla 8.0.28 inclusa. Per ulteriori informazioni, consulta Correzione dei bug di MySQL attraverso gli aggiornamenti del motore di database Aurora MySQL 3.x.

Aggiornamenti del motore di database Aurora MySQL 2024-03-15 (versione 3.04.2, compatibile con MySQL 8.0.28)

Versione: 3.04.2

Aurora MySQL 3.04.2 è disponibile a livello generale. Le versioni di Aurora MySQL 3.04 sono compatibili con MySQL 8.0.28. Per ulteriori informazioni sulle modifiche apportate alla community, consulta le Note di rilascio di MySQL 8.0.

Per informazioni sulle nuove funzionalità di Aurora MySQL versione 3, consulta Aurora MySQL versione 3 compatibile con MySQL 8.0. Per le differenze tra Aurora MySQL versione 3 e Aurora MySQL versione 2, vedi Confronto tra Aurora MySQL versione 2 e Aurora MySQL versione 3. Per un confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition, vedi Confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition.

> **Note**
>
> Questa versione è designata come versione di supporto a lungo termine (LTS). Per ulteriori informazioni, consulta Versioni con supporto a lungo termine (Long-Term Support, LTS) di Aurora MySQL nella Guida per l'utente di Amazon Aurora.
> Si consiglia di non impostare il **`AutoMinorVersionUpgrade`** parametro su **`true`** (o abilitare l'aggiornamento automatico della versione secondaria in) per le versioni LTS. Console di gestione AWS Ciò potrebbe comportare l'aggiornamento del cluster DB alla versione di destinazione successiva per la campagna Automatic Minor Version Upgrade, che potrebbe non essere una versione LTS.

Le versioni di Aurora MySQL attualmente supportate sono 2.07.9, 2.7.10, 2.11.*, 2.12.*, 3.03.*, 3.04.*, 3.05.* e 3.06.*.

Puoi eseguire un aggiornamento sul posto, ripristinare uno snapshot o avviare un aggiornamento gestito blue/green utilizzando Amazon RDS Blue/Green Deployments da qualsiasi cluster Aurora MySQL versione 2 attualmente disponibile in un cluster Aurora MySQL versione 3.04.2.

Per informazioni sulla pianificazione di un aggiornamento ad Aurora MySQL versione 3, consulta Pianificazione dell'aggiornamento per Aurora MySQL versione 3 nella Guida per l'utente di Amazon Aurora. Per informazioni generali sugli aggiornamenti di Aurora MySQL, consulta Aggiornamento dei cluster database Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Per informazioni sulla risoluzione dei problemi, consulta Risoluzione dei problemi di aggiornamento con Aurora MySQL versione 3.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Risolti i problemi di sicurezza e CVEs:

Le seguenti correzioni CVE sono incluse in questa versione:

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

Miglioramenti della disponibilità:

- È stato risolto un problema per cui un'istanza DB di replica in lettura non può essere avviata correttamente quando il carico di lavoro nell'istanza DB di Writer è elevato.
- È stato risolto un problema a causa del quale un'istanza DB di Aurora MySQL Writer poteva fallire a causa di un difetto nel componente che comunica con lo storage Aurora. Il difetto si verifica a seguito di un'interruzione della comunicazione tra l'istanza DB e lo storage sottostante a seguito di un aggiornamento del software.
- È stato risolto un problema che poteva causare il riavvio di un'istanza di database quando si eseguivano contemporaneamente le istruzioni SHOW STATUS e PURGE BINARY LOGS. `PURGE BINARY LOGS` è un'istruzione gestita che viene eseguita per rispettare il periodo di conservazione dei binlog configurato dall'utente.
- È stato risolto un problema che, durante il riavvio di un'istanza di database, può causare un ulteriore riavvio.
- È stato risolto un problema di conflitto di blocchi causato da un thread di log di audit che poteva portare a un utilizzo elevato della CPU e al timeout delle applicazioni client.
- È stato risolto un problema per cui un'istanza del database Aurora MySQL può subire riavvii multipli durante l'avvio dell'istanza mentre venivano inizializzati segmenti di rollback di grandi dimensioni.

- È stato risolto un problema che poteva causare il riavvio di un'istanza DB durante l'esecuzione di una query che faceva riferimento a una funzione aggregata.

Miglioramenti generali:

- È stato risolto un problema che poteva causare il fallimento di una query parallela a causa di problemi di rete transitori durante la lettura dei dati dal volume del cluster Aurora DB
- È stato risolto un problema per cui l'utente non era in grado di interrompere alcuna query o impostare i timeout di sessione per le query. `performance_schema`
- È stato risolto un problema a causa del quale la replica del log binario (binlog) configurata per l'utilizzo di certificati SSL personalizzati (mysql.rds_import_binlog_ssl_material) poteva fallire quando l'istanza di replica era in fase di sostituzione dell'host.
- È stato risolto un problema relativo alla gestione dei file di registro di controllo che poteva rendere i file di registro inaccessibili per il download o la rotazione e, in alcuni casi, aumentare l'utilizzo della CPU.
- Ripristino delle `AUTO_INCREMENT` chiavi ottimizzato per ridurre i tempi di completamento per il ripristino delle istantanee, l'esecuzione del point-in-time ripristino e la clonazione di cluster DB con un numero elevato di tabelle nel database.
- È stato risolto un problema per cui le istruzioni SQL riferite ad alcune `performance_schema` tabelle possono restituire un errore dovuto alla mancanza di tali tabelle dopo la migrazione da Community MySQL alle versioni 3.04.0 e 3.04.1 di Aurora MySQL.
- È stato risolto un problema a causa del quale le istanze di replica in lettura di piccole dimensioni possono subire un maggiore ritardo di replica dopo l'aggiornamento da versioni di Aurora MySQL precedenti alla 2.11.*.
- È stato risolto un problema che poteva causare errori chiave duplicati per le `AUTO_INCREMENT` colonne che utilizzavano indici decrescenti dopo un'operazione di ripristino di istantanee, backtrack o clonazione del database.
- È stato risolto un problema che poteva far sì che le modifiche del parametro del `table_open_cache` database non avessero effetto fino al riavvio dell'istanza DB.
- È stato risolto un problema a causa del quale l'istanza Reader DB non era in grado di aprire una tabella, con ERRORE 1146. Questo problema si verifica quando si eseguono determinati tipi di istruzioni DDL (Data Definition Language) online mentre l'`INPLACE`algoritmo viene utilizzato sull'istanza Writer DB.

- È stato risolto un problema per impedire il riavvio dell'istanza durante il dimensionamento di Aurora Serverless v2 quando il processo di monitoraggio interno inviava erroneamente richieste di dimensionamento duplicate.
- È stato risolto un problema che poteva causare il riavvio del database quando gli utenti di log binario (binlog) connessi utilizzano un server di replica binlog duplicato. IDs

Aggiornamenti e migrazioni:

- È stato risolto un problema che poteva causare il fallimento degli aggiornamenti delle versioni principali di Aurora MySQL versione 3 a causa della presenza di voci orfane per i tablespace già eliminati nelle tabelle di sistema InnoDB in Aurora MySQL versione 2.

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla 8.0.28 inclusa, oltre a quanto segue. Per ulteriori informazioni, consulta MySQL bugs fixed by Aurora MySQL 3.x database engine updates.

- È stato risolto un problema per cui il valore della riga della cache poteva essere calcolato in modo errato, causando un errore durante il riavvio del database su istanze basate su Graviton. (Correzione di bug della community #35479763)
- L'esecuzione ripetuta di una routine memorizzata, utilizzando come sottoquery un'istruzione SELECT contenente più `XOR` condizioni o `AND OR`, ha portato a un consumo eccessivo e probabilmente all'esaurimento della memoria virtuale. (Correzione di bug della community #33852530)

Aggiornamenti del motore di database Aurora MySQL 2023-11-13 (versione 3.04.1, compatibile con MySQL 8.0.28)

Versione: 3.04.1

Aurora MySQL 3.04.1 è disponibile a livello generale. Le versioni di Aurora MySQL 3.04 sono compatibili con MySQL 8.0.28. Per ulteriori informazioni sulle modifiche apportate alla community, consulta le Note di rilascio di MySQL 8.0.

> **Note**
>
> Questa versione è designata come versione di supporto a lungo termine (LTS). Per ulteriori informazioni, consulta Versioni con supporto a lungo termine (Long-Term Support, LTS) di Aurora MySQL nella Guida per l'utente di Amazon Aurora.
> Si consiglia di non impostare il `AutoMinorVersionUpgrade` parametro su `true` (o abilitare l'aggiornamento automatico della versione secondaria in Console di gestione AWS) per le versioni LTS. Ciò potrebbe comportare l'aggiornamento del cluster DB alla versione di destinazione successiva per la campagna Automatic Minor Version Upgrade, che potrebbe non essere una versione LTS.

Per informazioni sulle nuove funzionalità di Aurora MySQL versione 3, consulta Aurora MySQL versione 3 compatibile con MySQL 8.0. Per le differenze tra Aurora MySQL versione 3 e Aurora MySQL versione 2, vedi Confronto tra Aurora MySQL versione 2 e Aurora MySQL versione 3. Per un confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition, vedi Confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition.

Le versioni di Aurora MySQL attualmente supportate sono 2.07.9, 2.7.10, 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.*, 3.04.* e 3.05.*.

Puoi eseguire un aggiornamento sul posto, ripristinare uno snapshot o avviare un aggiornamento gestito blue/green utilizzando Amazon RDS Blue/Green Deployments da qualsiasi cluster Aurora MySQL versione 2 attualmente disponibile in un cluster Aurora MySQL versione 3.04.1.

Per informazioni sulla pianificazione di un aggiornamento ad Aurora MySQL versione 3, consulta Pianificazione dell'aggiornamento per Aurora MySQL versione 3 nella Guida per l'utente di Amazon Aurora. Per informazioni generali sugli aggiornamenti di Aurora MySQL, consulta Aggiornamento dei cluster database Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Per informazioni sulla risoluzione dei problemi, consulta Risoluzione dei problemi di aggiornamento con Aurora MySQL versione 3.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Miglioramenti della disponibilità:

- È stato risolto un problema per cui le istanze database di Aurora MySQL che utilizzavano query parallele potevano subire un riavvio del database durante l'esecuzione di un numero elevato di query parallele simultanee.
- È stato risolto un problema che poteva causare il ripristino errato del set GTID eseguito su un cluster di replica di log binario (binlog) con binlog avanzato abilitato quando una sorgente binlog era impostata su o. `gtid_mode` `ON` `ON_PERMISSIVE` Questo problema può causare il riavvio dell'istanza di scrittura del cluster di replica un'ulteriore volta durante il ripristino o portare a risultati errati durante l'esecuzione di query sul set GTID eseguito.
- È stato risolto un problema di gestione della memoria che può causare il riavvio di un'istanza database Aurora MySQL o un failover a causa di una diminuzione della memoria liberabile quando è abilitato il log binario avanzato.
- È stato risolto un problema che può causare il riavvio dell'istanza di lettura quando l'istanza di scrittura aumenta il volume del database fino a un multiplo di 160 GB.
- È stato risolto un problema a causa del quale un'istanza database Aurora MySQL con la funzionalità di registro binario avanzata abilitata poteva rimanere bloccata durante l'avvio dell'istanza database durante l'esecuzione del processo di ripristino del registro binario.
- È stato risolto un problema che poteva causare il riavvio di un'istanza database a causa di un deadlatch quando si eseguivano contemporaneamente le istruzioni `SHOW STATUS` e `PURGE BINARY LOGS`. L'eliminazione dei log binari è un'istruzione gestita che viene eseguita per rispettare il periodo di conservazione di binlog configurato dall'utente.
- È stato risolto un problema che poteva causare la non disponibilità del cluster di database al riavvio dell'istanza di scrittura mentre il database stava creando o eliminando i trigger sulle tabelle di sistema interne.
- È stato risolto un problema che può causare il riavvio di un'istanza database a causa di una lunga attesa del semaforo quando si utilizza la funzionalità binlog avanzata su un cluster con una replica Aurora.

Miglioramenti generali:

- È stato risolto un problema che può causare l'indisponibilità del database quando il binlog avanzato è abilitato su un cluster di database Aurora Serverless v2 in esecuzione su Aurora MySQL 3.04.0.
- Sono stati rimossi i metadati di archiviazione non utilizzati prima della scrittura su Aurora Storage quando è abilitata la funzionalità binlog avanzata. Ciò evita determinati scenari in cui potrebbe verificarsi un riavvio o un failover del database a causa della maggiore latenza di scrittura dovuta all'aumento dei byte trasmessi sulla rete.

- È stato risolto un problema per cui le tabelle degli schemi delle prestazioni specifiche di Aurora non venivano create durante un aggiornamento o una migrazione.
- È stato risolto un problema che poteva causare la visualizzazione di risultati errati nelle `NumBinaryLogFiles` metriche attivate quando CloudWatch il binlog avanzato è abilitato.

Aggiornamenti e migrazioni:

- L'aggiornamento da MySQL 5.7 a MySQL 8.0 con un numero molto elevato di tabelle in un unico database ha causato un consumo eccessivo di memoria nel server. È stato rilevato che, durante il processo di verifica della possibilità di aggiornare le tabelle, abbiamo recuperato in anticipo tutti `Table` gli oggetti del dizionario di dati, elaborandoli ciascuno e recuperandone il nome, quindi eseguendo l'operazione sull'elenco. `CHECK TABLE ... FOR UPGRADE` In questo caso non era necessario recuperare tutti gli oggetti in anticipo e l'operazione ha contribuito notevolmente al consumo di memoria. Per correggere questo problema, in questi casi recuperiamo un oggetto `Table` alla volta, eseguiamo tutti i controlli necessari, ne recuperiamo il nome e rilasciamo l'oggetto, prima di procedere con quello successivo. (Bug 34526001)

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla 8.0.28 inclusa, oltre a quanto riportato di seguito. Per ulteriori informazioni, consulta MySQL bugs fixed by Aurora MySQL 3.x database engine updates.

- È stato risolto un problema che può causare un maggiore utilizzo della CPU a causa della rotazione dei certificati TLS in background (Correzione di un bug della community 34284186)

Aggiornamenti del motore di database Aurora MySQL 2023-07-31 (versione 3.04.0, compatibile con MySQL 8.0.28)

Versione: 3.04.0

Aurora MySQL 3.04.0 è disponibile a livello generale. Le versioni 3.04 di Aurora MySQL sono compatibili con MySQL 8.0.28, le versioni 3.03 di Aurora MySQL sono compatibili con MySQL 8.0.26, mentre le versioni 3.02 di Aurora MySQL sono compatibili con MySQL 8.0.23. Per ulteriori informazioni sulle modifiche apportate alla community dalla versione 8.0.23 alla 8.0.28, consulta MySQL 8.0 Release Notes.

> **Note**
>
> Questa versione è designata come versione di supporto a lungo termine (LTS). Per ulteriori informazioni, consulta Versioni con supporto a lungo termine (Long-Term Support, LTS) di Aurora MySQL nella Guida per l'utente di Amazon Aurora.
> Si consiglia di non impostare il **`AutoMinorVersionUpgrade`** parametro su **`true`** (o abilitare l'aggiornamento automatico della versione secondaria in) per le versioni LTS. Console di gestione AWS Ciò potrebbe comportare l'aggiornamento del cluster DB alla versione di destinazione successiva per la campagna Automatic Minor Version Upgrade, che potrebbe non essere una versione LTS.

Per informazioni sulle nuove funzionalità di Aurora MySQL versione 3, consulta Aurora MySQL versione 3 compatibile con MySQL 8.0. Per le differenze tra Aurora MySQL versione 3 e Aurora MySQL versione 2, vedi Confronto tra Aurora MySQL versione 2 e Aurora MySQL versione 3. Per un confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition, vedi Confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition.

Le versioni di Aurora MySQL attualmente supportate sono 2.07.9, 2.11.1, 2.11.2, 3.01.*, 3.02.*, 3.03.* e 3.04.0.

Puoi eseguire un aggiornamento sul posto, ripristinare uno snapshot o avviare un aggiornamento gestito blue/green utilizzando Amazon RDS Blue/Green Deployments da qualsiasi cluster Aurora MySQL versione 2 attualmente supportato in un cluster Aurora MySQL versione 3.04.0.

Per informazioni sulla pianificazione di un aggiornamento ad Aurora MySQL versione 3, consulta Pianificazione dell'aggiornamento per Aurora MySQL versione 3 nella Guida per l'utente di Amazon Aurora. Per informazioni generali sugli aggiornamenti di Aurora MySQL, consulta Aggiornamento dei cluster database Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Per informazioni sulla risoluzione dei problemi, consulta Risoluzione dei problemi di aggiornamento con Aurora MySQL versione 3.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Il log binario avanzato (binlog) di Aurora MySQL non è attualmente supportato per l'istanza database Aurora Serverless v2 su Aurora MySQL versione 3.04.0. L'attivazione di questa funzionalità può comportare l'indisponibilità del database. Se è necessario l'uso del log binario avanzato su Aurora MySQL versione 3.04.0, consigliamo di utilizzare una classe di istanza database non serverless o di impostare l'ACU minima e massima dell'istanza database Serverless v2 sullo stesso valore.
> Ulteriori informazioni sulla registrazione binaria avanzata in Aurora MySQL sono disponibili nella Guida per l'utente di Aurora.

Miglioramenti

Nuove caratteristiche:

- Sono state migliorate le prestazioni delle query utilizzando gli indici testuali di InnoDB per cercare frasi in modalità linguaggio naturale. Per ulteriori informazioni sulle ricerche full-text in MySQL, consulta Full-Text Search Functions.
- Amazon Aurora MySQL supporta l'inoltro di scrittura locale (in un cluster). È ora possibile inoltrare operazioni di scrittura da un'istanza database di lettura a un'istanza database di scrittura all'interno di un cluster database Aurora MySQL. Per ulteriori informazioni, consulta la sezione relativa all'utilizzo dell'inoltro di scrittura locale in un cluster database Aurora MySQL.
- È stata aggiunta la possibilità di modificare il valore del parametro `aurora_replica_read_consistency` per la funzionalità Utilizzo dell'inoltro di scrittura in un database globale Amazon Aurora nelle sessioni con `autocommit` disattivato. Per ulteriori informazioni, consulta Configuration parameters for write forwarding.
- A partire da Aurora MySQL 3.04, per la funzionalità di inoltro di scrittura del database globale, è ora possibile impostare il valore del parametro `aurora_replica_read_consistency` tramite il cluster di database e i gruppi di parametri delle istanze database. Prima della versione 3.04 di Aurora MySQL, il valore di questo parametro poteva essere configurato solo a livello di sessione.

Risolti i problemi di sicurezza e CVEs:

- Cambiato il SSL/TLS provider da AWS-LCOpenSSL a. Ciò comporta una serie di modifiche tra cui, a titolo esemplificativo ma non esaustivo, le seguenti:

  - Le connessioni al database che utilizzano SSL possono ora essere ripristinate tramite il riavvio senza tempi di inattività (Zero Downtime Restart) e l'applicazione delle patch senza tempi di inattività (Zero Downtime Patching) durante l'aggiornamento da Aurora MySQL versione 3.04.0 a una versione successiva.
  - Supporto per TLSv1 .3 che include il supporto per i cifrari SSL TLS_AES_128_GCM_, TLS_AES_256_GCM_ SHA256 e TLS_ 0_ 05_. SHA384 CHACHA2 POLY13 SHA256
  - Rimozione del supporto per le crittografie DHE-RSA-* meno sicure.

  Per ulteriori informazioni, consulta Utilizzo di TLS con cluster database Aurora MySQL
- È stato aggiunto il privilegio dinamico `SHOW_ROUTINE` a `rds_superuser_role` che consente l'accesso alle definizioni e alle proprietà di tutte le routine archiviate, come le procedure archiviate e le funzioni. Per ulteriori dettagli, consulta SHOW_ROUTINE.
- È stato risolto un problema che poteva causare la mancata visualizzazione di eventi nel log di audit durante la rotazione dei file del log di audit.
- Abilitato il supporto per il protocollo Transport Layer Security (TLS) 1.3 sicuro e ad alte prestazioni mantenendo la compatibilità con la versione TLS 1.2.
- Le versioni TLS TLSv1 e TLSv1 .1 erano obsolete nella community MySQL 8.0.26 e, di conseguenza, in Aurora MySQL 3.03. Questi protocolli sono stati ora rimossi nella community MySQL 8.0.28 e di conseguenza in Aurora MySQL 3.04. Per impostazione predefinita, qualsiasi client sicuro che non può comunicare tramite TLS 1.2 o versioni successive verrà rifiutato. Per ulteriori informazioni sulla connessione alle istanze database tramite TLS, consulta Sicurezza con Amazon Aurora MySQL.

Le seguenti correzioni CVE sono incluse in questa versione:

- 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

Miglioramenti della disponibilità:

- È stato risolto un problema che poteva causare il riavvio del database durante il ripristino di transazioni prolungate.
- È stato risolto un problema relativo alla crittografia degli eventi dei flussi di attività del database che può causare il riavvio del database.
- È stato risolto un problema di gestione della memoria dovuto a errori di esaurimento della memoria durante l'inizializzazione del pool di buffer InnoDB durante l'avvio o durante il ridimensionamento in Aurora Serverless v2. Questo problema potrebbe aver causato il riavvio delle istanze del database o il peggioramento delle prestazioni, inclusa la riduzione della velocità effettiva o l'aumento della latenza.
- È stato risolto un problema che poteva causare il riavvio di un'istanza del lettore Aurora MySQL durante l'esecuzione di una query che utilizza un piano di esecuzione delle query parallele Aurora MySQL.
- È stato risolto un problema che, in determinate situazioni, poteva causare il riavvio delle istanze del lettore Aurora durante una stima dell'intervallo.
- È stato risolto un problema che poteva interrompere il ripristino del database durante l'avvio se il riavvio si verificava durante l'esecuzione di operazioni di inserimento pesanti che coinvolgevano l'incremento automatico delle colonne.
- È stato risolto un problema con l'audit avanzato di Aurora (Advanced Auditing) che causava un eccesso di registrazione dei messaggi informativi nel log degli errori di Aurora MySQL quando la variabile del server `server_audit_events` è impostata su `ALL` o `QUERY`. Questo problema potrebbe causare il riavvio dell'istanza del database.
- È stato risolto un problema che può causare il riavvio del database durante il rollback di un'`INSERT`istruzione quando la query parallela è abilitata.
- È stato risolto un problema che poteva causare il riavvio dell'istanza del database durante l'esecuzione dello strumento di `EXPLAIN ANALYZE` profilazione su una query che restituiva l'output `all select tables were optimized away` all'interno della colonna di `EXTRA` informazioni. Per ulteriori informazioni, consulta `EXPLAIN` Output Format.nella documentazione di MySQL.
- È stato risolto un problema che poteva causare il riavvio di un'istanza Region reader secondaria del database globale Aurora che utilizzava l'inoltro globale di scrittura quando un'istruzione di commit implicita inoltrata rilevava un errore.

- È stato risolto un problema che poteva causare il riavvio dell'istanza writer in una regione primaria del database globale Aurora quando una `SELECT FOR UPDATE` query veniva eseguita utilizzando l'inoltro globale di scrittura da una regione secondaria del database globale Aurora.

Miglioramenti generali:

- È stata aggiunta una nuova procedura archiviata, `mysql.rds_gtid_purged`, per consentire ai clienti di impostare la variabile di sistema `GTID_PURGED`. Per ulteriori informazioni, consulta mysql.rds_gtid_purged.
- Sono state aggiunte due nuove procedure archiviate, `mysql.rds_start_replication_until` e `mysql.rds_start_replication_until_gtid`, che consentono ai clienti di configurare una posizione per interrompere la replica dei log binari. Per ulteriori informazioni sulla configurazione di una posizione di arresto per la replica dei log binari in Aurora MySQL, consulta mysql.rds_start_replication_until.
- È stato risolto un problema che impediva alle procedure archivate di controllo della replica di Aurora MySQL di modificare la variabile `sql_log_bin`, se richiamate da una sessione con la modalità autocommit disabilitata.
- È stato aggiunto il supporto per la replica logica per le seguenti istruzioni DCL (Data Control Language): `GRANT/REVOKE` e `CREATE/DROP/ALTER/RENAME USER`.
- È stato risolto un problema per evitare che le statistiche di InnoDB diventassero obsolete, il che a volte può generare un piano di esecuzione delle query non ottimale che può portare a un aumento del tempo di esecuzione delle query.
- Sono state aggiunte due nuove viste di sistema, `information_schema.aurora_global_db_instance_status` e `information_schema.aurora_global_db_status`. Queste viste possono essere utilizzate per visualizzare lo stato e la topologia delle risorse primarie e secondarie in un cluster di database globale Aurora MySQL. I dettagli delle due viste di sistema sono disponibili qui, Aurora MySQL–specific information_schema tables.
- È stato risolto un problema per cui un utente non era in grado di accedere al database con un carattere jolly nel nome del database dopo aver eseguito l'istruzione `SET ROLE` con un carattere jolly di escape.
- È stato risolto un problema per cui gli eventi segnalati durante l'elaborazione delle rotazioni dei log di audit potevano non essere scritti nel log di audit.
- È stato risolto un problema per cui la creazione di una tabella temporanea interna, tramite un'esecuzione `TRIGGER`, può causare il riavvio di un'istanza database di scrittura.

- È stata aggiunta una nuova variabile di sistema, `innodb_aurora_max_partitions_for_range`. In alcuni casi in cui le statistiche persistenti non sono disponibili, è possibile utilizzare questo parametro per migliorare il tempo di esecuzione delle stime del numero di righe sulle tabelle partizionate. Ulteriori informazioni sono disponibili nella documentazione, Parametri di configurazione Aurora MySQL.
- È stato risolto un problema che consentiva erroneamente ai clienti di impostare `ROW_FORMAT` come `COMPRESSED` durante la creazione di tabelle partizionate. Le tabelle verranno convertite implicitamente in formato `COMPACT` con un avviso per informare che Aurora MySQL non supporta le tabelle compresse.
- È stato risolto un problema che poteva causare l'interruzione della replica multithread del log binario quando la `replica_parallel_type` variabile è impostata su e la variabile viene attivata. `LOGICAL_CLOCK` `replica_preserve_commit_order` `ON` Questo problema può verificarsi quando una transazione di dimensioni superiori a 500 MB viene eseguita sull'origine.
- È stato risolto un problema che si verificava quando era abilitata la funzionalità di inoltro di scrittura del database globale che poteva causare l'inoltro involontario delle modifiche alla configurazione `performance_schema` delle istanze di lettura nelle regioni secondarie all'istanza di scrittura nella regione primaria.
- È stato risolto un problema a causa del quale la variabile di stato del server `innodb_buffer_pool_reads` non poteva essere aggiornata dopo la lettura di una pagina di dati dal file system di archiviazione Aurora.
- La query parallela per Amazon Aurora MySQL non è supportata quando si sceglie la configurazione del cluster Aurora con ottimizzazione I/O. Per ulteriori informazioni, consulta Restrizioni delle query in parallelo per Aurora MySQL.
- È stato risolto un problema che, quando la query parallela è abilitata, faceva sì che l'ottimizzatore del piano di query scegliesse un piano di esecuzione inefficiente per determinate query `SELECT` che beneficiano di un indice primario o secondario.
- Sono state aggiornate le definizioni del fuso orario alla versione IANA 2023c.
- Sono state introdotte ottimizzazioni delle prestazioni di gestione dei file sulle repliche binlog per ridurre i conflitti durante la scrittura su file di log di relay.
- È stato risolto un problema per cui la `RPO_LAG_IN_MILLISECONDS` colonna nella `information_schema.aurora_global_db_status` tabella e nella `AuroraGlobalDBRPOLag` CloudWatch metrica mostrava sempre zero indipendentemente dal carico di lavoro dell'utente.
- È stato introdotto un nuovo parametro `aurora_tmptable_enable_per_table_limit`. Quando questo parametro è abilitato, la `tmp_table_size` variabile definisce la dimensione massima della

singola tabella temporanea interna in memoria creata dal TempTable motore di archiviazione. Per ulteriori informazioni, consulta Motore di storage per tabelle temporanee (implicite) interne.

- È stato risolto un problema a causa del quale veniva creata una connessione aggiuntiva quando la funzionalità di inoltro di scrittura del database globale era abilitata. Il problema si verifica quando le transazioni di sola lettura su un'istanza di lettura inoltrano erroneamente un commit implicito alla scrittura.
- È stato risolto un problema a causa del quale i campi `PROCESSLIST_USER` e `PROCESSLIST_HOST` della tabella `performance_schema.threads` non venivano compilati sulla scrittura nella regione primaria per le connessioni che utilizzavano la funzionalità di inoltro di scrittura del database globale. Ulteriori informazioni su questa tabella e sullo schema delle prestazioni sono disponibili in MySQL Reference Manual, The threads Table e nella Panoramica su Performance Schema nella Guida per l'utente di Amazon Aurora.
- È stato risolto un problema a causa del quale la metrica `CommitLatency` di Cloudwatch mostrava valori errati per le istanze di lettura nelle regioni secondarie quando veniva utilizzata la funzionalità di inoltro di scrittura del database globale. Per monitorare la latenza delle istruzioni DML inoltrate su cluster di database secondari, si consiglia di utilizzare le metriche `ForwardingReplicaDMLLatency` e `ForwardingWriterDMLLatency`. La latenza del commit può essere osservata anche utilizzando la metrica `CommitLatency` sull'istanza di scrittura della regione principale. Ulteriori informazioni sono disponibili nella Aurora User Guide, Amazon CloudWatch metrics for write forwarding.
- È stato risolto un problema a causa del quale le procedure archiviate di controllo delle repliche di Aurora MySQL utilizzate per gestire e configurare la replica dei log binari segnalavano erroneamente degli errori quando la replica di log binari multi-thread veniva configurata impostando il valore della variabile `replica_parallel_workers` maggiore di 0.
- È stato risolto un problema che poteva causare un elevato consumo di CPU quando più sessioni tentavano di accedere a una pagina che non esiste in memoria.

Aggiornamenti e migrazioni:

- Per eseguire un aggiornamento secondario della versione per un database globale Aurora da Aurora MySQL versione 3.01, 3.02 o 3.03 ad Aurora MySQL versione 3.04 o successiva, consulta Aggiornamento di Aurora MySQL modificando la versione del motore.
- È stato risolto un problema che poteva causare errori di controllo preliminare dell'aggiornamento a causa di errori di incoerenza dello schema segnalati per le tabelle `mysql.general_log_backup`, `mysql.general_log`, `mysql.slow_log_backup` e `mysql.slow_log` durante l'aggiornamento

da Aurora MySQL 2 ad Aurora MySQL 3. Per ulteriori informazioni sulla risoluzione dei problemi di aggiornamento, consulta Risoluzione dei problemi di aggiornamento con Aurora MySQL versione 3.

- È stato risolto un problema che poteva causare gravi errori di aggiornamento della versione durante l'aggiornamento ad Aurora MySQL 3 quando una definizione di trigger contiene una parola chiave riservata che non è racchiusa tra virgolette.

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla 8.0.28 inclusa, oltre a quanto riportato di seguito. Per ulteriori informazioni, consulta MySQL bugs fixed by Aurora MySQL 3.x database engine updates.

- È stato risolto un problema a causa del quale un blocco buffer contenente una pagina di tabella temporanea intrinseca veniva riposizionato durante l'attraversamento delle pagine, causando un errore di asserzione (Bug 33715694).
- InnoDB: impedisce alle operazioni DDL online di accedere alla out-of-bounds memoria (bug # 34750489, bug # 108925)
- È stato risolto un problema che a volte poteva produrre risultati di query errati durante l'elaborazione di istruzioni SQL complesse costituite da più espressioni di tabella comuni annidate () (Bug# 34572040, Bug# 34634469, Bug# 33856374) CTEs

Aggiornamenti del motore di database Aurora MySQL 2023-12-08 (versione 3.03.3) (obsoleta)

Versione: 3.03.3

Aurora MySQL 3.03.3 è disponibile a livello generale. Le versioni 3.03 di Aurora MySQL sono compatibili con MySQL 8.0.26. Per ulteriori informazioni sulle modifiche apportate alla community dalla versione 8.0.23 alla 8.0.28, consulta MySQL 8.0 Release Notes.

Per informazioni sulle nuove funzionalità di Aurora MySQL versione 3, consulta Aurora MySQL versione 3 compatibile con MySQL 8.0. Per le differenze tra Aurora MySQL versione 3 e Aurora MySQL versione 2, vedi Confronto tra Aurora MySQL versione 2 e Aurora MySQL versione 3. Per un confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition, vedi Confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition.

Le versioni di Aurora MySQL attualmente disponibili sono 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.*, 3.04.* e 3.05.*.

Puoi eseguire un aggiornamento sul posto, ripristinare uno snapshot o avviare un aggiornamento gestito blue/green utilizzando Amazon RDS Blue/Green Deployments da qualsiasi cluster Aurora MySQL versione 2 attualmente disponibile in un cluster Aurora MySQL versione 3.03.3.

Per informazioni sulla pianificazione di un aggiornamento ad Aurora MySQL versione 3, consulta Pianificazione dell'aggiornamento per Aurora MySQL versione 3 nella Guida per l'utente di Amazon Aurora. Per informazioni generali sugli aggiornamenti di Aurora MySQL, consulta Aggiornamento dei cluster database Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Per informazioni sulla risoluzione dei problemi, consulta Risoluzione dei problemi di aggiornamento con Aurora MySQL versione 3.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Risolti i problemi di sicurezza ed CVEs elencati di seguito:

Correzioni e altri miglioramenti per ottimizzare la gestione in un ambiente gestito. Ulteriori correzioni CVE sono riportate di seguito:

- CVE-2023-38545

Miglioramenti della disponibilità:

- È stato risolto un problema per cui le istanze database di Aurora MySQL che utilizzavano query parallele potevano subire un riavvio del database durante l'esecuzione di un numero elevato di query parallele simultanee.
- È stato risolto un problema che poteva causare il ripristino errato del set GTID eseguito su un cluster di replica di log binario (binlog) con binlog avanzato abilitato quando qualsiasi origine binlog disponeva di `gtid_mode` impostata su `ON` o `ON_PERMISSIVE`. Questo problema può causare il riavvio dell'istanza di scrittura del cluster di replica un'ulteriore volta durante il ripristino o portare a risultati errati durante l'esecuzione di query sul set GTID eseguito.

- È stato risolto un problema di gestione della memoria che può causare il riavvio di un'istanza database Aurora MySQL o un failover a causa di una diminuzione della memoria liberabile quando è abilitato il log binario avanzato.
- È stato risolto un problema che può causare il riavvio dell'istanza di lettura quando l'istanza di scrittura aumenta il volume del database fino a un multiplo di 160 GB.
- È stato risolto un problema a causa del quale un'istanza database Aurora MySQL con la funzionalità di registro binario avanzata abilitata poteva rimanere bloccata durante l'avvio dell'istanza database durante l'esecuzione del processo di ripristino del registro binario.
- È stato risolto un problema durante l'applicazione di patch senza tempi di inattività che causava il riavvio dell'istanza con conseguente chiusura imprevista delle connessioni al database.
- È stato risolto un problema che poteva causare il riavvio di un'istanza database a causa di un deadlatch quando si eseguivano contemporaneamente le istruzioni `SHOW STATUS` e `PURGE BINARY LOGS`. L'eliminazione dei log binari è un'istruzione gestita che viene eseguita per rispettare il periodo di conservazione di binlog configurato dall'utente.
- È stato risolto un problema che può causare il riavvio di un'istanza database a causa di una lunga attesa del semaforo quando si utilizza la funzionalità binlog avanzata su un cluster con una replica Aurora.

Miglioramenti generali:

- Sono stati rimossi i metadati di archiviazione non utilizzati prima di scrivere sull'archiviazione di Aurora quando è abilitata la funzionalità binlog avanzato. Ciò evita determinati scenari in cui potrebbe verificarsi un riavvio o un failover del database a causa della maggiore latenza di scrittura dovuta all'aumento dei byte trasmessi sulla rete.
- È stato risolto un problema che poteva causare la visualizzazione di risultati errati nelle `NumBinaryLogFiles` metriche attivate quando il binlog avanzato era abilitato. CloudWatch
- È stato risolto un problema che poteva far sì che la modifica del parametro del database `table_open_cache` non avesse effetto fino al riavvio dell'istanza database.
- È stato risolto un problema che poteva causare il riavvio del database quando gli utenti di log binario (binlog) connessi utilizzano un server di replica binlog duplicato. IDs

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla 8.0.26 inclusa, oltre a quanto riportato di seguito. Per ulteriori informazioni, consulta MySQL bugs fixed by Aurora MySQL 3.x database engine updates.

- È stato risolto un problema che può causare un maggiore utilizzo della CPU a causa della rotazione dei certificati TLS in background (Correzione di un bug della community 34284186)

Aggiornamenti del motore di database Aurora MySQL 2023-08-29 (versione 3.03.2) (obsoleto)

Versione: 3.03.2

Aurora MySQL 3.03.2 è disponibile a livello generale. Le versioni 3.04 di Aurora MySQL sono compatibili con MySQL 8.0.28, le versioni 3.03 di Aurora MySQL sono compatibili con MySQL 8.0.26, mentre le versioni 3.02 di Aurora MySQL sono compatibili con MySQL 8.0.23. Per ulteriori informazioni sulle modifiche apportate alla community dalla versione 8.0.23 alla 8.0.28, consulta MySQL 8.0 Release Notes.

Per informazioni sulle nuove funzionalità di Aurora MySQL versione 3, consulta Aurora MySQL versione 3 compatibile con MySQL 8.0. Per le differenze tra Aurora MySQL versione 3 e Aurora MySQL versione 2, vedi Confronto tra Aurora MySQL versione 2 e Aurora MySQL versione 3. Per un confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition, vedi Confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition.

Le versioni di Aurora MySQL attualmente disponibili sono 2.07.9, 2.07.10, 2.11.*, 3.01.*, 3.02.*, 3.03.* e 3.04.*.

Puoi eseguire un aggiornamento sul posto, ripristinare uno snapshot o avviare un aggiornamento gestito blue/green utilizzando Amazon RDS Blue/Green Deployments da qualsiasi cluster Aurora MySQL versione 2 attualmente disponibile in un cluster Aurora MySQL versione 3.03.2.

Per informazioni sulla pianificazione di un aggiornamento ad Aurora MySQL versione 3, consulta Pianificazione dell'aggiornamento per Aurora MySQL versione 3 nella Guida per l'utente di Amazon Aurora. Per informazioni generali sugli aggiornamenti di Aurora MySQL, consulta Aggiornamento dei cluster database Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Per informazioni sulla risoluzione dei problemi, consulta Risoluzione dei problemi di aggiornamento con Aurora MySQL versione 3.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Risolti i problemi di sicurezza e CVEs:

- È stato risolto un problema che poteva causare la mancata visualizzazione di eventi nel registro di controllo durante la rotazione del file del registro di controllo.

Le seguenti correzioni CVE sono incluse in questa versione:

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

Miglioramenti della disponibilità:

- È stato risolto un problema che poteva causare il riavvio del database durante il ripristino di transazioni prolungate.
- È stato risolto un problema che poteva causare la non disponibilità del cluster di database al riavvio dell'istanza di writer mentre il database stava creando o eliminando i trigger sulle tabelle di sistema interne.
- È stato risolto un problema che può causare il riavvio di un'istanza di database durante l'esecuzione di una query che fa riferimento a una funzione aggregata.
- È stato risolto un problema che può causare il riavvio del database durante il rollback di un'`INSERT`istruzione quando la query parallela è abilitata.
- L'inserimento rapido è abilitato solo per le normali tabelle InnoDB in Aurora MySQL versione 3.03.2 e successive. Questa ottimizzazione non funziona per le tabelle temporanee di InnoDB. Per

ulteriori informazioni sull'ottimizzazione rapida degli inserti, consulta Miglioramenti delle prestazioni di Amazon Aurora MySQL.

Miglioramenti generali:

- È stato risolto un problema per cui l'istanza di lettura non era in grado di aprire una tabella, con `ERROR 1146`. Questo problema si verifica quando si eseguono determinati tipi di Data Definition Language (DDL) online mentre l'algoritmo `INPLACE`viene utilizzato sull'istanza di scrittura.
- Sono state introdotte ottimizzazioni delle prestazioni di gestione dei file sulle repliche binlog per ridurre i conflitti durante la scrittura su file di log di relay.
- È stato risolto un problema che, quando la query parallela è abilitata, faceva sì che l'ottimizzatore del piano di query scegliesse un piano di esecuzione inefficiente per determinate query `SELECT` che beneficiano di un indice primario o secondario.
- È stato aggiunto il supporto per la replica logica per le seguenti istruzioni DCL (Data Control Language): `GRANT/REVOKE` e `CREATE/DROP/ALTER/RENAME USER`.
- La query parallela per Amazon Aurora MySQL non è supportata quando si sceglie la configurazione del cluster Aurora con ottimizzazione I/O. Per ulteriori informazioni, consulta Restrizioni delle query in parallelo per Aurora MySQL.

Aggiornamenti e migrazioni:

- Per eseguire un aggiornamento secondario della versione per un database globale Aurora da Aurora MySQL versione 3.01 o 3.02 ad Aurora MySQL versione 3.03 o successiva, consulta Aggiornamento di Aurora MySQL modificando la versione del motore.
- È stato risolto un problema che poteva causare gravi errori di aggiornamento della versione durante l'aggiornamento ad Aurora MySQL versione 3 quando una definizione di trigger contiene una parola chiave riservata che non è racchiusa tra virgolette.

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla 8.0.26 inclusa, oltre a quanto riportato di seguito. Per ulteriori informazioni, consulta MySQL bugs fixed by Aurora MySQL 3.x database engine updates.

- È stato risolto un problema che a volte poteva produrre risultati di query errati durante l'elaborazione di istruzioni SQL complesse costituite da più Common Table Expressions () annidate. CTEs (Bug 34572040, Bug 34634469, Bug 33856374)
- InnoDB: una race condition tra thread che tentavano di de-inizializzare e inizializzare le statistiche per la stessa tabella ha generato un errore di asserzione. (Bug 33135425)
- InnoDB: impedisce alle operazioni DDL online di accedere alla memoria. out-of-bounds (Bug 34750489, Bug 108925)

Aggiornamenti del motore di database Aurora MySQL 2023-05-11 (versione 3.03.1) (obsoleto)

Versione: 3,03.1

Aurora MySQL 3.03.1 è disponibile a livello generale. Le versioni 3.03 di Aurora MySQL sono compatibili con MySQL 8.0.26, mentre le versioni 3.02 di Aurora MySQL sono compatibili con MySQL 8.0.23. Per ulteriori informazioni sulle modifiche apportate alla community dalla versione 8.0.23 alla 8.0.26, consulta MySQL 8.0 Release Notes.

Per informazioni sulle nuove funzionalità di Aurora MySQL versione 3, consulta Aurora MySQL versione 3 compatibile con MySQL 8.0. Per le differenze tra Aurora MySQL versione 3 e Aurora MySQL versione 2, vedi Confronto tra Aurora MySQL versione 2 e Aurora MySQL versione 3. Per un confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition, vedi Confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition.

Le versioni di Aurora MySQL attualmente supportate sono 2.07.9, 2.11.1, 2.11.2, 3.01.*, 3.02.* e 3.03.*.

Puoi effettuare un aggiornamento locale o ripristinare uno snapshot da un cluster della versione 2 di Aurora MySQL attualmente supportata in Aurora MySQL 3.03.1.

Per informazioni sulla pianificazione di un aggiornamento ad Aurora MySQL versione 3, consulta Pianificazione dell'aggiornamento per Aurora MySQL versione 3 nella Guida per l'utente di Amazon Aurora. Per informazioni generali sugli aggiornamenti di Aurora MySQL, consulta Aggiornamento dei cluster database Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Per informazioni sulla risoluzione dei problemi, consulta Risoluzione dei problemi di aggiornamento con Aurora MySQL versione 3.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Nuove caratteristiche:

- La configurazione di I/O archiviazione ottimizzata Aurora è disponibile a partire dalla versione 3.03.1. Per ulteriori informazioni, consultare Configurazioni dell'archiviazione per i cluster database Amazon Aurora.
- È stata aggiunta una nuova variabile di sistema, `innodb_aurora_max_partitions_for_range`. In alcuni casi in cui le statistiche persistenti non sono disponibili, è possibile utilizzare questo parametro per migliorare il tempo di esecuzione delle stime del numero di righe sulle tabelle partizionate. Ulteriori informazioni sono disponibili nella documentazione, Parametri di configurazione Aurora MySQL.

Miglioramenti della disponibilità:

- È stato risolto un problema che può causare il riavvio dell'istanza database a causa dell'accesso errato alla memoria non valida quando una connessione viene chiusa immediatamente dopo il completamento di una transazione.
- È stato risolto un problema con Aurora Advanced Auditing che causava un eccesso di registrazione dei messaggi informativi nel log degli errori di Aurora MySQL quando la variabile del server `server_audit_events` è impostata su ALL o QUERY. Questo problema può causare il riavvio dell'istanza database.
- È stato risolto un problema che, in alcune situazioni, può causare il riavvio delle istanze del lettore Aurora quando si tenta di leggere una pagina che non è più accessibile durante una stima dell'intervallo.
- È stato risolto un problema che poteva causare il riavvio di un'istanza del lettore Aurora MySQL durante l'esecuzione di una query che utilizza un piano di esecuzione di query parallele di Aurora.
- È stato risolto un problema per cui le istanze database che utilizzano la replica di log binari potevano registrare un aumento dell'utilizzo della CPU ed errori di connessione quando collegati più consumer di replica di log binari.
- È stato risolto un problema per cui i metodi di accesso alla scansione degli indici non supportati venivano presi in considerazione per le espressioni di tabella comuni (CTE) durante la

materializzazione di tabelle temporanee intermedie, il che poteva portare a comportamenti indesiderati, tra cui riavvii del database o risultati di query errati. Risolviamo questo problema evitando l'uso di tali metodi di accesso alla scansione degli indici non supportati sulle tabelle che utilizzano il TempTable motore di archiviazione.

- L'inserimento rapido non è abilitato in questa versione di Aurora MySQL, a causa di un problema che può causare incongruenze durante l'esecuzione di query come, e. `INSERT INTO SELECT FROM` Per ulteriori informazioni sull'ottimizzazione rapida degli inserti, consulta Miglioramenti delle prestazioni di Amazon Aurora MySQL.

Miglioramenti generali:

- È stato risolto un problema che può causare tempi di esecuzione più lunghi del previsto per l'istruzione SHOW BINARY LOGS. Ciò potrebbe comportare una riduzione della velocità di trasmissione effettiva del commit del database.
- È stato risolto un problema che poteva causare la non riuscita dell'esportazione parallela per le tabelle utente con colonne aggiunte mediante la funzionalità ADD COLUMN immediata.
- È stato risolto un problema per cui gli eventi segnalati durante l'elaborazione delle rotazioni dei log di audit potevano non essere scritti nel log di audit.
- È stato risolto un problema che poteva causare l'esaurimento della memoria disponibile durante l'esecuzione di query sulla tabella INFORMATION_SCHEMA INNODB_TABLESPACES.
- È stato risolto un problema che consentiva erroneamente ai clienti di impostare ROW_FORMAT come COMPRESSED durante la creazione di tabelle partizionate. Le tabelle verranno convertite implicitamente in formato COMPACT con un avviso per informare che Aurora MySQL non supporta le tabelle compresse.

Aggiornamenti e migrazioni:

- Per eseguire un aggiornamento secondario della versione per un database globale Aurora da Aurora MySQL versione 3.01 o 3.02 ad Aurora MySQL versione 3.03 o successiva, consulta Aggiornamento di Aurora MySQL modificando la versione del motore.
- È stato risolto un problema che poteva causare errori di controllo preliminare dell'aggiornamento a causa di errori di incoerenza dello schema segnalati per le tabelle `mysql.general_log_backup`, `mysql.general_log`, `mysql.slow_log_backup` e `mysql.slow_log` durante l'aggiornamento da Aurora MySQL 2 ad Aurora MySQL 3. Per ulteriori informazioni sulla risoluzione dei problemi di

aggiornamento, consulta Risoluzione dei problemi di aggiornamento con Aurora MySQL versione 3.

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla 8.0.26 inclusa, oltre a quanto riportato di seguito. Per ulteriori informazioni, consulta MySQL bugs fixed by Aurora MySQL 3.x database engine updates.

- È stato risolto un problema a causa del quale un blocco buffer contenente una pagina di tabella temporanea intrinseca veniva riposizionato durante l'attraversamento delle pagine, causando un errore di asserzione. (Bug 33715694)

Aggiornamenti del motore di database Aurora MySQL (versione 3.03.0) (obsoleta)

Versione: 3.03.0

Aurora MySQL 3.03.0 è disponibile a livello generale. Le versioni 3.03 di Aurora MySQL sono compatibili con MySQL 8.0.26, mentre le versioni 3.02 di Aurora MySQL sono compatibili con MySQL 8.0.23. Per ulteriori informazioni sulle modifiche apportate alla community dalla versione 8.0.23 alla 8.0.26, consulta MySQL 8.0 Release Notes.

Per informazioni sulle nuove funzionalità di Aurora MySQL versione 3, consulta Aurora MySQL versione 3 compatibile con MySQL 8.0. Per le differenze tra Aurora MySQL versione 3 e Aurora MySQL versione 2, consulta la sezione Confronto tra Aurora MySQL versione 2 e Aurora MySQL versione 3. Per un confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition, vedi Confronto tra Aurora MySQL versione 3 e MySQL 8.0 Community Edition.

Le versioni di Aurora MySQL attualmente supportate sono 2.07.*, 2.11.*, 3.01.*, 3.02.* e 3.03.*.

Puoi effettuare un aggiornamento locale o ripristinare uno snapshot da un cluster della versione 2 di Aurora MySQL attualmente supportata in Aurora MySQL 3.03.0.

Per informazioni sulla pianificazione di un aggiornamento ad Aurora MySQL versione 3, consulta Pianificazione dell'aggiornamento per Aurora MySQL versione 3 nella Guida per l'utente di Amazon Aurora. Per informazioni generali sugli aggiornamenti di Aurora MySQL, consulta Aggiornamento dei cluster database Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Per informazioni sulla risoluzione dei problemi, consulta Risoluzione dei problemi di aggiornamento con Aurora MySQL versione 3.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Risolti i problemi di sicurezza ed CVEs elencati di seguito:

Correzioni e altri miglioramenti per ottimizzare la gestione in un ambiente gestito. Ulteriori correzioni CVE riportate di seguito:

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

Miglioramenti della disponibilità:

- È stato risolto un problema per cui le classi di istanza database più grandi potevano riscontrare problemi durante il riavvio a causa del tempo di inizializzazione del pool di buffer che richiedeva più tempo del previsto.
- È stato risolto un problema a causa del quale l'istanza database si poteva riavviare durante il processo di ripristino del database quando è abilitata la registrazione binaria.
- È stato risolto un problema che poteva causare errori di connessione sulle istanze di lettura durante l'esecuzione delle istruzioni Data Control Language (DCL), ad esempio `GRANT` e `REVOKE` o durante la creazione di nuove connessioni sull'istanza di scrittura.
- È stato risolto un problema a causa del quale la query parallela veniva utilizzata erroneamente per le operazioni DML (Data Manipulation Language), come le istruzioni `DELETE` e `UPDATE`, che al momento non sono supportate, che comportava il riavvio dell'istanza database. Per ulteriori informazioni sulle operazioni supportate nelle query parallele, consulta le limitazioni delle query parallele di Aurora MySQL.
- È stato risolto un problema che, in rari casi, poteva causare il riavvio delle repliche Aurora durante l'esecuzione simultanea di operazioni di aggiornamento di grandi dimensioni o carichi di lavoro DDL (Data Definition Language) sull'istanza di scrittura e operazioni di lettura sullo stesso set di tabelle sulla replica Aurora.

- È stato risolto un problema relativo all'operazione di riduzione dell'istanza di lettura Aurora Serverless v2 che poteva causare il riavvio dell'istanza di lettura e, in alcuni rari casi, causare incoerenza dei dati.
- È stato risolto un problema che poteva causare il riavvio dell'istanza database a causa di un accesso errato a una posizione di memoria non valida quando una connessione all'istanza database viene chiusa.
- È stato risolto un problema che, in rare condizioni, può causare il riavvio dell'istanza database durante l'elaborazione di una query con una clausola `GROUP BY` che tronca una colonna decimale a zero posizioni decimali.
- È stato risolto un problema che poteva causare il riavvio di un'istanza database a causa dell'accesso errato a un record durante l'esecuzione di una query di intervallo utilizzando un indice spaziale.
- È stato risolto un problema che può causare il riavvio dell'istanza database sulle istanze di replica di Aurora MySQL quando le tabelle temporanee interne superano i valori di memoria predefiniti o configurati dal cliente o i valori mmap.
- È stato risolto un problema per cui la rotazione dei log di audit avanzato poteva causare problemi di gestione della memoria.
- L'inserimento rapido non è abilitato in questa versione di Aurora MySQL, a causa di un problema che può causare incongruenze durante l'esecuzione di query come, e. `INSERT INTO SELECT FROM` Per ulteriori informazioni sull'ottimizzazione rapida degli inserti, consulta Miglioramenti delle prestazioni di Amazon Aurora MySQL.

Miglioramenti generali:

- È stata migliorata la latenza delle query di lettura delle sessioni di inoltro di scrittura del database globale utilizzando l'impostazione di coerenza di lettura GLOBAL.
- È stato risolto un problema per cui il valore del parametro `wait_timeout` non veniva rispettato dopo che una sessione client eseguiva i comandi `reset_connection` o `change_user`.
- È stato risolto un problema a causa del quale le applicazioni potevano registrare un aumento della latenza durante la connessione a un'istanza database quando l'istanza registrava un aumento improvviso delle connessioni in entrata. Sono state introdotte due nuove CloudWatch metriche, `AuroraSlowHandshakeCount` e`AuroraSlowConnectionHandleCount`, per aiutare a risolvere i ritardi nella creazione delle connessioni per le istanze DB Aurora MySQL. Ulteriori informazioni su questi parametri sono disponibili nella documentazione sulle definizioni dei parametri di CloudWatch Aurora, Amazon metrics for CloudWatch Amazon Aurora.

- Il parametro `temptable_use_mmap` è obsoleto e il relativo supporto verrà rimosso in una versione futura di MySQL. Per ulteriori informazioni, consulta Motore di storage per tabelle temporanee (implicite) interne.
- È stato risolto un problema che può causare tempi di esecuzione più lunghi del previsto per l'istruzione `SHOW BINARY LOGS`. Ciò potrebbe comportare una riduzione della velocità di trasmissione effettiva del commit del database.

Aggiornamenti e migrazioni:

- Per eseguire un aggiornamento secondario della versione per un database globale Aurora da Aurora MySQL versione 3.01 o 3.02 ad Aurora MySQL versione 3.03 o successiva, consulta Aggiornamento di Aurora MySQL modificando la versione del motore.
- È stato risolto un problema che poteva causare errori degli aggiornamenti delle versioni principali da Aurora MySQL versione 2 ad Aurora MySQL versione 3 in presenza di un numero elevato di tabelle (oltre 750.000) nel cluster.
- È stato risolto un problema che poteva causare errori degli aggiornamenti delle versioni principali da Aurora MySQL versione 2 ad Aurora MySQL versione 3 perché la migrazione delle tabelle `mysql.innodb_table_stats` e `mysql.innodb_index_stats` and richiedeva più tempo del previsto. Questo problema riguardava principalmente i cluster database con milioni di tabelle.
- È stato risolto un problema che poteva causare errori di aggiornamento da Aurora MySQL versione 2 ad Aurora MySQL versione 3 a causa di errori di incoerenza dello schema. Questi errori vengono segnalati dal controllo preliminare dell'aggiornamento per le tabelle `mysql.general_log_template` e `mysql.slow_log_template`. Per ulteriori informazioni sulla risoluzione dei problemi di aggiornamento, consulta Risoluzione dei problemi di aggiornamento con Aurora MySQL versione 3.
- È stato risolto un problema che può causare errori di aggiornamento da Aurora MySQL versione 2 ad Aurora MySQL versione 3 a causa dell'errore `schemaInconsistencyCheck`. Questo errore è causato da incongruenze dello schema all'interno della tabella `mysql.table_migration_index_info`, come segnalato da `upgrade-prechecks.log`. Per ulteriori informazioni sulla risoluzione dei problemi di aggiornamento ad Aurora MySQL versione 3, consulta Risoluzione dei problemi di aggiornamento con Aurora MySQL versione 3.

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla 8.0.26 inclusa, oltre a quanto riportato di seguito. Per ulteriori informazioni, consulta MySQL bugs fixed by Aurora MySQL 3.x database engine updates.

- È stato risolto un problema per cui alcuni tipi di colonne, tra cui `JSON` e `TEXT`, a volte esaurivano il buffer di ordinamento se la sua dimensione non era almeno 15 volte quella della riga più grande dell'ordinamento. Ora il buffer di ordinamento deve essere solo 15 volte più grande della chiave di ordinamento più grande. (Bug 103325, Bug 105532, Bug 32738705, Bug 33501541)
- Risolto un problema per cui InnoDB non gestiva sempre correttamente alcuni nomi legali per le partizioni di tabella. (Bug 32208630)
- È stato risolto un problema che, in determinate condizioni, poteva restituire risultati errati a causa di un calcolo impreciso della proprietà di nullabilità durante l'esecuzione di una query con una condizione `OR`. (Bug 34060289)
- È stato risolto un problema che, in determinate condizioni, poteva restituire risultati errati quando venivano soddisfatte le due condizioni seguenti:
  - una tabella derivata viene unita al blocco di query esterno
  - la query include un left join e una sottoquery `IN`

  (Bug 34060289)
- È stato risolto un problema a causa del quale venivano generati valori `AUTO_INCREMENT` errati quando veniva superato il valore massimo intero della colonna. L'errore era dovuto al fatto che il valore massimo della colonna non veniva considerato. In questo caso avrebbe dovuto essere restituito il precedente valore valido `AUTO_INCREMENT`, causando un errore di chiave duplicata. (Bug 87926, 26906787)
- È stato risolto un problema per cui non era possibile revocare il `DROP` privilegio sul Performance Schema. (Bug 33578113)
- È stato risolto un problema a causa del quale una procedura archiviata contenente un'istruzione `IF` che utilizzava `EXISTS`, che agiva su una o più tabelle eliminate e ricreate tra un'esecuzione e l'altra, non veniva eseguita correttamente per le invocazioni successive alla prima. (Bug 32855634).
- Risolto un problema per cui una query che fa riferimento a una vista in una sottoquery e in un blocco di query esterno può causare un riavvio imprevisto. (Bug 32324234)

Aggiornamenti del motore di database Aurora MySQL 2023-04-17 (versione 3.02.3) (obsoleta)

Versione: 3.02.3

Aurora MySQL 3.02.3 è disponibile a livello generale. Le versioni 3.02 di Aurora MySQL sono compatibili con MySQL 8.0.23, mentre le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7.

Per i dettagli sulle nuove funzionalità di Aurora MySQL versione 3 e sulle differenze tra Aurora MySQL versione 3 e Aurora MySQL versione 2 o la community MySQL 8.0, consulta Confronto tra Aurora MySQL versione 2 e Aurora MySQL versione 3 nell'utente Amazon Aurora. Guida.

Le versioni di Aurora MySQL attualmente supportate sono 2.07.*, 2.11.1, 2.11.2, 3.01.*, 3.02.* e 3.03.*.

Puoi effettuare un aggiornamento locale o ripristinare uno snapshot da un cluster della versione 2 di Aurora MySQL attualmente supportata in Aurora MySQL 3.02.3.

Per informazioni sulla pianificazione di un aggiornamento ad Aurora MySQL versione 3, consulta Pianificazione dell'aggiornamento per Aurora MySQL versione 3 nella Guida per l'utente di Amazon Aurora. Per la procedura di aggiornamento vera e propria, consulta Aggiornamento ad Aurora MySQL versione 3 nella Guida per l'utente di Amazon Aurora Per informazioni generali sugli aggiornamenti di Aurora MySQL, consulta Aggiornamento dei cluster database Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Per informazioni sulla risoluzione dei problemi, consulta Risoluzione dei problemi di aggiornamento con Aurora MySQL versione 3.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Miglioramenti della disponibilità:

- È stato risolto un problema che può causare il riavvio dell'istanza database a causa dell'accesso errato alla memoria non valida quando una connessione viene chiusa immediatamente dopo il completamento di una transazione.

- L'inserimento rapido non è abilitato in questa versione di Aurora MySQL, a causa di un problema che può causare incongruenze durante l'esecuzione di query come, e. `INSERT INTO SELECT FROM` Per ulteriori informazioni sull'ottimizzazione rapida degli inserti, consulta Miglioramenti delle prestazioni di Amazon Aurora MySQL.

Miglioramenti generali:

- È stato risolto un problema per cui i metodi di accesso alla scansione degli indici non supportati venivano presi in considerazione per le espressioni di tabella comuni (CTE) durante la materializzazione di tabelle temporanee intermedie, il che poteva portare a comportamenti indesiderati, tra cui riavvii del database o risultati di query errati. Questo problema è stato risolto evitando l'uso di tali metodi di accesso alla scansione degli indici non supportati sulle tabelle che utilizzano il motore di archiviazione. TempTable
- È stato risolto un problema che, in rari casi, può causare il riavvio di un'istanza di lettura di Aurora MySQL quando si accede a una tabella con grandi operazioni di aggiornamento o Data Definition Language (DDL) in esecuzione contemporaneamente sull'istanza di scrittura di Aurora MySQL.
- È stato risolto un problema che, in alcune situazioni, può causare il riavvio delle istanze del lettore Aurora quando tenta di leggere una pagina che non è più accessibile durante una stima dell'intervallo.
- È stato risolto un problema per cui le istanze database che utilizzavano la replica di log binari potevano registrare un aumento dell'utilizzo della CPU ed errori di connessione quando erano collegati più consumer di replica di log binari.
- È stato risolto un problema che poteva causare il riavvio di un'istanza del lettore Aurora MySQL durante l'esecuzione di una query che utilizza un piano di esecuzione di query parallele di Aurora.

Aggiornamenti del motore di database Aurora MySQL 2022-11-18 (versione 3.02.2) (obsoleto)

Versione: 3.02.2

Aurora MySQL 3.02.2 è disponibile a livello generale. Le versioni 3.02 di Aurora MySQL sono compatibili con MySQL 8.0.23, le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Per i dettagli sulle nuove funzionalità di Aurora MySQL versione 3 e le differenze tra Aurora MySQL versione 3 e Aurora MySQL versione 2 o la community MySQL 8.0, consulta Comparing Aurora MySQL versione 2 e Aurora MySQL versione 3 nella Amazon Aurora User Guide.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

È possibile ripristinare uno snapshot da un cluster della versione 2 di Aurora MySQL attualmente supportata in Aurora MySQL 3.02.2.

Per informazioni sulla pianificazione di un aggiornamento ad Aurora MySQL versione 3, consulta Pianificazione dell'aggiornamento per Aurora MySQL versione 3 nella Guida per l'utente di Amazon Aurora. Per la procedura di aggiornamento vera e propria, consulta Aggiornamento ad Aurora MySQL versione 3 nella Guida per l'utente di Amazon Aurora Per informazioni generali sugli aggiornamenti di Aurora MySQL, consulta Aggiornamento dei cluster database Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Per informazioni sulla risoluzione dei problemi, consulta Risoluzione dei problemi di aggiornamento con Aurora MySQL versione 3.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Aurora MySQL versione 3.02.2 è generalmente disponibile e compatibile con MySQL 8.0.23 della community.

Risolti i problemi di sicurezza ed CVEs elencati di seguito:

Correzioni e altri miglioramenti per ottimizzare la gestione in un ambiente gestito. Ulteriori correzioni CVE riportate di seguito:

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

- CVE-2022-21444

Miglioramenti della disponibilità:

- È stato risolto un problema che poteva causare il riavvio dell'istanza database a causa di un accesso errato alla memoria non valida quando una connessione all'istanza database viene chiusa in modo esplicito o implicito.
- È stato risolto un problema che poteva causare l'interruzione ripetuta dell'avvio del database su classi di istanza più grandi a causa dell'inizializzazione del pool di buffer che richiedeva più tempo del previsto.
- È stato risolto un problema che, in rare condizioni, può causare il riavvio dell'istanza database quando Aurora Serverless v2 tenta erroneamente di aggiornare la cache della tabella durante il dimensionamento.
- È stato risolto un problema che, in rare condizioni, può causare il riavvio del database durante l'elaborazione di una query con una clausola GROUP BY che tronca una colonna decimale a zero posizioni decimali.
- L'inserimento rapido non è abilitato in questa versione di Aurora MySQL, a causa di un problema che può causare incongruenze durante l'esecuzione di query come, e. `INSERT INTO SELECT FROM` Per ulteriori informazioni sull'ottimizzazione rapida degli inserti, consulta Miglioramenti delle prestazioni di Amazon Aurora MySQL.

Miglioramenti generali:

- È stato risolto un problema che può causare errori di aggiornamento da Aurora MySQL versione 2 (compatibile con MySQL 5.7) ad Aurora MySQL versione 3 (compatibile con MySQL 8.0) a causa di un'incoerenza dei metadati nella tabella mysql.host.
- Sono stati aggiunti miglioramenti delle prestazioni per ridurre i tempi di aggiornamento da Aurora MySQL versione 2 (compatibile con MySQL 5.7) ad Aurora MySQL versione 3 (compatibile con MySQL 8.0). Parallelizzando alcune fasi di aggiornamento, il tempo viene ulteriormente ridotto quando si utilizzano classi di istanza più grandi, ad esempio db.r6g.16xlarge o db.r5.24xlarge.
- È stato aggiunto il supporto per la visualizzazione di tutti gli errori durante l'aggiornamento da Aurora MySQL versione 2 (compatibile con MySQL 5.7) ad Aurora MySQL versione 3 (compatibile con MySQL 8.0) quando le versioni precedenti erano limitate a visualizzare solo 50 errori.
- È stato risolto un problema che, in rare condizioni, può causare errori dei contatori di incremento automatico dopo l'aggiornamento della versione principale da Aurora MySQL versione 2 (compatibile con MySQL 5.7) a Aurora MySQL versione 3 (compatibile con MySQL 8.0).

- È stato risolto un problema che poteva causare errori di aggiornamento delle versioni principali da Aurora MySQL versione 2 ad Aurora MySQL versione 3 perché la migrazione delle tabelle "mysql.innodb_table_stats" e "mysql.innodb_index_stats" richiedeva più tempo del previsto. Questo problema riguardava principalmente i cluster di database con un numero elevato di tabelle (>1,5 milioni).
- È stato risolto un problema che poteva causare errori di aggiornamento delle versioni principali da Aurora MySQL versione 2 ad Aurora MySQL versione 3 a causa di un difetto nel flusso di lavoro di aggiornamento del motore AMS 8.0, che causava l'accumulo dei record di log sul volume del cluster di archiviazione Aurora e interrompeva le normali operazioni di scrittura. Questo problema riguardava principalmente i cluster di database con un numero elevato di tabelle, circa >750.000.
- È stato risolto un problema che impediva alle istanze inattive di Aurora MySQL Serverless v2 di ridursi a 0,5 ACUs perché i thread di eliminazione di MySQL venivano mantenuti attivi in modo errato.
- È stato risolto un problema a causa del quale le applicazioni potevano registrare un aumento della latenza durante la connessione a un'istanza database quando l'istanza registrava un aumento improvviso delle connessioni in entrata.
- Sono state introdotte due nuove CloudWatch metriche Amazon per aiutare a risolvere i ritardi nella creazione della connessione per le istanze del database Aurora MySQL. Ulteriori informazioni sulle AuroraSlowHandshakeCount AuroraSlowConnectionHandleCount metriche sono disponibili nelle definizioni delle metriche CloudWatch Aurora.

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla 8.0.23 inclusa, oltre a quanto riportato di seguito. Per ulteriori informazioni, consulta MySQL bugs fixed by Aurora MySQL 3.x database engine updates.

- È stato risolto un problema che, in determinate condizioni, poteva restituire risultati errati a causa di un calcolo impreciso della proprietà di nullabilità durante l'esecuzione di una query con una condizione OR. (Bug 34060289)
- È stato risolto un problema che, in determinate condizioni, poteva restituire risultati errati quando venivano soddisfatte le due condizioni seguenti:
  - Una tabella derivata viene unita nel blocco di query esterno.
  - La query include un left join e una sottoquery IN. (Bug 34060289)

- È stato risolto un problema per cui non era possibile revocare il privilegio DROP sul Performance Schema. (Bug 33578113)
- È stato risolto un problema a causa del quale una procedura archiviata contenente un'istruzione IF che utilizzava EXISTS, che agiva su una o più tabelle eliminate e ricreate tra un'esecuzione e l'altra, non veniva eseguita correttamente per le invocazioni successive alla prima. (Bug MySQL 32855634).
- Sono stati generati valori AUTO_INCREMENT errati quando è stato superato il valore massimo intero della colonna. L'errore era dovuto al fatto che il valore massimo della colonna non veniva considerato. In questo caso avrebbe dovuto essere restituito il precedente valore AUTO_INCREMENT valido, causando un errore di chiave duplicato. (Bug 87926, 26906787)
- È stato risolto un problema che poteva causare un errore durante l'aggiornamento di un cluster di database Aurora MySQL versione 1 (compatibile con MySQL 5.6) contenente una tabella creata dall'utente con una determinata tabella. IDs L'assegnazione di queste tabelle IDs può causare conflitti nella tabella del dizionario dei dati IDs durante l'aggiornamento da Aurora MySQL versione 2 (compatibile con MySQL 5.7) ad Aurora MySQL versione 3 (compatibile con MySQL 8.0) (Bug #33919635)

Aggiornamenti del motore di database Aurora MySQL 2022-09-07 (versione 3.02.1) (obsoleto)

Versione: 3.02.1

Aurora MySQL 3.02.1 è disponibile a livello generale. Le versioni 3.02 di Aurora MySQL sono compatibili con MySQL 8.0.23, le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Per i dettagli sulle nuove funzionalità di Aurora MySQL versione 3 e le differenze tra Aurora MySQL versione 3 e Aurora MySQL versione 2 o la community MySQL 8.0, consulta Comparing Aurora MySQL versione 2 e Aurora MySQL versione 3 nella Amazon Aurora User Guide.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

Per informazioni sulla pianificazione di un aggiornamento ad Aurora MySQL versione 3, consulta Pianificazione dell'aggiornamento per Aurora MySQL versione 3 nella Guida per l'utente di Amazon Aurora. Per la procedura di aggiornamento vera e propria, consulta Aggiornamento ad Aurora MySQL

versione 3 nella Guida per l'utente di Amazon Aurora Per informazioni generali sugli aggiornamenti di Aurora MySQL, consulta Aggiornamento dei cluster database Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Per informazioni sulla risoluzione dei problemi, consulta Risoluzione dei problemi di aggiornamento con Aurora MySQL versione 3.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Aurora MySQL versione 3.02.1 è generalmente disponibile e compatibile con MySQL 8.0.23 della community.

Risolti i problemi di sicurezza ed CVEs elencati di seguito:

Correzioni e altri miglioramenti per ottimizzare la gestione in un ambiente gestito. Ulteriori correzioni CVE riportate di seguito:

- CVE-2022-0778

Miglioramenti della disponibilità:

- È stato risolto un problema che poteva causare errori di connessione e latenza elevata quando più repliche di log binario MySQL (binlog) sono collegate a un nodo di scrittura Aurora o quando c'è un gran numero di query simultanee di lunga durata in combinazione con un aumento di nuove richieste di connessione.
- È stato risolto un problema che causava un riavvio del database quando si attivava il controllo avanzato degli eventi `CONNECT`.
- È stato risolto un problema che può causare il riavvio del database sulle istanze di replica di lettura Aurora MySQL quando le tabelle temporanee interne esauriscono la dimensione allocata nella memoria e nei file mmap impostati come valore predefinito o configurato dal cliente.
- È stato risolto un problema che può causare un riavvio ripetuto di una replica di lettura durante operazioni DDL simultanee sulle procedure archiviate.
- L'inserimento rapido non è abilitato in questa versione di Aurora MySQL, a causa di un problema che può causare incongruenze durante l'esecuzione di query come, e. `INSERT INTO SELECT`

`FROM` Per ulteriori informazioni sull'ottimizzazione rapida degli inserti, consulta Miglioramenti delle prestazioni di Amazon Aurora MySQL.

Miglioramenti generali:

- Aggiunto il supporto per le istanze R6i.

Informazioni aggiuntive:

- Aurora MySQL versione 3.02.1 non supporta gli aggiornamenti a versioni principali direttamente da Aurora MySQL versione 2 (compatibile con MySQL 5.7). Per eseguire un aggiornamento a una versione principale di questa versione, esegui prima un aggiornamento della versione principale ad Aurora MySQL versione 3.02.0, quindi esegui un aggiornamento locale della versione secondaria ad Aurora MySQL versione 3.02.1.

Aggiornamenti del motore di database Aurora MySQL 2022-04-20 (versione 3.02.0) (obsoleto)

Versione: 3,02.0

Aurora MySQL 3.02.0 è disponibile a livello generale. Le versioni 3.02 di Aurora MySQL sono compatibili con MySQL 8.0.23, le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Per i dettagli sulle nuove funzionalità di Aurora MySQL versione 3 e le differenze tra Aurora MySQL versione 3 e Aurora MySQL versione 2 o la community MySQL 8.0, consulta Comparing Aurora MySQL versione 2 e Aurora MySQL versione 3 nella Amazon Aurora User Guide.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

È possibile ripristinare uno snapshot da un cluster della versione 2 di Aurora MySQL attualmente supportata in Aurora MySQL 3.02.0.

Per informazioni sulla pianificazione di un aggiornamento ad Aurora MySQL versione 3, consulta Pianificazione dell'aggiornamento per Aurora MySQL versione 3 nella Guida per l'utente di Amazon Aurora. Per la procedura di aggiornamento vera e propria, consulta Aggiornamento ad Aurora MySQL

versione 3 nella Guida per l'utente di Amazon Aurora Per informazioni generali sugli aggiornamenti di Aurora MySQL, consulta Aggiornamento dei cluster database Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Per informazioni sulla risoluzione dei problemi, consulta Risoluzione dei problemi di aggiornamento con Aurora MySQL versione 3.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Aurora MySQL versione 3.02.0 è generalmente disponibile e compatibile con MySQL 8.0.23 della community.

Risolti i problemi di sicurezza ed CVEs elencati di seguito:

Correzioni e altri miglioramenti per ottimizzare la gestione in un ambiente gestito. Ulteriori correzioni CVE riportate di seguito:

- CVE-2021-22946

Nuove caratteristiche:

- Amazon Aurora Serverless v2 2.07.2 è disponibile a livello generale. Per ulteriori informazioni, consulta la panoramica di Amazon Aurora serverless, il blog e la documentazione sull'utilizzo di Aurora Serverless v2. Inizia oggi stesso creando un database Aurora Serverless v2 utilizzando solo pochi passaggi nella Console di gestione AWS.

Miglioramenti della disponibilità:

- È stato risolto un problema a causa del quale il server poteva entrare in un ciclo di riavvio e non essere disponibile durante l'eliminazione di un record o di una tabella contenente due o più colonne a lunghezza variabile (tipi VARCHAR, VARBINARY, BLOB e TEXT). Per ulteriori dettagli sui tipi di colonna, consulta innodb-row-format.
- È stato risolto un problema a causa del quale non era possibile stabilire il timeout delle connessioni esistenti e stabilire nuove connessioni su un cluster con il log binario attivato e con almeno un consumer di log binario collegato che causava un conflitto di risorse tra l'applicazione e i consumer.

- La memoria liberabile è indicata dalla `FreeableMemory` CloudWatch metrica. Per ulteriori informazioni, consulta i CloudWatch parametri Amazon per Amazon Aurora.
  - È stato risolto un problema che può causare il riavvio di un'istanza database o un failover a causa di una diminuzione della memoria liberabile quando è abilitato il log binario avanzato.
  - È stato risolto un problema che può causare il riavvio di un'istanza database o un failover a causa di una diminuzione della memoria liberabile quando è abilitato il log binario avanzato.
  - È stato risolto un problema che può causare il riavvio di un'istanza database o un failover a causa di una diminuzione della memoria liberabile quando è abilitato il log binario avanzato.
- È stato risolto un problema che, in rare condizioni, può causare un errore di immissione duplicata durante l'inserimento di nuove righe in una tabella contenente una colonna `AUTO_INCREMENT` su un cluster ripristinato da uno snapshot.
- L'inserimento rapido non è abilitato in questa versione di Aurora MySQL, a causa di un problema che può causare incongruenze durante l'esecuzione di query come, e. `INSERT INTO SELECT FROM` Per ulteriori informazioni sull'ottimizzazione rapida degli inserti, consulta Miglioramenti delle prestazioni di Amazon Aurora MySQL.

Miglioramenti generali:

- È stato risolto un problema per cui lo stato del volume non veniva visualizzato quando si utilizzava il comando `SHOW VOLUME STATUS`. Per ulteriori informazioni, consulta SQL.Managing. AuroraMy VolumeStatus.
- È stato risolto un problema che causava un errore delle chiamate a mysql_rds_import_binlog_ssl_material con MySQL server ERROR 3512.
- È stato risolto un problema a causa del quale il ritardo della replica di Aurora veniva segnalato erroneamente per le istanze del lettore Aurora eliminate.

Aggiornamenti/migrazioni:

- È stato risolto un problema che poteva causare errori di migrazione dei database MySQL 8.0.x verso Aurora MySQL versione 3 a causa di un problema nella copia di file ibdata e tablespace nell'archiviazione di Aurora.
- È stato risolto un problema che poteva causare errori di aggiornamento dei cluster da Aurora MySQL versione 2 a Aurora MySQL versione 3 quando le tabelle del database contenevano una grande quantità di dati.

- È stato risolto un problema che poteva causare errori durante il ripristino dei cluster da Aurora MySQL versione 2 ad Aurora MySQL versione 3 a causa di un errore nella creazione di informazioni SDI (Serialized Data Dictionary Information) per una tabella.
- È stato risolto un problema che poteva causare errori di aggiornamento da Aurora MySQL versione 2 ad Aurora MySQL versione 3 a causa di errori di incoerenza dello schema segnalati nei controlli preliminari delle tabelle di sistema RDS.
- È stato risolto un problema che poteva causare errori durante la migrazione o il ripristino dai database RDS per MySQL 8.0 o Aurora MySQL versione 2 ai database Aurora MySQL versione 3 causa di una sintassi non valida in una procedura archiviata gestita da RDS.
- È stato risolto un problema che poteva causare errori di aggiornamento da Aurora MySQL 2 ad Aurora MySQL 3 a causa di errori di incoerenza dello schema segnalati dai controlli preliminari dell'aggiornamento per le tabelle di log generale e di log lento.

Integrazione delle correzioni di bug della community di MySQL

Questa versione include tutte le correzioni di bug della community fino alla 8.0.23 inclusa, oltre a quanto riportato di seguito. Per ulteriori informazioni, consulta MySQL bugs fixed by Aurora MySQL 3.x database engine updates.

- Risolta la gestione impropria delle tabelle temporanee utilizzate per i cursori all'interno delle procedure archiviate che poteva causare un comportamento imprevisto del server, mysqld-8-0-24-bug. (Bug 32416811)

Aggiornamenti del motore di database Aurora MySQL 2022-04-15 (versione 3.01.1) (obsoleto)

Versione: 3.01.1

Aurora MySQL 3.01.1 è disponibile a livello generale. Le versioni 3.01 di Aurora MySQL sono compatibili con MySQL 8.0.23, le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Per i dettagli sulle nuove funzionalità di Aurora MySQL versione 3 e le differenze tra Aurora MySQL versione 3 e Aurora MySQL versione 2 o la community MySQL 8.0, consulta Comparing Aurora MySQL versione 2 e Aurora MySQL versione 3 nella Amazon Aurora User Guide.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

È possibile ripristinare uno snapshot da un cluster della versione 2 di Aurora MySQL attualmente supportata in Aurora MySQL 3.01.1.

Per informazioni sulla pianificazione di un aggiornamento ad Aurora MySQL versione 3, consulta Pianificazione dell'aggiornamento per Aurora MySQL versione 3 nella Guida per l'utente di Amazon Aurora. Per la procedura di aggiornamento vera e propria, consulta Aggiornamento ad Aurora MySQL versione 3 nella Guida per l'utente di Amazon Aurora Per informazioni generali sugli aggiornamenti di Aurora MySQL, consulta Aggiornamento dei cluster database Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Per informazioni sulla risoluzione dei problemi, consulta Risoluzione dei problemi di aggiornamento con Aurora MySQL versione 3.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Aurora MySQL versione 3.01.1 è generalmente disponibile e compatibile con MySQL 8.0.23 della community.

Aurora MySQL versione 3.01.1 è consigliata per aggiornamenti e migrazioni a un database di Aurora compatibile con MySQL 8.0.

Risolti i problemi di sicurezza ed CVEs elencati di seguito:

Correzioni e altri miglioramenti per ottimizzare la gestione in un ambiente gestito. Ulteriori correzioni CVE riportate di seguito:

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

Miglioramenti della disponibilità:

- La memoria liberabile è indicata dalla `FreeableMemory` CloudWatch metrica. Per ulteriori informazioni, consulta i CloudWatch parametri Amazon per Amazon Aurora.
  - È stato risolto un problema che può causare il riavvio di un'istanza database o un failover a causa di una diminuzione della memoria liberabile quando è abilitato il log binario avanzato.
  - È stato risolto un problema che può causare il riavvio di un'istanza database o un failover a causa di una diminuzione della memoria liberabile quando è abilitato il log binario avanzato.
  - È stato risolto un problema che può causare il riavvio di un'istanza database o un failover a causa di una diminuzione della memoria liberabile quando è abilitato il log binario avanzato.
- È stato risolto un problema che, in rare condizioni, può causare un errore di immissione duplicata durante l'inserimento di nuove righe in una tabella contenente una colonna `AUTO_INCREMENT` su un cluster ripristinato da uno snapshot.
- L'inserimento rapido non è abilitato in questa versione di Aurora MySQL, a causa di un problema che può causare incongruenze durante l'esecuzione di query come, e. `INSERT INTO SELECT FROM` Per ulteriori informazioni sull'ottimizzazione rapida degli inserti, consulta Miglioramenti delle prestazioni di Amazon Aurora MySQL.

Miglioramenti generali:

- È stato risolto un problema per cui lo stato del volume non veniva visualizzato quando si utilizzava il comando `SHOW VOLUME STATUS`. Per ulteriori informazioni, consulta SQL.Managing. AuroraMy VolumeStatus.
- È stato risolto un problema che causava un errore delle chiamate a mysql_rds_import_binlog_ssl_material con MySQL server ERROR 3512.
- È stato risolto un problema a causa del quale il ritardo della replica di Aurora veniva segnalato erroneamente per le istanze del lettore Aurora eliminate.

Aggiornamenti/migrazioni:

- È stato risolto un problema che poteva causare errori di migrazione dei database MySQL 8.0.x verso Aurora MySQL versione 3 a causa di un problema nella copia di file ibdata e tablespace nell'archiviazione di Aurora.
- È stato risolto un problema che poteva causare errori di aggiornamento dei cluster da Aurora MySQL versione 2 a Aurora MySQL versione 3 quando le tabelle del database contenevano una grande quantità di dati.

- È stato risolto un problema che poteva causare errori durante il ripristino dei cluster da Aurora MySQL versione 2 ad Aurora MySQL versione 3 a causa di un errore nella creazione di informazioni SDI (Serialized Data Dictionary Information) per una tabella.
- È stato risolto un problema che poteva causare errori di aggiornamento da Aurora MySQL versione 2 ad Aurora MySQL versione 3 a causa di errori di incoerenza dello schema segnalati nei controlli preliminari delle tabelle di sistema RDS.
- È stato risolto un problema che poteva causare errori durante la migrazione o il ripristino dai database RDS per MySQL 8.0 o Aurora MySQL versione 2 ai database Aurora MySQL versione 3 causa di una sintassi non valida in una procedura archiviata gestita da RDS.
- È stato risolto un problema che poteva causare errori di aggiornamento da Aurora MySQL 2 ad Aurora MySQL 3 a causa di errori di incoerenza dello schema segnalati dai controlli preliminari dell'aggiornamento per le tabelle di log generale e di log lento.

Integrazione delle correzioni di bug della community di MySQL

Questa versione include tutte le correzioni di bug della community fino alla 8.0.23 inclusa, oltre a quanto riportato di seguito. Per ulteriori informazioni, consulta MySQL bugs fixed by Aurora MySQL 3.x database engine updates.

- Risolta la gestione impropria delle tabelle temporanee utilizzate per i cursori all'interno delle procedure archiviate che poteva causare un comportamento imprevisto del server, mysqld-8-0-24-bug. (Bug 32416811)

Aggiornamenti del motore di database Aurora MySQL 2021-11-18 (versione 3.01.0) (obsoleto)

Versione: 3.01.0

Aurora MySQL 3.01.0 è disponibile a livello generale. Le versioni 3.01 di Aurora MySQL sono compatibili con MySQL 8.0.23, le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Per i dettagli sulle nuove funzionalità di Aurora MySQL versione 3 e le differenze tra Aurora MySQL versione 3 e Aurora MySQL versione 2 o la community MySQL 8.0, consulta Comparing Aurora MySQL versione 2 e Aurora MySQL versione 3 nella Amazon Aurora User Guide.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

È possibile ripristinare uno snapshot da un cluster della versione 2 di Aurora MySQL attualmente supportata in Aurora MySQL 3.01.0.

Per informazioni sulla pianificazione di un aggiornamento ad Aurora MySQL versione 3, consulta Pianificazione dell'aggiornamento per Aurora MySQL versione 3 nella Guida per l'utente di Amazon Aurora. Per la procedura di aggiornamento vera e propria, consulta Aggiornamento ad Aurora MySQL versione 3 nella Guida per l'utente di Amazon Aurora Per informazioni generali sugli aggiornamenti di Aurora MySQL, consulta Aggiornamento dei cluster database Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Per informazioni sulla risoluzione dei problemi, consulta Risoluzione dei problemi di aggiornamento con Aurora MySQL versione 3.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Aurora MySQL versione 3.01.0 è generalmente compatibile con MySQL 8.0.23 della community. Questa versione include le correzioni di sicurezza per i problemi di vulnerabilità ed esposizione comuni (CVE) a partire dalla community di MySQL 8.0.23.

Aurora MySQL versione 3.01.0 contiene tutte le correzioni di bug specifiche di Aurora tramite Aurora MySQL versione 2.10.0.

Per i dettagli sulle nuove funzionalità di Aurora MySQL versione 3, consulta Funzionalità della community MySQL 8.0 e Nuove ottimizzazioni delle query parallele nella Guida per l'utente di Amazon Aurora.

Miglioramenti della disponibilità:

- L'inserimento rapido non è abilitato in questa versione di Aurora MySQL, a causa di un problema che può causare incongruenze durante l'esecuzione di query come, e. `INSERT INTO SELECT FROM` Per ulteriori informazioni sull'ottimizzazione rapida degli inserti, consulta Miglioramenti delle prestazioni di Amazon Aurora MySQL.

Aggiornamenti del motore del database Amazon Aurora MySQL versione 2

> **Important**
>
> Aurora MySQL versione 2 ha raggiunto la fine del supporto standard il 31 ottobre 2024. Per ulteriori informazioni, consulta Preparazione per la fine del supporto standard dell'edizione 2 compatibile con Amazon Aurora MySQL.

Di seguito sono riportati alcuni aggiornamenti del motore del database per Amazon Aurora MySQL versione 2.

- Aggiornamenti del motore di database Aurora MySQL 2025-04-09 (versione 2.12.5, compatibile con MySQL 5.7.44) - Versione RDS Extended Support
- Aggiornamenti del motore di database Aurora MySQL 2024-10-23 (versione 2.12.4, compatibile con MySQL 5.7.44) - Versione RDS Extended Support
- Aggiornamenti del motore di database Aurora MySQL 2024-07-09 (versione 2.12.3, compatibile con MySQL 5.7.44) - Versione RDS Extended Support
- Aggiornamenti del motore di database Aurora MySQL 2024-03-19 (versione 2.12.2, compatibile con MySQL 5.7.44) - Versione RDS Extended Support
- Aggiornamenti del motore di database Aurora MySQL 2023-12-28 (versione 2.12.1, compatibile con MySQL 5.7.40) - Versione RDS Extended Support
- Aggiornamenti del motore di database Aurora MySQL 2023-10-25 (versione 2.12.0.1, compatibile con MySQL 5.7.40) - Versione RDS Extended Support (Beta)
- Aggiornamenti del motore di database Aurora MySQL 2023-07-25 (versione 2.12.0, compatibile con MySQL 5.7.40) - Versione RDS Extended Support
- Aggiornamenti del motore di database Aurora MySQL 2024-07-19 (versione 2.11.6, compatibile con MySQL 5.7.12) - Versione RDS Extended Support
- Aggiornamenti del motore di database Aurora MySQL 2024-03-26 (versione 2.11.5, compatibile con MySQL 5.7.12) - Versione RDS Extended Support (predefinita)
- Aggiornamenti del motore di database Aurora MySQL 2023-10-17 (versione 2.11.4, compatibile con MySQL 5.7.12) - Versione RDS Extended Support

- Aggiornamenti del motore di database Aurora MySQL 2023-06-09 (versione 2.11.3, compatibile con MySQL 5.7.12) - Versione RDS Extended Support
- Aggiornamenti del motore di database Aurora MySQL 2023-03-24 (versione 2.11.2, compatibile con MySQL 5.7.12) - Versione RDS Extended Support
- Aggiornamenti del motore di database Aurora MySQL 2023-02-14 (versione 2.11.1, compatibile con MySQL 5.7.12) - Versione RDS Extended Support
- Aggiornamenti del motore di database Aurora MySQL 2022-10-25 (versione 2.11.0, compatibile con MySQL 5.7.12) - Versione RDS Extended Support
- Aggiornamenti del motore di database Aurora MySQL 01/11/2022 (versione 2.10.3) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 26/01/2022 (versione 2.10.2) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 21/10/2021 (versione 2.10.1) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 25/05/2021 (versione 2.10.0) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 12/11/2021 (versione 2.09.3) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 26/02/2021 (versione 2.09.2) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 11/12/2020 (versione 2.09.1) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 17/09/2020 (versione 2.09.0) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 06/01/2022 (versione 2.08.4) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 12/11/2020 (versione 2.08.3) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 28/08/2020 (versione 2.08.2) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 18/06/2020 (versione 2.08.1) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 02/06/2020 (versione 2.08.0) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 2023-08-15 (versione 2.07.10) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 2023-05-04 (versione 2.07.9) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 16/06/2022 (versione 2.07.8) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 24/11/2021 (versione 2.07.7) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 02/09/2021 (versione 2.07.6) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 06/07/2021 (versione 2.07.5) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 03/04/2021 (versione 2.07.4) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 10/11/2020 (versione 2.07.3) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 17/04/2020 (versione 2.07.2) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 23/12/2019 (versione 2.07.1) (obsoleta)

- Aggiornamenti del motore di database Aurora MySQL 25/11/2019 (versione 2.07.0) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 22/11/2019 (versione 2.06.0) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 11/11/2019 (versione 2.05.0) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 14/08/2020 (versione 2.04.9) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 20/11/2019 (versione 2.04.8) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 14/11/2019 (versione 2.04.7) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 19/09/2019 (versione 2.04.6) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 08/07/2019 (versione 2.04.5) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 29/05/2019 (versione 2.04.4) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 09/05/2019 (versione 2.04.3) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 02/05/2019 (versione 2.04.2) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 25/03/2019 (versione 2.04.1) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 25/03/2019 (versione 2.04.0) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 07/02/2019 (versione 2.03.4) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 18/01/2019 (versione 2.03.3) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 09/01/2019 (versione 2.03.2) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 24/10/2018 (versione 2.03.1) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 11/10/2018 (versione 2.03) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 08/10/2018 (versione 2.02.5) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 21/09/2018 (versione 2.02.4) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 23/08/2018 (versione 2.02.3) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 04/06/2018 (versione 2.02.2) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 03/05/2018 (versione 2.02) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 13/03/2018 (versione 2.01.1) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 06/02/2018 (versione 2.01) (obsoleta)

Aggiornamenti del motore di database Aurora MySQL 2025-04-09 (versione 2.12.5, compatibile con MySQL 5.7.44) - Versione RDS Extended Support

Versione: 2.12.5

Aurora MySQL 2.12.5 è disponibile a livello generale. Le versioni di Aurora MySQL 2.12 sono compatibili fino a MySQL 5.7.44. Per ulteriori informazioni sulle modifiche alla community, vedere Modifiche in MySQL 5.7.44 (2022-10-11, disponibilità generale).

Le versioni di Aurora MySQL attualmente supportate sono 2.11.*, 2.12.*, 3.04.*, 3.05.*, 3.06.*, 3.07.* e 3.08*.

È possibile aggiornare un cluster di database Aurora MySQL 2.* esistente ad Aurora MySQL 2.12.5. È inoltre possibile ripristinare un'istantanea da qualsiasi versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.12.5.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Risolti i problemi di sicurezza e CVEs:

Questa versione include tutte le correzioni CVE della community fino a MySQL 5.7.44 incluso. Sono incluse le seguenti correzioni CVE:

- 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

Miglioramenti della disponibilità:

- È stato risolto un problema sulla replica a causa del quale un'interruzione della rete poteva non ristabilire correttamente la connessione con il writer.

Aggiornamenti del motore di database Aurora MySQL 2024-10-23 (versione 2.12.4, compatibile con MySQL 5.7.44) - Versione RDS Extended Support

Versione: 2.12.4

Aurora MySQL 2.12.4 è disponibile a livello generale. Le versioni di Aurora MySQL 2.12 sono compatibili fino a MySQL 5.7.44. Per ulteriori informazioni sulle modifiche alla community, vedere Modifiche in MySQL 5.7.44 (2022-10-11, disponibilità generale).

Le versioni di Aurora MySQL attualmente supportate sono 2.11.*, 2.12.*, 3.04.*, 3.05.*, 3.06.* e 3.07.*.

È possibile aggiornare un cluster di database Aurora MySQL 2.* esistente ad Aurora MySQL 2.12.4. È inoltre possibile ripristinare un'istantanea da qualsiasi versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.12.4.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Risolti i problemi di sicurezza e CVEs:

Questa versione include tutte le correzioni CVE della community fino a MySQL 5.7.44 incluso. Sono incluse le seguenti correzioni CVE:

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

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla versione 5.7.44 inclusa. Per ulteriori informazioni, consulta Correzione dei bug di MySQL attraverso gli aggiornamenti del motore di database Aurora MySQL 2.x.

Funzionalità non supportate in Aurora MySQL versione 2

Le seguenti funzionalità non sono attualmente supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Raggruppamento delle scansioni

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa

degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Istruzione SQL `CREATE TABLESPACE`
- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Protocollo X

Aggiornamenti del motore di database Aurora MySQL 2024-07-09 (versione 2.12.3, compatibile con MySQL 5.7.44) - Versione RDS Extended Support

Versione: 2.12.3

Aurora MySQL 2.12.3 è disponibile a livello generale. Le versioni di Aurora MySQL 2.12 sono compatibili fino a MySQL 5.7.44. Per ulteriori informazioni sulle modifiche alla community, vedere Modifiche in MySQL 5.7.44 (2022-10-11, disponibilità generale).

Le versioni di Aurora MySQL attualmente supportate sono 2.11.*, 2.12.*, 3.03.*, 3.04.*, 3.05.*, 3.06.* e 3.07.*.

È possibile aggiornare un cluster di database Aurora MySQL 2.* esistente ad Aurora MySQL 2.12.3. È inoltre possibile ripristinare un'istantanea da qualsiasi versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.12.3.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Risolti i problemi di sicurezza e CVEs:

- È stato risolto un problema di sicurezza per le stored procedure MySQL.

Questa versione include tutte le correzioni CVE della community fino a MySQL 5.7.44 incluso. Sono incluse le seguenti correzioni CVE:

- CVE-2023-21912
- CVE-2023-44487
- CVE-2024-0853
- CVE-2024-20993
- 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

Miglioramenti della disponibilità:

- È stato risolto un problema che causava il riavvio di un'istanza DB Aurora MySQL durante l'esecuzione di una query parallela.
- È stato risolto un problema che poteva causare il riavvio di un server di database a causa dell'accesso simultaneo alle risorse di connessione durante la scalabilità senza interruzioni, il riavvio senza tempi di inattività (ZDR) e l'applicazione di patch senza tempi di inattività (ZDP).
- È stato risolto un problema che poteva causare il riavvio di un'istanza Reader DB durante la liberazione della memoria utilizzata per l'applicazione di registro.
- È stato risolto un problema in un processo in background che causava l'esecuzione prolungata o non riuscita delle query mentre l'operazione in background eliminava gli indici temporanei.
- È stato risolto un problema relativo alle routine di avvio che poteva causare il riavvio delle istanze DB di Writer a causa dell'incoerenza dei metadati.
- È stato aggiunto un indicatore per l'avanzamento del recupero delle transazioni. In questo modo si evita una potenziale indisponibilità, in rare situazioni in cui il completamento del ripristino delle transazioni richiede molto tempo.
- È stato risolto un problema che poteva causare il riavvio di un'istanza Reader DB durante la lettura di una tabella che veniva alterata o eliminata sull'istanza DB di Writer.
- È stato risolto un problema a causa del quale un valore di `thread_stack` parametro basso causava il riavvio ripetuto del database. Il `thread_stack` valore minimo consentito è stato aumentato da 131.072 a 136.192 per garantire il corretto avvio e prevenire problemi di avvio.
- È stato risolto un problema che causava il riavvio di un'istanza Reader DB durante l'esecuzione di una query parallela.
- È stato risolto un problema che poteva causare il riavvio delle repliche di lettura di Aurora in caso di alcuni rari ordini di commit delle transazioni sull'istanza Writer DB.
- È stato risolto un problema che, in rari casi, può causare il riavvio di un'istanza DB quando una transazione di sola lettura ottiene blocchi condivisi.
- È stato risolto un problema che poteva causare il riavvio di un'istanza Reader DB che utilizza l'inoltro di scrittura quando un'istruzione di commit implicita inoltrata rileva un errore.

Miglioramenti generali:

- È stato risolto un problema che può causare errori o avvisi imprevisti di violazione della chiave primaria nelle istruzioni SQL su alcune righe quando si eseguono `INSERT` istruzioni simultanee su una tabella con una colonna chiave `AUTO_INCREMENT` primaria e una colonna chiave univoca e quando un'`INSERT`istruzione presenta violazioni di chiave univoche su righe diverse.
- È stato risolto un problema che poteva causare risultati di query errati quando ZDR ripristinava erroneamente le variabili di sessione impostate come suggerimenti nelle query.
- È stato risolto un problema nella query parallela che causava la restituzione di un set di risultati incompleto quando si utilizzavano le funzioni built-in `LPAD` e `RPAD` string.
- È stato risolto un problema che causava la mancanza di indici di chiave esterna nelle istanze Reader DB quando un'`ALTER TABLE RENAME COLUMN`istruzione veniva eseguita sull'istanza Writer DB su una tabella con una chiave esterna.
- È stato risolto un problema che poteva causare il mancato completamento del processo di disabilitazione dell'inoltro di scrittura.
- È stato risolto un problema durante il Aurora Serverless v1 ridimensionamento che causava il riavvio dell'istanza DB a causa di un accesso errato a una struttura dati interna durante la ricerca di un punto di scalabilità.
- È stato risolto un problema per cui lo schema delle prestazioni non era abilitato quando la gestione automatizzata di Performance Insights era attivata per le istanze DB db.t4g.medium e db.t4g.large.
- Il timeout delle richieste per le operazioni di machine learning di Aurora verso SageMaker Amazon AI è stato aumentato da 3 a 30 secondi. Questo aiuta a risolvere un problema in cui i clienti potrebbero riscontrare un numero maggiore di tentativi o errori per le richieste ad Amazon SageMaker AI da Aurora Machine Learning quando utilizzano batch di dimensioni maggiori.
- È stato risolto un problema per cui `UPDATE` le query lente `INSERT` ed eseguite da MySQL Event Scheduler non venivano registrate nel registro delle query lente a meno che non fossero precedute da una query lenta. `DELETE SELECT`

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla 5.7.44 inclusa. Per ulteriori informazioni, consulta Correzione dei bug di MySQL attraverso gli aggiornamenti del motore di database Aurora MySQL 2.x.

- È stato risolto un problema per cui le tabelle temporanee associate ai trigger durante l'esecuzione delle istruzioni potevano causare un riavvio imprevisto del motore DB.

- È stato risolto un difetto che poteva causare la chiusura del server quando una singola tabella `UPDATE` e `DELETE` le istruzioni che utilizzano espressioni indicizzate venivano eseguite come istruzioni preparate. (Bug #29257254)

Funzionalità non supportate in Aurora MySQL versione 2

Le seguenti funzionalità non sono attualmente supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Raggruppamento delle scansioni

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Istruzione SQL `CREATE TABLESPACE`
- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Protocollo X

Aggiornamenti del motore di database Aurora MySQL 2024-03-19 (versione 2.12.2, compatibile con MySQL 5.7.44) - Versione RDS Extended Support

Versione: 2.12.2

Aurora MySQL 2.12.2 è disponibile a livello generale. Le versioni di Aurora MySQL 2.12 sono compatibili fino a MySQL 5.7.44. Per ulteriori informazioni sulle modifiche alla community, vedere Modifiche in MySQL 5.7.44 (2022-10-11, disponibilità generale).

Le versioni di Aurora MySQL attualmente supportate sono 2.11.*, 2.12.*, 3.03.*, 3.04.*, 3.05.* e 3.06.*.

È possibile aggiornare un cluster di database Aurora MySQL 2.* esistente ad Aurora MySQL 2.12.2. È inoltre possibile ripristinare un'istantanea da qualsiasi versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.12.2.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Risolti i problemi di sicurezza ed CVEs elencati di seguito:

Questa versione include tutte le correzioni CVE della community fino a MySQL 5.7.44 incluso. Sono incluse le seguenti correzioni CVE:

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

Problemi di sicurezza:

- È stata aggiunta una correzione che garantisce l'utilizzo predefinito delle repliche dei log binari SSL/TLS se l'origine supporta connessioni crittografate, indipendentemente dall'`MASTER_SSL`impostazione.

Miglioramenti della disponibilità:

- È stato risolto un problema che poteva impedire l'avvio corretto di un'istanza di replica di lettura in caso di carico di lavoro elevato sull'istanza writer.
- È stato risolto un problema che poteva causare il failover di un'istanza di Aurora MySQL Database Writer a causa di un difetto nel componente che comunica con lo storage Aurora. Il difetto si verifica a seguito di un'interruzione della comunicazione tra l'istanza di database e lo storage sottostante a seguito di un aggiornamento software dell'istanza di archiviazione Aurora.
- È stato risolto un problema che, in rare condizioni, può causare il riavvio delle istanze del lettore.
- È stato risolto un problema a causa del quale un utente privilegiato poteva modificare i limiti delle risorse associati all'utente rdsadmin. Se impostati in modo errato, questi limiti di risorse possono impedire all'agente di monitoraggio RDS di monitorare lo stato dell'istanza del database, con conseguente indisponibilità del database.

Aggiornamenti e migrazioni:

- È stato risolto un problema che si verificava durante il tentativo di avviare la replica dei log binari per i cluster Aurora MySQL migrati da Amazon RDS MySQL 5.7 e che contenevano procedure memorizzate non supportate.
- L'utilità di pianificazione degli eventi del database è stata disabilitata durante un aggiornamento della versione principale ad Aurora MySQL versione 3. Questo aggiornamento consente di evitare modifiche al database mediante l'esecuzione dell'evento mentre è in corso l'aggiornamento della versione principale.

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla versione 5.7.44 inclusa. Per ulteriori informazioni, consulta MySQL bugs fixed by Aurora MySQL 2.x database engine updates.

Funzionalità non supportate in Aurora MySQL versione 2

Le seguenti funzionalità non sono attualmente supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Raggruppamento delle scansioni

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Istruzione SQL `CREATE TABLESPACE`
- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Protocollo X

Aggiornamenti del motore di database Aurora MySQL 2023-12-28 (versione 2.12.1, compatibile con MySQL 5.7.40) - Versione RDS Extended Support

Versione: 2.12.1

Aurora MySQL 2.12.1 è disponibile a livello generale. Le versioni di Aurora MySQL 2.12 sono compatibili fino a MySQL 5.7.40. Per ulteriori informazioni sulle modifiche apportate alla community, consulta Changes in MySQL 5.7.40 (2022-10-11, General Availability).

Le versioni di Aurora MySQL attualmente supportate sono 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.*, 3.04.* e 3.05.*.

Puoi aggiornare un cluster di database Aurora MySQL 2.* esistente ad Aurora MySQL 2.12.1. Puoi anche ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.12.1.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Risolti i problemi di sicurezza ed CVEs elencati di seguito:

Questa versione include tutte le CVEs correzioni della community fino a MySQL 5.7.44 incluso.

Correzioni e altri miglioramenti per ottimizzare la gestione in un ambiente gestito. Ulteriori correzioni CVE sono riportate di seguito:

- 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
- Corretta l'elaborazione di token a carattere singolo da parte di un plug-in parser di ricerca full-text (FTS) (Bug 35432973)
- È stato risolto un problema per cui gli eventi segnalati durante l'elaborazione delle rotazioni del registro di controllo potevano non essere scritti nel registro di controllo

Nuove caratteristiche:

- È stato aggiunto il supporto per la replica di log binario multi-thread (binlog), in cui il thread SQL sulla replica binlog applicava gli eventi di log binario in parallelo quando possibile. Ulteriori informazioni sulle opzioni di configurazione per aiutare a ottimizzare la replica multi-thread nella Guida per l'utente di Aurora.

Miglioramenti della disponibilità:

- È stato risolto un problema per cui le istanze database di Aurora MySQL che utilizzavano query parallele potevano subire un riavvio del database durante l'esecuzione di un numero elevato di query parallele simultanee.
- È stato risolto un problema di conflitto di blocchi causato da un thread di log di audit che poteva portare a un utilizzo elevato della CPU e al timeout delle applicazioni client.
- È stato risolto un problema che può causare il riavvio di un'istanza database quando si tenta di leggere una pagina del database che appartiene a una tabella eliminata.
- È stato risolto un problema che può causare il riavvio dell'istanza di lettura quando l'istanza di scrittura aumenta il volume del database fino a un multiplo di 160 GB.
- È stato risolto un problema in gestore blocchi che poteva causare un riavvio o un failover durante la gestione dei commit in due fasi con il livello di isolamento impostato su `READ_COMMITED` o `READ_UNCOMMITED` e venivano utilizzate transazioni XA o era abilitato il log binario (binlog).
- È stato risolto un problema che poteva causare la non disponibilità del cluster di database al riavvio dell'istanza di scrittura mentre il database stava creando o eliminando i trigger sulle tabelle di sistema interne.
- È stato risolto un problema che poteva causare il riavvio dell'istanza database quando il numero di connessioni al database si avvicina al valore impostato dal parametro `max_connections`.

- È stato risolto un problema che poteva causare il riavvio di un'istanza di lettura di Aurora durante l'esecuzione di query DML (Data Manipulation Language) su una tabella contenente un indice full-text.
- L'inserimento rapido non è abilitato in questa versione di Aurora MySQL, a causa di un problema che può causare incongruenze durante l'esecuzione di query come, e. `INSERT INTO SELECT FROM` Per ulteriori informazioni sull'ottimizzazione rapida degli inserti, consulta Miglioramenti delle prestazioni di Amazon Aurora MySQL.

Miglioramenti generali:

- È stato risolto un problema che poteva causare un errore di una query parallela a causa di problemi transitori di rete durante la lettura dei dati dal volume del cluster Aurora.
- È stato risolto un problema relativo alla gestione dei file di log di audit che può rendere i file di log inaccessibili per il download o la rotazione e, in alcuni casi, aumentare l'utilizzo della CPU.
- È stato risolto un problema per cui le istanze di replica in lettura di piccole dimensioni potevano presentare un maggiore ritardo di replica dopo l'aggiornamento da versioni precedenti alla 2.11.*
- È stato risolto un problema che poteva causare un numero eccessivo di messaggi di log durante la consultazione della tabella delle autorizzazioni `procs_priv` per la verifica delle richieste che coinvolgono routine archiviate.
- È stato risolto un problema di gestione della memoria che poteva far sì che l'istanza database utilizzasse una quantità eccessiva di memoria durante l'esecuzione delle query utilizzando l'ottimizzazione dell'hash join.
- È stato risolto un problema che poteva produrre un valore errato della variabile `Threads_running` nelle tabelle di stato globali `information_schema` e `performance_schema` quando si utilizzava l'inoltro di scrittura.
- È stato risolto un problema che causava il riavvio del database durante l'esecuzione di istruzioni `SELECT` con tabelle partizionate (create in una versione di MySQL che supportava il vecchio gestore di partizioniha_`partition`) e la query parallela veniva scelta dal pianificatore di query.
- È stato risolto un problema che può impedire la creazione di nuove connessioni client al database quando è abilitato l'inoltro di scrittura.
- Ritardo di replica del log binario (binlog) ridotto quando una replica binlog di Aurora MySQL esegue eventi `QUERY` scritti nel file binlog di origine senza un database predefinito definito dal comando `USE`.

- È stato risolto un problema che poteva causare la segnalazione errata della `CommitLatency` CloudWatch metrica quando il `innodb_flush_log_at_trx_commit` parametro non è impostato su 1.
- È stato risolto un problema che può causare la chiusura delle connessioni al database prima che vengano stabilite. È più probabile che questo problema influisca sulle istanze database che aprono e chiudono le connessioni a una velocità elevata.
- È stato risolto un problema che poteva causare il riavvio del database quando gli utenti di log binario (binlog) connessi utilizzano un server di replica binlog duplicato. IDs
- È stato risolto un problema che poteva causare l'interruzione della replica multithread del log binario quando la `replica_parallel_type` variabile è impostata su e la variabile viene attivata. `LOGICAL_CLOCK` `replica_preserve_commit_order` `ON` Questo problema può verificarsi quando una transazione di dimensioni superiori a 500 MB viene eseguita sull'origine.

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla 5.7.40 inclusa, oltre a quanto riportato di seguito. Per ulteriori informazioni, consulta MySQL bugs fixed by Aurora MySQL 2.x database engine updates.

- È stato risolto un problema che poteva causare il blocco delle connessioni remote esistenti e nuove se eseguite contemporaneamente all'istruzione `SHOW PROCESSLIST` (Bug della community 34857411)
- Replica: alcuni log eventi binari non venivano sempre gestiti correttamente (Bug 34617506)

Funzionalità non supportate in Aurora MySQL versione 2

Le seguenti funzionalità non sono attualmente supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Raggruppamento delle scansioni

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa

degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Istruzione SQL `CREATE TABLESPACE`
- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Protocollo X

Aggiornamenti del motore di database Aurora MySQL 2023-10-25 (versione 2.12.0.1, compatibile con MySQL 5.7.40) - Versione RDS Extended Support (Beta)

Version: 2.12.0.1

Aurora MySQL 2.12.0.1 è generalmente disponibile nelle seguenti regioni: Stati Uniti orientali (Virginia settentrionale), Stati Uniti orientali (Ohio), Stati Uniti occidentali (California settentrionale), Stati Uniti occidentali (Oregon), (Stati Uniti orientali) e (Stati Uniti occidentali). AWS GovCloud AWS GovCloud Si tratta di una versione iniziale, che prevede solo correzioni di sicurezza. Queste correzioni verranno implementate in modo più ampio in tutte le regioni con la prossima versione della patch, 2.12.1. Le versioni 2.12 di Aurora MySQL sono compatibili con MySQL 5.7.40.

Le versioni Aurora MySQL attualmente supportate sono 2.07.*, 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.* e 3.04.*.

È possibile aggiornare un cluster di database di Aurora MySQL 2.* esistente ad Aurora MySQL 2.12.0.1. È anche possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.12.0.1.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Risolti i problemi di sicurezza ed CVEs elencati di seguito:

Questa versione include tutte le CVEs correzioni della community fino a MySQL 5.7.40 incluso.

- CVE-2023-38545

Miglioramenti della disponibilità:

- L'inserimento rapido non è abilitato in questa versione di Aurora MySQL, a causa di un problema che può causare incongruenze durante l'esecuzione di query come, e. `INSERT INTO SELECT FROM` Per ulteriori informazioni sull'ottimizzazione rapida degli inserti, consulta Miglioramenti delle prestazioni di Amazon Aurora MySQL.

Aggiornamenti del motore di database Aurora MySQL 2023-07-25 (versione 2.12.0, compatibile con MySQL 5.7.40) - Versione RDS Extended Support

Versione: 2.12.0

Aurora MySQL 2.12.0 è disponibile a livello generale. Le versioni di Aurora MySQL 2.12 sono compatibili fino a MySQL 5.7.40. Per ulteriori informazioni sulle modifiche apportate alla community, consulta Changes in MySQL 5.7.40 (2022-10-11, General Availability).

Le versioni di Aurora MySQL attualmente supportate sono 2.11.*, 2.12.*, 3.01.*, 3.02.* e 3.03.*.

È possibile aggiornare un cluster di database Aurora MySQL 2.* esistente ad Aurora MySQL 2.12.0. È anche possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.12.0.

In caso di domande o dubbi, l'AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Problemi di sicurezza risolti ed CVEs elencati di seguito:

Questa versione include tutte le CVEs correzioni della community fino a MySQL 5.7.40 incluso.

- I cifrari SSL predefiniti utilizzati da Aurora MySQL sono stati aggiornati per escludere i valori DES CBC3 - -SHA meno sicuri dal parametro del database SSL_CIPHER. Se riscontri problemi di connessione SSL dovuti alla rimozione del codice DES- CBC3 -SHA, utilizza un codice sicuro applicabile dal seguente elenco, Configurazione delle suite di crittografia per le connessioni ai cluster Aurora MySQL DB. Ulteriori informazioni sulla configurazione delle crittografia di connessione del client MySQL sono disponibili nella documentazione di MySQL.
- 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

Miglioramenti della disponibilità:

- È stato risolto un problema nella crittografia degli eventi dei flussi di attività del database che poteva causare il riavvio del database
- Sono stati risolti due problemi che potevano causare un errore di riavvio del database se ciò si verificava durante l'esecuzione di una query DDL (Data Definition Language)
- È stato risolto un problema per cui i picchi di connessione possono causare un aumento della latenza delle query o il riavvio dell'istanza database
- È stato risolto un problema che, in rari casi, poteva causare il riavvio di una replica Aurora durante l'esecuzione simultanea di operazioni di aggiornamento di grandi dimensioni o carichi di lavoro DDL (Data Definition Language) sull'istanza di scrittura e operazioni di lettura sullo stesso set di tabelle sulla replica Aurora
- È stato risolto un problema a causa del quale i picchi di connessione potevano richiedere più tempo per il completamento del processo di creazione della connessione o la non riuscita con errori di timeout
- È stato risolto un problema a causa del quale la rotazione dei log di audit avanzato poteva ridurre la memoria liberabile, con conseguente riavvio dell'istanza database
- È stato risolto un problema che poteva causare il riavvio di un'istanza del lettore Aurora MySQL durante l'esecuzione di una query che utilizza un piano di esecuzione di query parallele di Aurora
- È stato risolto un problema che poteva causare il riavvio dell'istanza di scrittura durante l'esecuzione della query `OPTIMIZE TABLE` su una tabella con un indice FTS (Full Text Search)
- È stato risolto un problema che poteva causare il riavvio dell'istanza writer in una AWS regione primaria del database globale Aurora quando una `SELECT FOR UPDATE` query veniva eseguita utilizzando l'inoltro globale di scrittura da una regione secondaria del database globale Aurora.
- È stato risolto un problema che poteva causare il riavvio di un'istanza AWS Region reader secondaria del database globale Aurora che utilizzava l'inoltro globale di scrittura quando un'istruzione di commit implicita inoltrata rilevava un errore.
- L'inserimento rapido non è abilitato in questa versione di Aurora MySQL, a causa di un problema che può causare incongruenze durante l'esecuzione di query come, e. `INSERT INTO SELECT FROM` Per ulteriori informazioni sull'ottimizzazione rapida degli inserti, consulta Miglioramenti delle prestazioni di Amazon Aurora MySQL.

Miglioramenti generali:

- Sono state introdotte ottimizzazioni delle prestazioni di gestione dei file sulle repliche binlog per ridurre i conflitti durante la scrittura su file di log di relay

- È stato risolto un problema che poteva causare una segnalazione errata del contatore `buffer_pool_read_requests` nelle metriche `information_schema`
- È stato risolto un problema che poteva causare il riempimento dell'archiviazione locale durante l'esecuzione di operazioni `LOAD FROM S3` o `SELECT INTO S3`. Il problema può anche comportare un maggiore utilizzo della CPU, il riavvio del database a causa di memoria insufficiente e una maggiore latenza per queste query.
- È stato risolto un problema per cui le istanze database che utilizzavano la replica di log binari potevano registrare un aumento dell'utilizzo della CPU e errori di connessione quando erano collegati più consumer di replica di log binari
- È stato risolto un problema a causa del quale le variabili di stato del server SSL non venivano popolate
- È stato risolto un problema a causa del quale le istruzioni DML (Data Manipulation Language) che eseguivano scritture duplicate potevano causare un'eccessiva registrazione degli errori e un aumento delle latenze delle query
- Le definizioni del fuso orario sono state aggiornate alla versione IANA 2023c
- È stato aggiunto il supporto per l'abilitazione e la disabilitazione della registrazione binaria a livello di sessione. Consulta Stored Procedures - Replicating nella Guida per l'utente di Amazon Aurora
- È stato aggiunto il supporto per l'impostazione del formato di log binario a livello di sessione. Consulta Stored Procedures - Replicating nella Guida per l'utente di Amazon Aurora
- È stato risolto un problema a causa del quale l'impostazione del parametro `aurora_disable_hash_join` su `1` o `ON` poteva non impedire all'ottimizzatore di utilizzare un hash join
- È stato risolto un problema relativo alle scansioni degli indici in cui poteva essere restituito un risultato impreciso durante l'esecuzione di una query `SELECT` con la clausola `GROUP BY` e il parametro `aurora_parallel_query` impostati su `ON`
- È stato risolto un problema che, in rari casi, può causare il riavvio di un'istanza di lettura di Amazon Aurora quando si accede a una tabella con grandi operazioni di aggiornamento o Data Definition Language (DDL) in esecuzione contemporaneamente sull'istanza di scrittura
- È stato risolto un problema che poteva causare una segnalazione errata del contatore `buffer_pool_read_requests` nelle metriche `information_schema`
- È stato risolto un problema che poteva causare il riavvio di una replica binlog se la variabile di sistema server_uuid dell'origine è mancante o ha un valore non valido

- È stato risolto un problema per evitare che le statistiche di InnoDB diventassero obsolete, il che a volte può generare un piano di esecuzione delle query non ottimale che può portare a un aumento del tempo di esecuzione delle query
- È stato risolto un problema per cui le `AuroraGlobalDBRPOLag` CloudWatch metriche mostravano sempre zero indipendentemente dal carico di lavoro dell'utente

Aggiornamenti e migrazioni:

- Per eseguire un aggiornamento della versione secondaria per un database globale Aurora da Aurora MySQL versione 2.07 o 2.11 ad Aurora MySQL versione 2.12 o successiva, consulta Aggiornamento di Aurora MySQL modificando la versione del motore.

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla 5.7.40 inclusa, oltre a quanto riportato di seguito. Per ulteriori informazioni, consulta MySQL bugs fixed by Aurora MySQL 2.x database engine updates.

- È stato risolto un problema che può causare un maggiore utilizzo della CPU a causa della rotazione dei certificati TLS in background (Correzione di un bug della community 34284186)

Funzionalità non supportate in Aurora MySQL versione 2

Le seguenti funzionalità non sono attualmente supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Raggruppare le scansioni.

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Istruzione SQL `CREATE TABLESPACE`
- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Protocollo X

Aggiornamenti del motore di database Aurora MySQL 2024-07-19 (versione 2.11.6, compatibile con MySQL 5.7.12) - Versione RDS Extended Support

Versione: 2.11.6

Aurora MySQL 2.11.6 è disponibile a livello generale. Le versioni 2.11 di Aurora MySQL sono compatibili con MySQL 5.7.12. Per ulteriori informazioni sulle modifiche apportate alla community, consulta Changes in MySQL 5.7.12 (2016-04-11, General Availability).

Le versioni di Aurora MySQL attualmente supportate sono 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.*, 3.04.*, 3.05.*, 3.06.* e 3.07.*.

È possibile aggiornare un cluster di database Aurora MySQL 2.* esistente ad Aurora MySQL 2.11.6. È inoltre possibile ripristinare un'istantanea da qualsiasi versione precedente di Aurora MySQL versione 2 attualmente supportata in Aurora MySQL 2.11.6.

Se aggiorni un database globale Aurora MySQL alla versione 2.11.*, devi aggiornare i tuoi cluster di database primari e secondari alla stessa identica versione, incluso il livello di patch. Per ulteriori informazioni sull'aggiornamento della versione secondaria di un database globale Aurora, consulta Aggiornamenti delle versioni secondarie.

Immediatamente dopo l'aggiornamento locale della versione del motore ad Aurora MySQL 2.11.*, viene applicato automaticamente un aggiornamento del sistema operativo a tutte le istanze interessate sulle classi di istanze database db.r4, db.r5, db.t2 e db.t3, se le istanze eseguono una vecchia versione del sistema operativo. In un cluster database Multi-AZ, tutte le istanze di lettura applicano prima l'aggiornamento del sistema operativo. Al termine dell'aggiornamento del sistema operativo sulla prima istanza di lettura, si verifica un failover e l'istanza di scrittura precedente viene aggiornata.

> **Note**
>
> L'aggiornamento del sistema operativo non viene applicato automaticamente ai database globali di Aurora durante gli aggiornamenti delle versioni principali.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Risolti i problemi di sicurezza e CVEs:

- È stato risolto un problema di sicurezza nelle stored procedure MySQL.

Questa versione include tutte le correzioni CVE della community fino a MySQL 5.7.12 incluso. Le seguenti correzioni CVE sono incluse in questa versione:

- CVE-2023-21912
- CVE-2023-38545
- CVE-2023-39975

- CVE-2023-44487
- CVE-2024-0853
- CVE-2024-20963
- CVE-2024-20993
- 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

Miglioramenti della disponibilità:

- È stato risolto un problema che poteva causare il riavvio di un server di database a causa dell'accesso simultaneo alle risorse di connessione durante la scalabilità senza interruzioni, il riavvio senza interruzioni (ZDR) e l'applicazione di patch senza interruzioni (ZDP).
- È stato risolto un problema che poteva causare il riavvio di un'istanza Reader DB durante la liberazione della memoria utilizzata per l'applicazione di registro.
- È stato risolto un problema che causava il riavvio di un'istanza Reader DB durante l'esecuzione di una query parallela.
- È stato risolto un problema che poteva causare il riavvio di un'istanza Reader DB che utilizza l'inoltro di scrittura quando un'istruzione di commit implicita inoltrata rileva un errore.

Miglioramenti generali:

- È stato risolto un problema che può causare errori o avvisi imprevisti di violazione della chiave primaria nelle istruzioni SQL su alcune righe quando si eseguono `INSERT` istruzioni simultanee su

una tabella con una colonna chiave `AUTO_INCREMENT` primaria e una colonna chiave univoca e quando un'`INSERT`istruzione presenta violazioni di chiave univoche su righe diverse.

- È stato risolto un problema che poteva causare risultati di query errati quando ZDR ripristinava erroneamente le variabili di sessione impostate come suggerimenti nelle query.
- È stato risolto un problema durante il Aurora Serverless v1 ridimensionamento che causava il riavvio dell'istanza DB a causa di un accesso errato a una struttura dati interna durante la ricerca di un punto di scalabilità.
- È stato risolto un problema per cui `UPDATE` le query lente `INSERT` ed eseguite da MySQL Event Scheduler non venivano registrate nel registro delle query lente a meno che non fossero precedute da una query lenta. `DELETE SELECT`

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla 5.7.12 inclusa. Per ulteriori informazioni, consulta Correzione dei bug di MySQL attraverso gli aggiornamenti del motore di database Aurora MySQL 2.x.

Funzionalità non supportate in Aurora MySQL versione 2

Le seguenti funzionalità non sono attualmente supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi

- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 2024-03-26 (versione 2.11.5, compatibile con MySQL 5.7.12) - Versione RDS Extended Support (predefinita)

Versione: 2.11.5

Aurora MySQL 2.11.5 è disponibile a livello generale. Le versioni 2.11 di Aurora MySQL sono compatibili con MySQL 5.7.12. Per ulteriori informazioni sulle modifiche apportate alla community, consulta Changes in MySQL 5.7.12 (2016-04-11, General Availability).

La versione 2.11.5 è l'attuale versione predefinita per Aurora MySQL versione 2 quando si crea un cluster DB. Per ulteriori informazioni, consulta Versioni predefinite di Amazon Aurora.

Le versioni di Aurora MySQL attualmente supportate sono 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.*, 3.04.*, 3.05.* e 3.06.*.

È possibile aggiornare un cluster di database Aurora MySQL 2.* esistente ad Aurora MySQL 2.11.5. È inoltre possibile ripristinare un'istantanea da qualsiasi versione precedente di Aurora MySQL versione 2 attualmente supportata in Aurora MySQL 2.11.5.

Se aggiorni un database globale Aurora MySQL alla versione 2.11.*, devi aggiornare i tuoi cluster di database primari e secondari alla stessa identica versione, incluso il livello di patch. Per ulteriori informazioni sull'aggiornamento della versione secondaria di un database globale Aurora, consulta Aggiornamenti delle versioni secondarie.

Immediatamente dopo l'aggiornamento locale della versione del motore ad Aurora MySQL 2.11.*, viene applicato automaticamente un aggiornamento del sistema operativo a tutte le istanze interessate sulle classi di istanze database db.r4, db.r5, db.t2 e db.t3, se le istanze eseguono una vecchia versione del sistema operativo. In un cluster database Multi-AZ, tutte le istanze di lettura applicano prima l'aggiornamento del sistema operativo. Al termine dell'aggiornamento del sistema operativo sulla prima istanza di lettura, si verifica un failover e l'istanza di scrittura precedente viene aggiornata.

> **Note**
>
> L'aggiornamento del sistema operativo non viene applicato automaticamente ai database globali di Aurora durante gli aggiornamenti delle versioni principali.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Risolti i problemi di sicurezza e CVEs:

Le seguenti correzioni CVE sono incluse in questa versione:

- 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

Miglioramenti della disponibilità:

- È stato risolto un problema a causa del quale un'istanza DB di Aurora MySQL Writer poteva fallire a causa di un difetto nel componente che comunica con lo storage Aurora. Il difetto si verifica a seguito di un'interruzione della comunicazione tra l'istanza DB e lo storage sottostante a seguito di un aggiornamento del software.
- È stato risolto un problema che, in rare condizioni, poteva causare il riavvio delle istanze DB del lettore.
- È stato risolto un problema di conflitto di blocchi causato da un thread di log di audit che poteva portare a un utilizzo elevato della CPU e al timeout delle applicazioni client.

Miglioramenti generali:

- È stato risolto un problema che poteva causare il fallimento di una query parallela a causa di problemi transitori di rete durante la lettura dei dati dal volume del cluster Aurora DB.
- È stato risolto un problema relativo alla gestione dei file di registro di controllo che può rendere i file di registro inaccessibili per il download o la rotazione e, in alcuni casi, aumentare l'utilizzo della CPU.
- È stato risolto un problema che poteva produrre un valore errato della `Threads_running` variabile nelle tabelle di stato `information_schema` e nelle tabelle di stato `performance_schema` globali quando si utilizzava l'inoltro di scrittura.

Aggiornamenti e migrazioni:

- È stato risolto un problema che impediva l'avvio della replica dei log binari sui cluster Aurora MySQL DB migrati da RDS for MySQL 5.7.
- L'utilità di pianificazione degli eventi del database è stata disabilitata durante gli aggiornamenti delle versioni principali di Aurora MySQL versione 3. Ciò consente di evitare modifiche al database mediante l'esecuzione di eventi mentre è in corso l'aggiornamento della versione principale.

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla 5.7.12 inclusa, oltre a quanto riportato di seguito. Per ulteriori informazioni, consulta MySQL bugs fixed by Aurora MySQL 2.x database engine updates.

Funzionalità non supportate in Aurora MySQL versione 2

Le seguenti funzionalità non sono attualmente supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 2023-10-17 (versione 2.11.4, compatibile con MySQL 5.7.12) - Versione RDS Extended Support

Versione: 2.11.4

Aurora MySQL 2.11.4 è disponibile a livello generale. Le versioni 2.11 di Aurora MySQL sono compatibili con MySQL 5.7.12. Per ulteriori informazioni sulle modifiche apportate alla community, consulta Changes in MySQL 5.7.12 (2016-04-11, General Availability).

Le versioni di Aurora MySQL attualmente supportate sono 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.* e 3.04.*.

È possibile aggiornare un cluster di database di Aurora MySQL 2.* esistente ad Aurora MySQL 2.11.4. È anche possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.11.4.

Se aggiorni un database globale Aurora MySQL alla versione 2.11.*, devi aggiornare i tuoi cluster di database primari e secondari alla stessa identica versione, incluso il livello di patch. Per ulteriori informazioni sull'aggiornamento della versione secondaria di un database globale Aurora, consulta Aggiornamenti delle versioni secondarie.

Immediatamente dopo l'aggiornamento locale della versione del motore ad Aurora MySQL 2.11.*, viene applicato automaticamente un aggiornamento del sistema operativo a tutte le istanze interessate sulle classi di istanze database db.r4, db.r5, db.t2 e db.t3, se le istanze eseguono una vecchia versione del sistema operativo. In un cluster database Multi-AZ, tutte le istanze di lettura applicano prima l'aggiornamento del sistema operativo. Al termine dell'aggiornamento del sistema operativo sulla prima istanza di lettura, si verifica un failover e l'istanza di scrittura precedente viene aggiornata.

> **Note**
>
> L'aggiornamento del sistema operativo non viene applicato automaticamente ai database globali di Aurora durante gli aggiornamenti delle versioni principali.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Risolti i problemi di sicurezza ed CVEs elencati di seguito:

- È stato risolto un problema per cui gli eventi segnalati durante l'elaborazione delle rotazioni dei log di audit potevano non essere scritti nel log di audit.
- CVE-2022-24407

Miglioramenti della disponibilità:

- È stato risolto un problema per cui le istanze database di Aurora MySQL che utilizzavano query parallele potevano subire un riavvio del database durante l'esecuzione di un numero elevato di query parallele simultanee.
- È stato risolto un problema che poteva causare il riavvio di un'istanza di database durante l'esecuzione di carichi di lavoro di lettura I/O intensivi.
- È stato risolto un problema che può causare il riavvio di un'istanza database quando si tenta di leggere una pagina del database che appartiene a una tabella eliminata.
- È stato risolto un problema che può causare il riavvio dell'istanza di lettura quando l'istanza di scrittura aumenta il volume del database fino a un multiplo di 160 GB.
- È stato risolto un problema che poteva causare la non disponibilità del cluster di database al riavvio dell'istanza di scrittura mentre il database stava creando o eliminando i trigger sulle tabelle di sistema interne.
- È stato risolto un problema che poteva causare il riavvio di un'istanza di lettura durante l'esecuzione di query DML (Data Manipulation Language) su una tabella contenente un indice full-text.

- È stato risolto un problema che poteva causare il riavvio di un'istanza di lettura durante l'esecuzione di una query che utilizza un piano di esecuzione di query parallele di Aurora.
- È stato risolto un problema che poteva causare il riavvio dell'istanza di scrittura durante l'esecuzione della query `OPTIMIZE TABLE` su una tabella con un indice FTS (Full Text Search).
- L'inserimento rapido non è abilitato in questa versione di Aurora MySQL, a causa di un problema che può causare incongruenze durante l'esecuzione di query come, e. `INSERT INTO SELECT FROM` Per ulteriori informazioni sull'ottimizzazione rapida degli inserti, consulta Miglioramenti delle prestazioni di Amazon Aurora MySQL.

Miglioramenti generali:

- È stato risolto un problema per cui le istanze di replica in lettura di piccole dimensioni potevano presentare un maggiore ritardo di replica dopo l'aggiornamento da versioni precedenti alla 2.11.*.
- È stato risolto un problema che poteva causare un numero eccessivo di messaggi di log durante la consultazione di procs_priv grant table per la verifica delle richieste che coinvolgono routine archiviate.
- È stato risolto un problema di gestione della memoria che poteva far sì che l'istanza database utilizzasse una quantità eccessiva di memoria durante l'esecuzione delle query utilizzando l'ottimizzazione dell'hash join.
- È stato risolto un problema che causava il riavvio del database durante l'esecuzione di istruzioni `SELECT` con tabelle partizionate (create in una versione di MySQL che supportava il vecchio gestore di partizioni`ha_partition`) e la query parallela veniva scelta dal pianificatore di query.
- È stato risolto un problema che può impedire la creazione di nuove connessioni client al database quando è abilitato l'inoltro di scrittura.
- Ritardo di replica del log binario (binlog) ridotto quando una replica binlog di Aurora MySQL esegue eventi `QUERY` scritti nel file binlog di origine senza un database predefinito definito dal comando `USE`.
- È stato risolto un problema relativo alle scansioni degli indici in cui poteva essere restituito un risultato impreciso durante l'esecuzione di una query `SELECT` con la clausola `GROUP BY` e il parametro `aurora_parallel_query` impostati su `ON`.
- È stato aggiunto il supporto per l'abilitazione e la disabilitazione della registrazione binaria a livello di sessione. Consulta Stored Procedures - Replicating nella Guida per l'utente di Amazon Aurora.
- È stato risolto un problema che poteva causare il riavvio di una replica binlog se la variabile di sistema server_uuid dell'origine è mancante o ha un valore non valido.

- È stato aggiunto il supporto per l'impostazione del formato di log binario a livello di sessione. Consulta Stored Procedures - Replicating nella Guida per l'utente di Amazon Aurora.
- È stato risolto un problema che poteva causare la segnalazione errata della `CommitLatency` CloudWatch metrica quando il `innodb_flush_log_at_trx_commit` parametro non è impostato su 1.
- È stato risolto un problema per evitare che le statistiche di InnoDB diventassero obsolete, il che a volte può generare un piano di esecuzione delle query non ottimale che può portare a un aumento del tempo di esecuzione delle query.
- È stato risolto un problema che poteva causare il riavvio del database quando gli utenti di log binario (binlog) connessi utilizzano un server di replica binlog duplicato. IDs

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla 5.7.12 inclusa, oltre a quanto riportato di seguito. Per ulteriori informazioni, consulta MySQL bugs fixed by Aurora MySQL 2.x database engine updates.

- Replica: alcuni eventi di log binario non venivano sempre gestiti correttamente. (Bug 34617506)
- È stato risolto un problema che può causare un maggiore utilizzo della CPU a causa della rotazione dei certificati TLS in background (Correzione di un bug della community 34284186).
- Nelle istruzioni preparate, alcuni tipi di query secondarie potrebbero causare la chiusura del server. (Bug 33100586)

Funzionalità non supportate in Aurora MySQL versione 2

Le seguenti funzionalità non sono attualmente supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa

degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 2023-06-09 (versione 2.11.3, compatibile con MySQL 5.7.12) - Versione RDS Extended Support

Versione: 2.11.3

Aurora MySQL 2.11.3 è disponibile a livello generale. Le versioni 2.11 di Aurora MySQL sono compatibili con MySQL 5.7.12. Per ulteriori informazioni sulle modifiche apportate alla community, consulta Changes in MySQL 5.7.12 (2016-04-11, General Availability).

Le versioni di Aurora MySQL attualmente supportate sono 2.07.*, 2.11.*, 3.01.*, 3.02.* e 3.03.*.

È possibile aggiornare un cluster di database di Aurora MySQL 2.* esistente ad Aurora MySQL 2.11.3. È anche possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.11.3.

Se aggiorni un database globale Aurora MySQL alla versione 2.11.*, devi aggiornare i tuoi cluster di database primari e secondari alla stessa identica versione, incluso il livello di patch. Per ulteriori

informazioni sull'aggiornamento della versione secondaria di un database globale Aurora, consulta Aggiornamenti delle versioni secondarie.

Immediatamente dopo l'aggiornamento locale della versione del motore ad Aurora MySQL 2.11.*, viene applicato automaticamente un aggiornamento del sistema operativo a tutte le istanze interessate sulle classi di istanze database db.r4, db.r5, db.t2 e db.t3, se le istanze eseguono una vecchia versione del sistema operativo. In un cluster database Multi-AZ, tutte le istanze di lettura applicano prima l'aggiornamento del sistema operativo. Al termine dell'aggiornamento del sistema operativo sulla prima istanza di lettura, si verifica un failover e l'istanza di scrittura precedente viene aggiornata.

> **Note**
>
> L'aggiornamento del sistema operativo non viene applicato automaticamente ai database globali di Aurora durante gli aggiornamenti delle versioni principali.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Risolti i problemi di sicurezza ed CVEs elencati di seguito:

- Sono state aggiornate le crittografie SSL predefinite utilizzate da Aurora MySQL per escludere i valori `DES-CBC3-SHA` meno sicuri dal parametro del database SSL_CIPHER. Se riscontri problemi di connessione SSL dovuti alla rimozione del `DES-CBC3-SHA` codice, utilizza un codice sicuro applicabile da questo elenco,. ConfiguringCipherSuites Ulteriori informazioni sulla configurazione delle crittografia di connessione del client MySQL sono disponibili nella documentazione di MySQL.
- CVE-2023-21963

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

Miglioramenti della disponibilità:

- È stato risolto un problema relativo alla crittografia degli eventi dei flussi di attività del database (DAS, database activity streams) che poteva causare il riavvio del database.
- Sono stati risolti due problemi che potevano causare un errore di riavvio del database se ciò si verificava durante l'esecuzione di una query DDL (Data Definition Language).
- L'inserimento rapido non è abilitato in questa versione di Aurora MySQL, a causa di un problema che può causare incongruenze durante l'esecuzione di query come, e. `INSERT INTO SELECT FROM` Per ulteriori informazioni sull'ottimizzazione rapida degli inserti, consulta Miglioramenti delle prestazioni di Amazon Aurora MySQL.

Miglioramenti generali:

- Sono state introdotte ottimizzazioni delle prestazioni di gestione dei file sulle repliche binlog per ridurre i conflitti durante la scrittura su file di log di relay.
- È stato risolto un problema a causa del quale l'impostazione del parametro `aurora_disable_hash_join` su 1 o su ON poteva non impedire all'ottimizzatore di utilizzare un hash join.
- È stato risolto un problema che poteva causare una segnalazione errata del contatore `buffer_pool_read_requests` nelle metriche `information_schema`.
- È stato risolto un problema che poteva causare il riempimento dell'archiviazione locale durante l'esecuzione di operazioni `LOAD FROM S3` o `SELECT INTO S3`. Il problema può anche comportare un maggiore utilizzo della CPU, il riavvio del database a causa di memoria insufficiente e una maggiore latenza per queste query.

Funzionalità non supportate in Aurora MySQL versione 2

Le seguenti funzionalità non sono attualmente supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 2023-03-24 (versione 2.11.2, compatibile con MySQL 5.7.12) - Versione RDS Extended Support

Versione: 2.11.2

Aurora MySQL 2.11.2 è disponibile a livello generale. Le versioni 2.11 di Aurora MySQL sono compatibili con MySQL 5.7.12. Per ulteriori informazioni sulle modifiche apportate alla community, consulta Changes in MySQL 5.7.12 (2016-04-11, General Availability).

Le versioni di Aurora MySQL attualmente supportate sono 2.07.*, 2.11.*, 3.01.*, 3.02.* e 3.03.*.

È possibile aggiornare un cluster di database Aurora MySQL 2.* esistente ad Aurora MySQL 2.11.2. È anche possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.11.2.

Se aggiorni un database globale Aurora MySQL alla versione 2.11.*, devi aggiornare i tuoi cluster di database primari e secondari alla stessa identica versione, incluso il livello di patch. Per ulteriori informazioni sull'aggiornamento della versione secondaria di un database globale Aurora, consulta Aggiornamenti delle versioni secondarie.

Immediatamente dopo l'aggiornamento locale della versione del motore ad Aurora MySQL 2.11.*, viene applicato automaticamente un aggiornamento del sistema operativo a tutte le istanze interessate sulle classi di istanze database db.r4, db.r5, db.t2 e db.t3, se le istanze eseguono una vecchia versione del sistema operativo. In un cluster database Multi-AZ, tutte le istanze di lettura applicano prima l'aggiornamento del sistema operativo. Al termine dell'aggiornamento del sistema operativo sulla prima istanza di lettura, si verifica un failover e l'istanza di scrittura precedente viene aggiornata.

> **Note**
>
> L'aggiornamento del sistema operativo non viene applicato automaticamente ai database globali di Aurora durante gli aggiornamenti delle versioni principali.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Miglioramenti generali:

- È stato risolto un problema per cui le istanze database che utilizzavano la replica di log binari potevano registrare un aumento dell'utilizzo della CPU e errori di connessione quando erano collegati più consumer di replica di log binari.
- È stato risolto un problema che poteva causare la mancata sincronizzazione di un'istanza di lettura in una regione secondaria del database globale dopo l'aggiornamento alla versione 2.11 di Aurora MySQL se l'istanza di scrittura principale del database si trova su Aurora MySQL versione 2.10.

Miglioramenti della disponibilità:

- L'inserimento rapido non è abilitato in questa versione di Aurora MySQL, a causa di un problema che può causare incongruenze durante l'esecuzione di query come, e. `INSERT INTO SELECT FROM` Per ulteriori informazioni sull'ottimizzazione rapida degli inserti, consulta Miglioramenti delle prestazioni di Amazon Aurora MySQL.

Funzionalità non supportate in Aurora MySQL versione 2

Le seguenti funzionalità non sono attualmente supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB

- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 2023-02-14 (versione 2.11.1, compatibile con MySQL 5.7.12) - Versione RDS Extended Support

Versione: 2.11.1

Aurora MySQL 2.11.1 è disponibile a livello generale. Le versioni 2.11 di Aurora MySQL sono compatibili con MySQL 5.7.12. Per ulteriori informazioni sulle modifiche apportate alla community, consulta Changes in MySQL 5.7.12 (2016-04-11, General Availability).

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.07.*, 2.09.*, 2.10.*, 2.11.*, 3.01.* e 3.02.*.

È possibile aggiornare un cluster di database di Aurora MySQL 2.* esistente ad Aurora MySQL 2.11.1. Per i cluster che eseguono la versione 1 di Aurora MySQL, è possibile aggiornare un cluster di Aurora MySQL 1.23 o superiore esistente a 2.11.1. È anche possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.11.1.

Se un database globale Aurora MySQL viene aggiornato alla versione 2.11.* e l'inoltro di scrittura è attivato, è necessario aggiornare i cluster di database primario e secondario alla stessa versione, incluso il livello di patch, per continuare a utilizzare l'inoltro di scrittura. Per ulteriori informazioni sull'aggiornamento della versione secondaria di un database globale Aurora, consulta Aggiornamenti delle versioni secondarie.

Immediatamente dopo l'aggiornamento locale della versione del motore ad Aurora MySQL 2.11.*, viene applicato automaticamente un aggiornamento del sistema operativo a tutte le istanze interessate sulle classi di istanze database db.r4, db.r5, db.t2 e db.t3, se le istanze eseguono una vecchia versione del sistema operativo. In un cluster database Multi-AZ, tutte le istanze di lettura

applicano prima l'aggiornamento del sistema operativo. Al termine dell'aggiornamento del sistema operativo sulla prima istanza di lettura, si verifica un failover e l'istanza di scrittura precedente viene aggiornata.

> **Note**
>
> L'aggiornamento del sistema operativo non viene applicato automaticamente ai database globali di Aurora durante gli aggiornamenti delle versioni principali.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Risolti i problemi di sicurezza ed CVEs elencati di seguito:

Correzioni e altri miglioramenti per ottimizzare la gestione in un ambiente gestito. Ulteriori correzioni CVE riportate di seguito:

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

Miglioramenti della disponibilità:

- È stato risolto un problema per cui i picchi di connessione possono causare un aumento della latenza delle query o il riavvio dell'istanza database.

- È stato risolto un problema che, in rari casi, poteva causare il riavvio di una replica Aurora durante l'esecuzione simultanea di operazioni di aggiornamento di grandi dimensioni o carichi di lavoro DDL (Data Definition Language) sull'istanza di scrittura e operazioni di lettura sullo stesso set di tabelle sulla replica Aurora.
- È stato risolto un problema a causa del quale i picchi di connessione potevano richiedere più tempo per il completamento del processo di creazione della connessione o la non riuscita con errori di timeout.
- È stato risolto un problema a causa del quale la rotazione dei log di audit avanzato poteva ridurre la memoria liberabile, con conseguente riavvio dell'istanza database.
- L'inserimento rapido non è abilitato in questa versione di Aurora MySQL, a causa di un problema che può causare incongruenze durante l'esecuzione di query come, e. `INSERT INTO SELECT FROM` Per ulteriori informazioni sull'ottimizzazione rapida degli inserti, consulta Miglioramenti delle prestazioni di Amazon Aurora MySQL.

Miglioramenti generali:

- È stato risolto un problema a causa del quale le variabili di stato del server SSL non venivano popolate.
- È stato risolto un problema a causa del quale le istruzioni DML (Data Manipulation Language) che eseguivano scritture duplicate potevano causare un'eccessiva registrazione degli errori e un aumento delle latenze delle query.

Confronto con Aurora MySQL Versione 1

Le seguenti caratteristiche di Amazon Aurora MySQL sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).

- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 2022-10-25 (versione 2.11.0, compatibile con MySQL 5.7.12) - Versione RDS Extended Support

Versione: 2.11.0

Aurora MySQL 2.11.0 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7.12. Per ulteriori informazioni sulle modifiche apportate alla community, consulta Changes in MySQL 5.7.12 (2016-04-11, General Availability).

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 2.11.*, 3.01.* e 3.02.*.

È possibile aggiornare un cluster di database Aurora MySQL 2.* esistente ad Aurora MySQL 2.11.0. Per i cluster che eseguono la versione Aurora MySQL 1, è possibile aggiornare un cluster Aurora MySQL 1.23 o superiore esistente a 2.11.0. È anche possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.11.0.

Se un database globale Aurora MySQL viene aggiornato alla versione 2.11.* e l'inoltro di scrittura è attivato, è necessario aggiornare i cluster di database primario e secondario alla stessa versione, incluso il livello di patch, per continuare a utilizzare l'inoltro di scrittura. Per ulteriori informazioni sull'aggiornamento della versione secondaria di un database globale Aurora, consulta Aggiornamenti delle versioni secondarie.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Risolti i problemi di sicurezza e CVEs:

- Cambiato il SSL/TLS provider da AWS-LCOpenSSL a.

  Ciò comporta una serie di modifiche tra cui, a titolo esemplificativo ma non esaustivo, la rimozione del supporto per i cifrari DHE-RSA-* meno sicuri.

  Per ulteriori informazioni, consulta Utilizzo di TLS con cluster database Aurora MySQL

Le seguenti correzioni CVE sono incluse in questa versione:

- 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

Nuove caratteristiche:

- Con Aurora MySQL versione 2.11, è disponibile un nuovo aggiornamento del sistema operativo. Si consiglia di applicare questo aggiornamento del sistema operativo in sospeso a tutte le istanze database di Aurora MySQL dopo l'aggiornamento alla versione 2.11. Per ulteriori informazioni, Utilizzo degli aggiornamenti del sistema operativo.
- È possibile utilizzare una nuova opzione di configurazione dinamica, innodb_deadlock_detect, per disabilitare il rilevamento dei deadlock. Sui sistemi ad alta simultaneità, il rilevamento del

deadlock può causare un rallentamento quando numerosi thread attendono lo stesso blocco. A volte, può essere più efficiente disabilitare il rilevamento dei deadlock e affidarsi all'impostazione innodb_lock_wait_timeout per il rollback delle transazioni quando si verifica un deadlock. (Bug 23477773) Ulteriori informazioni sul rilevamento del deadlock di Innodb sono disponibili nella documentazione di MySQL.

- Sono state aggiunte le fuzioni `UUID_TO_BIN`, `BIN_TO_UUID` e `IS_UUID` di MySQL 8.0. Ulteriori informazioni sull'utilizzo di queste funzioni sono disponibili nella funzione MySQL Miscellaneous.
- È stato aggiunto il supporto per i suggerimenti per l'ottimizzazione che consentono all'utente di abilitare o disabilitare una query parallela di Aurora MySQL per tabella o per query.
  - Utilizzo di query in parallelo per Amazon Aurora MySQL
  - Suggerimenti di Aurora MySQL
- È stato rimosso il supporto per il tipo di istanza R3.
- Aggiunto il supporto per le istanze R6i.

Miglioramenti della disponibilità:

- È stato risolto un problema che poteva impedire la replica logica tra regioni in un cluster di database a causa di un file binlog e di una posizione errati scritti nei log degli errori. Questo problema si può verificare quando il motore viene riavviato dopo l'esecuzione di un'istruzione DDL.
- È stato risolto un problema che, in rare condizioni, può causare il riavvio delle istanze di lettura di Aurora durante l'esecuzione di istruzioni della lista di controllo degli accessi (ACL) come GRANT e FLUSH sull'istanza di scrittura. È più probabile che questo problema influisca sulle istanze di lettura con un numero elevato di utenti e operazioni ACL (ad esempio, modifiche delle autorizzazioni).
- È stato risolto un problema che, in rare condizioni, può causare il riavvio o il failover dell'istanza di scrittura quando una transazione accede a una riga che viene eliminata da un'altra transazione.
- Sono state migliorate le prestazioni di ricerca delle frasi full-text per ridurre significativamente il tempo impiegato per cercare le frasi in una tabella con indici full-text.
- Risolto un problema per cui, dopo il riavvio, un'istanza di scrittura si bloccava in un lento ripristino e successivamente si riavviava nuovamente. Questo problema si verifica quando nel database è presente un numero elevato di righe di cui non è stato eseguito il commit al momento del riavvio iniziale.
- È stato risolto un problema che, in rari casi, causa il riavvio del server di database a causa di una lunga attesa del semaforo quando il thread del rilevatore di deadlock si blocca.

- È stato risolto un problema che, in rari casi, può causare il riavvio del database a causa di una lunga attesa del semaforo quando I/O i thread vengono bloccati.
- L'inserimento rapido non è abilitato in questa versione di Aurora MySQL, a causa di un problema che può causare incongruenze durante l'esecuzione di query come, e. `INSERT INTO SELECT FROM` Per ulteriori informazioni sull'ottimizzazione rapida degli inserti, consulta Miglioramenti delle prestazioni di Amazon Aurora MySQL.

Miglioramenti generali:

- È stato risolto un problema che può causare il riavvio del server di database quando sono soddisfatte tutte le seguenti condizioni:
  - ALLOW_INVALID_DATES è disabilitato in MODALITÀ SQL.
  - Il server del database sta elaborando un'istruzione INSERT, UPDATE, DELETE o SELECT con un valore DATETIME non valido, in modo che il mese non è compreso tra 1 e 12.
- È stato risolto un problema per cui il periodo di conservazione dei log binari non veniva rispettato quando log-bin era impostato su OFF, con conseguente utilizzo superiore al previsto dell'archiviazione. Dopo questa correzione, i log binari verranno eliminati in base al periodo di conservazione. Ulteriori informazioni su come configurare il periodo di conservazione dei log binari sono disponibili nella Guida per l'utente di Aurora MySQL.
- È stato risolto un problema che poteva causare la riduzione della memoria liberabile sull'istanza database quando alcune istruzioni SQL del Data Control Language (DCL) come GRANT, FLUSH PRIVILEGES e così via vengono eseguite su quell'istanza. L'uso frequente di tali istruzioni può causare una continua riduzione della memoria disponibile e il riavvio dell'istanza del database a causa di problemi. out-of-memory L'uso di tali istruzioni sull'istanza di scrittura può inoltre causare una riduzione della memoria liberabile sulle istanze di lettura.
- È stata introdotta una dimensione maggiore del buffer di lettura per le letture eseguite dai log di inoltro per ridurre al minimo il numero di I/O operazioni di lettura, il che riduce il conflitto tra i thread e SQL. I/O
- È stato risolto un problema che poteva causare un errore della procedura archiviata mysql.rds_rotate_slow_log con il messaggio di errore "Table 'mysql.slow_log_backup' doesn't exist".
- È stato risolto un problema a causa del quale l'eccessiva invalidazione della cache delle query causava un utilizzo della CPU e latenze sulla replica di lettura superiori al previsto, a causa del fatto che la replica di lettura doveva leggere i dati dal disco anziché dalla cache delle query.

- È stato risolto un problema che consentiva agli utenti di eseguire i comandi INSTALL PLUGIN e UNINSTALL PLUGIN su un'istanza del lettore, che poteva causare un deadlock su LOCK_plugin, LOCK_system_variables_hash, LOCK_global_system_variables. Queste istruzioni possono ora essere eseguite solo sull'istanza scrittura in un cluster di database.
- È stato risolto un problema a causa del quale i cluster possono presentare una latenza di commit superiore a quella prevista in seguito all'attivazione della registrazione binaria. Ciò influisce su tutte le transazioni che generano eventi binlog di grandi dimensioni (dimensioni superiori a 500 MB).
- È stato risolto un problema che poteva causare un valore errato della metrica trx_active_transactions nella tabella INFORMATION_SCHEMA.INNODB_METRICS.
- È stato risolto un problema che poteva interrompere la replica logica a causa della mancata coerenza del file binlog durante l'esecuzione di un rollback al punto di salvataggio (savepoint) per una transazione di grandi dimensioni.
- Hash delle credenziali mascherati in general-log e audit-log per impostazione predefinita utilizzando una maschera segreta coerente slow-query-log. Ciò è configurabile tramite il parametro aurora_mask_password_hashes_type.
- È stato risolto un problema per cui la durata Zero-Downtime-Restart (ZDR) veniva riportata erroneamente negli eventi osservati dal cliente.
- È stato risolto un problema che poteva causare un errore delle chiamate a mysql_rds_import_binlog_ssl_material con MySQL server ERROR 1457.
- È stato risolto un problema per cui l'inizializzazione del thread di dump poteva andare in deadkock con il thread per l'eliminazione dei log binari. Ciò può impedire la rotazione del file binlog attivo e continuare invece ad aumentare o causare problemi con le nuove connessioni di replica binlog.
- È stato risolto un problema per cui la cache delle query poteva restituire risultati obsoleti su una replica di lettura di Aurora.

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla 5.7 inclusa, oltre a quanto riportato di seguito. Per ulteriori informazioni, consulta MySQL bugs fixed by Aurora MySQL 2.x database engine updates.

- È stato risolto un problema a causa del quale il codice per la lettura delle informazioni dalle tabelle degli eventi dell'istruzione Performance Schema (ad esempio, events_statements_current) non impediva la scrittura simultanea delle informazioni sul set di caratteri. Di conseguenza, il set di caratteri di testo della query SQL potrebbe non essere valido, con conseguente chiusura del

server. Con questa correzione, un set di caratteri non valido causa il troncamento delle colonne SQL_TEXT e impedisce le chiusure del server. (Bug 23540008)

- InnoDB: backport di una correzione dei bug della community 25189192, 84038. È stato risolto un problema per cui dopo un'operazione RENAME TABLE che spostava una tabella in uno schema diverso, InnoDB non riusciva ad aggiornare la tabella del dizionario dati INNODB_SYS_DATAFILES. Ciò ha provocato un errore al riavvio che indicava che non era possibile individuare il file di dati dello spazio di tabella.
- InnoDB: risolto un problema per cui il server eliminava un indice di chiave esterna definito internamente durante l'aggiunta di un nuovo indice e tentava di utilizzare un indice secondario definito su una colonna generata virtuale come indice di chiave esterna, causando la chiusura del server. InnoDB ora consente a un vincolo di chiave esterna di fare riferimento a un indice secondario definito su una colonna generata virtuale. (Bug #23533396)
- È stato risolto un problema che si verificava quando due sessioni eseguivano contemporaneamente un'istruzione INSERT ... L'operazione ON DUPLICATE KEY UPDATE ha generato un deadlock. Durante il rollback parziale di una tupla, un'altra sessione potrebbe aggiornarla. La correzione di questo bug ripristina le correzioni per il bug 11758237, il bug 17604730 e il bug 20040791. (Bug 25966845)
- Backport di una correzione per il bug della community 27407480: è stato risolto un problema per cui i privilegi EXECUTE e ALTER ROUTINE non venivano concessi correttamente ai creatori della routine anche con automatic_sp_privileges abilitato.
- Backport di una correzione per il bug della community 24671968: risolto un problema per cui una query poteva produrre risultati errati se la clausola WHERE conteneva una sottoquery dipendente, la tabella aveva un indice secondario sulle colonne dell'elenco di selezione seguito dalle colonne nella sottoquery e `GROUP BY` o `DISTINCT` consentiva alla query di utilizzare una scansione degli indici debole.
- È stato risolto un problema a causa del quale la replica si interrompeva se veniva emessa un'istruzione di eliminazione multi tabella su più tabelle con chiavi esterne. (Bug 80821)
- È stato risolto un problema per cui in casi speciali alcuni errori slave non venivano ignorati anche con slave_skip_errors abilitato. Nei casi in cui l'apertura e il blocco di una tabella o le conversioni dei campi non sono riusciti su un server che esegue la replica basata su righe, l'errore viene considerato critico e lo stato slave_skip_errors viene ignorato. La correzione garantisce che, con slave_skip_errors abilitato, tutti gli errori segnalati durante l'applicazione di una transazione vengano gestiti correttamente. (Bug 70640, Bug 17653275)
- Risolto un problema per cui quando un'istruzione `SET PASSWORD` veniva replicata da un master MySQL 5.6 a uno slave MySQL 5.7 o da un master MySQL 5.7 con la variabile di sistema

log_builtin_as_identified_by_password impostata su ON su uno slave MySQL 5.7, anche l'hash della password veniva sottoposto ad hash prima di essere archiviato sullo slave. Il problema è stato risolto e l'hash della password replicata viene archiviato come originariamente passato allo slave. (Bug 24687073)

- È stato risolto un problema a causa del quale la serializzazione di un valore JSON costituito da un documento secondario di grandi dimensioni racchiuso in molti livelli di array JSON, oggetti o entrambi richiedeva a volte un tempo eccessivo per essere completata. (Bug 23031146)
- Le istruzioni che non possono essere analizzate (a causa, ad esempio, di errori di sintassi) non vengono più scritte nel log delle query lente. (Bug 33732907)

Confronto con Aurora MySQL Versione 1

Le seguenti caratteristiche di Amazon Aurora MySQL sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 01/11/2022 (versione 2.10.3) (obsoleta)

Versione: 2.10.3

Aurora MySQL 2.10.3 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 2.11.*, 3.01.* e 3.02.*.

È possibile aggiornare un cluster di database di Aurora MySQL 2.* esistente ad Aurora MySQL 2.10.3. Per i cluster che eseguono la versione 1 di Aurora MySQL, è possibile aggiornare un cluster di Aurora MySQL 1.23 o superiore esistente a 2.10.3. È anche possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.10.3.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Risolti i problemi di sicurezza ed CVEs elencati di seguito:

Correzioni e altri miglioramenti per ottimizzare la gestione in un ambiente gestito. Ulteriori correzioni CVE riportate di seguito:

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

Miglioramenti generali:

- È stato risolto un problema che, in rare condizioni, causa il riavvio del server di database a causa di una lunga attesa del semaforo quando il thread del rilevatore di deadlock si blocca.
- È stato risolto un problema che poteva causare la riduzione della memoria liberabile sull'istanza database quando alcune istruzioni SQL del Data Control Language (DCL) come GRANT, FLUSH PRIVILEGES e così via vengono eseguite su quell'istanza. L'uso frequente di tali istruzioni può causare una continua riduzione della memoria disponibile e il riavvio dell'istanza del database a causa di out-of-memory problemi. L'uso di tali istruzioni sull'istanza di scrittura può inoltre causare una riduzione della memoria liberabile sulle istanze di lettura.
- È stato risolto un problema a causa del quale le query sulla tabella «performance_schema.events_waits_summary_global_by_event_name» potevano diventare lente quando un'istanza di database era sottoposta a un carico intenso con l'evento di attesa «_respond_to_client» abilitato. wait/io/aurora

- È stato risolto un problema che, in rare condizioni, può causare l'arresto e il riavvio del server di database quando le transazioni vengono ripristinate parzialmente a causa di una violazione dei vincoli sugli indici secondari.
- È stato risolto un problema che, in rare condizioni, può causare il riavvio o il failover dell'istanza di scrittura quando una transazione accede a una riga che viene eliminata da un'altra transazione.
- È stato risolto un problema che, in rare condizioni, poteva causare il riavvio del database a causa di una lunga attesa del semaforo quando i thread si bloccavano. I/O
- È stato risolto un problema che poteva causare il riavvio della replica di lettura durante il failover in rare condizioni in cui era in uso il file di blocco del socket Unix.
- È stato risolto un problema a causa del quale l'eccessiva invalidazione della cache delle query causava un utilizzo della CPU e latenze sulla replica di lettura superiori al previsto, a causa del fatto che la replica di lettura doveva leggere i dati dal disco anziché dalla cache delle query.

Integrazione delle correzioni di bug di MySQL Community Edition

Questa versione include tutte le correzioni di bug della community fino alla 5.7 inclusa, oltre a quanto riportato di seguito. Per ulteriori informazioni, consulta MySQL bugs fixed by Aurora MySQL 2.x database engine updates.

- È stato risolto un problema a causa del quale il codice per la lettura delle informazioni dalle tabelle degli eventi dell'istruzione Performance Schema (ad esempio, events_statements_current) non impediva la scrittura simultanea delle informazioni sul set di caratteri. Di conseguenza, il set di caratteri di testo della query SQL potrebbe non essere valido, con conseguente chiusura del server. Con questa correzione, un set di caratteri non valido causa il troncamento delle colonne SQL_TEXT e impedisce le chiusure del server. (Bug 23540008)
- Risolto un problema per cui quando una istruzione UPDATE richiedeva una tabella temporanea con una chiave primaria più grande di 1024 byte e quella tabella veniva creata utilizzando InnoDB, il server poteva chiudersi. (Bug 25153670)
- È stato risolto un problema che si verificava quando due sessioni eseguivano contemporaneamente un'istruzione INSERT ... L'operazione ON DUPLICATE KEY UPDATE ha generato un deadlock. Durante il rollback parziale di una tupla, un'altra sessione potrebbe aggiornarla. La correzione di questo bug ripristina le correzioni per il bug 11758237, il bug 17604730 e il bug 20040791. (Bug 25966845)

Confronto con Aurora MySQL Versione 1

Le seguenti caratteristiche di Amazon Aurora MySQL sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query

- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 26/01/2022 (versione 2.10.2) (obsoleta)

Versione: 2.10.2

Aurora MySQL 2.10.2 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

È possibile aggiornare un cluster di database Aurora MySQL 2.* esistente a Aurora MySQL 2.10.0. Per i cluster che eseguono la versione Aurora MySQL 1, è possibile aggiornare un cluster Aurora MySQL 1.23 o superiore esistente a 2.10.0. È anche possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.10.0.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Risolti i problemi di sicurezza ed CVEs elencati di seguito:

Correzioni e altri miglioramenti per ottimizzare la gestione in un ambiente gestito. Ulteriori correzioni CVE riportate di seguito:

- 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

Miglioramenti generali:

- È stata aggiunta un'ottimizzazione delle prestazioni per ridurre la latenza di I/O del database nelle classi di istanza 24XL.
- È stato aggiunto il supporto per le crittografie SSL ECDHE. Per ulteriori informazioni sulla configurazione dei client per l'utilizzo di queste crittografie SSL, consulta la seguente documentazione MySQL, crittografia di protocolli di connessione
- Risolti i problemi di sicurezza relativi all'integrazione di Aurora MySQL con altri servizi AWS come Amazon S3, Amazon ML e. AWS Lambda
- È stato risolto un problema che può causare il fallimento del riavvio di un'istanza di database quando il database ha più di 1 GB circa di combinazioni di utenti e privilegi.
- È stato risolto un problema relativo alla query parallela che poteva causare la restituzione da parte del database di gruppi o ordinamento errati durante l'esecuzione di query con una clausola GROUP BY e una clausola WHERE contenente un predicato di range.
- È stato risolto un problema che causa l'inaccessibilità delle tabelle general_log e slow_log dopo l'aggiornamento della versione principale sul posto da Aurora-MySQL 1.x (compatibile con MySQL 5.6) a Aurora MySQL 2.x (compatibile con MySQL 5.7).
- È stato risolto un problema che, in rari casi, causa il riavvio dell'istanza del database quando le tabelle innodb_trx, innodb_locks o innodb_lockwaits vengono sottoposte a query mentre il database è sottoposto a carichi di lavoro pesanti. Strumenti di monitoraggio come Performance Insights possono sottoporre a query tali tabelle.
- È stato risolto un problema per cui il valore di una colonna TIMESTAMP di una riga esistente è aggiornato all'ultimo timestamp quando sono soddisfatte tutte le seguenti condizioni:

1. Esiste un'attivazione per la tabella.
2. Viene eseguito un INSERT sulla tabella che dispone di una clausola ON DUPLICATE KEY UPDATE.
3. La riga inserita causa una violazione del valore duplicato in un indice UNIQUE o PRIMARY KEY.
4. Una o più colonne sono di tipo di dati TIMESTAMP e hanno un valore predefinito di CURRENT_TIMESTAMP.

- È stato risolto un problema che, in rari casi, poteva impedire a una replica binlog di connettersi a un'istanza con binlog abilitato.
- È stato risolto un problema per cui, in rare condizioni, le transazioni non erano in grado di eseguire il commit durante l'esecuzione su un'istanza con binlog abilitato.
- È stato risolto un problema per cui non era possibile stabilire nuove connessioni a un'istanza con binlog abilitato.
- È stato risolto un problema che può causare un'eccessiva registrazione interna quando si tenta di eseguire patch e riavvii con tempi di inattività pari a zero causando il riempimento dell'archiviazione locale.
- È stato risolto un problema che causa l'interruzione di una replica binlog con un errore HA_ERR_FOUND_DUPP_KEY durante la replica di determinate istruzioni DDL e DCL. Il problema si verifica quando l'istanza sorgente è configurata con il formato di registrazione binario MIXED e il livello di isolamento READ COMMITTED o READ UNCOMMITTED.
- È stato risolto un problema per cui il I/O thread di replica binlog non era in grado di tenere il passo con l'istanza principale, quando la replica multithread era abilitata
- È stato risolto un problema per cui, in rare condizioni, un numero elevato di connessioni attive all'istanza del database poteva causare la segnalazione errata della CloudWatch CommitLatency metrica.
- È stato risolto un problema che causa il riempimento dell'archiviazione locale sulle istanze Graviton durante l'esecuzione di LOAD FROM S3 o SELECT INTO S3.
- È stato risolto un problema che causa risultati di query errati durante le query di una tabella con una chiave esterna e sono soddisfatte entrambe le seguenti condizioni:
  1. La cache delle query è abilitata
  2. Viene ripristinata una transazione con eliminazione o aggiornamento a cascata su quella tabella
- È stato risolto un problema che, in rare condizioni, può causare il riavvio delle istanze del lettore Aurora. La probabilità che questo problema si verifichi aumenta con l'aumentare del numero di ripristini dello stato precedente delle transazioni.

- È stato risolto un problema per cui il numero di occorrenze mutex “LOCK_epoch_id_master” in Performance Schema aumenta quando una sessione viene aperta e chiusa.
- È stato risolto un problema che può causare un numero crescente di deadlock per i carichi di lavoro che hanno molte transazioni che aggiornano contemporaneamente la stessa serie di righe.
- È stato risolto un problema che, in rare condizioni, può causare il riavvio delle istanze quando il volume del database aumenta fino a un multiplo di 160 GB.
- È stato risolto un problema con una query parallela che potrebbe causare il riavvio del database durante l'esecuzione di istruzioni SQL con clausola LIMIT.
- È stato risolto un problema che, in rare condizioni, può causare il riavvio dell'istanza del database quando si utilizzano transazioni XA nel livello di isolamento READ COMMITTED.
- È stato risolto un problema per cui, dopo il riavvio di un'istanza Aurora Read, l’istanza poteva riavviarsi nuovamente se, durante il riavvio, si presentava un carico di lavoro DDL pesante.
- È stato risolto un problema con la segnalazione errata del ritardo di replica del lettore Aurora.
- È stato risolto un problema che, in rare condizioni, può causare il riavvio di un'istanza di scrittura quando un controllo dell'integrità dei dati in memoria non riesce.
- È stato risolto un problema che, in rare condizioni, mostra erroneamente il grafico 'Caricamento del database' nelle sessioni di Performance Insights (PI) che utilizza attivamente la CPU anche se le sessioni hanno terminato l'elaborazione e sono inattive.
- È stato risolto un problema che, in rare condizioni, può causare il riavvio del server di database quando una query viene elaborata utilizzando Parallel Query.
- È stato risolto un problema che, in rare condizioni, può causare il riavvio dell'istanza di scrittura in un cluster di database globale principale a causa di una condizione di competizione durante la replica globale del database.
- È stato risolto un problema che può verificarsi durante il riavvio di un'istanza di database, che può causare più di un riavvio.

Integrazione delle correzioni di bug di MySQL Community Edition

- È stato risolto un problema in InnoDB per cui un errore nel codice relativo alle statistiche delle tabelle generava un'asserzione nel file di origine dict0stats.cc. (Bug 24585978)
- È stato risolto un problema per cui un indice secondario su una colonna virtuale veniva danneggiato quando l'indice veniva creato online. Per le istruzioni UPDATE, applichiamo la seguente correzione: se il valore della colonna virtuale del registro indice è impostato su NULL, allora generiamo questo valore dal registro dell'indice del cluster. (Bug 30556595)

- È stato risolto un problema in InnoDB per cui l'eliminazione di righe contrassegnate era in grado di acquisire un blocco di lettura esterno prima del completamento di un ripristino dello stato precedente parziale. Il blocco di lettura esterno impediva la conversione di un blocco implicito in un blocco esplicito durante il ripristino dello stato precedente parziale, causando un errore di asserzione. (Bug 29195848)
- È stato risolto un problema per cui i nomi host vuoti negli account potevano causare un comportamento errato del server. (Bug 28653104)
- È stato risolto un problema in InnoDB per cui un'interruzione della query durante un'attesa di blocco causava un errore. (Bug 28068293)
- È stato risolto un problema nella replica in cui le transazioni interleave potevano talvolta bloccare l'applier slave quando il livello di isolamento delle transazioni era impostato su REPEATABLE READ. (Bug 25040331)
- È stato risolto un problema che può causare l'arresto delle repliche binlog a causa del timeout di attesa del blocco. (Bug 27189701)

Confronto con Aurora MySQL Versione 1

Le seguenti caratteristiche di Amazon Aurora MySQL sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 21/10/2021 (versione 2.10.1) (obsoleta)

versione: 2.10.1

Aurora MySQL 2.10.1 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

È possibile aggiornare un cluster di database Aurora MySQL 2.* esistente a Aurora MySQL 2.10.0. Per i cluster che eseguono la versione Aurora MySQL 1, è possibile aggiornare un cluster Aurora MySQL 1.23 o superiore esistente a 2.10.0. È anche possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.10.0.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Risolti i problemi di sicurezza ed CVEs elencati di seguito:

Correzioni e altri miglioramenti per ottimizzare la gestione in un ambiente gestito. Ulteriori correzioni CVE riportate di seguito:

- 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

Miglioramenti della disponibilità:

- Aggiunta la possibilità di chiudere in modo pulito il cluster per i futuri aggiornamenti delle versioni principali.

Miglioramenti generali:

- È stato risolto un problema che poteva causare un elevato consumo della CPU nelle istanze del lettore a causa dell'eccessiva registrazione dei messaggi informativi nei file di log diagnostici interni.
- È stato risolto un problema per cui il valore di una colonna TIMESTAMP di una riga esistente è aggiornato all'ultimo timestamp quando sono soddisfatte tutte le seguenti condizioni:
  1. Esiste un'attivazione per la tabella.
  2. Viene eseguito un INSERT sulla tabella che dispone di una clausola ON DUPLICATE KEY UPDATE.
  3. La riga inserita causa una violazione del valore duplicato in un indice UNIQUE o PRIMARY KEY.
  4. Una o più colonne sono di tipo di dati TIMESTAMP e hanno un valore predefinito di CURRENT_TIMESTAMP.
- È stato risolto un problema introdotto nella versione 2.10.0 che portava l'uso della funzione json_merge a generare un codice di errore in alcuni casi. In particolare, quando la funzione json_merge viene utilizzata in un DDL contenente colonne generate, può restituire il codice di errore 1305.
- È stato risolto un problema per cui, in rare condizioni, le repliche di lettura vengono riavviate quando la cronologia degli aggiornamenti di un oggetto di grandi dimensioni viene convalidata per la visualizzazione di lettura di una transazione sulla replica di lettura.
- È stato risolto un problema che, in rare condizioni, causava il riavvio di un'istanza di scrittura quando un controllo dell'integrità dei dati in memoria non riesce.

Integrazione delle correzioni di bug della community di MySQL

- CURRENT_TIMESTAMP PRODUCE ZERI NELL'ATTIVAZIONE. (Bug 25209512)

Confronto con Aurora MySQL Versione 1

Le seguenti caratteristiche di Amazon Aurora MySQL sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.

- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 25/05/2021 (versione 2.10.0) (obsoleta)

Versione: 2.10.0

Aurora MySQL 2.10.0 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

È possibile aggiornare un cluster di database Aurora MySQL 2.* esistente a Aurora MySQL 2.10.0. Per i cluster che eseguono la versione Aurora MySQL 1, è possibile aggiornare un cluster Aurora MySQL 1.23 o superiore esistente a 2.10.0. È anche possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.10.0.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Risolti i problemi di sicurezza ed CVEs elencati di seguito:

Correzioni e altri miglioramenti per ottimizzare la gestione in un ambiente gestito. Ulteriori correzioni CVE riportate di seguito:

- 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

Nuove caratteristiche:

- La classe di istanza `db.t3.large` è ora supportata per Aurora MySQL.
- Replica dei log binari:
  - È stata introdotta la I/O cache binlog per migliorare le prestazioni di binlog riducendo la contesa tra thread di scrittura e thread di dump. Per ulteriori informazioni, consulta Ottimizzazione della replica dei log binari nella Guida per l'utente di Amazon Aurora.
  - Nella versione Aurora MySQL 2.08, abbiamo introdotto una elaborazione log binaria (binlog) migliorata per ridurre il tempo di recupero degli arresti anomali e la latenza del tempo di commit quando sono coinvolte transazioni molto grandi. Questi miglioramenti sono ora supportati per i cluster con GTID abilitato.
- Maggiore disponibilità delle istanze di lettura:
  - In precedenza, al riavvio di un'istanza di scrittura, tutte le istanze di lettura in un cluster Aurora MySQL venivano riavviate. Con il lancio di oggi, le istanze di lettura nella regione continuano a soddisfare le richieste di lettura durante il riavvio di un'istanza di scrittura, migliorando la disponibilità di lettura nel cluster. Per ulteriori informazioni, consulta la pagina relativa al riavvio di un cluster Aurora MySQL (versione 2.10 e successive) nella Guida per l'utente di Amazon Aurora.

    > **Important**
    >
    > Dopo l'aggiornamento a Aurora MySQL 2.10, il riavvio dell'istanza di scrittura non esegue il riavvio dell'intero cluster. Per riavviare l'intero cluster, ora si riavvia qualsiasi istanza di lettura nel cluster dopo il riavvio dell'istanza di scrittura.
- Migliorate le prestazioni delle letture della pagina di lettura anticipata richieste dalla tecnica di lettura anticipata logica (LRA). Ciò è stato fatto raggruppando le letture di più pagine in un'unica

richiesta inviata allo storage Aurora. Di conseguenza, le query che utilizzano l'ottimizzazione LRA vengono eseguite fino a 3 volte più velocemente.

- Riavvio e applicazione di patch senza tempi di inattività:
  - Miglioramento del riavvio senza tempi di inattività (ZDR) e dell'applicazione di patch senza tempi di inattività (ZDP) per abilitare ZDR e ZDP in una gamma più ampia di scenari, incluso il supporto aggiunto per i casi in cui sono abilitati i log binari. Inoltre, è stata migliorata la visibilità degli eventi ZDR e ZDP. Per i dettagli, consulta la documentazione: Zero-downtime restart (ZDR) per Amazon Aurora MySQL e Applicazione di patch senza tempi di inattività (ZDP) nella Guida per l'utente di Amazon Aurora.

Miglioramenti della disponibilità:

- Miglioramenti per un avvio più rapido quando il database ha un numero elevato di indici temporanei e tabelle creati durante una precedente attività DDL interrotta.
- Risolti diversi problemi relativi ai riavvii ripetuti durante il ripristino in caso di arresto di operazioni DDL interrotte, come `DROP TRIGGER`, `ALTER TABLE` e in particolare `ALTER TABLE` che modifica il tipo di partizionamento o il numero di partizioni in una tabella.
- È stato risolto un problema che poteva causare il riavvio del server durante l'elaborazione dei log DAS (Database Activity Streams).
- È stato risolto un problema che comportava la stampa di un messaggio di errore durante l'elaborazione di una query `ALTER` sulle tabelle di sistema.

Miglioramenti generali:

- È stato risolto un problema per cui la cache delle query poteva restituire risultati non obsoleti su un'istanza di lettura.
- È stato risolto un problema per cui alcuni parametri di commit Aurora non venivano aggiornati quando la variabile di sistema `innodb_flush_log_at_trx_commit` era impostata su 0 o 2.
- È stato risolto un problema per cui un risultato di query memorizzato nella cache delle query non veniva aggiornato dalle transazioni multi-istruzione.
- È stato risolto un problema che poteva causare l'aggiornamento corretto dell'ultimo timestamp dei file di log binari. Ciò poteva comportare la rimozione prematura dei file di log binari prima di raggiungere il periodo di conservazione configurato dal cliente.
- Risolto il nome e la posizione del file binlog segnalati errati da InnoDB dopo il ripristino in caso di arresto.

- Risolto un problema che poteva causare transazioni di grandi dimensioni per generare eventi binlog non corretti se il parametro `binlog_checksum` era impostato su `NONE`.
- È stato risolto un problema che causava l'interruzione di una replica binlog con un errore se la transazione replicata conteneva un'istruzione DDL e un numero elevato di modifiche di riga.
- È stato risolto un problema che causava un riavvio in un'istanza di lettura quando si lasciava cadere una tabella.
- È stato risolto un problema che causava il guasto dei connettori open source quando si tentava di utilizzare un file binlog con una transazione di grandi dimensioni.
- È stato risolto un problema che poteva causare risultati di query errati sulla colonna geometrica di grandi dimensioni dopo aver creato un indice spaziale sulla tabella con i valori geometrici di grandi dimensioni.
- Il database ora ricrea lo spazio tabella temporaneo durante il riavvio, che consente di liberare e ripristinare lo spazio di archiviazione associato.
- È stato risolto un problema che impediva il troncamento delle tabelle `performance_schema` sulle istanze di lettura Aurora.
- Risolto un problema che causava l'interruzione di una replica binlog con un errore `HA_ERR_KEY_NOT_FOUND`.
- È stato risolto un problema che causava il riavvio del database durante l'esecuzione dell'istruzione `FLUSH TABLES WITH READ LOCK`.
- È stato risolto un problema che impediva l'uso di funzioni di blocco a livello utente sulle istanze di lettura Aurora.

Integrazione delle correzioni di bug della community di MySQL

- Le transazioni interleave potevano talvolta bloccare l'applier replica quando il livello di isolamento delle transazioni era impostato su REPEATABLE READ. (Bug 25040331)
- Quando una procedure archiviata conteneva un'istruzione riferita a una vista che a sua volta faceva riferimento a un'altra vista, la procedura non poteva essere richiamata correttamente più di una volta. (Bug 87858, Bug 26864199)
- Per le query con molte condizioni `OR`, l'ottimizzatore ora è più efficiente in termini di memoria e ha meno probabilità di superare il limite di memoria imposto dalla variabile di sistema range_optimizer_max_mem_size. Inoltre, il valore predefinito per tale variabile è stato incrementato da 1.536.000 a 8.388.608. (Bug 79450, Bug 22283790)

- Replica: nella funzione `next_event()`, che viene chiamata dal thread SQL di una replica per leggere l'evento successivo dal log di relay, il thread SQL non ha rilasciato l'`relaylog.log_lock` acquisito quando si è verificato un errore (ad esempio, a causa di un log di relay chiuso), causando il blocco di tutti gli altri thread in attesa di acquisire un blocco sul log di relay. Con questa correzione, il blocco viene rilasciato prima che il thread SQL lasci la funzione nella situazione. (Bug 21697821)
- Correzione di un danneggiamento della memoria per `ALTER TABLE` con colonna virtuale. (Bug 24961167, Bug 24960450)
- Replica: non è stato possibile configurare le repliche multithread con dimensioni di coda ridotte utilizzando slave_pending_jobs_size_max se fosse stato necessario elaborare transazioni più grandi di quelle dimensioni. Qualsiasi pacchetto più grande di slave_pending_jobs_size_max è stato rifiutato con l'errore `ER_MTS_EVENT_BIGGER_PENDING_JOBS_SIZE_MAX`, anche se il pacchetto era più piccolo del limite impostato da slave_max_allowed_packet. Con questa correzione, slave_pending_jobs_size_max diventa un limite flessibile anziché un limite rigido. Se la dimensione di un pacchetto supera slave_pending_jobs_size_max ma è inferiore a slave_max_allowed_packet, la transazione viene trattenuta fino a quando tutti i dipendenti replica non hanno code vuote e quindi elaborate. Tutte le transazioni successive vengono mantenute fino al completamento della transazione di grandi dimensioni. Le dimensioni della coda per i dipendenti replica possono quindi essere limitate, pur consentendo transazioni occasionali più grandi. (Bug 21280753, Bug 77406)
- Replica: quando si utilizzava una replica multithread, gli errori dell'applicatore visualizzavano i dati dell'ID dipendente che non erano coerenti con i dati esternalizzati nelle tabelle di replica dello schema di prestazioni. (Bug 25231367)
- Replica: su una replica di replica basata su GTID in esecuzione con -gtid-mode=on, -log-bin=off e che utilizzava - slave-skip-errors, quando si verificava un errore che doveva essere ignorato non veniva aggiornata correttamente, causando una perdita di sincronia con. `Exec_Master_Log_Pos` `Exec_Master_Log_Pos` `Read_master_log_pos` Se `GTID_NEXT` non era stato specificato, la replica non aggiornerebbe mai lo stato GTID durante il rollback da una singola transazione di rendiconto. `Exec_Master_Log_Pos` non verrebbe aggiornato perché anche se la transazione è stata completata, il suo stato GTID mostrerebbe altro. La correzione rimuove il vincolo dell'aggiornamento dello stato GTID quando una transazione viene ripristinata solo se viene specificato `GTID_NEXT`. (Bug 22268777)
- Replica: un'istruzione parzialmente fallita non consumava correttamente un GTID generato automaticamente o specificato quando la registrazione binaria è stata disattivata. La correzione assicura che un DROP TABLE parzialmente non riuscito, un DROP USER parzialmente non

riuscito o un DROP VIEW parzialmente non riuscito consumino rispettivamente il GTID pertinente e lo salvino nella tabella `@@GLOBAL.GTID_EXECUTED` e `mysql.gtid_executed` quando la registrazione binaria è disabilitata. (Bug 21686749)

- Replica: le repliche che eseguono MySQL 5.7 non sono in grado di connettersi a un'origine MySQL 5.5 a causa di un errore nel recupero del server_uuid, che non fa parte di MySQL 5.5. Ciò è stato causato da modifiche nel metodo di recupero del `server_uuid`. (Bug 22748612)
- Replica: il meccanismo di salto delle transazioni GTID che ignora silenziosamente una transazione GTID precedentemente eseguita non ha funzionato correttamente per le transazioni XA. (Bug 25041920)
- Le istruzioni “>ROLLBACK XA che non sono andate a buon fine a causa di un ID transazione errato, potrebbero essere registrate nel log binario con l'ID della transazione corretto e potrebbero quindi essere eseguite da repliche di replica. Viene ora effettuato un controllo della situazione di errore prima che si verifichi la registrazione binaria e le istruzioni XA `ROLLBACK` non riuscite non vengono registrate. (Bug 26618925)
- Replica: se una replica è stata configurata utilizzando un'istruzione CHANGE MASTER TO che non specificava il nome del file di registro di origine e la posizione del registro di origine, veniva chiusa prima dell'emissione di START SLAVE, quindi riavviata con l'opzione - relay-log-recovery set, la replica non veniva avviata. Ciò accadeva perché il thread del ricevitore non era stato avviato prima del tentativo di recupero del log di inoltro, quindi nel log di inoltro non era disponibile alcun evento di rotazione del log per fornire il nome del file di log di origine e la posizione del log di origine. In questa situazione, la replica ora salta il recupero del log di inoltro e registra un avviso, quindi procede all'avvio della replica. (Bug 28996606, Bug 93397)
- Replica: nella replica basata su riga, veniva restituito un messaggio che visualizzava in modo errato le lunghezze dei campi durante la replica da una tabella con una colonna `utf8mb3` a una tabella con la stessa definizione in cui la colonna era stata definita con un set di caratteri `utf8mb4`. (Bug 25135304, Bug 83918)
- Replica: quando veniva emessa un'istruzione RESET SLAVE su una replica di replica GTIDs in uso, i file di log di inoltro esistenti venivano eliminati, ma il nuovo file di registro di inoltro sostitutivo veniva generato prima che il set di dati ricevuti GTIDs per il canale fosse cancellato. Il precedente set GTID è stato quindi scritto nel nuovo file di registro di inoltro come `PREVIOUS_GTIDS` evento, causando un errore fatale nella replica in cui si affermava che la replica aveva GTIDs più dell'origine, anche se il set gtid_executed per entrambi i server era vuoto. Ora, quando `RESET SLAVE` viene emesso, il set di dati ricevuti GTIDs viene cancellato prima della generazione del nuovo file di log di inoltro, in modo che questa situazione non si verifichi. (Bug 27411175)

- Replica: utilizzata per la replica, le transazioni che includono le istruzioni che hanno causato un errore di analisi (ER_PARSE_ERROR) non possono essere ignorate manualmente con GTIDs il metodo consigliato di iniettare una transazione vuota o sostitutiva con lo stesso GTID. Questa azione dovrebbe avere come esito che la replica identifichi il GTID come già utilizzato e quindi salti la transazione indesiderata che ha condiviso il suo GTID. Tuttavia, nel caso di un errore di analisi, poiché l'istruzione è stata analizzata prima che il GTID venisse controllato per verificare se dovesse essere saltato, il thread dell'applicatore di replica si è arrestato a causa dell'errore di analisi, anche se l'intenzione era di saltare comunque la transazione. Con questa correzione, il thread dell'applicatore di replica ora ignora gli errori di analisi se la transazione interessata deve essere saltata perché il GTID era già stato utilizzato. Notare che questa modifica di comportamento non si applica nel caso di carichi di lavoro costituiti da output di log binario prodotto da `mysqlbinlog`. In tale situazione, esisterebbe il rischio che anche una transazione con un errore di analisi che segue immediatamente una transazione saltata venga ignorata in modo silenzioso, quando dovrebbe invece generare un errore. (Bug 27638268)
- Replica: abilitare il thread SQL a GTID salta una transazione parziale. (Bug 25800025)
- Replica: quando è stato fornito un parametro di timeout negativo o frazionario`WAIT_UNTIL_SQL_THREAD_AFTER_GTIDS()`, il server si è comportato in modi imprevisti. Con questa correzione:
  - Un valore di timeout frazionario viene letto così com'è, senza arrotondamento.
  - Un valore di timeout negativo viene rifiutato con un errore se il server è in modalità SQL rigorosa; se il server non è in una modalità SQL rigorosa, il valore restituisce `NULL` immediatamente la funzione senza alcuna attesa e quindi emette un avviso. (Bug 24976304, Bug 83537)
- Replica: se la funzione `WAIT_FOR_EXECUTED_GTID_SET()` è stata utilizzata con un valore di timeout che include una parte frazionaria (ad esempio 1,5), un errore nella logica di casting significava che il timeout è stato arrotondato al secondo intero più vicino e a zero per valori inferiori a 1 secondo (ad esempio 0,1). La logica di casting è stata ora corretta in modo che il valore di timeout venga applicato come originariamente specificato senza arrotondamento. Grazie a Dirkjan Bussink per il contributo. (Bug 29324564, Bug 94247)
- GTIDs Se abilitato, XA COMMIT su una transazione XA disconnessa all'interno di una transazione con più dichiarazioni generava un'asserzione. (Bug 22173903)
- Replica: è stata generata un'asserzione nelle build di debug se è stata emessa un'istruzione ROLLBACK XA per un identificatore di transazione sconosciuto quando il valore gtid_next era stato impostato manualmente. Il server ora non tenta di aggiornare lo stato GTID se un'istruzione `ROLLBACK` XA non va a buon fine con un errore. (Bug 27928837, Bug 90640)

- Risolve il problema dell'ordinamento errato quando vengono utilizzate più `CASE` funzioni nella clausola `ORDER BY` (Bug 22810883).
- Alcune query che utilizzavano l'ordinamento potevano accedere a una colonna non inizializzata durante l'ottimizzazione e causare la chiusura del server. (Bug 27389294)

Confronto con Aurora MySQL Versione 1

Le seguenti caratteristiche di Amazon Aurora MySQL sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB

- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 12/11/2021 (versione 2.09.3) (obsoleta)

Versione: 2.09.3

Aurora MySQL 2.09.3 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

È possibile aggiornare un cluster di database Aurora MySQL 2.* esistente a Aurora MySQL 2.10.0. Per i cluster che eseguono la versione Aurora MySQL 1, è possibile aggiornare un cluster Aurora MySQL 1.23 o superiore esistente a 2.10.0. È anche possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.10.0.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore utilizzando la Console di gestione AWS, la AWS CLI o l'API di Amazon RDS.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Correzioni di sicurezza:

Correzioni e altri miglioramenti per ottimizzare la gestione in un ambiente gestito. Ulteriori correzioni CVE riportate di seguito:

- 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

Miglioramenti della disponibilità:

- Introdotta un'ottimizzazione che può ridurre la contesa per le query eseguite su tabelle in information_schema.
- Aggiungi il supporto per i codici SSL ECDHE.

Miglioramenti generali:

- È stato risolto un problema che, in rare condizioni, può causare il riavvio di un'istanza di scrittura quando un controllo dell'integrità dei dati in memoria non riesce.
- È stato risolto un problema che, in rare condizioni, può causare il riavvio dell'istanza del database quando il volume del cluster si espande mentre è abilitata la registrazione binaria.
- È stata risolta una rara condizione di competizione durante il riavvio di un'istanza di database, che può causare più di un riavvio.
- È stato risolto un problema che può causare il fallimento del riavvio di un'istanza di database quando il database ha un numero elevato di combinazioni di utenti e privilegi.
- È stato risolto un problema con una query parallela che può causare il riavvio del database durante l'esecuzione di istruzioni SQL con clausola LIMIT.
- È stato risolto un problema con la segnalazione errata del ritardo di replica di Aurora.
- È stato risolto un problema che può causare l'inaccessibilità delle tabelle general_log e slow_log dopo l'aggiornamento della versione principale sul posto da Aurora-MySQL 1.x (basato su MySQL 5.6) a Aurora-MySQL 2.x (basato su MySQL 5.7).
- È stato risolto un problema che, in rari casi, può causare il riavvio dell'istanza del database quando le tabelle innodb_trx, innodb_locks o innodb_lockwaits vengono interrogate mentre il database è sottoposto a carichi di lavoro pesanti. Strumenti e funzionalità di monitoraggio come Performance Insights possono interrogare tali tabelle.
- È stato risolto un problema che può causare il riavvio di un'istanza di database quando viene eseguita l'istruzione SQL «FLUSH TABLES WITH READ LOCK».
- È stato risolto un problema per il quale il processo di eliminazione di InnoDB si interrompe durante l'eliminazione di un'istanza di lettore, con conseguente aumento temporaneo della lunghezza dell'elenco della cronologia.
- È stato risolto un problema relativo alla query parallela che può causare il riavvio del database durante l'esecuzione di un'istruzione SQL su una tabella contenente una colonna virtuale.
- È stato risolto un problema relativo alla query parallela che può causare la restituzione da parte del database di gruppi o ordinamento errati durante l'esecuzione di query con la clausola GROUP BY e una clausola WHERE contenente un predicato di range.
- È stato risolto un problema nella query parallela che, in rare condizioni, può causare il riavvio del database durante l'esecuzione di istruzioni SQL con funzioni JSON.
- È stato risolto un problema che, in rare condizioni, può causare il riavvio dell'istanza di scrittura nel cluster di database globale principale a causa di una condizione di competizione durante la replica globale del database.

- È stato risolto un problema che può causare l'interruzione di una replica Binlog con un errore HA_ERR_FOUND_DUPP_KEY durante la replica di determinate istruzioni DDL e DCL. Il problema si verifica quando l'istanza sorgente è configurata con il formato di registrazione binario MIXED e il livello di isolamento READ COMMITTED o READ UNCOMMITTED.
- È stato risolto un problema che, in rare condizioni, può causare il riavvio dell'istanza del database quando si utilizzavano transazioni XA nel livello di isolamento READ COMMITTED.
- È stato risolto un problema per cui il valore di una colonna TIMESTAMP di una riga esistente viene aggiornato all'ultimo timestamp quando sono soddisfatte tutte le seguenti condizioni: 1. Esiste un'attivazione per la tabella; 2. Viene eseguito un INSERT sulla tabella con una clausola ON DUPLICATE KEY UPDATE; 3. La riga inserita può causare una violazione del valore duplicato in un indice UNIQUE o PRIMARY KEY; e 4. Una o più colonne sono di tipo di dati TIMESTAMP e hanno un valore predefinito di CURRENT_TIMESTAMP.
- È stato risolto un problema che, in rare condizioni, può causare il riavvio di un'istanza del lettore a causa di un'elaborazione errata del controllo.
- È stato risolto un problema che può causare il riavvio dell'istanza del lettore quando l'istanza writer aumenta il volume del database per superare limiti specifici delle dimensioni del volume.
- È stato risolto un problema che può causare tempi di riavvio più lunghi per le istanze di database utilizzando volumi di cluster clonati.
- È stato risolto un problema per il quale il riavvio di un'istanza di database potrebbe non riuscire una o più volte dopo l'esecuzione di un'operazione TRUNCATE TABLE sull'istanza writer.
- È stato risolto un problema che, in rare condizioni, può causare il riavvio dell'istanza del database.
- È stato risolto un problema che, in rare condizioni, può causare il riavvio dell'istanza di scrittura quando il volume del database aumenta fino a un multiplo di 160 GB.

Integrazione delle correzioni di bug della community di MySQL

- Bug #23533396 - Durante l’aggiunta di un nuovo indice, il server ha eliminato un indice di chiave esterna definito internamente e ha tentato di utilizzare un indice secondario definito su una colonna generata virtuale come indice di chiave esterna, causando un’uscita dal server. InnoDB ora consente a un vincolo di chiave esterna di fare riferimento a un indice secondario definito su una colonna generata virtuale.
- Bug #29550513 - Replica: un problema di blocco nella funzione WAIT_FOR_EXECUTED_GTID_SET() può causare il blocco del server in determinate circostanze. Il problema ora è stato corretto.

Confronto con Aurora MySQL Versione 1

Le seguenti caratteristiche di Amazon Aurora MySQL sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online

- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 26/02/2021 (versione 2.09.2) (obsoleta)

Versione: 2.09.2

Aurora MySQL 2.09.2 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

È possibile aggiornare un cluster di database Aurora MySQL 2.* esistente a Aurora MySQL 2.09.2. Per i cluster che eseguono la versione Aurora MySQL 1, è possibile aggiornare un cluster Aurora MySQL 1.23 o superiore esistente a 2.09.2. È anche possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.09.2.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore utilizzando la Console di gestione AWS, la AWS CLI o l'API di Amazon RDS.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Nuove caratteristiche:

- I cluster Aurora MySQL ora supportano le seguenti istanze EC2 R6g alimentate da processori Graviton2 basati su AWS ARM:,,,,,,. `r6g.large` `r6g.xlarge` `r6g.2xlarge` `r6g.4xlarge` `r6g.8xlarge` `r6g.12xlarge` `r6g.16xlarge` Per ulteriori informazioni, consulta la pagina relativa alle classi di istanze database di Aurora nella Guida per l'utente di Amazon Aurora.

Correzioni di sicurezza:

Correzioni e altri miglioramenti per ottimizzare la gestione in un ambiente gestito. Ulteriori correzioni CVE riportate di seguito:

- 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

Miglioramenti della disponibilità:

- Risolto un problema introdotto nella versione 2.09.0 che può causare una latenza di scrittura elevata durante il ridimensionamento del volume di storage del cluster.
- Risolto un problema nella funzione di ridimensionamento dinamico che poteva causare il riavvio delle repliche di lettura di Aurora.
- Risolto un problema che poteva causare tempi di inattività più lunghi durante l'aggiornamento da 1.23.* a 2.09.*.
- Risolto un problema per il quale una DDL o una DML poteva causare il riavvio del motore durante una richiesta di precaricamento della pagina.
- È stato risolto un problema che causava l'interruzione di una replica binlog con un errore se la transazione replicata conteneva un'istruzione DDL e un numero elevato di modifiche di riga.
- Risolto un problema per cui un database che funge da replica binlog poteva riavviarsi durante la replica di un evento DDL sulla tabella mysql `time_zone`.

- Risolto un problema che poteva causare transazioni di grandi dimensioni per generare eventi binlog non corretti se il parametro `binlog_checksum` era impostato su `NONE`.
- Risolto un problema che causava l'interruzione di una replica binlog con un errore `HA_ERR_KEY_NOT_FOUND`.
- Miglioramento della stabilità complessiva.

Confronto con Aurora MySQL Versione 1

Le caratteristiche Amazon Aurora MySQL seguenti sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per AWS Lambda richiamare funzioni in modo sincrono. Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 11/12/2020 (versione 2.09.1) (obsoleta)

Versione: 2.09.1

Aurora MySQL 2.09.1 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

È possibile aggiornare un cluster di database Aurora MySQL 2.* esistente a Aurora MySQL 2.09.1. Per i cluster che eseguono la versione Aurora MySQL 1, è possibile aggiornare un cluster Aurora MySQL 1.23 o superiore esistente a 2.09.1. È anche possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.09.1.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore utilizzando la Console di gestione AWS, la AWS CLI o l'API RDS.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Correzioni di sicurezza:

Correzioni e altri miglioramenti per ottimizzare la gestione in un ambiente gestito. Ulteriori correzioni CVE riportate di seguito:

- 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

Modifiche non compatibili:

Questa versione introduce una modifica dell'autorizzazione che influisce sul comportamento del `mysqldump` comando. Gli utenti devono disporre del `PROCESS` privilegio per accedere alla `INFORMATION_SCHEMA.FILES` tabella. Per eseguire il `mysqldump` comando senza alcuna modifica, concedere il `PROCESS` privilegio all'utente del database a cui si connette il `mysqldump` comando. Inoltre puoi eseguire il`mysqldump`comando con l'`--no-tablespaces`opzione. Con questa opzione,l'`mysqldump` output non include alcuna`CREATE`

`LOGFILE GROUP`o`CREATE TABLESPACE`istruzioni. In tal caso, il `mysqldump` comando non accede alla `INFORMATION_SCHEMA.FILES` tabella e non è necessario che tu conceda l' `PROCESS` autorizzazione.

Miglioramenti della disponibilità:

- Risolto un problema che poteva causare il blocco di una sessione client quando il motore del database rilevava un errore durante la lettura o la scrittura nella rete.
- Corretta una perdita di memoria nella funzione di ridimensionamento dinamico, introdotta in 2.09.0.

Database globali.

- Risolti diversi problemi per cui le repliche di una regione secondaria di un database globale potevano riavviarsi quando veniva effettuato l'aggiornamento alla versione 2.09.0 mentre il writer di una regione primaria era in una versione precedente.

Integrazione delle correzioni di bug della community di MySQL

- Replica: le transazioni interleave potevano talvolta bloccare l'applier slave quando il livello di isolamento delle transazioni era impostato su REPEATABLE READ. (Bug 25040331)
- Per una tabella con una colonna TIMESTAMP o DATETIME con un valore di default di CURRENT_TIMESTAMP, la colonna poteva essere inizializzata su `0000-00-00 00:00:00` se la tabella aveva un'attivazione `BEFORE INSERT`. (Bug 25209512, bug 84077)
- Per un'istruzione INSERT per la quale l'elenco `VALUES` produceva valori per la seconda riga o successive utilizzando una sottoquery contenente un join, il server poteva uscire dopo la mancata risoluzione dei privilegi richiesti. (Bug 23762382)

Confronto con Aurora MySQL Versione 1

Le caratteristiche Amazon Aurora MySQL seguenti sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.

- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 17/09/2020 (versione 2.09.0) (obsoleta)

Versione: 2.09.0

Aurora MySQL 2.09.0 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

È possibile ripristinare uno snapshot da Aurora MySQL 1.23.* a Aurora MySQL 2.09.0. Hai anche la possibilità di aggiornare i cluster di database Aurora MySQL 2.* esistenti ad Aurora MySQL 2.09.0. Non è possibile aggiornare un cluster Aurora MySQL 1.23.* esistente direttamente a 2.09.0; tuttavia, è possibile ripristinarne lo snapshot in Aurora MySQL 2.09.0.

Important

I miglioramenti apportati allo storage Aurora in questa versione limitano i percorsi di aggiornamento disponibili da Aurora MySQL 1.* a Aurora MySQL 2.09. Quando si aggiorna un cluster Aurora MySQL 1.* alla 2.09, è necessario eseguire l'aggiornamento da Aurora MySQL 1.23.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite l'API Console di gestione AWS RDS. AWS CLI

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Note

Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Nuove caratteristiche:

- Con questa versione, puoi creare istanze database MySQL di Amazon Aurora con un massimo di 128 TiB di storage. Il nuovo limite di storage è un aumento rispetto al precedente 64 TiB. La dimensione di storage di 128 TiB supporta database più grandi. Questa funzionalità non è supportata su istanze Small (db.t2 o db.t3). Un singolo spazio di tabella non può superare i 64 TiB a causa delle limitazioni di InnoDB con dimensioni di pagina di 16 KB.

  Aurora avvisa l'utente quando la dimensione del volume del cluster è vicina a 128 TiB, in modo da poter intervenire prima di raggiungere il limite di dimensione. Gli avvisi vengono visualizzati nel registro mysql e negli eventi RDS in Console di gestione AWS.
- È ora possibile attivare o disattivare la query parallela per un cluster esistente modificando il valore del parametro cluster DB `aurora_parallel_query`. Non è necessario utilizzare l'impostazione `parallelquery` per il parametro `--engine-mode` durante la creazione del cluster.

  La query parallela è ora espansa per essere disponibile in tutte le regioni in cui Aurora MySQL è disponibile.

  Sono disponibili numerosi altri miglioramenti alle funzionalità e modifiche alle procedure per l'aggiornamento e l'abilitazione di query parallele in un cluster Aurora. Per ulteriori informazioni, consulta Utilizzo di query in parallelo per Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Aurora ridimensiona dinamicamente lo spazio di storage del cluster. Con il ridimensionamento dinamico, lo spazio di storage per il cluster DB Aurora diminuisce automaticamente quando si rimuovono i dati dal cluster DB. Per ulteriori informazioni, consulta Storage scaling nella Guida per l'utente di Amazon Aurora.

  > **Note**
  >
  > La funzionalità di ridimensionamento dinamico viene implementata in fasi nelle regioni in AWS cui Aurora è disponibile. A seconda della regione in cui si trova il cluster, questa funzionalità potrebbe non essere ancora disponibile. Per ulteriori informazioni, consulta l'annuncio delle novità.

Correzioni con alta priorità:

- Bug Backport of Community #27659490: SELECT USING DYNAMIC RANGE AND INDEX MERGE USE TOO MUCH MEMORY (OOM)
- Bug #26881508: MYSQL #1: DISABLE_ABORT_ON_ERROR IN AUTH_COMMON.H
- Bug Backport of Community #24437124: POSSIBLE BUFFER OVERFLOW ON CREATE TABLE
- Backport di Bug #27158030: INNODB ONLINE ALTER CRASHES WITH CONCURRENT DML
- Bug #29770705: SERVER CRASHED WHILE EXECUTING SELECT WITH SPECIFIC WHERE CLAUSE
- Backport di BUG #26502135: MYSQLD SEGFAULTS IN MDL_CONTEXT::TRY_ACQUIRE_LOCK_IMPL
- Backport di Bug #26935001: ALTER TABLE AUTO_INCREMENT TRIES TO READ INDEX FROM DISCARDED TABLESPACE
- Bug #28491099: [FATAL] MEMORY BLOCK IS INVALID | INNODB: ASSERTION FAILURE: UT0UT.CC:670
- Bug #30499288: GCC 9.2.1 REPORTS A NEW WARNING FOR OS_FILE_GET_PARENT_DIR
- Bug #29952565 where MYSQLD GOT SIGNAL 11 WHILE EXECUTING A QUERY(UNION + ORDER BY + SUB-QUERY)
- Bug #30628268: OUT OF MEMORY CRASH
- Bug #30441969: BUG #29723340: MYSQL SERVER CRASH AFTER SQL QUERY WITH DATA ? AST
- Bug #30569003: 5.7 REPLICATION BREAKAGE WITH SYNTAX ERROR WITH GRANT MANAGEMENT
- Bug #29915479: RUNNING COM_REGISTER_SLAVE WITHOUT COM_BINLOG_DUMP CAN RESULTS IN SERVER EXIT
- Bug #30569003: 5.7 REPLICATION BREAKAGE WITH SYNTAX ERROR WITH GRANT MANAGEMENT
- Bug #29915479: RUNNING COM_REGISTER_SLAVE WITHOUT COM_BINLOG_DUMP CAN RESULTS IN SERVER EXIT
- Bug #20712046: SHOW PROCESSLIST AND PERFORMANCE_SCHEMA TABLES DO NOT MASK PASSWORD FROM QUERY
- Backport bug #18898433: EXTREMELY SLOW PERFORMANCE WITH OUTER JOINS AND JOIN BUFFER (fixed in 5.7.21). Le query con molti join rimanenti erano lente se veniva utilizzato il buffering di join (ad esempio, utilizzando l'algoritmo di loop nidificato del blocco). (Bug #18898433, Bug #72854)"

- Bug di Backport #26402045: MYSQLD CRASHES ON QUERY (fixed in MySQL 5.7.23). Alcuni casi di materializzazione delle query secondarie potrebbero causare l'uscita del server. Queste query ora producono un errore che suggerisce che la materializzazione deve essere disabilitata. (Bug #26402045)
- [Backport da MySQL] users other than rdsadmin is disallowed to update pfs table in the reader replica.
- Fix the issue where the customer can not update the perfschema in the reader replica
- Bug #26666274: INFINITE LOOP IN PERFORMANCE SCHEMA BUFFER CONTAINER
- Bug #26997096: relay_log_space value is not updated in a synchronized manner so that its value sometimes much higher than the actual disk space used by relay logs.
- BUG #25082593: FOREIGN KEY VALIDATION DOESN'T NEED TO ACQUIRE GAP LOCK IN READ COMMITTED
- CVE-2019-2731
- CVE-2018-2645
- CVE-2019-2581
- CVE-2018-2787
- CVE-2019-2482
- CVE-2018-2640
- CVE-2018-2784
- CVE-2019-2628
- CVE-2019-2911
- CVE-2019-2628
- CVE-2018-3284
- CVE-2018-3065
- CVE-2019-2537
- CVE-2019-2948
- CVE-2019-2434
- CVE-2019-2420

Miglioramenti della disponibilità:

- Attiva la correzione ABA di gestione blocco per impostazione predefinita.

- È stato risolto un problema nella gestione dei blocchi per cui una race condition può causare la condivisione di un blocco da parte di due transazioni, causando il riavvio del database.
- È stato risolto un problema durante la creazione di una tabella temporanea con formato di riga compresso che poteva causare un riavvio.
- Correggere il valore predefinito di `table_open_cache` su istanze 16XL e 24XL che potrebbero causare failover ripetuti e un elevato utilizzo della CPU su classi di istanze di grandi dimensioni (R4/R5-16XL, R5-12XL, R5-24XL). Questo ha avuto un impatto su 2.07.x.
- Risolto un problema per cui il ripristino di un cluster da Amazon S3 a Aurora MySQL versione 2.08.0 richiede più tempo del previsto quando il backup S3 non includeva la tabella `mysql.host`.
- Risolto un problema che poteva causare ripetuti failover a causa degli aggiornamenti delle colonne virtuali con indici secondari.
- È stato risolto un problema relativo alla gestione della memoria del blocco delle transazioni con transazioni di scrittura a esecuzione prolungata che causavano il riavvio del database.
- Sono stati risolti diversi problemi per cui il motore poteva bloccarsi durante l'applicazione di patch a tempo zero durante il controllo del punto sicuro per l'applicazione delle patch.
- Risolto un problema per ignorare il redo logging per le tabelle temporanee, che in precedenza causava un arresto anomalo.
- È stata risolta una condizione di gara nel gestore delle serrature tra l'uccisione connection/query e la sessione interrotta.
- Risolto un problema per cui il database potrebbe arrestarsi se si tratta di una replica binlog e ricevere un evento DDL per la tabella MySQL `time_zone`.

Database globali.

- La vista `INFORMATION_SCHEMA.REPLICA_HOST_STATUS` MySQL in una regione secondaria ora mostra le voci per le repliche appartenenti a tale regione.
- Corretti errori di query imprevisti che potevano verificarsi in un'area secondaria di DB globale dopo problemi temporanei di connettività di rete tra le aree primarie e secondarie.
-

Query parallela:

- È stato risolto il piano `EXPLAIN` per una query di query parallela, che non è corretto per una semplice query a tabella singola.

- Corretto il self-deadlatch che può verificarsi quando è attivata la query parallela.

Miglioramenti generali:

- Esporta in S3 ora supporta la parola chiave `ENCRYPTION`.
- Il parametro `aurora_binlog_replication_max_yield_seconds` ora ha un valore massimo di 36.000. Il precedente valore massimo accettato era 45. Questo parametro funziona solo quando il parametro `aurora_binlog_use_large_read_buffer` è impostato su 1.
- Modificato il comportamento da mappare `MIXED` `binlog_format` a `ROW` invece che `STATEMENT` durante l'esecuzione di `LOAD DATA FROM INFILE | S3`.
- Risolto un problema per cui una replica binlog connessa a un binlog primario di Aurora MySQL potrebbe mostrare dati incompleti quando il primario ha eseguito `LOAD DATA FROM S3` e `binlog_format` è impostato su `STATEMENT`.
- Aumento della lunghezza massima consentita per le variabili di sistema di audit `server_audit_incl_users` e `server_audit_excl_users` da 1024 byte a 2000 byte.
- È stato risolto un problema per cui gli utenti potrebbero perdere l'accesso al database quando si abbassava il parametro `max_connections` nel gruppo di parametri quando le connessioni correnti sono maggiori del valore impostato.
- È stato risolto un problema nei flussi di attività dati in cui il carattere escape non veniva inserito correttamente in una singola citazione e in una barra rovesciata.

Integrazione delle correzioni di bug della community di MySQL

- Bug #27659490: SELECT USING DYNAMIC RANGE AND INDEX MERGE USE TOO MUCH MEMORY(OOM)
- Bug #26881508: MYSQL #1: DISABLE_ABORT_ON_ERROR IN AUTH_COMMON.H
- Bug #24437124: POSSIBLE BUFFER OVERFLOW ON CREATE TABLE
- Bug #27158030: INNODB ONLINE ALTER CRASHES WITH CONCURRENT DML
- Bug #29770705: SERVER CRASHED WHILE EXECUTING SELECT WITH SPECIFIC WHERE CLAUSE
- Bug #26502135: MYSQLD SEGFAULTS IN MDL_CONTEXT::TRY_ACQUIRE_LOCK_IMPL
- Bug #26935001: ALTER TABLE AUTO_INCREMENT TRIES TO READ INDEX FROM DISCARDED TABLESPACE

- Bug #28491099: [FATAL] MEMORY BLOCK IS INVALID | INNODB: ASSERTION FAILURE: UT0UT.CC:670
- Bug #30499288: GCC 9.2.1 REPORTS A NEW WARNING FOR OS_FILE_GET_PARENT_DIR
- Bug #29952565: where MYSQLD GOT SIGNAL 11 WHILE EXECUTING A QUERY(UNION + ORDER BY + SUB-QUERY)
- Bug #30628268: OUT OF MEMORY CRASH
- Bug #30441969: BUG #29723340: MYSQL SERVER CRASH AFTER SQL QUERY WITH DATA ? AST
- Bug #30569003: 5.7 REPLICATION BREAKAGE WITH SYNTAX ERROR WITH GRANT MANAGEMENT
- Bug #29915479: RUNNING COM_REGISTER_SLAVE WITHOUT COM_BINLOG_DUMP CAN RESULTS IN SERVER EXIT
- Bug #30569003: 5.7 REPLICATION BREAKAGE WITH SYNTAX ERROR WITH GRANT MANAGEMENT
- Bug #29915479: RUNNING COM_REGISTER_SLAVE WITHOUT COM_BINLOG_DUMP CAN RESULTS IN SERVER EXIT
- Bug #20712046: SHOW PROCESSLIST AND PERFORMANCE_SCHEMA TABLES DO NOT MASK PASSWORD FROM QUERY
- Bug #18898433: EXTREMELY SLOW PERFORMANCE WITH OUTER JOINS AND JOIN BUFFER (fixed in 5.7.21)
- Bug #26402045: MYSQLD CRASHES ON QUERY (fixed in MySQL 5.7.23)
- Bug #23103937: PS_TRUNCATE_ALL_TABLES () NON FUNZIONA IN MODALITÀ SUPER_READ
- Bug #26666274: INFINITE LOOP IN PERFORMANCE SCHEMA BUFFER CONTAINER
- Bug #26997096: relay_log_space value is not updated in a synchronized manner so that its value sometimes much higher than the actual disk space used by relay logs. (https://github.com/mysql/mysql-server/commit/78f25d2809ad457e81f90342239c9bc32a36cdfa)
- Bug #25082593: FOREIGN KEY VALIDATION DOESN'T NEED TO ACQUIRE GAP LOCK IN READ COMMITTED
- Bug #24764800: REPLICATION FAILING ON SLAVE WITH XAER_RMFAIL ERROR.
- Bug #81441: AVVISO SU LOCALHOST DURANTE L'UTILIZZO SKIP-NAME-RESOLVE.

Confronto con Aurora MySQL Versione 1

Le caratteristiche Amazon Aurora MySQL seguenti sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare AWS Lambda funzioni in modo sincrono. Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online

- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 06/01/2022 (versione 2.08.4) (obsoleta)

Versione: 2.08.4

Aurora MySQL 2.08.4 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

È possibile aggiornare un cluster di database Aurora MySQL 2.* esistente a Aurora MySQL 2.10.0. Per i cluster che eseguono la versione Aurora MySQL 1, è possibile aggiornare un cluster Aurora MySQL 1.23 o superiore esistente a 2.10.0. È anche possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.10.0.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore utilizzando la Console di gestione AWS, la AWS CLI o l'API di Amazon RDS.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Correzioni di sicurezza e miglioramenti generali:

- Risolti i problemi di sicurezza relativi all'integrazione di Aurora MySQL con altri servizi AWS come Amazon S3, Amazon ML e. AWS Lambda
- È stato risolto un problema per cui il valore di una colonna TIMESTAMP di una riga esistente viene aggiornato all'ultimo timestamp quando sono soddisfatte tutte le seguenti condizioni: 1. Esiste un'attivazione per la tabella; 2. Viene eseguito un INSERT sulla tabella con una clausola ON DUPLICATE KEY UPDATE; 3. La riga inserita può causare una violazione del valore duplicato in un indice UNIQUE o PRIMARY KEY; e 4. Una o più colonne sono di tipo di dati TIMESTAMP e hanno un valore predefinito di CURRENT_TIMESTAMP.
- È stato risolto un problema che, in rare condizioni, causa il riavvio di un'istanza di scrittura quando un controllo dell'integrità dei dati in memoria non riesce.
- È stato risolto un problema con una query parallela che potrebbe causare il riavvio del database durante l'esecuzione di istruzioni SQL con clausola LIMIT.

Confronto con Aurora MySQL Versione 1

Le caratteristiche Amazon Aurora MySQL seguenti sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 12/11/2020 (versione 2.08.3) (obsoleta)

Versione: 2.08.3

Aurora MySQL 2.08.3 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

È possibile aggiornare i cluster di database Aurora MySQL 2.* esistenti direttamente a Aurora MySQL 2.08.3. È possibile aggiornare un cluster Aurora MySQL 1.* esistente direttamente a 2.07.3 o versione successiva e quindi eseguire direttamente l'aggiornamento a 2.08.3.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite l'API Console di gestione AWS RDS. AWS CLI

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Correzioni di sicurezza:

Correzioni e altri miglioramenti per ottimizzare la gestione in un ambiente gestito. Ulteriori correzioni CVE riportate di seguito:

- 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

Modifiche non compatibili:

Questa versione introduce una modifica dell'autorizzazione che influisce sul comportamento del `mysqldump` comando. Gli utenti devono disporre del `PROCESS` privilegio per accedere alla `INFORMATION_SCHEMA.FILES` tabella. Per eseguire il `mysqldump` comando senza alcuna modifica, concedere il `PROCESS` privilegio all'utente del database a cui si connette il `mysqldump` comando. Inoltre puoi eseguire il`mysqldump`comando con l'`--no-tablespaces`opzione. Con questa opzione,l'`mysqldump` output non include alcuna`CREATE LOGFILE GROUP`o`CREATE TABLESPACE`istruzioni. In tal caso, il `mysqldump` comando non accede alla `INFORMATION_SCHEMA.FILES` tabella e non è necessario che tu conceda l' `PROCESS` autorizzazione.

Integrazione delle correzioni di bug della community di MySQL

- Bug 23762382 - INSERISCI VALORI QUERY CON JOIN IN SELECT CAUSA UN COMPORTAMENTO ERRATO.
- Bug 25209512 - CURRENT_TIMESTAMP PRODUCE ZERI NEL TRIGGER.

Confronto con Aurora MySQL Versione 1

Le caratteristiche Amazon Aurora MySQL seguenti sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 28/08/2020 (versione 2.08.2) (obsoleta)

Versione: 2.08.2

Aurora MySQL 2.08.2 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

È possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.08.2. È anche possibile aggiornare i cluster database di Aurora MySQL 2.* in Aurora MySQL 2.08.2. Non è possibile aggiornare un cluster Aurora MySQL 1.* esistente direttamente a 2.08.2; tuttavia, è possibile ripristinarne lo snapshot in Aurora MySQL 2.08.2. Per ulteriori

informazioni sul ripristino degli snapshot, consulta Ripristino da uno snapshot cluster database nella Guida per l'utente di Amazon Aurora.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite l'API Console di gestione AWS RDS. AWS CLI

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Correzioni importanti:

- È stato risolto un problema che potrebbe causare un'interruzione non pianificata e influire sulla disponibilità del database.

Correzioni disponibilità:

- Risolto un problema per cui il database Aurora MySQL poteva riavviarsi se si tratta di una replica binlog e replicare un evento DDL sulla tabella mysql `time_zone`.

Confronto con Aurora MySQL Versione 1

Le caratteristiche Amazon Aurora MySQL seguenti sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.

- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 18/06/2020 (versione 2.08.1) (obsoleta)

Versione: 2.08.1

Aurora MySQL 2.08.1 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

È possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.08.1. È anche possibile aggiornare i cluster database di Aurora MySQL 2.* in Aurora MySQL 2.08.1. Non è possibile aggiornare un cluster Aurora MySQL 1.* esistente direttamente a 2.08.1; tuttavia, è possibile ripristinarne lo snapshot in Aurora MySQL 2.08.1.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite l'API Console di gestione AWS RDS. AWS CLI

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Nuove caratteristiche:

- Inoltro di scrittura del database globale. In un Aurora Global Database è ora possibile eseguire determinate operazioni di scrittura, ad esempio istruzioni DML, mentre si è connessi a un cluster secondario. Le operazioni di scrittura vengono inoltrate al cluster primario e le eventuali modifiche vengono replicate ai cluster secondari. Per ulteriori informazioni, consulta Utilizzo dell'inoltro di scrittura in un database globale Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Correzioni per la stabilità generale:

- Risolto un problema per cui il ripristino di un cluster da Amazon S3 a Aurora MySQL versione 2.08.0 richiedeva più tempo del previsto se il backup S3 non includeva la tabella `mysql.host`.

Confronto con Aurora MySQL Versione 1

Le caratteristiche Amazon Aurora MySQL seguenti sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi

- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 02/06/2020 (versione 2.08.0) (obsoleta)

Versione: 2.08.0

Aurora MySQL 2.08.0 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

È possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.08.0. È anche possibile aggiornare i cluster database di Aurora MySQL 2.* in Aurora MySQL 2.08.0. Non è possibile aggiornare un cluster Aurora MySQL 1.* esistente direttamente a 2.08.0; tuttavia, è possibile ripristinarne lo snapshot in Aurora MySQL 2.08.0.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite l'API Console di gestione AWS RDS. AWS CLI

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Nuove caratteristiche:

- Elaborazione log binaria (binlog) migliorata per ridurre il tempo di recupero degli arresti anomali e la latenza del tempo di commit quando sono coinvolte transazioni molto grandi.
- Avvio della funzione DAS (Database Activity Streams) per Aurora MySQL. Questa funzionalità fornisce un flusso di near-real-time dati dell'attività del database nel database relazionale per aiutarti a monitorare l'attività. Per ulteriori informazioni, consulta Monitoraggio di Amazon Aurora tramite i flussi di attività del database nella Guida per l'utente di Amazon Aurora.
- File di fuso orario aggiornati per supportare l'ultima modifica del fuso orario del Brasile.
- Sono state introdotte nuove parole chiave in SQL per utilizzare la funzionalità hash join per una tabella and/or interna specifica:`HASH_JOIN`,`HASH_JOIN_PROBING`, e. `HASH_JOIN_BUILDING` Per ulteriori dettagli, consulta Suggerimenti Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Introdotto il supporto di hint order join in Aurora MySQL 5.7 eseguendo il backporting di una funzionalità MySQL 8.0. I nuovi suggerimenti sono `JOIN_FIXED_ORDER`, `JOIN_ORDER`, `JOIN_PREFIX` e `JOIN_SUFFIX`. Per una documentazione dettagliata sul supporto degli hint order join, vedere WL #9158: Join order hints.
- Aurora Machine Learning ora supporta le funzioni definite dall'utente con `MEDIUMINT` come tipo di ritorno.
- La procedura archiviata `lambda_async()` ora supporta tutti i caratteri `utf8` MySQL.

Correzioni con alta priorità:

- Risolto un problema per il quale un'istanza DB del lettore restituiva risultati incompleti per una query FTS dopo che la tabella `INFORMATION_SCHEMA.INNODB_SYS_TABLES` è stata interrogata sull'istanza DB writer.

- CVE-2019-5443
- CVE-2019-3822

Miglioramenti della disponibilità:

- Risolto un problema che causava il riavvio del database dopo l'esecuzione di un'istruzione multi-query che accede a più tabelle o database con la cache delle query attivata.
- È stata risolta una condizione di competizione nel gestore di blocchi che causava un riavvio o un failover del database durante il rollback della transazione.
- Risolto un problema che causava il riavvio o il failover del database quando più connessioni tentavano di aggiornare la stessa tabella con un indice di ricerca full-text.
- È stato risolto un problema che poteva attivare il riavvio o il failover del database durante un comando `kill session`. Se riscontri questo problema, contatta l' AWS assistenza per attivare questa correzione sulla tua istanza.
- Risolto un problema che causava il riavvio dell'istanza DB del lettore durante una transazione multi-istruzione con più istruzioni `SELECT` e un carico di lavoro di scrittura pesante sull'istanza DB writer con `AUTOCOMMIT` abilitato.
- Risolto un problema che causava il riavvio dell'istanza DB del lettore dopo l'esecuzione di query di lunga durata mentre l'istanza DB writer era sotto un carico di lavoro di scrittura OLTP pesante.

Miglioramenti generali:

- Miglioramento del tempo di ripristino del database e della latenza di commit per le transazioni di lunga durata quando binlog è abilitato.
- Migliorato l'algoritmo per generare statistiche migliori per stimare conteggi di valori distinti su colonne indicizzate, incluse le colonne con distribuzioni di dati distorte.
- Riduzione dei tempi di risposta e dell'utilizzo della CPU delle query di join che accedono alle tabelle temporanee MyISAM e dei risultati versati nello storage locale.
- Risolto un problema che impediva il ripristino di snapshot Aurora MySQL 5.6 con nomi di database o tabelle contenenti spazi in un nuovo cluster Aurora MySQL 5.7.
- Incluse le informazioni sulle transazioni della vittima quando il deadlock viene risolto in `show engine innodb status`.
- Risolto un problema che causava il blocco delle connessioni quando client di più versioni diverse sono connessi allo stesso database e accedevano alla cache delle query.

- È stata risolta una perdita di memoria risultante da più invocazioni del flusso di lavoro Zero-Downtime Patch (ZDP) o Zero-Downtime Restart (ZDR) per tutta la durata di un'istanza di database.
- È stato corretto un messaggio di errore nelle operazioni Zero-Downtime Patch (ZDP) o Zero-Downtime Restart (ZDR) che indica erroneamente che l'ultima transazione è stata interrotta se il flag di commit automatico è disattivato.
- Risolto un problema nelle operazioni Zero-Downtime Patch (ZDP) che poteva portare a un messaggio di errore del server durante il ripristino delle variabili di sessione utente nel nuovo processo di database.
- Risolto un problema nelle operazioni ZDP (Zero Downtime Patch) che poteva causare errori intermittenti del database quando durante l'applicazione delle patch sono presenti query a lungo termine.
- È stato risolto un problema per cui le query che includevano una funzione Aurora Machine Learning restituivano messaggi di errore vuoti a causa di una risposta di errore gestita in modo errato da servizi di Machine Learning come Amazon AI SageMaker e Amazon Comprehend.
- È stato risolto un problema nella funzionalità di out-of-memory monitoraggio che non rispettava un valore personalizzato del parametro. `table_definition_cache`
- Il messaggio di errore «Esecuzione query interrotta» viene restituito se viene interrotta una query di Aurora Machine Learning. In precedenza, è stato restituito il messaggio generico «Errore interno nell'elaborazione della richiesta ML».
- Risolto un problema che poteva causare un timeout di connessione a uno slave binlog quando il parametro `slave_net_timeout` è inferiore al parametro `aurora_binlog_replication_max_yield_seconds` e c'è un carico di lavoro contenuto sul cluster master binlog.
- Miglioramento del monitoraggio dell'avanzamento del recupero binlog mediante l'output di messaggi informativi nel log degli errori con una frequenza di un messaggio al minuto.
- Risolto un problema per il quale le transazioni attive non venivano segnalate dalla query `SHOW ENGINE INNODB STATUS`.

Integrazione delle correzioni di bug della community di MySQL

- Bug #25289359: un blocco della cache full-text eseguito quando i dati vengono sincronizzati non è stato rilasciato se la dimensione della cache full-text superava il limite di dimensione della cache full-text.

- Bug #29138644: la modifica manuale dell'ora di sistema durante l'esecuzione del server MySQL ha causato ritardi nel thread del cleaner delle pagine.
- Bug #25222337: un nome di campo colonna virtuale NULL in un indice virtuale ha causato l'uscita del server durante un confronto tra nomi di campo che si verifica durante il popolamento di colonne virtuali interessate da un vincolo di chiave esterna.
- Bug #25053286: l'esecuzione di una procedura archiviata contenente una query che ha effettuato l'accesso a una vista potrebbe allocare memoria che non è stata liberata fino al termine della sessione.
- Bug #25586773: l'esecuzione di una procedura archiviata contenente un'istruzione che ha creato una tabella dal contenuto di alcune istruzioni SELECT potrebbe causare una perdita di memoria.
- Bug #28834208: durante l'applicazione di log, dopo un'operazione OPTIMIZE TABLE, InnoDB non ha popolato le colonne virtuali prima di controllare gli aggiornamenti dell'indice delle colonne virtuali.
- Bug #26666274: ciclo infinito nel container del buffer dello schema delle prestazioni a causa di un overflow di numeri interi non firmati a 32 bit.

Confronto con Aurora MySQL Versione 1

Le caratteristiche Amazon Aurora MySQL seguenti sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 2023-08-15 (versione 2.07.10) (obsoleta)

Versione: 2,07.10

Aurora MySQL 2.07.10 è disponibile a livello generale. Le versioni 2.07 di Aurora MySQL sono compatibili con MySQL 5.7.12. Per ulteriori informazioni sulle modifiche apportate alla community, consulta Changes in MySQL 5.7.12 (2016-04-11, General Availability).

Le versioni Aurora MySQL attualmente supportate sono 2.07.*, 2.11.*, 3.01.*, 3.02.*, 3.03.* e 3.04.*.

È possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.07.10. È inoltre possibile aggiornare i cluster di database Aurora MySQL 2.* esistenti ad Aurora MySQL 2.07.10. L'aggiornamento locale è disponibile per i cluster Aurora MySQL

1.* ad Aurora MySQL 2.* (consulta Aggiornamento da Aurora MySQL 1.x a 2.x). È anche disponibile per i cluster Aurora MySQL 2.* ad Aurora MySQL 3.* (consulta Aggiornamento da Aurora MySQL 2.x a 3.x).

Immediatamente dopo l'aggiornamento locale della versione del motore ad Aurora MySQL 2.07.10, viene applicato automaticamente un aggiornamento del sistema operativo a tutte le istanze interessate sulle classi di istanze database db.r4, db.r5, db.t2 e db.t3, se le istanze eseguono una vecchia versione del sistema operativo. In un cluster database Multi-AZ, tutte le istanze di lettura applicano prima l'aggiornamento del sistema operativo. Al termine dell'aggiornamento del sistema operativo sulla prima istanza di lettura, si verifica un failover e l'istanza di scrittura precedente viene aggiornata.

> **Note**
>
> L'aggiornamento del sistema operativo non viene applicato automaticamente ai database globali di Aurora durante gli aggiornamenti delle versioni principali.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Risolti i problemi di sicurezza ed CVEs elencati di seguito:

Correzioni e altri miglioramenti per ottimizzare la gestione in un ambiente gestito. Ulteriori correzioni CVE riportate di seguito:

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

- È stato risolto un problema per cui gli eventi segnalati durante l'elaborazione delle rotazioni dei log di audit potevano non essere scritti nel log di audit.
- Sono stati aggiornati i codici SSL predefiniti utilizzati da Aurora MySQL per escludere i valori DES CBC3 - -SHA meno sicuri dal parametro del database SSL_CIPHER. Se riscontri problemi di connessione SSL dovuti alla rimozione del codice DES- CBC3 -SHA, utilizza un codice sicuro applicabile tra le seguenti informazioni, Configurazione delle suite di crittografia per le connessioni ai cluster Aurora MySQL DB.
- OpenSSL è stato aggiornato alla versione 1.0.2zh.

Miglioramenti generali:

- È stato aggiunto il supporto per le cifrature SSL ECDHE-RSA che utilizzano chiavi di dimensioni inferiori per la crittografia.
- È stato risolto un problema di gestione della memoria durante l'esecuzione di query con hash join.

Funzionalità non supportate in Aurora MySQL versione 2

Le seguenti funzionalità non sono attualmente supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP).
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Istruzione SQL `CREATE TABLESPACE`
- Plugin replica gruppi

- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Protocollo X

Aggiornamenti del motore di database Aurora MySQL 2023-05-04 (versione 2.07.9) (obsoleta)

Versione: 2.07.9

Aurora MySQL 2.07.9 è disponibile a livello generale. Le versioni 2.07 di Aurora MySQL sono compatibili con MySQL 5.7.12. Per ulteriori informazioni sulle modifiche apportate alla community, consulta Changes in MySQL 5.7.12 (2016-04-11, General Availability).

Le versioni di Aurora MySQL attualmente supportate sono 2.07.*, 2.11.*, 3.01.*, 3.02.* e 3.03.*.

È possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.07.9. È inoltre possibile aggiornare i cluster di database Aurora MySQL 2.* esistenti ad Aurora MySQL 2.07.9. L'aggiornamento locale è disponibile per i cluster Aurora MySQL 1.* ad Aurora MySQL 2.* (consulta Aggiornamento da Aurora MySQL 1.x a 2.x). È anche disponibile per i cluster Aurora MySQL 2.* ad Aurora MySQL 3.* (consulta Aggiornamento da Aurora MySQL 2.x a 3.x).

Immediatamente dopo l'aggiornamento locale della versione del motore ad Aurora MySQL 2.07.9, viene applicato automaticamente un aggiornamento del sistema operativo a tutte le istanze interessate sulle classi di istanze database db.r4, db.r5, db.t2 e db.t3, se le istanze eseguono una vecchia versione del sistema operativo. In un cluster database Multi-AZ, tutte le istanze di lettura applicano prima l'aggiornamento del sistema operativo. Al termine dell'aggiornamento del sistema operativo sulla prima istanza di lettura, si verifica un failover e l'istanza di scrittura precedente viene aggiornata.

> **Note**
>
> L'aggiornamento del sistema operativo non viene applicato automaticamente ai database globali di Aurora durante gli aggiornamenti delle versioni principali.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Risolti i problemi di sicurezza CVEs elencati di seguito:

Correzioni e altri miglioramenti per ottimizzare la gestione in un ambiente gestito. Ulteriori correzioni CVE riportate di seguito:

- CVE-2022-32221

Miglioramenti della disponibilità:

- È stato risolto un problema a causa del quale la rotazione dei log di audit avanzato poteva ridurre la memoria liberabile, con conseguente riavvio dell'istanza database.
- È stato risolto un problema che poteva verificarsi durante il riavvio del database e che impediva l'avvio corretto del database per un periodo prolungato.

Miglioramenti generali:

- È stato risolto un problema che, in rare condizioni, può causare il riavvio delle istanze quando il volume del database aumenta fino a un multiplo di 160 GB.

Funzionalità non supportate in Aurora MySQL versione 2

Le seguenti funzionalità non sono attualmente supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP).
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 16/06/2022 (versione 2.07.8) (obsoleta)

Versione: 2.07.8

Aurora MySQL 2.07.8 è disponibile a livello generale. Le versioni 2.* di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.* di Aurora MySQL sono compatibili con MySQL 5.6.

> **Note**
>
> Questa versione è designata come versione di supporto a lungo termine (LTS). Per ulteriori informazioni, consulta Versioni con supporto a lungo termine (Long-Term Support, LTS) di Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

È possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.07.8. È inoltre possibile aggiornare i cluster di database Aurora MySQL 2.* esistenti ad Aurora MySQL 2.07.8. L'aggiornamento locale è disponibile per i cluster Aurora MySQL 1.* ad Aurora MySQL 2.* (consulta Aggiornamento da Aurora MySQL 1.x a 2.x). È anche disponibile per i cluster Aurora MySQL 2.* ad Aurora MySQL 3.* (consulta Aggiornamento da Aurora MySQL 2.x a 3.x).

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite l'API Console di gestione AWS RDS. AWS CLI

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Correzioni di sicurezza:

Correzioni e altri miglioramenti per ottimizzare la gestione in un ambiente gestito. Ulteriori correzioni CVE riportate di seguito:

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

Miglioramenti generali:

- È stato risolto un problema che, in rari casi, causa il riavvio del server di database quando il thread del rilevatore di deadlock si blocca a causa di una condizione di gara.

Integrazione delle correzioni di bug della community di MySQL

- Quando una istruzione UPDATE richiedeva una tabella temporanea con una chiave primaria più grande di 1024 byte e quella tabella veniva creata utilizzando InnoDB, il server poteva chiudersi. (Bug 25153670)

Confronto con Aurora MySQL Versione 1

Le caratteristiche Amazon Aurora MySQL seguenti sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 24/11/2021 (versione 2.07.7) (obsoleta)

Versione: 2.07.7

Aurora MySQL 2.07.7 è disponibile a livello generale. Le versioni 2.* di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.* di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

È possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.07.7. È inoltre possibile aggiornare i cluster di database Aurora MySQL 2.*

esistenti ad Aurora MySQL 2.07.7. Non è possibile aggiornare un cluster Aurora MySQL 1.* esistente direttamente a 2.07.7; tuttavia, è possibile ripristinarne lo snapshot in Aurora MySQL 2.07.7.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite l'API Console di gestione AWS RDS. AWS CLI

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Correzioni di sicurezza:

Correzioni e altri miglioramenti per ottimizzare la gestione in un ambiente gestito. Ulteriori correzioni CVE riportate di seguito:

- CVE-2019-17543
- CVE-2019-2960

Miglioramenti generali:

- Risolti i problemi di sicurezza relativi all'integrazione di Aurora MySQL con altri servizi AWS come Amazon S3, Amazon ML, Lambda.
- È stato risolto un problema con la segnalazione errata di un ritardo di replica di Aurora.
- È stato risolto un problema che può causare il fallimento del riavvio di un'istanza di database quando il database ha un numero elevato di combinazioni di utenti e privilegi.
- È stato risolto un problema che può causare l'inaccessibilità delle tabelle general_log e slow_log dopo l'aggiornamento locale della versione principale da Aurora-MySQL 1.x (basato su MySQL 5.6) ad Aurora MySQL 2.x (basato su MySQL 5.7).

- È stato risolto un problema che, in rare condizioni, può causare il riavvio di un'istanza del lettore a causa di un'elaborazione errata del controllo.
- È stato risolto un problema che, in rare condizioni, mostra il grafico “Caricamento del database” nelle sessioni di Performance Insights (PI) utilizzando attivamente la CPU anche se le sessioni hanno terminato l'elaborazione e sono inattive.
- È stato risolto un problema con una query parallela che può causare il riavvio del database durante l'esecuzione di istruzioni SQL con clausola LIMIT.
- È stato risolto un problema per cui il valore di una colonna TIMESTAMP di una riga esistente è aggiornato all'ultimo timestamp quando sono soddisfatte tutte le seguenti condizioni: 1. Esiste un’attivazione per la tabella; 2. viene eseguito un INSERT sulla tabella con una clausola ON DUPLICATE KEY UPDATE; 3. la riga inserita può causare una violazione del valore duplicato in un indice UNIQUE o PRIMARY KEY; e 4. una o più colonne sono di tipo di dati TIMESTAMP e hanno un valore predefinito di CURRENT_TIMESTAMP.
- È stato risolto un problema che, in rare condizioni, può causare il riavvio dell'istanza del database quando si utilizzavano transazioni XA nel livello di isolamento READ COMMITTED.

Confronto con Aurora MySQL Versione 1

Le caratteristiche Amazon Aurora MySQL seguenti sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 02/09/2021 (versione 2.07.6) (obsoleta)

Versione 2.07.6

Aurora MySQL 2.07.6 è disponibile a livello generale. Le versioni 2.* di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.* di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

È possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.07.6. È inoltre possibile aggiornare i cluster di database Aurora MySQL 2.* esistenti ad Aurora MySQL 2.07.6. Non è possibile aggiornare un cluster Aurora MySQL 1.* esistente direttamente a 2.07.6; tuttavia, è possibile ripristinarne lo snapshot in Aurora MySQL 2.07.6.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite l'API Console di gestione AWS RDS. AWS CLI

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Integrazione delle correzioni di bug della community di MySQL

- L'INSERIMENTO DI REGISTRI DI DIMENSIONI 64K RICHIEDE TROPPO TEMPO. (Bug#23031146)

Confronto con Aurora MySQL Versione 1

Le caratteristiche Amazon Aurora MySQL seguenti sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa

degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 06/07/2021 (versione 2.07.5) (obsoleta)

Versione: 2.07.5

Aurora MySQL 2.07.5 è disponibile a livello generale. Le versioni 2.* di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.* di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

È possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.07.5. È anche possibile aggiornare i cluster database di Aurora MySQL 2.* in Aurora MySQL 2.07.5. Non è possibile aggiornare un cluster Aurora MySQL 1.* esistente direttamente a 2.07.5; tuttavia, è possibile ripristinarne lo snapshot in Aurora MySQL 2.07.5.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite l'API Console di gestione AWS RDS. AWS CLI

> **Note**
>
> Questa versione è designata come versione di supporto a lungo termine (LTS). Per ulteriori informazioni, consulta Versioni con supporto a lungo termine (Long-Term Support, LTS) di Aurora MySQL nella Guida per l'utente di Amazon Aurora.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Miglioramenti della disponibilità:

- È stato risolto un problema a causa del quale i blocchi a livello utente non erano consentiti su una replica Aurora.
- È stato risolto un problema che poteva causare il riavvio di un database durante l'utilizzo di transazioni XA nel livello di isolamento `READ COMMITTED`.
- Lunghezza massima consentita estesa fino a 2000 per parametri `server_audit_incl_users` e `server_audit_excl_users` globali.

Confronto con Aurora MySQL Versione 1

Le caratteristiche Amazon Aurora MySQL seguenti sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.

- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 03/04/2021 (versione 2.07.4) (obsoleta)

Versione: 2.07.4

Aurora MySQL 2.07.4 è disponibile a livello generale. Le versioni 2.* di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.* di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

È possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.07.4. È inoltre possibile aggiornare i cluster di database Aurora MySQL 2.* esistenti ad Aurora MySQL 2.07.4. Non è possibile aggiornare un cluster Aurora MySQL 1.* esistente direttamente a 2.07.4; tuttavia, è possibile ripristinarne lo snapshot in Aurora MySQL 2.07.4.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite l'API Console di gestione AWS RDS. AWS CLI

> **Note**
>
> Questa versione è designata come versione di supporto a lungo termine (LTS). Per ulteriori informazioni, consulta Versioni con supporto a lungo termine (Long-Term Support, LTS) di Aurora MySQL nella Guida per l'utente di Amazon Aurora.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Correzioni di sicurezza:

- 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

Miglioramenti della disponibilità:

- Risolto un problema che poteva causare il blocco di un client in caso di errore di rete durante la lettura o la scrittura di un pacchetto di rete.
- Miglioramento dei tempi di riavvio del motore in alcuni casi dopo una DDL interrotta.
- Risolto un problema per il quale una DDL o una DML poteva causare il riavvio del motore durante una richiesta di precaricamento della pagina.
- È stato risolto un problema a causa del quale una replica poteva riavviarsi durante l'esecuzione di una scansione inversa di una table/index replica Aurora Read.
- Risolto un problema nell'operazione di clonazione del cluster che poteva far sì che la clonazione impiegasse più tempo.
- Risolto un problema che poteva causare il riavvio di un database quando si utilizzava l'ottimizzazione delle query parallele per le colonne geo-spaziali.
- Risolto un problema che causava l'interruzione di una replica binlog con un errore `HA_ERR_KEY_NOT_FOUND`.

Integrazione delle correzioni di bug della community di MySQL

- Risolto un problema nel parser ngram full-text quando si trattava di token contenenti " (spazio), '%' o ','. Se utilizzano il parser ngram, i clienti dovrebbero rigenerare i loro indici FTS. (Bug 25873310)
- Risolto un problema che poteva causare il riavvio del motore durante l'esecuzione della query con viste SQL nidificate. (Bug 27214153, bug 26864199)

Confronto con Aurora MySQL Versione 1

Le caratteristiche Amazon Aurora MySQL seguenti sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.

- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 10/11/2020 (versione 2.07.3) (obsoleta)

Versione: 2.07.3

Aurora MySQL 2.07.3 è disponibile a livello generale. Le versioni 2.* di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.* di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

È possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.07.3. È inoltre possibile aggiornare i cluster di database Aurora MySQL 2.* esistenti ad Aurora MySQL 2.07.3. Non è possibile aggiornare un cluster Aurora MySQL 1.* esistente direttamente a 2.07.3; tuttavia, è possibile ripristinarne lo snapshot in Aurora MySQL 2.07.3.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite l'API Console di gestione AWS RDS. AWS CLI

> **Note**
>
> Questa versione è designata come versione di supporto a lungo termine (LTS). Per ulteriori informazioni, consulta Versioni con supporto a lungo termine (Long-Term Support, LTS) di Aurora MySQL nella Guida per l'utente di Amazon Aurora.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Correzioni di sicurezza:

Correzioni e altri miglioramenti per ottimizzare la gestione in un ambiente gestito.

- 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

Modifiche non compatibili:

Questa versione introduce una modifica dell'autorizzazione che influisce sul comportamento del `mysqldump` comando. Gli utenti devono disporre del `PROCESS` privilegio per accedere alla `INFORMATION_SCHEMA.FILES` tabella. Per eseguire il `mysqldump` comando senza alcuna modifica, concedere il `PROCESS` privilegio all'utente del database a cui si connette il `mysqldump` comando. Inoltre puoi eseguire il`mysqldump`comando con l'`--no-tablespaces`opzione. Con questa opzione,l'`mysqldump` output non include alcuna`CREATE LOGFILE GROUP`o`CREATE TABLESPACE`istruzioni. In tal caso, il `mysqldump` comando non accede alla `INFORMATION_SCHEMA.FILES` tabella e non è necessario che tu conceda l' `PROCESS` autorizzazione.

Miglioramenti della disponibilità:

- È stata risolta una condizione di gara nel gestore dei blocchi tra l'eliminazione di a connection/query e la chiusura della sessione con conseguente riavvio del database.
- Risolto un problema che causa il riavvio del database dopo l'esecuzione di un'istruzione multi-query che accede a più tabelle o database con la cache delle query attivata.
- Risolto un problema che poteva causare ripetuti riavvii a causa degli aggiornamenti delle colonne virtuali con indici secondari.

Integrazione delle correzioni di bug della community di MySQL

- InnoDB: le transazioni XA simultanee che venivano eseguite correttamente allo stadio di preparazione XA sul master entravano in conflitto quando riprodotte sullo slave, causando un timeout di attesa di blocco nel thread dell'applier. Il conflitto era dovuto all'intervallo di blocco GAP che differiva quando le transazioni venivano riprodotte serialmente sullo slave. Per evitare

questo tipo di conflitto, i blocchi GAP eseguiti dalle transazioni XA nel livello di isolamento READ COMMITTED vengono ora rilasciati (e non più ereditati) quando le transazioni XA raggiungono la fase di preparazione. (Bug 27189701, bug 25866046)

- InnoDB: un blocco gap veniva preso inutilmente durante la convalida della chiave esterna durante l'utilizzo del livello di isolamento READ COMMITTED. (Bug 25082593)
- Replica: quando si utilizzavano transazioni XA, se si verificava un timeout di attesa di blocco o un deadlock per il thread (SQL) dell'applier su uno slave di replica, il tentativo automatico non funzionava. La causa era che mentre il thread SQL eseguiva il rollback, non eseguiva il rollback della transazione XA. Ciò significa che quando la transazione veniva ritentata, il primo evento era XA START, che non era valido poiché la transazione XA era già in corso, causando un errore XAER_RMFAIL. (Bug 24764800)
- Replica: le transazioni interleave potevano talvolta bloccare l'applier slave quando il livello di isolamento delle transazioni era impostato su REPEATABLE READ. (Bug 25040331)
- Replica: il valore restituito da un'istruzione SHOW SLAVE STATUS per la dimensione totale combinata di tutti i file di log di inoltro esistenti (Relay_Log_Space) poteva diventare molto più grande dello spazio su disco effettivo utilizzato dai file di log di inoltro. Il I/O thread non bloccava la variabile mentre aggiornava il valore, quindi il thread SQL poteva eliminare automaticamente un file di log di inoltro e scrivere un valore ridotto prima che il I/O thread completasse l'aggiornamento del valore. Il thread I/O scriveva quindi il suo calcolo delle dimensioni originali, ignorando l'aggiornamento del thread SQL e aggiungendo pertanto lo spazio per il file eliminato. Il valore Relay_Log_Space è ora bloccato durante gli aggiornamenti per impedire aggiornamenti simultanei e garantire un calcolo accurato. (Bug 26997096, bug 87832)
- Per un'istruzione INSERT per la quale l'elenco VALUES produceva valori per la seconda riga o successive utilizzando una sottoquery contenente un join, il server poteva uscire dopo la mancata risoluzione dei privilegi richiesti. (Bug 23762382)
- Per una tabella con una colonna TIMESTAMP o DATETIME con un valore di default di CURRENT_TIMESTAMP, la colonna poteva essere inizializzata su `0000-00-00 00:00:00` se la tabella aveva un'attivazione `BEFORE INSERT`. (Bug 25209512, bug 84077)
- Un'uscita dal server poteva essere causata da tentativi simultanei di più thread di registrare e annullare la registrazione degli oggetti dello schema delle prestazioni dei metadati. (Bug 26502135)
- L'esecuzione di una procedura archiviata contenente un'istruzione che ha creato una tabella dal contenuto di alcune istruzioni SELECT potrebbe causare una perdita di memoria. (Bug 25586773)
- L'esecuzione di una procedura archiviata contenente una query che ha effettuato l'accesso a una visualizzazione poteva allocare memoria che non veniva liberata fino al termine della sessione. (Bug 25053286)

- Alcuni casi di materializzazione delle query secondarie potrebbero causare l'uscita del server. Queste query ora producono un errore che suggerisce che la materializzazione deve essere disabilitata. (Bug #26402045)
- Le query con molti join rimanenti erano lente se veniva utilizzato il buffering di join (ad esempio, utilizzando l'algoritmo di loop nidificato del blocco). (Bug 18898433, bug 72854)
- L'ottimizzatore saltava la seconda colonna in un indice composito durante l'esecuzione di un join interno con una clausola `LIKE` per la seconda colonna. (Bug 28086754)

Confronto con Aurora MySQL Versione 1

Le caratteristiche Amazon Aurora MySQL seguenti sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare AWS Lambda funzioni in modo sincrono. Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 17/04/2020 (versione 2.07.2) (obsoleta)

Versione: 2.07.2

Aurora MySQL 2.07.2 è disponibile a livello generale. Le versioni 2.* di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.* di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

È possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.07.2. È anche possibile aggiornare i cluster database di Aurora MySQL 2.* in Aurora MySQL 2.07.2. Non è possibile aggiornare un cluster Aurora MySQL 1.* esistente direttamente a 2.07.2; tuttavia, è possibile ripristinarne lo snapshot in Aurora MySQL 2.07.2.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite l'API Console di gestione AWS RDS. AWS CLI

> **Note**
>
> Questa versione è designata come versione di supporto a lungo termine (LTS). Per ulteriori informazioni, consulta Versioni con supporto a lungo termine (Long-Term Support, LTS) di Aurora MySQL nella Guida per l'utente di Amazon Aurora.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Correzioni di sicurezza:

- CVE-2016-8287
- CVE-2016-5634

Correzioni con alta priorità:

- È stato risolto un problema per cui la clonazione richiedeva più tempo in alcuni cluster di database con carichi di scrittura elevati.
- Risolto un problema che poteva causare query su un'istanza database di lettura con piani di esecuzione, utilizzando indici secondari per restituire dati non salvati. Il problema è limitato ai dati interessati da operazioni di Data Manipulation Language (DML) che modificano le colonne chiave dell'indice primario o secondario.

Miglioramenti generali:

- Risolto un problema che causava un lento ripristino di un cluster DB Aurora 1.x contenente indici FTS (Full Text Search) in un cluster DB Aurora 2.x.
- Risolto un problema che causava il ripristino più lento di uno snapshot di database Aurora 1.x contenente tabelle partizionate con caratteri speciali nei nomi di tabella in un cluster Aurora 2.x DB.
- Risolto un problema che causava errori durante l'interrogazione dei log delle query lenti e dei log generali nelle istanze database del lettore.

Integrazione delle correzioni di bug della community di MySQL

- Bug n. 23104498: Risolto un problema in Performance Schema nella segnalazione della memoria totale utilizzata. (https://github.com/mysql/mysql-server/commit/20b6840df5452f47313c6f9a6ca075bfbc00a96b)
- Bug n. 22551677: Risolto un problema in Performance Schema che poteva causare il crash del motore del database nel tentativo di portarlo offline. (https://github.com/mysql/mysql-server/commit/05e2386eccd32b6b444b900c9f8a87a1d8d531e9)
- Bug n. 23550835, Bug n. 23298025, Bug n. 81464: Risolto un problema in Performance Schema che causava un crash del motore di database dovuto al superamento della capacità di un buffer interno. (https://github.com/mysql/mysql-server/commit/b4287f93857bf2f99b18fd06f555bbe5b12debfc)

Confronto con Aurora MySQL Versione 1

Le caratteristiche Amazon Aurora MySQL seguenti sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 23/12/2019 (versione 2.07.1) (obsoleta)

Versione: 2.07.1

Aurora MySQL 2.07.1 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

È possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.07.1. È anche possibile aggiornare i cluster database di Aurora MySQL 2.* in Aurora

MySQL 2.07.1. Non è possibile aggiornare un cluster Aurora MySQL 1.* esistente direttamente a 2.07.1; tuttavia, è possibile ripristinarne lo snapshot in Aurora MySQL 2.07.1.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite l'API Console di gestione AWS RDS. AWS CLI

> **Note**
>
> Questa versione non è attualmente disponibile nelle seguenti AWS regioni: AWS GovCloud (Stati Uniti orientali) [us-gov-east-1], (Stati Uniti occidentali) [-1], Cina AWS GovCloud (Ningxia) [us-gov-westcn-northwest-1], Asia Pacifico (Hong Kong) [ap-east-1] e Medio Oriente (Bahrein) [me-south-1]. Non appena disponibile verrà comunicato l'annuncio.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Correzioni con alta priorità:

- Corretta una perdita di memoria lenta in un sottosistema Aurora specifico di analisi e registrazione del database che riduce la memoria libera.

Correzioni per la stabilità generale:

- Corretto un arresto anomalo durante l'esecuzione di una query complessa che coinvolgeva join multi-tabella e aggregazione che utilizzano tabelle intermedie internamente.

Confronto con Aurora MySQL versione 1

Le caratteristiche Amazon Aurora MySQL seguenti sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 25/11/2019 (versione 2.07.0) (obsoleta)

Versione: 2.07.0

Aurora MySQL 2.07.0 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

È possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.07.0. È anche possibile aggiornare i cluster database di Aurora MySQL 2.* in Aurora MySQL 2.07.0. Non è possibile aggiornare un cluster Aurora MySQL 1.* esistente direttamente a 2.07.0; tuttavia, è possibile ripristinarne lo snapshot in Aurora MySQL 2.07.0.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite l'API Console di gestione AWS RDS. AWS CLI

> **Note**
>
> Questa versione non è attualmente disponibile nelle seguenti AWS regioni: AWS GovCloud (Stati Uniti orientali) [us-gov-east-1], (Stati Uniti occidentali) [-1], Cina AWS GovCloud (Ningxia) [us-gov-westcn-northwest-1], Asia Pacifico (Hong Kong) [ap-east-1], Medio Oriente (Bahrein) [me-south-1] e Sud America (San Paolo) [sa-east-1] east-1]. Non appena disponibile verrà comunicato l'annuncio.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Nuove caratteristiche:

- I database globali ora consentono di aggiungere aree di replica secondarie di sola lettura per i cluster di database distribuiti in queste AWS regioni: Stati Uniti orientali (Virginia settentrionale) [us-east-1], Stati Uniti orientali (Ohio) [us-east-2], Stati Uniti occidentali (California settentrionale) [us-west-1], Stati Uniti occidentali (Oregon) [us-west-2], Europa (Irlanda) [eu-west-1] Europa (Londra) [eu-west-2], Europa (Parigi) [eu-west-3], Asia Pacifico (Tokyo) [ap-northeast-1], Asia Pacifico (Seoul) [ap-northeast-2], Asia Pacifico (Singapore) [ap-southeast-1], Asia Pacifico (Singapore) [ap-southeast-1], Asia Pacifico (Sydney) [ap-southeast-2], Canada (Central) [ca-central-1], Europa (Francoforte) [eu-central-1] e Asia Pacifico (Mumbai) [ap-south-1].
- L'apprendimento automatico di Amazon Aurora è un'integrazione altamente ottimizzata tra il database Aurora MySQL e i servizi di machine learning (ML). AWS Aurora Machine Learning consente agli sviluppatori di aggiungere una varietà di previsioni basate su ML alle loro applicazioni di database invocando i modelli di ML con il linguaggio di programmazione SQL che già conoscono e utilizzano per lo sviluppo del database, senza dover creare integrazioni personalizzate o apprendere strumenti separati. Per ulteriori informazioni, consulta Utilizzo delle funzionalità di Machine Learning (ML) con Amazon Aurora.
- Aggiunto supporto per il livello di isolamento ANSI `READ COMMITTED` nelle repliche di lettura. Questo livello di isolamento consente l'esecuzione di query a esecuzione prolungata nella replica di lettura senza influire sull'elevato throughput delle scritture sul nodo di scrittura. Per ulteriori informazioni, consulta Livelli di isolamento di Aurora MySQL.

Correzioni importanti:

- CVE-2019-2922
- CVE-2019-2923
- CVE-2019-2924
- CVE-2019-2910

Correzioni con alta priorità:

- È stato risolto un problema nel recupero DDL che causava tempi di inattività prolungati del database. I cluster che non sono disponibili dopo l'esecuzione dell'istruzione drop su più tabelle, ad esempio `DROP TABLE t1, t2, t3`, devono essere aggiornati a questa versione.

- È stato risolto un problema nel recupero DDL che causava tempi di inattività prolungati del database. I cluster che non sono disponibili dopo l'esecuzione delle istruzioni DDL `INPLACE ALTER TABLE` devono essere aggiornati a questa versione.

Correzioni per la stabilità generale:

- È stato risolto un problema che generava dati incoerenti nella tabella `information_schema.replica_host_status`.

Integrazione delle correzioni di bug della community di MySQL

- Bug #26251621: INCORRECT BEHAVIOR WITH TRIGGER AND GCOL
- Bug #22574695: ASSERTION `!TABLE || (!TABLE->READ_SET || BITMAP_IS_SET(TABLE->READ_SET, FIEL
- Bug #25966845: INSERT ON DUPLICATE KEY GENERATE A DEADLOCK
- Bug #23070734: CONCURRENT TRUNCATE TABLES CAUSE STALL
- Bug #26191879: FOREIGN KEY CASCADES USE EXCESSIVE MEMORY
- Bug #20989615: INNODB AUTO_INCREMENT PRODUCES SAME VALUE TWICE

Confronto con Aurora MySQL Versione 1

Le caratteristiche Amazon Aurora MySQL seguenti sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Aurora MySQL 2.07.0 è compatibile con MySQL 5.7 e include caratteristiche come il supporto di JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Al momento, Aurora MySQL 2.07.0 non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 22/11/2019 (versione 2.06.0) (obsoleta)

Versione: 2.06.0

Aurora MySQL 2.06.0 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni di Aurora MySQL correntemente supportate sono 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 2.01.*, 2.02.*, 2.03.*, 2.04.*, 2.05.* e 2.06.*.

È possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.06.0. È anche possibile aggiornare i cluster database di Aurora MySQL 2.* in Aurora MySQL 2.06.0. Non è possibile aggiornare un cluster Aurora MySQL 1.* esistente direttamente a 2.06.0; tuttavia, è possibile ripristinarne lo snapshot in Aurora MySQL 2.06.0.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite l'API Console di gestione AWS RDS. AWS CLI

> **Note**
>
> Questa versione non è attualmente disponibile nelle seguenti AWS regioni: AWS GovCloud (Stati Uniti orientali) [us-gov-east-1], (Stati Uniti occidentali) [-1], Cina AWS GovCloud (Ningxia) [us-gov-westcn-northwest-1], Asia Pacifico (Hong Kong) [ap-east-1] e Medio Oriente (Bahrein) [me-south-1]. Non appena disponibile verrà comunicato l'annuncio.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Nuove caratteristiche:

- I cluster Aurora MySQL ora supportano i tipi di istanza db.r5.8xlarge, db.r5.16xlarge e db.r5.24xlarge. Per ulteriori informazioni sui tipi di istanza per i cluster Aurora MySQL, consulta Aurora DB instance classes nella Guida per l'utente di Amazon Aurora.
- La caratteristica hash join è ora generalmente disponibile e non richiede l'attivazione della modalità lab Aurora. Questa caratteristica può migliorare le prestazioni delle query se devi eseguire il join di un'ingente quantità di dati tramite una query equi-join. Per ulteriori informazioni sull'utilizzo di questa funzionalità, consulta Utilizzo dell'API dati per Aurora Serverless nella Guida per l'utente di Amazon Aurora.
- La caratteristica di conflitto delle righe è ora generalmente disponibile e non richiede l'attivazione della modalità lab Aurora. Questa caratteristica migliora significativamente il throughput per i carichi di lavoro in cui più transazioni si contendono le righe della stessa pagina.

- Aurora MySQL 2.06 e versioni successive supportano il "riavvolgimento" di un cluster DB a un'ora specifica, senza ripristinare i dati di un backup. Questa caratteristica, nota come backtrack, offre un modo rapido per recuperare gli errori degli utenti, come ad esempio il rilascio della tabella sbagliata o l'eliminazione della riga sbagliata. Il backtrack viene completato in pochi secondi, anche per database di grandi dimensioni. Leggi il AWS blog per una panoramica e consulta la sezione Backtracking di un cluster Aurora DB nella Amazon Aurora User Guide per maggiori dettagli.
- Aurora 2.06 e versioni successive supportano le AWS Lambda chiamate sincrone tramite la funzione nativa. `lambda_sync()` Inoltre, è disponibile la funzione nativa `lambda_async()`, che può essere utilizzata come alternativa alla procedura di archiviazione esistente per l'invocazione Lambda asincrona. Per ulteriori informazioni sulle chiamate delle funzioni Lambda, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Correzioni importanti:

Nessuna.

Correzioni con alta priorità:

Correzioni di sicurezza

- 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

Gestione delle connessioni

- La disponibilità del database è stata migliorata per rispondere meglio all'aumento delle connessioni client durante l'esecuzione di una o più connessioni. DDLs Viene gestita creando temporaneamente thread aggiuntivi quando necessario. Ti consigliamo di eseguire l'aggiornamento se il database non risponde a seguito di un aumento delle connessioni durante l'elaborazione di DDL.

Riavvio del motore

- È stato risolto un problema di indisponibilità prolungata durante il riavvio del motore. Questo risolve un problema nell'inizializzazione del pool di buffer. Questo problema si verifica raramente ma può potenzialmente influire su qualsiasi versione supportata.
- È stato risolto un problema che causava il riavvio di un database configurato come log binario master (binlog) mentre era in esecuzione un carico di lavoro di scrittura elevato.

Correzioni per la stabilità generale:

- Apportati miglioramenti per le query che accedono ai dati non memorizzati nella cache e sono più lente del solito. I clienti che hanno riscontrato una inspiegabile latenza di lettura elevata durante l'accesso ai dati non memorizzati nella cache sono incoraggiati a eseguire l'aggiornamento perché potrebbero riscontrare questo problema.
- È stato risolto un problema a causa del quale non si riusciva a ripristinare le tabelle partizionate da uno snapshot del database. I clienti che riscontrano errori durante l'accesso alle tabelle partizionate in un database ripristinato dallo snapshot di un database Aurora MySQL 1.* sono invitati a utilizzare questa versione.
- Stabilità migliorata delle repliche di Aurora correggendo il conflitto tra i thread che servono le query di lettura e quello che applica le modifiche allo schema mentre è in corso una query DDL sull'istanza database di scrittura.
- È stato risolto un problema di stabilità relativo all'aggiornamento della tabella `mysql.innodb_table_stats` attivato dalle operazioni DDL.
- È stato risolto un problema che riportava erroneamente `ERROR 1836` quando una query nidificata veniva eseguita su una tabella temporanea nella replica di Aurora.

Miglioramenti alle prestazioni:

- Prestazioni migliorate della replica binlog impedendo le chiamate API non necessarie alla cache, se la cache delle query è stata disabilitata sullo slave binlog.

Confronto con Aurora MySQL Versione 1

Le caratteristiche Amazon Aurora MySQL seguenti sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Aurora MySQL 2.06.0 è compatibile con MySQL 5.7 e include caratteristiche come il supporto di JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Al momento, Aurora MySQL 2.06.0 non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query

- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 11/11/2019 (versione 2.05.0) (obsoleta)

Versione: 2.05.0

Aurora MySQL 2.05.0 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni Aurora MySQL attualmente supportate sono 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 2.01.*, 2.02.*, 2.03.* e 2.04.*.

È possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.05.0. È possibile eseguire l'aggiornamento dei cluster database di Aurora MySQL 2.* esistenti fino alla versione 2.04.6 a Aurora MySQL 2.05.0. Non è possibile aggiornare un cluster Aurora MySQL 1.* esistente direttamente a 2.05.0; tuttavia, è possibile ripristinarne lo snapshot in Aurora MySQL 2.05.0.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite l'API Console di gestione AWS RDS. AWS CLI

> **Note**
>
> Questa versione non è attualmente disponibile nelle seguenti AWS regioni: AWS GovCloud (Stati Uniti orientali) [us-gov-east-1], (Stati Uniti occidentali) [-1], Cina AWS GovCloud (Ningxia) [us-gov-westcn-northwest-1], Asia Pacifico (Hong Kong) [ap-east-1], Europa (Stoccolma) [eu-north-1] e Medio Oriente (Bahrein) [me-south-1]. Non appena disponibile verrà comunicato l'annuncio.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Correzioni importanti:

- 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
- È stato risolto un problema a causa del quale gli eventi nel file binlog corrente sul master non erano replicati sullo slave se il valore del parametro `sync_binlog` non era impostato su 1.

Correzioni con alta priorità:

- Consigliamo vivamente ai clienti con dimensioni del database vicine a 64 TiB di eseguire l'aggiornamento a questa versione per evitare tempi di inattività a causa di bug di stabilità che incidono sui volumi vicini al limite di storage Aurora.
- Il valore predefinito del parametro `aurora_binlog_replication_max_yield_seconds` è stato modificato a zero per impedire un incremento del ritardo di replica a favore delle prestazioni delle query in primo piano sul master binlog.

Integrazione delle correzioni di bug di MySQL.

- Bug #23054591: PURGE BINARY LOGS TO sta leggendo l'intero file binlog e causando lo stallo MySql

Confronto con Aurora MySQL Versione 1

Le caratteristiche Amazon Aurora MySQL seguenti sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Aurora MySQL 2.05.0 è compatibile con MySQL 5.7 e include caratteristiche come il supporto di JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Al momento, Aurora MySQL 2.05.0 non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina

- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 14/08/2020 (versione 2.04.9) (obsoleta)

Versione: 2.04.9

Aurora MySQL 2.04.9 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite l'API Console di gestione AWS RDS. AWS CLI

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Questa versione non è attualmente disponibile nelle seguenti AWS regioni: AWS GovCloud (Stati Uniti orientali) [us-gov-east-1], AWS GovCloud (Stati Uniti occidentali) [us-gov-west-1], Asia Pacifico (Hong Kong) [ap-east-1] e Medio Oriente (Bahrein) [me-south-1]. Non appena disponibile verrà comunicato l'annuncio.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Correzioni con alta priorità:

Correzioni importanti:

- 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

Miglioramenti della disponibilità:

- È stato risolto un problema che poteva causare il riavvio o il failover del database a causa dell'esecuzione di un comando `kill session`. Se riscontri questo problema, contatta AWS l'assistenza per attivare questa correzione sulla tua istanza.
- È stato risolto un problema che causa il riavvio del database durante l'esecuzione di una query complessa che coinvolge join multi-tabella e aggregazione che utilizzano tabelle intermedie internamente.
- È stato risolto un problema che causa il riavvio del database per un'interruzione `DROP TABLE` su più tabelle.
- È stato risolto un problema che causava un failover del database durante il ripristino del database.
- Corretto un riavvio del database causato da segnalazioni errate di threads_running quando log di controllo e query lente sono abilitati.
- Risolto un problema per cui un comando `kill query` poteva bloccarsi durante l'esecuzione.
- È stata risolta una condizione di competizione nel gestore di blocchi che causava un riavvio o un failover del database durante il rollback della transazione.
- Risolto un problema che causava il riavvio o il failover del database quando più connessioni tentavano di aggiornare la stessa tabella con un indice di ricerca full-text.
- È stato risolto un problema che può causare un deadlatch durante la rimozione di un indice con conseguente failover o riavvio.

Miglioramenti generali:

- Sono stati risolti problemi che potevano causare l'utilizzo dei dati di una transazione non eseguita da query sulle repliche di lettura. Questo problema è limitato alle transazioni avviate immediatamente dopo il riavvio del database.

- Risolto un problema riscontrato durante `INPLACE ALTER TABLE` per una tabella con trigger definiti e quando il DDL non conteneva una clausola `RENAME`.
- È stato risolto un problema per cui la clonazione richiedeva più tempo in alcuni cluster di database con carichi di scrittura elevati.
- È stato risolto un problema riscontrato durante un aggiornamento quando una tabella partizionata contiene spazi incorporati nel nome.
- È stato risolto un problema per cui la replica di lettura poteva visualizzare in modo transitorio i risultati parziali di una transazione recentemente impegnata nel writer.
- È stato risolto un problema per cui le query su una replica di lettura rispetto a una tabella FTS potrebbe produrre risultati obsoleti. Ciò si verifica solo quando la query FTS sulla replica di lettura segue da vicino una query `INFORMATION_SCHEMA.INNODB_SYS_TABLES` per la stessa tabella FTS nel writer.
- Risolto un problema che causava un lento ripristino di un cluster database Aurora 1.x contenente indici FTS (Full Text Search) in un cluster database Aurora 2.x.
- Lunghezza massima consentita estesa fino a 2000 per parametri `server_audit_incl_users` e `server_audit_excl_users` globali.
- Risolto un problema per cui il ripristino da Aurora 1.x a Aurora 2.x potrebbe richiedere un lungo periodo di tempo.
- Risolto un problema per cui una chiamata `lambda_async` tramite stored procedure non funzionava con Unicode.
- È stato risolto un problema che si verificava quando un indice spaziale non gestisce correttamente una colonna di geometria fuori record.
- È stato risolto un problema che poteva causare il fallimento di una query su un'istanza database del lettore con errore `InternalFailureException` con il messaggio "Operazione terminata (errore interno)".

Integrazione delle correzioni di bug di MySQL.

- Bug #23070734, Bug #80060: il TRUNCATE simultaneo causa stalli TABLEs
- Bug #23103937: PS_TRUNCATE_ALL_TABLES () NON FUNZIONA IN MODALITÀ SUPER_READ
- Bug #22551677: quando si disconnette il server, una race condition all'interno dello schema delle prestazioni potrebbe causare l'uscita del server.
- Bug #27082268: sincronizzazione di sincronizzazione FTS non valida.

- BUG #12589870: risolto un problema che causa un riavvio con l'istruzione multi-query quando la cache delle query è abilitata.
- Bug #26402045: Alcuni casi di materializzazione delle query secondarie potrebbero causare l'uscita del server. Queste query ora producono un errore che suggerisce che la materializzazione deve essere disabilitata.
- Bug #18898433: le query con molti join di sinistra erano lente se è stato utilizzato il buffering di join (ad esempio, utilizzando l'algoritmo di loop nidificato del blocco).
- Bug #25222337: un nome di campo colonna virtuale NULL in un indice virtuale ha causato l'uscita del server durante un confronto tra nomi di campo che si verifica durante il popolamento di colonne virtuali interessate da un vincolo di chiave esterna. (https://github.com/mysql/mysql-server/commit/273d5c9d7072c63b6c47dbef6963d7dc491d5131)
- Bug #25053286: l'esecuzione di una procedura archiviata contenente una query che ha effettuato l'accesso a una vista potrebbe allocare memoria che non è stata liberata fino al termine della sessione. (https://github.com/mysql/mysql-server/commit/d7b37d4d141a95f577916448650c429f0d6e193d)
- Bug #25586773: L'esecuzione di una procedura memorizzata contenente un'istruzione che ha creato una tabella dal contenuto di determinate istruzioni SELECT (https://dev.mysql.com/doc/refman/5.7/en/select.html) potrebbe causare una perdita di memoria. () https://github.com/mysql/mysql-server/commit/88301e5adab65f6750f66af284be410c4369d0c1
- Bug #26666274: INFINITO LOOP IN PERFORMANCE SCHEMA BUFFER CONTAINER
- Bug #23550835, Bug #23298025, Bug #81464: tabelle SELECT Performance Schema quando un buffer interno era pieno potrebbero causare l'uscita del server.

Confronto con Aurora MySQL Versione 1

Le caratteristiche Amazon Aurora MySQL seguenti sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.

- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Aurora MySQL 2.04.9 è compatibile con MySQL 5.7 e include caratteristiche come il supporto di JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Al momento, Aurora MySQL 2.04.9 non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 20/11/2019 (versione 2.04.8) (obsoleta)

Versione: 2.04.8

Aurora MySQL 2.04.8 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

Puoi ripristinare uno snapshot di qualsiasi versione Aurora MySQL 2.* in Aurora MySQL 2.04.8. È anche possibile aggiornare i cluster database di Aurora MySQL 2.* in Aurora MySQL 2.04.8.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite l'API Console di gestione AWS RDS. AWS CLI

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Questa versione non è attualmente disponibile nelle seguenti AWS regioni: AWS GovCloud (Stati Uniti orientali) [us-gov-east-1], AWS GovCloud (Stati Uniti occidentali) [us-gov-west-1], Asia Pacifico (Hong Kong) [ap-east-1] e Medio Oriente (Bahrein) [me-south-1]. Non appena disponibile verrà comunicato l'annuncio.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Nuove caratteristiche:

- Miglioramenti della replica di lettura:
  - Traffico di rete ridotto dall'istanza di scrittura con la trasmissione efficiente dei dati alle istanze di lettura all'interno del cluster DB Aurora. Questo miglioramento è abilitato per impostazione

predefinita poiché consente di evitare che le repliche restino indietro e vengano riavviate. Il parametro per questa caratteristica è `aurora_enable_repl_bin_log_filtering`.

- Traffico di rete ridotto dalle istanze di scrittura alle istanze di lettura all'interno del cluster DB Aurora mediante la compressione. Questo miglioramento è abilitato per impostazione predefinita solo per le classi di istanze 8xlarge e 16xlarge poiché queste istanze possono tollerare un sovraccarico aggiuntivo della CPU per la compressione. Il parametro per questa caratteristica è `aurora_enable_replica_log_compression`.

Correzioni con alta priorità:

- Gestione della memoria migliorata nell'istanza di scrittura Aurora che impedisce il riavvio del writer a causa di condizioni di memoria insufficiente durante un carico di lavoro elevato in presenza di istanze del lettore all'interno del cluster DB Aurora.
- Correzione di una condizione non deterministica nel pianificatore che determina il riavvio del motore durante l'accesso simultaneo all'oggetto dello schema delle prestazioni.

Confronto con Aurora MySQL Versione 1

Le caratteristiche Amazon Aurora MySQL seguenti sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per AWS Lambda richiamare funzioni in modo sincrono. Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Aurora MySQL 2.04.8 è compatibile con MySQL 5.7 e include caratteristiche come il supporto di JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Al momento, Aurora MySQL 2.04.8 non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 14/11/2019 (versione 2.04.7) (obsoleta)

Versione: 2.04.7

Aurora MySQL 2.04.7 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

È possibile ripristinare uno snapshot da una versione di Aurora MySQL attualmente supportata in Aurora MySQL 2.04.7. È anche possibile aggiornare i cluster database di Aurora MySQL 2.* in Aurora MySQL 2.04.7. Non è possibile aggiornare un cluster Aurora MySQL 1.* esistente direttamente a 2.04.7; tuttavia, è possibile ripristinarne lo snapshot in Aurora MySQL 2.04.7.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite l'API Console di gestione AWS RDS. AWS CLI

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Questa versione non è attualmente disponibile nelle seguenti AWS regioni: AWS GovCloud (Stati Uniti orientali) [us-gov-east-1], AWS GovCloud (Stati Uniti occidentali) [us-gov-west-1], Asia Pacifico (Hong Kong) [ap-east-1] e Medio Oriente (Bahrein) [me-south-1]. Non appena disponibile verrà comunicato l'annuncio.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Correzioni con alta priorità:

Gestione delle connessioni

- La disponibilità del database è stata migliorata per rispondere meglio all'aumento delle connessioni client durante l'esecuzione di una o più connessioni. DDLs Viene gestita creando temporaneamente thread aggiuntivi quando necessario. Ti consigliamo di eseguire l'aggiornamento se il database non risponde a seguito di un aumento delle connessioni durante l'elaborazione di DDL.
- È stato risolto un problema che generava un valore non corretto per la variabile di stato globale `Threads_running`.

Riavvio del motore

- È stato risolto un problema di indisponibilità prolungata durante il riavvio del motore. Questo risolve un problema nell'inizializzazione del pool di buffer. Questo problema si verifica raramente ma può potenzialmente influire su qualsiasi versione supportata.

Correzioni per la stabilità generale:

- Apportati miglioramenti per le query che accedono ai dati non memorizzati nella cache e sono più lente del solito. I clienti che hanno riscontrato inspiegabili latenze di lettura elevate durante l'accesso ai dati non memorizzati nella cache sono incoraggiati a eseguire l'aggiornamento perché potrebbero riscontrare questo problema.

Confronto con Aurora MySQL Versione 1

Le caratteristiche Amazon Aurora MySQL seguenti sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare AWS Lambda funzioni in modo sincrono. Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Aurora MySQL 2.04.7 è compatibile con MySQL 5.7 e include caratteristiche come il supporto di JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa

degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Al momento, Aurora MySQL 2.04.7 non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 19/09/2019 (versione 2.04.6) (obsoleta)

Versione: 2.04.6

Aurora MySQL 2.04.6 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

È possibile eseguire l'aggiornamento dei cluster di database di Aurora MySQL 2.* esistenti a Aurora MySQL 2.04.6. Non è invece possibile eseguire un aggiornamento in loco dei cluster di Aurora MySQL 1.*. Questa restrizione sarà revocata in una successiva versione di Aurora MySQL 2.*. Puoi ripristinare gli snapshot di Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 2.01.*, 2.02.*, 2.03.* e 2.04.* in Aurora MySQL 2.04.6.

Per utilizzare una versione precedente di Aurora MySQL, puoi creare nuovi cluster di database specificando la versione del motore tramite l'API Amazon RDS Console di gestione AWS. AWS CLI

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Questa versione non è attualmente disponibile nelle seguenti AWS regioni: Europa (Londra) [eu-west-2] AWS GovCloud , (Stati Uniti orientali) [-1], (Stati Uniti occidentali) us-gov-east [-1] AWS GovCloud , Cina (Ningxia) [us-gov-westcn-northwest-1] e Asia Pacifico (Hong Kong) [ap-east-1]. Non appena disponibile verrà comunicato l'annuncio.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

- È stato risolto un problema a causa del quale gli eventi nel file binlog corrente sul master non erano replicati sullo slave se il valore del parametro `sync_binlog` non era impostato su 1.
- Il valore predefinito del parametro `aurora_binlog_replication_max_yield_seconds` è stato modificato a zero per impedire un incremento del ritardo di replica a favore delle prestazioni delle query in primo piano sul master binlog.

Integrazione delle correzioni di bug di MySQL.

- Bug #23054591: PURGE BINARY LOGS TO sta leggendo l'intero file binlog e causa lo stallo MySql

Confronto con Aurora MySQL Versione 1

Le caratteristiche Amazon Aurora MySQL seguenti sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Aurora MySQL 2.04.6 è compatibile con MySQL 5.7 e include caratteristiche come il supporto di JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Al momento, Aurora MySQL 2.04.6 non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online

- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 08/07/2019 (versione 2.04.5) (obsoleta)

Versione: 2.04.5

Aurora MySQL 2.04.5 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

È possibile eseguire l'aggiornamento dei cluster di database di Aurora MySQL 2.* esistenti a Aurora MySQL 2.04.5. Non è invece possibile eseguire un aggiornamento in loco dei cluster di Aurora MySQL 1.*. Questa restrizione sarà revocata in una successiva versione di Aurora MySQL 2.*. Puoi ripristinare gli snapshot di Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 2.01.*, 2.02.*, 2.03.* e 2.04.* in Aurora MySQL 2.04.5.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Correzioni di sicurezza:

- CVE-2016-3518

Correzioni generali:

- È stata risolta una race condition durante la crescita del volume di storage che causava il riavvio del database.
- È stata risolto un errore di comunicazione interna durante l'apertura del volume che causava il riavvio del database.
- È stato aggiunto supporto per il ripristino DDL per `ALTER TABLE ALGORITHM=INPLACE` sulle tabelle partizionate.
- È stata risolto un problema con il ripristino DDL di `ALTER TABLE ALGORITHM=COPY` che causava il riavvio del database.
- È stata migliorata la stabilità delle repliche Aurora in presenza di un carico di lavoro di eliminazione pesante sullo scrittore.
- È stato risolto un problema di riavvio del database causato da un deadlatch tra il thread che eseguiva la sincronizzazione dell'indice di ricerca full-text e il thread che eseguiva l'espulsione della tabella di ricerca full-text dalla cache dei dizionari.
- È stato risolto un problema di stabilità sullo slave binlog durante la replica DDL mentre la connessione al master binlog era instabile.
- È stato risolto un out-of-memory problema nel codice di ricerca completo che causava il riavvio del database.
- È stato risolto un problema su Aurora Writer che ne comportava il riavvio quando veniva usato l'intero volume da 64 tebibyte (TiB).
- È stata risolta una race condition nella funzione dello schema di performance che causava il riavvio del database.
- È stato risolto un problema che causava l'interruzione delle connessioni durante la gestione di un errore nella gestione del protocollo di rete.

Confronto con Aurora MySQL Versione 1

Le caratteristiche Amazon Aurora MySQL seguenti sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.

- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare AWS Lambda funzioni in modo sincrono. Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Aurora MySQL 2.04.5 è compatibile con MySQL 5.7 e include caratteristiche come il supporto di JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Al momento, Aurora MySQL 2.04.5 non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 29/05/2019 (versione 2.04.4) (obsoleta)

Versione: 2.04.4

Aurora MySQL 2.04.4 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

Quando si crea un nuovo cluster di database Aurora MySQL, compresi i ripristini di una snapshot, è possibile scegliere la compatibilità con MySQL 5.7 o MySQL 5.6. Non è invece possibile eseguire un aggiornamento locale dei cluster Aurora MySQL 1.* o ripristinare i cluster Aurora MySQL 1.* da un backup Amazon S3 ad Aurora MySQL 2.04.4. Prevediamo di rimuovere queste limitazioni in una versione successiva di Aurora MySQL 2.*.

Puoi ripristinare le snapshot di Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 2.01.*, 2.02.*, 2.03.* e 2.04.* in Aurora MySQL 2.04.4.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Questa versione non è attualmente disponibile nelle regioni AWS GovCloud (Stati Uniti occidentali) [us-gov-west-1], Europa (Stoccolma) [eu-north-1], Cina (Ningxia) [cn-northwest-1] e Asia Pacifico (Hong Kong) [ap-east-1]. AWS Non appena disponibile verrà comunicato l'annuncio.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

- È stato risolto un problema che poteva causare errori durante il caricamento di dati in Aurora da S3.
- È stato risolto un problema che poteva causare errori durante il caricamento di dati Aurora in S3.
- È stato risolto un problema che causava l'interruzione delle connessioni durante la gestione di un errore nella gestione del protocollo di rete.
- È stato risolto un problema che poteva causare un arresto anomalo durante la gestione di tabelle partizionate.
- È stato risolto un problema relativo alla mancata disponibilità della funzionalità Performance Insights in alcune regioni.

Confronto con Aurora MySQL Versione 1

Le caratteristiche Amazon Aurora MySQL seguenti sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per AWS Lambda richiamare funzioni in modo sincrono. Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Aurora MySQL 2.04.4 è compatibile con MySQL 5.7 e include caratteristiche come il supporto di JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Al momento, Aurora MySQL 2.04.4 non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 09/05/2019 (versione 2.04.3) (obsoleta)

Versione: 2.04.3

Aurora MySQL 2.04.3 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

Quando si crea un nuovo cluster di database Aurora MySQL, compresi i ripristini di una snapshot, è possibile scegliere la compatibilità con MySQL 5.7 o MySQL 5.6. Non è invece possibile eseguire un aggiornamento locale dei cluster Aurora MySQL 1.* o ripristinare i cluster Aurora MySQL 1.* da un backup Amazon S3 ad Aurora MySQL 2.04.3. Prevediamo di rimuovere queste limitazioni in una versione successiva di Aurora MySQL 2.*.

Puoi ripristinare le snapshot di Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 2.01.*, 2.02.*, 2.03.* e 2.04.* in Aurora MySQL 2.04.3.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Questa versione non è attualmente disponibile nelle regioni AWS GovCloud (Stati Uniti occidentali) [us-gov-west-1] e Cina (Ningxia) [cn-northwest-1]. AWS Non appena disponibile verrà comunicato l'annuncio.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

- Corretto un bug nella replica binlog che può causare un problema sulle istanze Aurora configurate come slave binlog.
- È stata risolta una out-of-memory condizione che si verificava durante la gestione di routine memorizzate di grandi dimensioni.
- Corretto un errore nella gestione di determinati tipi di comandi `ALTER TABLE`.
- Corretto un problema di connessioni interrotte per un errore di gestione del protocollo di rete.

Confronto con Aurora MySQL Versione 1

Le caratteristiche Amazon Aurora MySQL seguenti sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare AWS Lambda funzioni in modo sincrono. Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Aurora MySQL 2.04.3 è compatibile con MySQL 5.7 e include caratteristiche come il supporto di JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Al momento, Aurora MySQL 2.04.3 non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 02/05/2019 (versione 2.04.2) (obsoleta)

Versione: 2.04.2

Aurora MySQL 2.04.2 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

Quando si crea un nuovo cluster di database Aurora MySQL, compresi i ripristini di una snapshot, è possibile scegliere la compatibilità con MySQL 5.7 o MySQL 5.6. Non è invece possibile eseguire un aggiornamento locale dei cluster Aurora MySQL 1.* o ripristinare i cluster Aurora MySQL 1.* da un backup Amazon S3 ad Aurora MySQL 2.04.2. Prevediamo di rimuovere queste limitazioni in una versione successiva di Aurora MySQL 2.*.

Puoi ripristinare le snapshot di Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 2.01.*, 2.02.*, 2.03.*, 2.04.0 e 2.04.1 in Aurora MySQL 2.04.2.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Questa versione non è attualmente disponibile nelle regioni AWS GovCloud (Stati Uniti occidentali) [us-gov-west-1] e Cina (Ningxia) [cn-northwest-1]. AWS Non appena disponibile verrà comunicato l'annuncio.

> **Note**
>
> Per informazioni su come aggiornare il cluster di database Aurora MySQL, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

- Aggiunto supporto per la replica binlog SSL che utilizza certificati personalizzati. Per informazioni sull'utilizzo della replica binlog in Aurora MySQL, consultamysql_rds_import_binlog_ssl_material.
- Correzione di un deadlatch sull'istanza primaria Aurora che si verifica quando viene ottimizzata una tabella con un indice di ricerche full text.
- Correzione di un errore sulle repliche Aurora in cui le prestazioni di determinate query che utilizzano `SELECT(*)` potrebbero influenzate relativamente alle tabelle con indici secondari.
- Correzione di una condizione che ha prodotto un errore 1032 durante il post.
- Miglioramento della stabilità delle repliche Aurora grazie alla correzione di più deadlatch.

Integrazione delle correzioni di bug di MySQL.

- Bug #24829050 - L'OTTIMIZZAZIONE DI INDEX_MERGE_INTERSECTION GENERA RISULTATI QUERY ERRATI

Confronto con Aurora MySQL Versione 1

Le caratteristiche Amazon Aurora MySQL seguenti sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Aurora MySQL 2.04.2 è compatibile con MySQL 5.7 e include caratteristiche come il supporto di JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Al momento, Aurora MySQL 2.04.2 non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 25/03/2019 (versione 2.04.1) (obsoleta)

Versione: 2.04.1

Aurora MySQL 2.04.1 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

Quando si crea un nuovo cluster di database Aurora MySQL, compresi i ripristini di una snapshot, è possibile scegliere la compatibilità con MySQL 5.7 o MySQL 5.6. Non è invece possibile eseguire un aggiornamento locale dei cluster Aurora MySQL 1.* o ripristinare i cluster Aurora MySQL 1.* da

un backup Amazon S3 ad Aurora MySQL 2.04.1. Prevediamo di rimuovere queste limitazioni in una versione successiva di Aurora MySQL 2.*.

Puoi ripristinare gli snapshot di Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 2.01.*, 2.02.*, 2.03.*, 2.04.0 in Aurora MySQL 2.04.1.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Questa versione non è attualmente disponibile nella regione AWS GovCloud (Stati Uniti occidentali) [us-gov-west-1]. Non appena disponibile verrà comunicato l'annuncio.

> **Note**
>
> La procedura per aggiornare il cluster di database è cambiata. Per ulteriori informazioni, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

- Corretto un problema in cui uno snapshot Aurora MySQL 5.6 per le versioni inferiori rispetto alla 1.16 non può essere ripristinato all'ultimo cluster Aurora MySQL 5.7.

Confronto con Aurora MySQL Versione 1

Le caratteristiche Amazon Aurora MySQL seguenti sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.

- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare AWS Lambda funzioni in modo sincrono. Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 25/03/2019 (versione 2.04.0) (obsoleta)

Versione: 2.04

Aurora MySQL 2.04 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

Quando si crea un nuovo cluster di database Aurora MySQL, compresi i ripristini di una snapshot, è possibile scegliere la compatibilità con MySQL 5.7 o MySQL 5.6. Non è invece possibile eseguire un aggiornamento locale dei cluster Aurora MySQL 1.* o ripristinare i cluster Aurora MySQL 1.* da un backup Amazon S3 ad Aurora MySQL 2.04.0. Prevediamo di rimuovere queste limitazioni in una versione successiva di Aurora MySQL 2.*.

Puoi ripristinare le snapshot di Aurora MySQL 1.19*, 2.01*, 2.02* e 2.03* in Aurora MySQL 2.04.0. Non è possibile ripristinare gli snapshot di Aurora MySQL 1.14.* o versione inferiore, 1.15.*, 1.16.*, 1.17.*, 1.18.* in Aurora MySQL 2.04.0. Questa restrizione viene rimossa in Aurora MySQL 2.04.1.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Questa versione non è attualmente disponibile nella regione AWS GovCloud (Stati Uniti occidentali) [us-gov-west-1]. Non appena disponibile verrà comunicato l'annuncio.

> **Note**
>
> La procedura per aggiornare il cluster di database è cambiata. Per ulteriori informazioni, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

- Supporta la replica basata su GTID. Per informazioni sull'utilizzo della replica basata su GTID con Aurora MySQL, consulta Utilizzo della replica basata su GTID per Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Correzione di un errore in cui la replica Aurora genera erroneamente un errore `Running in read-only mode` quando un'istruzione di eliminazione o di aggiornamento delle righe contiene una query secondaria InnoDB.

Integrazione delle correzioni di bug di MySQL.

- Bug #26225783: MYSQL CRASH ON CREATE TABLE (REPRODUCEABLE) -> INNODB: ALONG SEMAPHORE WAIT.

Confronto con Aurora MySQL Versione 1

Le caratteristiche Amazon Aurora MySQL seguenti sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare AWS Lambda funzioni in modo sincrono. Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Questa versione Aurora MySQL è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Questa versione di Aurora MySQL al momento non supporta le seguenti caratteristiche di MySQL 5.7:

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`

Aggiornamenti del motore di database Aurora MySQL 07/02/2019 (versione 2.03.4) (obsoleta)

Versione: 2.03.4

Aurora MySQL 2.03.4 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Quando si crea un nuovo cluster di database Aurora MySQL, compresi quelli ripristinati da snapshot, è possibile scegliere la compatibilità con MySQL 5.7 o MySQL 5.6.

Non è invece possibile eseguire un aggiornamento in loco dei cluster Aurora MySQL 1.* in Aurora MySQL 2.03.4 né ripristinare Aurora MySQL 2.03.4 da un backup di Amazon S3. Prevediamo di rimuovere queste limitazioni in una versione successiva di Aurora MySQL 2.*.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Questa versione non è attualmente disponibile nelle regioni AWS GovCloud (Stati Uniti occidentali) [us-gov-west-1] e Cina (Pechino) [cn-north-1]. Non appena disponibile verrà comunicato l'annuncio.

> **Note**
>
> La procedura per aggiornare il cluster di database è cambiata. Per ulteriori informazioni, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

- Support per le regole di confronto UTF8 MB4 Unicode 9.0 sensibili agli accenti e senza distinzione tra maiuscole e minuscole,. `utf8mb4_0900_as_ci`

Confronto con Aurora MySQL Versione 1

Le seguenti caratteristiche di Amazon Aurora MySQL sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare funzioni AWS Lambda in modo sincrono. Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.

- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Aurora MySQL 2.03.4 è compatibile con MySQL 5.7 e include caratteristiche come il supporto di JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Al momento, Aurora MySQL 2.03.4 non supporta le seguenti caratteristiche di MySQL 5.7:

- Identificatori globali di transazione (). GTIDs Aurora MySQL supporta GTIDs la versione 2.04 e successive.
- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`
- Protocollo X

Aggiornamenti del motore di database Aurora MySQL 18/01/2019 (versione 2.03.3) (obsoleta)

Versione: 2.03.3

Aurora MySQL 2.03.3 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Quando si crea un nuovo cluster di database Aurora MySQL, compresi quelli ripristinati da snapshot, è possibile scegliere la compatibilità con MySQL 5.7 o MySQL 5.6.

Non è invece possibile eseguire un aggiornamento in loco dei cluster Aurora MySQL 1.* in Aurora MySQL 2.03.3 né ripristinare Aurora MySQL 2.03.3 da un backup di Amazon S3. Prevediamo di rimuovere queste limitazioni in una versione successiva di Aurora MySQL 2.*.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Questa versione non è attualmente disponibile nelle regioni AWS GovCloud (Stati Uniti occidentali) [us-gov-west-1] e Cina (Pechino) [cn-north-1]. Non appena disponibile verrà comunicato l'annuncio.

> **Note**
>
> La procedura per aggiornare il cluster di database è cambiata. Per ulteriori informazioni, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Correzioni CVE

- CVE-2016-5436

Correzioni importanti:

- Correzione di un problema per cui una Replica Aurora potrebbe diventare dead latched durante l'esecuzione dell'indice o della scansione precedente.

- Correzione di un problema per cui una Replica Aurora potrebbe essere riavviata quando l'istanza della chiave primaria Aurora esegue operazioni DDL in sede su tabelle con partizioni.
- Correzione di un problema per cui una Replica Aurora potrebbe essere riavviata durante l'invalidazione della cache delle query dopo un'operazione DDL sull'istanza primaria Aurora.
- Corretto un problema per cui una Replica Aurora potrebbe essere riavviata durante una query `SELECT` su una tabella, mentre l'istanza primaria Aurora esegue il troncamento di quella tabella.
- Correzione di un problema di risultato non valido con tabelle temporanee MyISAM in cui è possibile accedere solo alle colonne indicizzate.
- Correzione di un problema di log lenti che hanno generato valori grandi errati periodicamente per `query_time` e `lock_time` dopo circa 40.000 query.
- Correzione di un problema in cui uno schema denominato "tmp" potrebbe causare l'interruzione della migrazione da RDS for MySQL ad Aurora MySQL.
- Correzione di un problema per cui i log di audit potrebbero avere eventi mancanti durante la rotazione dei log.
- Correzione di un problema per cui l'istanza primaria Aurora ripristinata da uno snapshot Aurora 5.6 potrebbe essere riavviata quando è abilitata la caratteristica Fast DDL in modalità lab.
- Correzione di un problema per cui l'utilizzo della CPU è causato al 100% dal thread di statistiche del dizionario.
- Correzione di un problema per cui una Replica Aurora potrebbe essere riavviata durante l'esecuzione di un'istruzione `CHECK TABLE`.

Integrazione delle correzioni di bug di MySQL.

- Bug #25361251: INCORRECT BEHAVIOR WITH INSERT ON DUPLICATE KEY IN SP
- Bug #26734162: INCORRECT BEHAVIOR WITH INSERT OF BLOB + ON DUPLICATE KEY UPDATE
- Bug #27460607: INCORRECT BEHAVIOR OF IODKU WHEN INSERT SELECT's SOURCE TABLE IS EMPTY
- Una query che utilizza una clausola `DISTINCT` o `GROUP BY` potrebbe restituire risultati non corretti. (MySQL 5.7 Bug #79591, Bug #22343910)
- Un `DELETE` da tabelle unite che utilizzano una tabella derivata nella clausola `WHERE` non riesce con errore 1093 (bug 23074801).
- GCOLS: INCORRECT BEHAVIOR WITH CHARSET CHANGES (Bug #25287633).

Confronto con Aurora MySQL Versione 1

Le seguenti caratteristiche di Amazon Aurora MySQL sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Aurora MySQL 2.03.3 è compatibile con MySQL 5.7 e include caratteristiche come il supporto di JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Al momento, Aurora MySQL 2.03.3 non supporta le seguenti caratteristiche di MySQL 5.7:

- Identificatori globali di transazione (). GTIDs Aurora MySQL supporta GTIDs la versione 2.04 e successive.
- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB

- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`
- Protocollo X

Aggiornamenti del motore di database Aurora MySQL 09/01/2019 (versione 2.03.2) (obsoleta)

Versione: 2.03.2

Aurora MySQL 2.03.2 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Quando si crea un nuovo cluster di database Aurora MySQL, compresi quelli ripristinati da snapshot, è possibile scegliere la compatibilità con MySQL 5.7 o MySQL 5.6.

Non è invece possibile eseguire un aggiornamento in loco dei cluster Aurora MySQL 1.* in Aurora MySQL 2.03.2 né ripristinare Aurora MySQL 2.03.2 da un backup di Amazon S3. Prevediamo di rimuovere queste limitazioni in una versione successiva di Aurora MySQL 2.*.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Questa versione non è attualmente disponibile nelle regioni AWS GovCloud (Stati Uniti occidentali) [us-gov-west-1] e Cina (Pechino) [cn-north-1]. Non appena disponibile verrà comunicato l'annuncio.

> **Note**
>
> La procedura per aggiornare il cluster di database è cambiata. Per ulteriori informazioni, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

- Selezione della versione di Aurora: a partire da Aurora MySQL 2.03.2, puoi scegliere tra più versioni di Aurora compatibili con MySQL 5.7 nella Console di gestione AWS. Per ulteriori informazioni, consulta Verifica o specifica delle versioni del motore Aurora MySQL tramite AWS nella Guida per l'utente di Amazon Aurora.

Correzioni importanti:

- CVE-2016-3495

Confronto con Aurora MySQL Versione 1

Le seguenti caratteristiche di Amazon Aurora MySQL sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).

- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Aurora MySQL 2.03.2 è compatibile con MySQL 5.7 e include caratteristiche come il supporto di JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Al momento, Aurora MySQL 2.03.2 non supporta le seguenti caratteristiche di MySQL 5.7:

- Identificatori globali di transazione (). GTIDs Aurora MySQL supporta GTIDs la versione 2.04 e successive.
- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`
- Protocollo X

Aggiornamenti del motore di database Aurora MySQL 24/10/2018 (versione 2.03.1) (obsoleta)

Versione: 2.03.1

Aurora MySQL 2.03.1 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Quando si crea un nuovo cluster di database Aurora MySQL, è possibile scegliere la compatibilità con MySQL 5.7 o MySQL 5.6. Quando si ripristina una snapshot compatibile con MySQL 5.6 si può scegliere la compatibilità con MySQL 5.7 o MySQL 5.6.

Puoi ripristinare le snapshot di Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 2.01.*, 2.02.* e 2.03 in Aurora MySQL 2.03.1.

Non è invece possibile eseguire un aggiornamento in loco dei cluster Aurora MySQL 1.* in Aurora MySQL 2.03.1 né ripristinare Aurora MySQL 2.03.1 da un backup di Amazon S3. Prevediamo di rimuovere queste limitazioni in una versione successiva di Aurora MySQL 2.*.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Questa versione non è attualmente disponibile nelle regioni AWS GovCloud (Stati Uniti occidentali) [us-gov-west-1] e Cina (Pechino) [cn-north-1]. Non appena disponibile verrà comunicato l'annuncio.

Miglioramenti

- È stato risolto un problema a causa del quale Aurora Writer si riavviava quando si eseguiva un rilevamento del deadlock della transazione.

Confronto con Aurora MySQL Versione 1

Le seguenti caratteristiche di Amazon Aurora MySQL sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.

- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Aurora MySQL 2.03.1 è compatibile con MySQL 5.7 e include caratteristiche come il supporto di JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Al momento, Aurora MySQL 2.03.1 non supporta le seguenti caratteristiche di MySQL 5.7:

- Identificatori globali di transazione (). GTIDs Aurora MySQL supporta GTIDs la versione 2.04 e successive.
- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`
- Protocollo X

Aggiornamenti del motore di database Aurora MySQL 11/10/2018 (versione 2.03) (obsoleta)

Versione: 2.03

Aurora MySQL 2.03 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Quando si crea un nuovo cluster di database Aurora MySQL, è possibile scegliere la compatibilità con MySQL 5.7 o MySQL 5.6. Quando si ripristina una snapshot compatibile con MySQL 5.6 si può scegliere la compatibilità con MySQL 5.7 o MySQL 5.6.

Puoi ripristinare le snapshot di Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 2.01.* e 2.02.* in Aurora MySQL 2.03.

Non è invece possibile eseguire un aggiornamento in loco dei cluster Aurora MySQL 1.* in Aurora MySQL 2.03 né ripristinare Aurora MySQL 2.03 da un backup di Amazon S3. Prevediamo di rimuovere queste limitazioni in una versione successiva di Aurora MySQL 2.*.

> **Note**
>
> Questa versione non è attualmente disponibile nelle regioni AWS GovCloud (Stati Uniti occidentali) [us-gov-west-1] e Cina (Pechino) [cn-north-1]. Non appena disponibile verrà comunicato l'annuncio.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

- Lo schema di performance è disponibile.
- È stato risolto un problema a causa del quale le sessioni zombie con stati terminati potevano consumare più CPU.
- È stato risolto un problema relativo al latch non attivo nel quale le transazioni di sola lettura acquisivano un blocco su un record di Aurora Writer.

- È stato risolto un problema a causa del quale la replica di Aurora senza carico di lavoro del cliente faceva un elevato uso di CPU.
- Sono stati risolti diversi problemi che causavano il riavvio della replica di Aurora o di Aurora Writer.
- Aggiunta la capacità di saltare la registrazione per scopi diagnostici quando viene raggiunto il limite di throughput del disco.
- È stato risolto un problema di perdita di memoria quando binlog è abilitato su Aurora Writer.

Integrazione delle correzioni di bug della community di MySQL

- LA SCANSIONE INVERSA SU UNA TABELLA PARTIZIONATA ESEGUE ICP - ORDER BY DESC (bug 24929748).
- L'OGGETTO JSON CREA CODICE JSON NON VALIDO (Bug#26867509).
- L'INSERIMENTO DI GRANDI QUANTITÀ DI DATI JSON PRENDE UN'ECCESSIVA QUANTITÀ DI TEMPO (bug 22843444).
- LE TABELLE PARTIZIONATE USANO PIÙ MEMORIA NELLA VERSIONE 5.7 CHE IN QUELLA 5.6 (bug 25080442).

Confronto con Aurora MySQL Versione 1

Le seguenti caratteristiche di Amazon Aurora MySQL sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).

- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Aurora MySQL 2.03 è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Al momento, Aurora MySQL 2.03 non supporta le seguenti caratteristiche di MySQL 5.7:

- Identificatori globali di transazione (). GTIDs Aurora MySQL supporta GTIDs la versione 2.04 e successive.
- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`
- Protocollo X

Aggiornamenti del motore di database Aurora MySQL 08/10/2018 (versione 2.02.5) (obsoleta)

Versione: 2.02.5

Aurora MySQL 2.02.5 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Quando si crea un nuovo cluster di database Aurora MySQL, è possibile scegliere la compatibilità con MySQL 5.7 o MySQL 5.6. Quando si ripristina una snapshot compatibile con MySQL 5.6 si può scegliere la compatibilità con MySQL 5.7 o MySQL 5.6.

Puoi ripristinare le snapshot di Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 2.01.* e 2.02.* in Aurora MySQL 2.02.5. È anche possibile eseguire un aggiornamento in loco da Aurora MySQL 2.01.* o 2.02.* a Aurora MySQL 2.02.5.

Non è invece possibile eseguire un aggiornamento in loco dei cluster Aurora MySQL 1.* in Aurora MySQL 2.02.5 né ripristinare Aurora MySQL 2.02.5 da un backup di Amazon S3. Prevediamo di rimuovere queste limitazioni in una versione successiva di Aurora MySQL 2.*.

Lo schema delle prestazioni è disabilitato per questa release di Aurora MySQL 5.7. Esegui l'aggiornamento a Aurora 2.03 per avere assistenza sullo schema di performance.

> **Note**
>
> Questa versione non è attualmente disponibile nelle regioni AWS GovCloud (Stati Uniti occidentali) [us-gov-west-1] e Cina (Pechino) [cn-north-1]. Non appena disponibile verrà comunicato l'annuncio.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

- È stato risolto un problema a causa del quale una replica di Aurora si riavviava durante l'esecuzione di una scansione inversa su una tabella.

Confronto con Aurora MySQL Versione 1

Le seguenti caratteristiche di Amazon Aurora MySQL sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Aurora MySQL 2.02.5 è compatibile con MySQL 5.7 e include caratteristiche come il supporto di JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Al momento, Aurora MySQL 2.02.5 non supporta le seguenti caratteristiche di MySQL 5.7:

- Identificatori globali di transazione (). GTIDs Aurora MySQL supporta GTIDs la versione 2.04 e successive.
- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica

- Istruzione SQL `CREATE TABLESPACE`
- Protocollo X

Aggiornamenti del motore di database Aurora MySQL 21/09/2018 (versione 2.02.4) (obsoleta)

Versione: 2.02.4

Aurora MySQL 2.02.4 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Quando si crea un nuovo cluster di database Aurora MySQL, è possibile scegliere la compatibilità con MySQL 5.7 o MySQL 5.6. Quando si ripristina una snapshot compatibile con MySQL 5.6 si può scegliere la compatibilità con MySQL 5.7 o MySQL 5.6.

Puoi ripristinare le snapshot di Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 2.01.* e 2.02.* in Aurora MySQL 2.02.4. È anche possibile eseguire un aggiornamento in loco da Aurora MySQL 2.01.* o 2.02.* a Aurora MySQL 2.02.4.

Non è invece possibile eseguire un aggiornamento in loco dei cluster Aurora MySQL 1.* in Aurora MySQL 2.02.4 né ripristinare Aurora MySQL 2.02.4 da un backup di Amazon S3. Prevediamo di rimuovere queste limitazioni in una versione successiva di Aurora MySQL 2.*.

Lo schema delle prestazioni è disabilitato per questa release di Aurora MySQL 5.7. Esegui l'aggiornamento a Aurora 2.03 per avere assistenza sullo schema di performance.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

- È stato risolto un problema di stabilità relativo agli indici Full Text Search su tabelle ripristinate da una snapshot di Aurora MySQL 5.6.

Integrazione delle correzioni di bug della community di MySQL

- `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

Confronto con Aurora MySQL Versione 1

Le seguenti caratteristiche di Amazon Aurora MySQL sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.

- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Compatibilità MySQL 5.7

Aurora MySQL 2.02.4 è compatibile con MySQL 5.7 e include caratteristiche come il supporto di JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Al momento, Aurora MySQL 2.02.4 non supporta le seguenti caratteristiche di MySQL 5.7:

- Identificatori globali di transazione (). GTIDs Aurora MySQL supporta GTIDs la versione 2.04 e successive.
- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`
- Protocollo X

Aggiornamenti del motore di database Aurora MySQL 23/08/2018 (versione 2.02.3) (obsoleta)

Versione: 2.02.3

Aurora MySQL 2.02.3 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Quando si crea un nuovo cluster di database Aurora MySQL, è possibile scegliere la compatibilità con MySQL 5.7 o MySQL 5.6. Quando si ripristina una snapshot compatibile con MySQL 5.6 si può scegliere la compatibilità con MySQL 5.7 o MySQL 5.6.

Puoi ripristinare le snapshot di Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 2.01.* e 2.02.* in Aurora MySQL 2.02.3. È anche possibile eseguire un aggiornamento in loco da Aurora MySQL 2.01.* o 2.02.* a Aurora MySQL 2.02.3.

Non è invece possibile eseguire un aggiornamento in loco dei cluster Aurora MySQL 1.* in Aurora MySQL 2.02.3 né ripristinare Aurora MySQL 2.02.3 da un backup di Amazon S3. Prevediamo di rimuovere queste limitazioni in una versione successiva di Aurora MySQL 2.*.

Lo schema delle prestazioni è disabilitato per questa release di Aurora MySQL 5.7. Esegui l'aggiornamento a Aurora 2.03 per avere assistenza sullo schema di performance.

> **Note**
>
> Questa versione non è attualmente disponibile nelle regioni AWS GovCloud (Stati Uniti occidentali) [us-gov-west-1] e Cina (Pechino) [cn-north-1]. Non appena disponibile verrà comunicato l'annuncio.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Confronto con Aurora MySQL Versione 1

Le seguenti caratteristiche di Amazon Aurora MySQL sono supportate in Aurora MySQL versione 1 (compatibile con MySQL 5.6), ma non sono al momento supportate in Aurora MySQL versione 2 (compatibile con MySQL 5.7).

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Al momento Aurora MySQL 2.01 non supporta le caratteristiche aggiunte nella versione 1.16 di Aurora MySQL e nelle versioni successive. Per informazioni su Aurora MySQL versione 1.16, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).

Compatibilità MySQL 5.7

Aurora MySQL 2.02.3 è compatibile con MySQL 5.7 e include caratteristiche come il supporto di JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Al momento, Aurora MySQL 2.02.3 non supporta le seguenti caratteristiche di MySQL 5.7:

- Identificatori globali di transazione (). GTIDs Aurora MySQL supporta GTIDs la versione 2.04 e successive.
- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online

- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`
- Protocollo X

Differenze di CLI tra Aurora MySQL 2.x e Aurora MySQL 1.x

- Il nome del motore di Aurora MySQL 2.x è `aurora-mysql`; mentre quello di Aurora MySQL 1.x continua a essere `aurora`.
- Il gruppo di parametri predefiniti di Aurora MySQL 2.x è `default.aurora-mysql5.7`; mentre quello di Aurora MySQL 1.x continua a essere `default.aurora5.6`.
- Il nome di famiglia del gruppo di parametri del cluster di database di Aurora MySQL 2.x è `aurora-mysql5.7`; mentre quello di Aurora MySQL 1.x continua a essere `aurora5.6`.

Consulta la documentazione di Aurora per la serie completa di comandi CLI e le differenze tra Aurora MySQL 2.x e Aurora MySQL 1.x.

Miglioramenti

- È stato risolto un problema a causa del quale una replica Aurora si riavviava quando si usavano ripristini di cursore ottimistici durante la lettura dei record.
- Il valore predefinito del parametro `innodb_stats_persistent_sample_pages` è stato aggiornato a 128 per migliorare le statistiche di indice.
- È stato risolto un problema a causa del quale una replica di Aurora si riavviava durante l'accesso a una piccola tabella contemporaneamente modificata sull'istanza primaria di Aurora.
- Risolto il problema relativo a `ANALYZE TABLE` per arrestare lo svuotamento della cache di definizione della tabella.
- È stato risolto un problema a causa del quale l'istanza primaria di Aurora o la replica di Aurora si riavviavano durante la conversione di una point query per la ricerca geospaziale in un intervallo di ricerca.

Aggiornamenti del motore di database Aurora MySQL 04/06/2018 (versione 2.02.2) (obsoleta)

Versione: 2.02.2

Aurora MySQL 2.02.2 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Quando si crea un nuovo cluster di database Aurora MySQL, è possibile scegliere la compatibilità con MySQL 5.7 o MySQL 5.6. Quando si ripristina una snapshot compatibile con MySQL 5.6 si può scegliere la compatibilità con MySQL 5.7 o MySQL 5.6.

Puoi ripristinare le snapshot di Aurora MySQL 1.14*, 1.15*, 1.16*, 1.17*, 2.01* e 2.02 in Aurora MySQL 2.02.2. È anche possibile eseguire un aggiornamento in loco da Aurora MySQL 2.01* o 2.02 a Aurora MySQL 2.02.2.

Non è invece possibile eseguire un aggiornamento in loco dei cluster Aurora MySQL 1.* in Aurora MySQL 2.02.2 né ripristinare Aurora MySQL 2.02.2 da un backup di Amazon S3. Prevediamo di rimuovere queste limitazioni in una versione successiva di Aurora MySQL 2.*.

Lo schema delle prestazioni è disabilitato per questa release di Aurora MySQL 5.7. Esegui l'aggiornamento a Aurora 2.03 per avere assistenza sullo schema di performance.

> **Note**
>
> Questa versione non è attualmente disponibile nelle regioni AWS GovCloud (Stati Uniti occidentali) [us-gov-west-1] e Cina (Pechino) [cn-north-1]. Non appena disponibile verrà comunicato l'annuncio.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Correzioni importanti:

- CVE-2016-3486

Confronto con Aurora MySQL 5.6

Le seguenti caratteristiche di Amazon Aurora MySQL sono supportate in Aurora MySQL 5.6, ma attualmente non in Aurora MySQL 5.7.

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Al momento Aurora MySQL 2.01 non supporta le caratteristiche aggiunte nella versione 1.16 di Aurora MySQL e nelle versioni successive. Per informazioni su Aurora MySQL versione 1.16, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).

Compatibilità MySQL 5.7

Aurora MySQL 2.02.2 è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Al momento, Aurora MySQL 2.02.2 non supporta le seguenti caratteristiche di MySQL 5.7:

- Identificatori globali di transazione (). GTIDs Aurora MySQL supporta GTIDs la versione 2.04 e successive.
- Plugin replica gruppi
- Maggiori dimensioni pagina

- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`
- Protocollo X

Differenze di CLI tra Aurora MySQL 2.x e Aurora MySQL 1.x

- Il nome del motore di Aurora MySQL 2.x è `aurora-mysql`; mentre quello di Aurora MySQL 1.x continua a essere `aurora`.
- Il gruppo di parametri predefiniti di Aurora MySQL 2.x è `default.aurora-mysql5.7`; mentre quello di Aurora MySQL 1.x continua a essere `default.aurora5.6`.
- Il nome di famiglia del gruppo di parametri del cluster di database di Aurora MySQL 2.x è `aurora-mysql5.7`; mentre quello di Aurora MySQL 1.x continua a essere `aurora5.6`.

Consulta la documentazione di Aurora per la serie completa di comandi CLI e le differenze tra Aurora MySQL 2.x e Aurora MySQL 1.x.

Miglioramenti

- È stato risolto un problema a causa del quale Aurora Writer si riavviava occasionalmente quando teneva traccia dei progressi di Aurora Replica.
- È stato risolto un problema a causa del quale una replica di Aurora si riavviava o generava un errore all'accesso di una tabella partizionata dopo la creazione di un indice o l'esecuzione di un'istruzione DROP nella tabella in Aurora Writer.
- È stato risolto un problema per cui una tabella su una replica Aurora era inaccessibile mentre applicava le modifiche causate dall'esecuzione delle istruzioni ALTER table ADD/DROP column su Aurora Writer.

Aggiornamenti del motore di database Aurora MySQL 03/05/2018 (versione 2.02) (obsoleta)

Versione: 2.02

Aurora MySQL 2.02 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Quando si crea un nuovo cluster di database Aurora MySQL, è possibile scegliere la compatibilità con MySQL 5.7 o MySQL 5.6. Quando si ripristina una snapshot compatibile con MySQL 5.6 si può scegliere la compatibilità con MySQL 5.7 o MySQL 5.6.

Puoi ripristinare le snapshot di Aurora MySQL 1.14*, 1.15*, 1.16*, 1.17* e 2.01* in Aurora MySQL 2.02. È anche possibile eseguire un aggiornamento in loco da Aurora MySQL 2.01* a Aurora MySQL 2.02.

Non è invece possibile eseguire un aggiornamento in loco dei cluster Aurora MySQL 1.x ad Aurora MySQL 2.02 o ripristinare Aurora MySQL 2.02 da un backup Amazon S3. Prevediamo di rimuovere queste limitazioni in una versione successiva di Aurora MySQL 2.x.

Lo schema delle prestazioni è disabilitato per questa release di Aurora MySQL 5.7. Esegui l'aggiornamento a Aurora 2.03 per avere assistenza sullo schema di performance.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Confronto con Aurora MySQL 5.6

Le seguenti caratteristiche di Amazon Aurora MySQL sono supportate in Aurora MySQL 5.6, ma attualmente non in Aurora MySQL 5.7.

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.

- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Al momento Aurora MySQL 2.01 non supporta le caratteristiche aggiunte nella versione 1.16 di Aurora MySQL e nelle versioni successive. Per informazioni su Aurora MySQL versione 1.16, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).

Compatibilità MySQL 5.7

Aurora MySQL 2.02 è compatibile con MySQL 5.7 e include caratteristiche come il supporto di JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Al momento, Aurora MySQL 2.02 non supporta le seguenti caratteristiche di MySQL 5.7:

- Identificatori globali di transazione (). GTIDs Aurora MySQL supporta GTIDs la versione 2.04 e successive.
- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`
- Protocollo X

Differenze di CLI tra Aurora MySQL 2.x e Aurora MySQL 1.x

- Il nome del motore di Aurora MySQL 2.x è `aurora-mysql`; mentre quello di Aurora MySQL 1.x continua a essere `aurora`.
- Il gruppo di parametri predefiniti di Aurora MySQL 2.x è `default.aurora-mysql5.7`; mentre quello di Aurora MySQL 1.x continua a essere `default.aurora5.6`.
- Il nome di famiglia del gruppo di parametri del cluster di database di Aurora MySQL 2.x è `aurora-mysql5.7`; mentre quello di Aurora MySQL 1.x continua a essere `aurora5.6`.

Consulta la documentazione di Aurora per la serie completa di comandi CLI e le differenze tra Aurora MySQL 2.x e Aurora MySQL 1.x.

Miglioramenti

- È stato risolto un problema a causa del quale Aurora Writer si riavviava quando si eseguivano istruzioni INSERT e si utilizzava l'ottimizzazione dell'inserimento rapido.
- È stato risolto un problema a causa del quale una replica di Aurora si riavviava quando si eseguivano istruzioni ALTER DATABASE nella replica di Aurora.
- È stato risolto un problema a causa del quale una replica di Aurora si riavviava quando si eseguivano query nelle tabelle eliminate in Aurora Writer.
- È stato risolto un problema a causa del quale una replica di Aurora si riavviava quando si impostava `innodb_adaptive_hash_index` su `OFF` nella replica di Aurora.
- È stato risolto un problema a causa del quale una replica di Aurora si riavviava quando si eseguivano query TRUNCATE TABLE in Aurora Writer.
- È stato risolto un problema a causa del quale Aurora Writer si bloccava in determinate circostanze dopo l'esecuzione delle istruzioni INSERT. In un cluster a più nodi, la conseguenza è un failover.
- È stato risolto un problema di perdita di memoria associato all'impostazione di variabili di sessione.
- È stato risolto un problema a causa del quale Aurora Writer si bloccava in determinate circostanze associate alla rimozione di un annullamento nelle tabelle con colonne generate.
- È stato risolto un problema a causa del quale Aurora Writer si riavviava occasionalmente in seguito all'attivazione della registrazione binaria.

Integrazione delle correzioni di bug di MySQL.

- Il join sinistro genera risultati errati nell'outer (bug 22833364).

Aggiornamenti del motore di database Aurora MySQL 13/03/2018 (versione 2.01.1) (obsoleta)

Versione: 2.01.1

Aurora MySQL 2.01.1 è disponibile a livello generale. Le versioni 2.x di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL sono compatibili con MySQL 5.6.

Quando si crea un nuovo cluster di database Aurora MySQL, è possibile scegliere la compatibilità con MySQL 5.7 o MySQL 5.6. Quando si ripristina una snapshot compatibile con MySQL 5.6 si può scegliere la compatibilità con MySQL 5.7 o MySQL 5.6.

Puoi ripristinare le snapshot di Aurora MySQL 1.14*, 1.15*, 1.16* e 1.17* in Aurora MySQL 2.01.1.

Non è invece possibile eseguire un aggiornamento in loco dei cluster Aurora MySQL 1.x ad Aurora MySQL 2.01.1 o ripristinare Aurora MySQL 2.01.1 da un backup Amazon S3. Prevediamo di rimuovere queste limitazioni in una versione successiva di Aurora MySQL 2.x.

Lo schema delle prestazioni è disabilitato per questa release di Aurora MySQL 5.7. Esegui l'aggiornamento a Aurora 2.03 per avere assistenza sullo schema di performance.

Confronto con Aurora MySQL 5.6

Le seguenti caratteristiche di Amazon Aurora MySQL sono supportate in Aurora MySQL 5.6, ma attualmente non in Aurora MySQL 5.7.

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.

- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Al momento Aurora MySQL 2.01.1 non supporta le caratteristiche aggiunte nella versione 1.16 di Aurora MySQL e nelle versioni successive. Per informazioni su Aurora MySQL versione 1.16, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).

Compatibilità MySQL 5.7

Aurora MySQL 2.01.1 è compatibile con MySQL 5.7 e include caratteristiche come il supporto di JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Al momento, Aurora MySQL 2.01.1 non supporta le seguenti caratteristiche di MySQL 5.7:

- Identificatori globali di transazione (). GTIDs Aurora MySQL supporta GTIDs la versione 2.04 e successive.
- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`
- Protocollo X

Differenze di CLI tra Aurora MySQL 2.x e Aurora MySQL 1.x

- Il nome del motore di Aurora MySQL 2.x è `aurora-mysql`; mentre quello di Aurora MySQL 1.x continua a essere `aurora`.
- Il gruppo di parametri predefiniti di Aurora MySQL 2.x è `default.aurora-mysql5.7`; mentre quello di Aurora MySQL 1.x continua a essere `default.aurora5.6`.
- Il nome di famiglia del gruppo di parametri del cluster di database di Aurora MySQL 2.x è `aurora-mysql5.7`; mentre quello di Aurora MySQL 1.x continua a essere `aurora5.6`.

Consulta la documentazione di Aurora per la serie completa di comandi CLI e le differenze tra Aurora MySQL 2.x e Aurora MySQL 1.x.

Miglioramenti

- È stato risolto un problema relativo al ripristino degli snapshot a causa del quale i privilegi del database specifici per Aurora venivano creati in modo errato nel caso di ripristino di uno snapshot compatibile con MySQL 5.6 attraverso la compatibilità di MySQL 5.7.
- È stato aggiunto il supporto per il ripristino degli snapshot di 1.17.

Aggiornamenti del motore di database Aurora MySQL 06/02/2018 (versione 2.01) (obsoleta)

Versione: 2.01

Aurora MySQL 2.01 è disponibile a livello generale. In futuro, le versioni 2.x di Aurora MySQL saranno compatibili con MySQL 5.7, mentre le versioni 1.x di Aurora MySQL saranno compatibili con MySQL 5.6.

Quando si crea un nuovo cluster di database Aurora MySQL, compresi quelli ripristinati da snapshot, è possibile scegliere la compatibilità con MySQL 5.7 o MySQL 5.6.

Puoi ripristinare le snapshot di Aurora MySQL 1.14*, 1.15* e 1.16* in Aurora MySQL 2.01.

Non è invece possibile eseguire un aggiornamento in loco dei cluster Aurora MySQL 1.x ad Aurora MySQL 2.01 o ripristinare Aurora MySQL 2.01 da un backup Amazon S3. Prevediamo di rimuovere queste limitazioni in una versione successiva di Aurora MySQL 2.x.

Lo schema delle prestazioni è disabilitato per questa release di Aurora MySQL 5.7. Esegui l'aggiornamento a Aurora 2.03 per avere assistenza sullo schema di performance.

Confronto con Aurora MySQL 5.6

Le seguenti caratteristiche di Amazon Aurora MySQL sono supportate in Aurora MySQL 5.6, ma attualmente non in Aurora MySQL 5.7.

- Prefetch asincrono delle chiavi (AKP). Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Hash join. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Funzioni native per richiamare funzioni in modo sincrono. AWS Lambda Per ulteriori informazioni, consulta Chiamare una funzione Lambda con una funzione nativa Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Raggruppare le scansioni. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).
- Migrazione dei dati da MySQL utilizzando un bucket Amazon S3. Per ulteriori informazioni, consulta Migrating data from MySQL by using an Amazon S3 bucket nella Guida per l'utente di Amazon Aurora.

Al momento Aurora MySQL 2.01 non supporta le caratteristiche aggiunte nella versione 1.16 di Aurora MySQL e nelle versioni successive. Per informazioni su Aurora MySQL versione 1.16, consulta Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta).

Compatibilità MySQL 5.7

Aurora MySQL 2.01 è compatibile con MySQL 5.7 e include funzionalità come il supporto JSON, gli indici spaziali e le colonne generate. Aurora MySQL utilizza un'implementazione nativa degli indici spaziali attraverso curve di ordine z per offrire prestazioni di scrittura migliorate di 20 volte e prestazioni di lettura migliorate di 10 volte rispetto a MySQL 5.7 per i set di dati spaziali.

Al momento, Aurora MySQL 2.01 non supporta le seguenti caratteristiche di MySQL 5.7:

- Identificatori globali di transazione (). GTIDs Aurora MySQL supporta GTIDs la versione 2.04 e successive.

- Plugin replica gruppi
- Maggiori dimensioni pagina
- Caricamento buffer pool InnoDB all'avvio
- Plugin parser full-text InnoDB
- Replica multi-source
- Ridimensionamento buffer pool online
- Plugin convalida password
- Plugin riscrittura query
- Filtri replica
- Istruzione SQL `CREATE TABLESPACE`
- Protocollo X

Differenze di CLI tra Aurora MySQL 2.x e Aurora MySQL 1.x

- Il nome del motore di Aurora MySQL 2.x è `aurora-mysql`; mentre quello di Aurora MySQL 1.x continua a essere `aurora`.
- Il gruppo di parametri predefiniti di Aurora MySQL 2.x è `default.aurora-mysql5.7`; mentre quello di Aurora MySQL 1.x continua a essere `default.aurora5.6`.
- Il nome di famiglia del gruppo di parametri del cluster di database di Aurora MySQL 2.x è `aurora-mysql5.7`; mentre quello di Aurora MySQL 1.x continua a essere `aurora5.6`.

Consulta la documentazione di Aurora per la serie completa di comandi CLI e le differenze tra Aurora MySQL 2.x e Aurora MySQL 1.x.

Aggiornamenti del motore del database per Amazon Aurora MySQL versione 1 (obsoleta)

Di seguito sono riportati alcuni aggiornamenti del motore di database Amazon Aurora versione 1:

- Aggiornamenti del motore di database Aurora MySQL 30/09/2021 (versione 1.23.4) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 28/06/2021 (versione 1.23.3) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 18/03/2021 (versione 1.23.2) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 24/11/2020 (versione 1.23.1) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 02/09/2020 (versione 1.23.0) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 03/06/2021 (versione 1.22.5) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 03/04/2021 (versione 1.22.4) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 11/09/2020 (versione 1.22.3) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 05/03/2020 (versione 1.22.2) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 23/12/2019 (versione 1.22.1) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 25/11/2019 (versione 1.22.0) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 25/11/2019 (versione 1.21.0) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 05/03/2020 (versione 1.20.1) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 11/11/2019 (versione 1.20.0) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 03/05/2020 (versione 1.19.6) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 19/09/2019 (versione 1.19.5) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 05/06/2019 (versione 1.19.2) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 09/05/2019 (versione 1.19.1) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 07/02/2019 (versione 1.19.0) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 20/09/2018 (versione 1.18.0) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 03/05/2020 (versione 1.17.9) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 17/01/2019 (versione 1.17.8) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 08/10/2018 (versione 1.17.7) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 06/09/2018 (versione 1.17.6) (obsoleta)

- Aggiornamenti del motore di database Aurora MySQL 14/08/2018 (versione 1.17.5) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 07/08/2018 (versione 1.17.4) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 05/06/2018 (versione 1.17.3) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 27/04/2018 (versione 1.17.2) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 23/03/2018 (versione 1.17.1) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 13/03/2018 (versione 1.17) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 20/11/2017 (versione 1.15.1) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 24/10/2017 (versione 1.15) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 13/03/2018 (versione 1.14.4) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 22/09/2017 (versione 1.14.1) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 07/08/2017 (versione 1.14) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 15/05/2017 (versione 1.13) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 04/05/2017 (versione 1.12) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 23/02/2017 (versione 1.11) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 12/01/2017 (versione 1.10.1) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 14/12/2016 (versione 1.10) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL: 10/11/2016 (versioni 1.9.0, 1.9.1) (obsolete)
- Aggiornamenti del motore di database Aurora MySQL 26/10/2016 (versione 1.8.1) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 18/10/2016 (versione 1.8) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 20/09/2016 (versione 1.7.1) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 30/08/2016 (versione 1.7.0) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 01/06/2016 (versione 1.6.5) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 06/04/2016 (versione 1.6) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 11/01/2016 (versione 1.5) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 03/12/2015 (versione 1.4) (obsoleta)
- Aggiornamenti del motore di database Aurora MySQL 16/10/2015 (versioni 1.2, 1.3) (obsolete)
- Aggiornamenti del motore di database Aurora MySQL 24/08/2015 (versione 1.1) (obsoleta)

Aggiornamenti del motore di database Aurora MySQL 30/09/2021 (versione 1.23.4) (obsoleta)

Versione: 1.23.4

Aurora MySQL 1.23.4 è disponibile a livello generale. Le versioni 2.* di Aurora MySQL sono compatibili con MySQL 5.7, mentre le versioni 1.* di Aurora MySQL sono compatibili con MySQL 5.6.

Questa versione del motore è pianificata per essere resa obsoleta il 28 febbraio 2023. Per ulteriori informazioni, consulta Preparazione per la fine del ciclo di vita di Amazon Aurora edizione compatibile con MySQL versione 1.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite la console RDS, la AWS CLI o l'API Amazon RDS.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Miglioramenti generali:

- È stato risolto un problema che poteva causare un elevato consumo della CPU nelle istanze del lettore a causa dell'eccessiva registrazione dei messaggi informativi nei file di log diagnostici interni.

Correzioni con alta priorità:

- CVE-2021-2307
- CVE-2021-2226
- CVE-2021-2160
- CVE-2021-2154
- CVE-2021-2060

- CVE-2021-2032
- CVE-2021-2001

Aggiornamenti del motore di database Aurora MySQL 28/06/2021 (versione 1.23.3) (obsoleta)

Versione: 1.23.3

Aurora MySQL 1.23.3 è disponibile a livello generale. Le versioni 1.* di Aurora MySQL sono compatibili con MySQL 5.6, mentre le versioni 2.* di Aurora MySQL sono compatibili con MySQL 5.7.

Questa versione del motore è pianificata per essere resa obsoleta il 28 febbraio 2023. Per ulteriori informazioni, consulta Preparazione per la fine del ciclo di vita di Amazon Aurora edizione compatibile con MySQL versione 1.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite la console RDS, la AWS CLI o l'API Amazon RDS.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Miglioramenti generali della stabilità e della disponibilità.

Correzioni di sicurezza:

- CVE-2021-23841
- CVE-2021-3449
- CVE-2020-28196

Aggiornamenti del motore di database Aurora MySQL 18/03/2021 (versione 1.23.2) (obsoleta)

Versione: 1.23.2

Aurora MySQL 1.23.2 è disponibile a livello generale. Le versioni 1.* di Aurora MySQL sono compatibili con MySQL 5.6, mentre le versioni 2.* di Aurora MySQL sono compatibili con MySQL 5.7.

Questa versione del motore è pianificata per essere resa obsoleta il 28 febbraio 2023. Per ulteriori informazioni, consulta Preparazione per la fine del ciclo di vita di Amazon Aurora edizione compatibile con MySQL versione 1.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite la console RDS, la AWS CLI o l'API Amazon RDS.

> **Note**
>
> Questa versione non è attualmente disponibile nelle seguenti regioni: AWS GovCloud (Stati Uniti orientali) [us-gov-east-1], AWS GovCloud (Stati Uniti occidentali) [-1]. us-gov-west Non appena disponibile verrà comunicato l'annuncio.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Correzioni con alta priorità:

- CVE-2020-14867
- CVE-2020-14812
- CVE-2020-14769
- CVE-2020-14765
- CVE-2020-14793
- CVE-2020-14672
- CVE-2020-1971

- CVE-2018-3143

Miglioramenti della disponibilità:

- Risolto un problema della funzionalità di ridimensionamento dello storage del cluster dinamico che poteva causare il riavvio delle istanze database del lettore.
- Risolto un problema di failover a causa di una race condition nella dichiarazione `RESET QUERY CACHE`.
- Risolto un arresto anomalo in una chiamata di una stored procedure nidificata con cache di query.
- Risolto un problema per impedire il riavvio ripetuto di `mysqld` durante il recupero da un troncamento incompleto di tabelle partizionate o sottopartizionate.
- Risolto un problema che poteva causare la mancata riuscita della migrazione da on-premise o RDS for MySQL a Aurora MySQL.
- Corretta una rara race condition in cui il database veniva riavviato durante il ridimensionamento del volume di storage.
- È stato risolto un problema nella gestione dei blocchi per cui una race condition può causare la condivisione di un blocco da parte di due transazioni, causando il riavvio del database.
- È stato risolto un problema relativo alla gestione della memoria del blocco delle transazioni con transazioni di scrittura a esecuzione prolungata che causavano il riavvio del database.
- È stata risolta una condizione di competizione nel gestore di blocchi che causava un riavvio o un failover del database durante il rollback della transazione.
- Risolto un problema durante l'aggiornamento da 5.6 a 5.7 quando la tabella aveva la DDL Fast Online abilitata in modalità lab nella versione 5.6.
- Sono stati risolti diversi problemi per cui il motore potrebbe riavviarsi durante l'applicazione di patch con tempo di inattività zero durante la verifica della presenza di un punto inattivo nell'attività del database per l'applicazione delle patch.
- Risolti diversi problemi relativi ai riavvii ripetuti dovuti a operazioni DDL interrotte, come `DROP TRIGGER`, `ALTER TABLE` e in particolare `ALTER TABLE` che modifica il tipo di partizionamento o il numero di partizioni in una tabella.
- Aggiornato il valore predefinito di `table_open_cache` su istanze 16XL e 24XL per evitare riavvii ripetuti e un elevato utilizzo della CPU su classi di istanze di grandi dimensioni (R4/R5-16XL, R5-12XL, R5-24XL). Ciò ha avuto un impatto sulle versioni 1.21.x e 1.22.x.
- Risolto un problema che causava l'interruzione di una replica binlog con un errore `HA_ERR_KEY_NOT_FOUND`.

Integrazione delle correzioni di bug della community di MySQL

- Replica: durante l'esecuzione di un' `SHOW BINLOG EVENTS` istruzione, qualsiasi transazione parallela è stata bloccata. La correzione assicura che il `SHOW BINLOG EVENTS` processo acquisisca ora solo un blocco per la durata del calcolo della posizione finale del file, pertanto le transazioni parallele non vengono bloccate per lunghe durate. (Bug #76618, Bug #20928790)

Aggiornamenti del motore di database Aurora MySQL 24/11/2020 (versione 1.23.1) (obsoleta)

Versione: 1.23.1

Aurora MySQL 1.23.1 è disponibile a livello generale. Le versioni 1.* di Aurora MySQL sono compatibili con MySQL 5.6, mentre le versioni 2.* di Aurora MySQL sono compatibili con MySQL 5.7.

Questa versione del motore è pianificata per essere resa obsoleta il 28 febbraio 2023. Per ulteriori informazioni, consulta Preparazione per la fine del ciclo di vita di Amazon Aurora edizione compatibile con MySQL versione 1.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite la console RDS, la AWS CLI o l'API Amazon RDS.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Correzioni di sicurezza:

Correzioni e altri miglioramenti per ottimizzare la gestione in un ambiente gestito. Ulteriori correzioni CVE riportate di seguito:

- CVE-2020-14559
- CVE-2020-14539

Modifiche non compatibili:

Questa versione introduce una modifica dell'autorizzazione che influisce sul comportamento del `mysqldump` comando. Gli utenti devono disporre del `PROCESS` privilegio per accedere alla `INFORMATION_SCHEMA.FILES` tabella. Per eseguire il `mysqldump` comando senza alcuna modifica, concedere il `PROCESS` privilegio all'utente del database a cui si connette il `mysqldump` comando. Inoltre puoi eseguire il`mysqldump`comando con l'`--no-tablespaces`opzione. Con questa opzione,l'`mysqldump` output non include alcuna`CREATE LOGFILE GROUP`o`CREATE TABLESPACE`istruzioni. In tal caso, il `mysqldump` comando non accede alla `INFORMATION_SCHEMA.FILES` tabella e non è necessario che tu conceda l' `PROCESS` autorizzazione.

Miglioramenti della disponibilità:

- Risolto un problema che causava il riavvio ripetuto di un'istanza di lettura Aurora in un cluster secondario di database globale che esegue la versione 1.23.0.
- Risolto un problema per il quale le repliche di una regione secondaria di un database globale potevano riavviarsi quando veniva effettuato l'aggiornamento alla versione 1.23.0 mentre il writer di una regione primaria era in una versione precedente.
- Risolta una perdita di memoria nella funzione di ridimensionamento dinamico, introdotta in Aurora MySQL 1.23.0.
- Risolto un problema che poteva causare il riavvio del server durante l'esecuzione di una query utilizzando la funzionalità di query parallela.
- Risolto un problema che poteva causare il blocco di una sessione client quando il motore del database rilevava un errore durante la lettura o la scrittura nella rete.

Aggiornamenti del motore di database Aurora MySQL 02/09/2020 (versione 1.23.0) (obsoleta)

Versione: 1.23.0

Aurora MySQL 1.23.0 è disponibile a livello generale. Le versioni 1.* di Aurora MySQL sono compatibili con MySQL 5.6, mentre le versioni 2.* di Aurora MySQL sono compatibili con MySQL 5.7.

Questa versione del motore è pianificata per essere resa obsoleta il 28 febbraio 2023. Per ulteriori informazioni, consulta Preparazione per la fine del ciclo di vita di Amazon Aurora edizione compatibile con MySQL versione 1.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

Puoi ripristinare lo snapshot di un database Aurora MySQL 1.* in Aurora MySQL 1.23.0.

> **Important**
>
> I miglioramenti apportati allo storage Aurora in questa versione limitano i percorsi di aggiornamento disponibili da Aurora MySQL 1.23 a Aurora MySQL 2.*. Quando si aggiorna un cluster Aurora MySQL 1.23 a 2.*, è necessario eseguire l'aggiornamento a Aurora MySQL 2.09.0 o versione successiva.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite la console RDS, la AWS CLI o l'API Amazon RDS.

> **Note**
>
> Questa versione non è attualmente disponibile nelle seguenti regioni: AWS GovCloud (Stati Uniti orientali) [us-gov-east-1], AWS GovCloud (Stati Uniti occidentali) [-1]. us-gov-west Non appena disponibile verrà comunicato l'annuncio.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Nuove caratteristiche:

- È ora possibile attivare o disattivare la query parallela per un cluster esistente modificando il valore del parametro cluster DB `aurora_parallel_query`. Non è necessario utilizzare l'impostazione `parallelquery` per il parametro `--engine-mode` durante la creazione del cluster.

  La query parallela è ora espansa per essere disponibile in tutte le regioni in cui Aurora MySQL è disponibile.

  Sono disponibili numerosi altri miglioramenti alle funzionalità e modifiche alle procedure per l'aggiornamento e l'abilitazione di query parallele in un cluster Aurora. Per ulteriori informazioni,

consulta Utilizzo di query in parallelo per Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

- Con questa versione, puoi creare istanze database MySQL di Amazon Aurora con un massimo di 128 TiB di storage. Il nuovo limite di storage è un aumento rispetto al precedente 64 TiB. La dimensione di storage di 128 TiB supporta database più grandi. Questa funzionalità non è supportata su istanze Small (db.t2 o db.t3). Un singolo spazio di tabella non può superare i 64 TiB a causa delle limitazioni di InnoDB con dimensioni di pagina di 16 KB.

  Aurora avvisa l'utente quando la dimensione del volume del cluster è vicina a 128 TiB, in modo da poter intervenire prima di raggiungere il limite di dimensione. Gli avvisi vengono visualizzati nel registro mysql e negli eventi RDS in Console di gestione AWS.
- Elaborazione log binaria (binlog) migliorata per ridurre il tempo di recupero degli arresti anomali e la latenza del tempo di commit quando sono coinvolte transazioni molto grandi.
- Aurora ridimensiona dinamicamente lo spazio di storage del cluster. Con il ridimensionamento dinamico, lo spazio di storage per il cluster DB Aurora diminuisce automaticamente quando si rimuovono i dati dal cluster DB. Per ulteriori informazioni, consulta Storage scaling nella Guida per l'utente di Amazon Aurora.

  > **Note**
  >
  > La funzionalità di ridimensionamento dinamico viene implementata in fasi nelle regioni in AWS cui Aurora è disponibile. A seconda della regione in cui si trova il cluster, questa funzionalità potrebbe non essere ancora disponibile. Per ulteriori informazioni, consulta l'annuncio delle novità.

Correzioni con alta priorità:

- CVE-2019-2911
- CVE-2019-2537
- CVE-2018-2787
- CVE-2018-2784
- CVE-2018-2645
- CVE-2018-2640

Miglioramenti della disponibilità:

- È stato risolto un problema nella gestione dei blocchi per cui una race condition può causare la condivisione di un blocco da parte di due transazioni, causando il riavvio del database.
- È stato risolto un problema relativo alla gestione della memoria del blocco delle transazioni con transazioni di scrittura a esecuzione prolungata che causavano il riavvio del database.
- È stata risolta una condizione di competizione nel gestore di blocchi che causava un riavvio o un failover del database durante il rollback della transazione.
- Risolto un problema durante l'aggiornamento da 5.6 a 5.7 quando `innodb_file_format` veniva modificato su una tabella con Fast DDL abilitato.
- Sono stati risolti diversi problemi per cui il motore potrebbe riavviarsi durante l'applicazione di patch con tempo di inattività zero durante la verifica della presenza di un punto inattivo nell'attività del database per l'applicazione delle patch.
- È stato risolto un problema relativo al ripristino DDL che influisce sul riavvio dell'istanza database durante il ripristino di un'operazione `DROP TRIGGER` interrotta.
- Risolto un problema che poteva causare l'indisponibilità del database se si verificava un arresto anomalo durante l'esecuzione di alcune operazioni di partizionamento. In particolare, un'operazione `ALTER TABLE` interrotta che modifica il tipo di partizionamento o il numero di partizioni in una tabella.
- Correggere il valore predefinito di `table_open_cache` su istanze 16XL e 24XL che potrebbero causare failover ripetuti e un elevato utilizzo della CPU su classi di istanze di grandi dimensioni (R4/R5-16XL, R5-12XL, R5-24XL). Ciò ha avuto un impatto su 1.21.x e 1.22.x.

Database globali.

- Compila i dati mancanti nella vista `INFORMATION_SCHEMA.REPLICA_HOST_STATUS` MySQL sulle AWS regioni primarie e secondarie in un database globale Aurora.
- Sono stati corretti errori di query imprevisti che potevano verificarsi in una regione secondaria di del database globale a causa della garbage collection dei record UNDO nella regione primaria, dopo problemi temporanei di connettività di rete tra le regioni primarie e secondarie.

Query parallela:

- È stato risolto un problema per cui una query parallela potrebbe causare la restituzione di un risultato vuoto da parte di una query a esecuzione prolungata.
- È stato risolto un problema per cui una query su una tabella di piccole dimensioni della replica di lettura di Aurora potrebbe richiedere più di un secondo.

- È stato risolto un problema che potrebbe causare un riavvio quando una query parallela e un'istruzione DML vengono eseguite contemporaneamente con un carico di lavoro elevato.

Miglioramenti generali:

- Risolto un problema per cui le query che utilizzano l'indice spaziale potevano restituire risultati parziali se l'indice spaziale veniva creato su tabelle con valori spaziali di grandi dimensioni già esistenti.
- Aumento della lunghezza massima consentita per le variabili di sistema di audit `server_audit_incl_users` e `server_audit_excl_users` da 1024 byte a 2000 byte.
- Risolto un problema per cui una replica binlog connessa a un binlog primario di Aurora MySQL potrebbe mostrare dati incompleti quando il binlog primario di Aurora MySQL carica i dati da S3 in `statement` `binlog_format`.
- Rispettare il comportamento della community per mappare binlog_format `mixed` su `row` anziché `statement` per caricare i dati.
- È stato risolto un problema che causava il blocco della replica binlog quando l'utente chiude la connessione e la sessione utilizza tabelle temporanee.
- Miglioramento del tempo di risposta di una query che coinvolge tabelle temporanee MyISAM.
- Risolto il problema di autorizzazione quando il worker binlog esegue una funzione Lambda nativa.
- Risolto un problema relativo alle repliche di lettura di Aurora quando si tenta di eseguire query o ruotare il log lento o il log generale.
- È stato risolto un problema che interrompeva la replica logica quando il parametro `binlog_checksum` è impostato su valori diversi sul master e sulla replica.
- È stato risolto un problema per cui la replica di lettura poteva visualizzare in modo transitorio i risultati parziali di una transazione recentemente impegnata nel writer.
- Includere le informazioni sulla transazione sottoposta a rollback `show engine innodb status` quando viene risolto un deadlock.

Integrazione delle correzioni di bug della community di MySQL

- Gli eventi Binlog con `ALTER TABLE ADD COLUMN ALGORITHM=QUICK` verranno riscritti come `ALGORITHM=DEFAULT` da essere compatibili con l'edizione della community.
- BUG #22350047: IF CLIENT KILLED AFTER ROLLBACK TO SAVEPOINT PREVIOUS STMTS COMMITTED

- Bug #29915479: RUNNING COM_REGISTER_SLAVE WITHOUT COM_BINLOG_DUMP CAN RESULTS IN SERVER EXIT
- Bug #30441969: BUG #29723340: MYSQL SERVER CRASH AFTER SQL QUERY WITH DATA ? AST
- Bug #30628268: OUT OF MEMORY CRASH
- Bug #27081349: UNEXPECTED BEHAVIOUR WHEN DELETE WITH SPATIAL FUNCTION
- Bug #27230859: UNEXPECTED BEHAVIOUR WHILE HANDLING INVALID POLYGON"
- Bug #27081349: UNEXPECTED BEHAVIOUR WHEN DELETE WITH SPATIAL"
- Bug #26935001: ALTER TABLE AUTO_INCREMENT TRIES TO READ INDEX FROM DISCARDED TABLESPACE
- Bug #29770705: SERVER CRASHED WHILE EXECUTING SELECT WITH SPECIFIC WHERE CLAUSE
- Bug #27659490: SELECT USING DYNAMIC RANGE AND INDEX MERGE USE TOO MUCH MEMORY(OOM)
- Bug #24786290: REPLICATION BREAKS AFTER BUG #74145 HAPPENS IN MASTER
- Bug #27703912: EXCESSIVE MEMORY USAGE WITH MANY PREPARE
- Bug #20527363: TRONCATE TEMPORARY TABLE CRASH:! DICT_ TF2 _FLAG_IS_SET (TABELLA, DICT_ _TEMPORANEO) TF2
- Bug#23103937 PS_TRUNCATE_ALL_TABLES() DOES NOT WORK IN SUPER_READ_ONLY MODE
- Bug #25053286: USE VIEW WITH CONDITION IN PROCEDURE CAUSES INCORRECT BEHAVIOR (risolto nella versione 5.6.36)
- Bug #25586773: INCORRECT BEHAVIOR FOR CREATE TABLE SELECT IN A LOOP IN SP (fixed in 5.6.39)
- Bug #27407480: AUTOMATIC_SP_PRIVILEGES REQUIRES NEED THE INSERT PRIVILEGES FOR MYSQL.USER TABLE
- Bug #26997096: `relay_log_space` valore non viene aggiornato in modo sincronizzato in modo che il suo valore è talvolta molto più alto dello spazio su disco effettivo utilizzato dai log di inoltro.
- Bug#15831300 SLAVE_TYPE_CONVERSIONS=ALL_NON_LOSSY NOT WORKING AS EXPECTED
- SSL Bug backport Bug #17087862, Bug #20551271
- Bug #16894092: PERFORMANCE REGRESSION IN 5.6.6+ FOR INSERT INTO ... SELECT ... FROM (risolto nella versione 5.6.15).

- Porta una correzione di bug relativa a `SLAVE_TYPE_CONVERSIONS`.

Aggiornamenti del motore di database Aurora MySQL 03/06/2021 (versione 1.22.5) (obsoleta)

Versione: 1.22.5

Aurora MySQL 1.22.5 è disponibile a livello generale. Le versioni 1.* di Aurora MySQL sono compatibili con MySQL 5.6, mentre le versioni 2.* di Aurora MySQL sono compatibili con MySQL 5.7.

Questa versione del motore è pianificata per essere resa obsoleta il 28 febbraio 2023. Per ulteriori informazioni, consulta Preparazione per la fine del ciclo di vita di Amazon Aurora edizione compatibile con MySQL versione 1.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

Per creare un cluster con una versione precedente di Aurora MySQL, specificare la versione del motore utilizzando la console RDS, AWS CLI o l'API di Amazon RDS.

> **Note**
>
> Questa versione è designata come versione di supporto a lungo termine (LTS). Per ulteriori informazioni, consulta Versioni con supporto a lungo termine (Long-Term Support, LTS) di Aurora MySQL nella Guida per l'utente di Amazon Aurora.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Miglioramenti della disponibilità:

- È stato risolto un problema che poteva causare l'arresto del database e successivamente il riavvio o il failover a causa di un conflitto di concorrenza tra thread di pulizia interni.
- È stato risolto un problema che poteva causare la mancata disponibilità del cluster se il database si fosse riavviato mentre manteneva le transazioni XA in stato preparato e quindi si fosse riavviato

di nuovo prima che venissero eseguiti il commit o il rollback di tali transazioni. Prima di questa correzione, è possibile risolvere il problema ripristinando il cluster a un point-in-time precedente al primo riavvio.

- È stato risolto un problema che poteva causare il blocco dell'eliminazione di InnoDB se il database si riavvia durante l'elaborazione di un'istruzione DDL. Di conseguenza, la lunghezza dell'elenco della cronologia di InnoDB aumenterebbe e il volume di archiviazione del cluster continuerebbe a crescere fino a quando non si riempie, rendendo il database non disponibile. Prima di questa correzione, è possibile contenere il problema riavviando nuovamente il database per sbloccare l'eliminazione.

Aggiornamenti del motore di database Aurora MySQL 03/04/2021 (versione 1.22.4) (obsoleta)

Versione: 1.22.4

Aurora MySQL 1.22.4 è disponibile a livello generale. Le versioni 1.* di Aurora MySQL sono compatibili con MySQL 5.6, mentre le versioni 2.* di Aurora MySQL sono compatibili con MySQL 5.7.

Questa versione del motore è pianificata per essere resa obsoleta il 28 febbraio 2023. Per ulteriori informazioni, consulta Preparazione per la fine del ciclo di vita di Amazon Aurora edizione compatibile con MySQL versione 1.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite la console RDS, la AWS CLI o l'API Amazon RDS.

> **Note**
>
> Questa versione è designata come versione di supporto a lungo termine (LTS). Per ulteriori informazioni, consulta Versioni con supporto a lungo termine (Long-Term Support, LTS) di Aurora MySQL nella Guida per l'utente di Amazon Aurora.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Correzioni di sicurezza:

Correzioni e altri miglioramenti per ottimizzare la gestione in un ambiente gestito. Ulteriori correzioni CVE riportate di seguito:

- CVE-2020-14867
- CVE-2020-14812
- CVE-2020-14793
- CVE-2020-14769
- CVE-2020-14765
- CVE-2020-14672
- CVE-2020-1971

Miglioramenti della disponibilità:

- È stato risolto un problema che poteva attivare il riavvio o il failover del database durante un comando `kill session`. Se riscontri questo problema, contatta l' AWS assistenza per attivare questa correzione sulla tua istanza.
- Registrazione binaria migliorata per ridurre il tempo di recupero da arresto anomalo e la latenza del tempo di commit quando sono coinvolte transazioni di grosse dimensioni.
- Risolto un problema che causava l'interruzione di una replica binlog con un errore `HA_ERR_KEY_NOT_FOUND`.

Aggiornamenti del motore di database Aurora MySQL 11/09/2020 (versione 1.22.3) (obsoleta)

Versione: 1.22.3

Aurora MySQL 1.22.3 è disponibile a livello generale. Le versioni 1.* di Aurora MySQL sono compatibili con MySQL 5.6, mentre le versioni 2.* di Aurora MySQL sono compatibili con MySQL 5.7.

Questa versione del motore è pianificata per essere resa obsoleta il 28 febbraio 2023. Per ulteriori informazioni, consulta Preparazione per la fine del ciclo di vita di Amazon Aurora edizione compatibile con MySQL versione 1.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite la console RDS, la AWS CLI o l'API Amazon RDS.

> **Note**
>
> Questa versione è designata come versione di supporto a lungo termine (LTS). Per ulteriori informazioni, consulta Versioni con supporto a lungo termine (Long-Term Support, LTS) di Aurora MySQL nella Guida per l'utente di Amazon Aurora.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Correzioni di sicurezza:

Correzioni e altri miglioramenti per ottimizzare la gestione in un ambiente gestito. Ulteriori correzioni CVE riportate di seguito:

- CVE-2020-14559
- CVE-2020-14539
- CVE-2020-2579
- CVE-2020-2812
- CVE-2020-2780
- CVE-2020-2763

Modifiche non compatibili:

Questa versione introduce una modifica dell'autorizzazione che influisce sul comportamento del `mysqldump` comando. Gli utenti devono disporre del `PROCESS` privilegio per accedere alla `INFORMATION_SCHEMA.FILES` tabella. Per eseguire il `mysqldump` comando senza alcuna modifica, concedere il `PROCESS` privilegio all'utente del database a cui si

connette il `mysqldump` comando. Inoltre puoi eseguire il`mysqldump`comando con l'`--no-tablespaces`opzione. Con questa opzione,l'`mysqldump` output non include alcuna`CREATE LOGFILE GROUP`o`CREATE TABLESPACE`istruzioni. In tal caso, il `mysqldump` comando non accede alla `INFORMATION_SCHEMA.FILES` tabella e non è necessario che tu conceda l' `PROCESS` autorizzazione.

Miglioramenti della disponibilità:

- Risolti i problemi che potevano causare il riavvio del server durante il ripristino di un'istruzione DDL di cui non era stato eseguito il commit.
- Risolte le race condition nel gestore dei blocchi che possono causare il riavvio del server.
- Risolto un problema per il quale l'agente di monitoraggio riavviava il server durante il ripristino di una transazione di grandi dimensioni

Miglioramenti generali:

- Modificato il comportamento da mappare `MIXED` `binlog_format` a `ROW` invece che `STATEMENT` durante l'esecuzione di `LOAD DATA FROM INFILE | S3`.
- Risolto un problema per cui una replica binlog connessa a un binlog primario di Aurora MySQL potrebbe mostrare dati incompleti quando il primario ha eseguito `LOAD DATA FROM S3` e `binlog_format` è impostato su `STATEMENT`.

Integrazione delle correzioni di bug della community di MySQL

- Bug 26654685: Un ID indice danneggiato rilevato durante un controllo di chiave esterna generava un'asserzione
- Bug 15831300: Per impostazione predefinita, quando si promuovono numeri interi da un tipo più piccolo sul master a un tipo più grande sullo slave (ad esempio, da una colonna SMALLINT sul master a una colonna BIGINT sullo slave), i valori promossi vengono trattati come se fossero firmati. Ora in questi casi è possibile modificare o sovrascrivere questo comportamento utilizzando uno o entrambi `ALL_SIGNED`, `ALL_UNSIGNED` nell'insieme di valori specificati per la variabile di sistema server slave_type_conversions . Per ulteriori informazioni, consultare Replica basata su righe: promozione e riduzione di livello degli attributi, nonché la descrizione della variabile.
- Bug 17449901: Con `foreign_key_checks=0`, InnoDB consentiva l'eliminazione di un indice richiesto da un vincolo di chiave esterna, posizionando la tabella in uno stato incoerente e causando il fallimento del controllo della chiave esterna che si verifica al caricamento della tabella.

Ora InnoDB impedisce di eliminare un indice richiesto da un vincolo di chiave esterna, anche con foreign_key_checks=0. Il vincolo di chiave esterna deve essere rimosso prima di eliminare l'indice di chiave esterna.

- BUG n. 20768847: Una operazione ALTER TABLE ... DROP INDEX su una tabella con dipendenze da chiavi esterne ha generato un'asserzione.

Aggiornamenti del motore di database Aurora MySQL 05/03/2020 (versione 1.22.2) (obsoleta)

Versione: 1.22.2

Aurora MySQL 1.22.2 è disponibile a livello generale. Le versioni 1.* di Aurora MySQL sono compatibili con MySQL 5.6, mentre le versioni 2.* di Aurora MySQL sono compatibili con MySQL 5.7.

Questa versione del motore è pianificata per essere resa obsoleta il 28 febbraio 2023. Per ulteriori informazioni, consulta Preparazione per la fine del ciclo di vita di Amazon Aurora edizione compatibile con MySQL versione 1.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite la console RDS, la AWS CLI o l'API Amazon RDS.

> **Note**
>
> Questa versione non è attualmente disponibile nelle seguenti regioni: AWS GovCloud (Stati Uniti orientali) [us-gov-east-1], AWS GovCloud (Stati Uniti occidentali) [-1]. us-gov-west Non appena disponibile verrà comunicato l'annuncio.
> Questa versione è designata come versione di supporto a lungo termine (LTS). Per ulteriori informazioni, consulta Versioni con supporto a lungo termine (Long-Term Support, LTS) di Aurora MySQL nella Guida per l'utente di Amazon Aurora.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Correzioni con alta priorità:

- È stato risolto un problema di errori di connessione intermittenti dopo la rotazione del certificato.
- È stato risolto un problema per cui la clonazione richiedeva più tempo in alcuni cluster di database con carichi di scrittura elevati.
- È stato risolto un problema che interrompeva la replica logica quando il parametro `binlog_checksum` è impostato su valori diversi sul master e sulla replica.
- È stato risolto un problema per cui log lenti e il log generale possono non ruotare correttamente su repliche di lettura.
- È stato risolto un problema con il comportamento ANSI Read Commit Isolation Level.

Aggiornamenti del motore di database Aurora MySQL 23/12/2019 (versione 1.22.1) (obsoleta)

Versione: 1.22.1

Aurora MySQL 1.22.1 è disponibile a livello generale. Le versioni 1.* di Aurora MySQL sono compatibili con MySQL 5.6, mentre le versioni 2.* di Aurora MySQL sono compatibili con MySQL 5.7.

Questa versione del motore è pianificata per essere resa obsoleta il 28 febbraio 2023. Per ulteriori informazioni, consulta Preparazione per la fine del ciclo di vita di Amazon Aurora edizione compatibile con MySQL versione 1.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite, the o l'API RDS Console di gestione AWS. AWS CLI È possibile eseguire l'aggiornamento dei cluster database di Aurora MySQL 1.* esistenti a Aurora MySQL 1.22.1.

> **Note**
>
> Questa versione non è attualmente disponibile nelle seguenti AWS regioni: AWS GovCloud (Stati Uniti orientali) [us-gov-east-1], (Stati Uniti occidentali) [-1], Cina AWS GovCloud

(Ningxia) [us-gov-westcn-northwest-1], Asia Pacifico (Hong Kong) [ap-east-1] e Medio Oriente (Bahrein) [me-south-1. Non appena disponibile verrà comunicato l'annuncio.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> La procedura per aggiornare il cluster di database è cambiata. Per ulteriori informazioni, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Correzioni importanti:

- Risolti i problemi che impedivano il ripristino del motore e coinvolgevano blocchi di tabelle e tabelle temporanee.
- Migliorata la stabilità del registro binario quando vengono utilizzate tabelle temporanee.

Correzioni con alta priorità:

- Corretta una perdita di memoria lenta in un sottosistema Aurora specifico di analisi e registrazione del database che riduce la memoria libera.

Aggiornamenti del motore di database Aurora MySQL 25/11/2019 (versione 1.22.0) (obsoleta)

Versione: 1.22.0

Aurora MySQL 1.22.0 è disponibile a livello generale. Le versioni 1.* di Aurora MySQL sono compatibili con MySQL 5.6, mentre le versioni 2.* di Aurora MySQL sono compatibili con MySQL 5.7.

Questa versione del motore è pianificata per essere resa obsoleta il 28 febbraio 2023. Per ulteriori informazioni, consulta Preparazione per la fine del ciclo di vita di Amazon Aurora edizione compatibile con MySQL versione 1.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite, the o l'API RDS Console di gestione AWS. AWS CLI È possibile eseguire l'aggiornamento dei cluster database di Aurora MySQL 1.* esistenti a Aurora MySQL 1.22.0.

> **Note**
>
> Questa versione non è attualmente disponibile nelle seguenti AWS regioni: AWS GovCloud (Stati Uniti orientali) [us-gov-east-1], (Stati Uniti occidentali) [-1], Cina AWS GovCloud (Ningxia) [us-gov-westcn-northwest-1], Asia Pacifico (Hong Kong) [ap-east-1], Medio Oriente (Bahrein) [me-south-1] e Sud America (San Paolo) [sa-east-1] east-1]. Non appena disponibile verrà comunicato l'annuncio.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> La procedura per aggiornare il cluster di database è cambiata. Per ulteriori informazioni, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Nuove caratteristiche:

- I cluster Aurora MySQL ora supportano i tipi di istanza r5.8xlarge, r5.16xlarge e r5.24xlarge.
- Binlog ha nuovi miglioramenti per migliorare la latenza del tempo di commit quando sono coinvolte transazioni di dimensioni molto grandi.

- Aurora MySQL ora ha un meccanismo per ridurre al minimo la finestra temporale durante la quale gli eventi di una transazione di grandi dimensioni vengono scritti su binlog in fase di commit. Ciò previene efficacemente l'esecuzione di un lungo ripristino offline agli arresti anomali del database durante tale intervallo di tempo. Questa caratteristica risolve anche il problema in cui una transazione di grandi dimensioni blocca le piccole transazioni sul commit di binlog. Questa caratteristica è disattivata per impostazione predefinita e può essere abilitata dal team di assistenza se necessario per il carico di lavoro. Se abilitata, viene attivata quando la dimensione della transazione è > 500 MB.
- Aggiunto supporto per il livello di isolamento ANSI `READ COMMITTED` nelle repliche di lettura. Questo livello di isolamento consente l'esecuzione di query a esecuzione prolungata nella replica di lettura senza influire sull'elevato throughput delle scritture sul nodo di scrittura. Per ulteriori informazioni, consulta Livelli di isolamento di Aurora MySQL.
- I database globali ora consentono di aggiungere aree di replica secondarie di sola lettura per i cluster di database distribuiti in queste AWS regioni: Stati Uniti orientali (Virginia settentrionale) [us-east-1], Stati Uniti orientali (Ohio) [us-east-2], Stati Uniti occidentali (California settentrionale) [us-west-1], Stati Uniti occidentali (Oregon) [us-west-2], Europa (Irlanda) [eu-west-1] Europa (Londra) [eu-west-2], Europa (Parigi) [eu-west-3], Asia Pacifico (Tokyo) [ap-northeast-1], Asia Pacifico (Seoul) [ap-northeast-2], Asia Pacifico (Singapore) [ap-southeast-1], Asia Pacifico (Singapore) [ap-southeast-1], Asia Pacifico (Sydney) [ap-southeast-2], Canada (Central) [ca-central-1], Europa (Francoforte) [eu-central-1] e Asia Pacifico (Mumbai) [ap-south-1].
- La caratteristica di conflitto delle righe è ora generalmente disponibile e non richiede l'attivazione della modalità lab Aurora. Questa caratteristica migliora significativamente il throughput per i carichi di lavoro in cui più transazioni si contendono le righe della stessa pagina.
- Questa versione ha aggiornato i file del fuso orario per supportare l'ultimo aggiornamento del fuso orario del Brasile per i nuovi cluster.

Correzioni importanti:

- CVE-2019-2922
- CVE-2019-2923
- CVE-2019-2924
- CVE-2019-2910

Correzioni con alta priorità:

- 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
- È stato risolto un problema nel componente di recupero DDL che causava tempi di inattività prolungati del database. I cluster che non sono disponibili dopo l'esecuzione della query `TRUNCATE TABLE` su una tabella con una colonna `AUTO_INCREMENT` devono essere aggiornati.
- È stato risolto un problema nel componente di recupero DDL che causava tempi di inattività prolungati del database. I cluster che non sono disponibili dopo l'esecuzione della query `DROP TABLE` su più tabelle in parallelo devono essere aggiornati.

Correzioni per la stabilità generale:

- È stato risolto un problema che causava il riavvio delle repliche di lettura durante una transazione a esecuzione prolungata. I clienti che riscontrano riavvii di replica che coincidono con un calo accelerato della memoria disponibile devono prendere in considerazione l'aggiornamento a questa versione.
- È stato risolto un problema che riportava erroneamente `ERROR 1836` quando una query nidificata veniva eseguita su una tabella temporanea nella replica di lettura.
- È stato risolto un errore di interruzione della query in parallelo nell'istanza di lettura Aurora durante l'esecuzione di un carico di lavoro di scrittura elevato nell'istanza di scrittura Aurora.
- È stato risolto un problema che causava il riavvio di un database configurato come master binlog mentre era in esecuzione un carico di lavoro di scrittura elevato.
- È stato risolto un problema di indisponibilità prolungata durante il riavvio del motore. Questo risolve un problema nell'inizializzazione del pool di buffer. Questo problema si verifica raramente ma può potenzialmente influire su qualsiasi versione supportata.
- È stato risolto un problema che generava dati incoerenti nella tabella `information_schema.replica_host_status`.

- È stato risolto un problema relativo alla race condition tra la query in parallelo e i percorsi di esecuzione standard che causava il riavvio intermittente dei nodi di lettura.
- Stabilità del database migliorata quando il numero di connessioni client supera il valore del parametro `max_connections`.
- Stabilità migliorata delle istanze di lettura bloccando DDL e query `LOAD FROM S3` non supportate.

Integrazione delle correzioni di bug della community di MySQL

- Bug#16346241 - SERVER CRASH IN ITEM_PARAM::QUERY_VAL_STR
- Bug#17733850 - NAME_CONST() CRASH IN ITEM_NAME_CONST::ITEM_NAME_CONST()
- Bug #20989615 - INNODB AUTO_INCREMENT PRODUCES SAME VALUE TWICE
- Bug #20181776 - ACCESS CONTROL DOESN'T MATCH MOST SPECIFIC HOST WHEN IT CONTAINS WILDCARD
- Bug #27326796 - MYSQL CRASH WITH INNODB ASSERTION FAILURE IN FILE PARS0PARS.CC
- Bug #20590013 - IF YOU HAVE A FULLTEXT INDEX AND DROP IT YOU CAN NO LONGER PERFORM ONLINE DDL

Aggiornamenti del motore di database Aurora MySQL 25/11/2019 (versione 1.21.0) (obsoleta)

Versione: 1.21.0

Aurora MySQL 1.21.0 è disponibile a livello generale. Le versioni 1.* di Aurora MySQL sono compatibili con MySQL 5.6, mentre le versioni 2.* di Aurora MySQL sono compatibili con MySQL 5.7.

Le versioni Aurora MySQL attualmente supportate sono 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.* e 2.07.*. Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite, the o l'API RDS Console di gestione AWS. AWS CLI È possibile eseguire l'aggiornamento dei cluster database di Aurora MySQL 1.* esistenti a Aurora MySQL 1.21.0.

> **Note**
>
> Questa versione non è attualmente disponibile nelle seguenti AWS regioni: AWS GovCloud (Stati Uniti orientali) [us-gov-east-1], (Stati Uniti occidentali) [-1], Cina AWS GovCloud (Ningxia) [us-gov-westcn-northwest-1], Asia Pacifico (Hong Kong) [ap-east-1], Europa (Stoccolma) [eu-north-1] e Medio Oriente (Bahrein) [me-south-1]. Non appena disponibile verrà comunicato l'annuncio.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> La procedura per aggiornare il cluster di database è cambiata. Per ulteriori informazioni, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Correzioni importanti:

- CVE-2018-0734
- CVE-2019-2534
- CVE-2018-2612
- CVE-2017-3599
- CVE-2018-2562
- CVE-2017-3329
- CVE-2018-2696
- CVE-2015-4737

Correzioni con alta priorità:

- Consigliamo vivamente ai clienti con dimensioni del database vicine a 64 TiB di eseguire l'aggiornamento a questa versione per evitare tempi di inattività a causa di bug di stabilità che incidono sui volumi vicini al limite di storage Aurora.

Correzioni per la stabilità generale:

- È stato risolto un errore di interruzione query parallela su istanze Aurora Reader durante l'esecuzione di un carico di lavoro di scrittura intenso sull'istanza Aurora Writer.
- È stato risolto un problema su istanze Aurora Reader che riduceva la quantità di memoria libera durante le transazioni di lunga durata in presenza di intenso traffico di commit della transazione sull'istanza writer.
- Il valore del parametro `aurora_disable_hash_join` viene ora mantenuto dopo il riavvio del database o la sostituzione dell'host.
- È stato risolto un problema correlato alla cache Full Text Search che causava l'esaurimento della memoria dell'istanza Aurora. I clienti che utilizzano la ricerca full-text devono eseguire l'aggiornamento.
- È stata migliorata la stabilità del database quando la caratteristica hash join è abilitata e l'istanza ha poca memoria. I clienti che utilizzano l'hash join devono eseguire l'aggiornamento.
- È stato risolto un problema nella cache delle query a causa del quale l'errore "Too many connections" (Troppe connessioni) poteva causare un riavvio.
- Risolto il problema relativo al calcolo della memoria libera nelle istanze T2 per includere lo spazio di memoria swap per evitare riavvii non necessari.

Integrazione delle correzioni di bug della community di MySQL

- Bug #19929406: HANDLE_FATAL_SIGNAL (SIG=11) IN __MEMMOVE_ _BACK FROM STRING: :COPY SSSE3
- Bug #17059925: per istruzioni UNION, il valore esaminato dalle righe è stato calcolato erroneamente. Questo bug si manifestava con valori troppo grandi per la colonna `ROWS_EXAMINED` delle tabelle di istruzioni schema di prestazioni (ad esempio events_statements_current).
- Bug #11827369: alcune query con sottoquery `SELECT ... FROM DUAL` nidificate generavano un'asserzione.
- Bug #16311231: restituiti risultati errati se una query conteneva una sottoquery in una clausola `IN` che conteneva un'operazione XOR nella clausola `WHERE`.

Aggiornamenti del motore di database Aurora MySQL 05/03/2020 (versione 1.20.1) (obsoleta)

Versione: 1.20.1

Aurora MySQL 1.20.1 è disponibile a livello generale. Le versioni 1.* di Aurora MySQL sono compatibili con MySQL 5.6, mentre le versioni 2.* di Aurora MySQL sono compatibili con MySQL 5.7.

Le versioni Aurora MySQL attualmente supportate sono 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.* e 2.07.*. Puoi ripristinare lo snapshot di un database Aurora MySQL 1.* in Aurora MySQL 1.20.1.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite la console RDS, la AWS CLI o l'API Amazon RDS.

> **Note**
>
> Questa versione non è attualmente disponibile nelle seguenti regioni: AWS GovCloud (Stati Uniti orientali) [us-gov-east-1], AWS GovCloud (Stati Uniti occidentali) [-1]. us-gov-west Non appena disponibile verrà comunicato l'annuncio.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Correzioni con alta priorità:

- È stato risolto un problema di errori di connessione intermittenti dopo la rotazione del certificato.
- È stato risolto un problema relativo alla simultaneità della chiusura della connessione che causava un failover in condizioni di carico di lavoro elevato.

Correzioni per la stabilità generale:

- È stato corretto un arresto anomalo durante l'esecuzione di una query complessa che coinvolgeva join multi-tabella e aggregazione che utilizza tabelle intermedie internamente.

Aggiornamenti del motore di database Aurora MySQL 11/11/2019 (versione 1.20.0) (obsoleta)

Versione: 1.20.0

Aurora MySQL 1.20.0 è disponibile a livello generale. Le versioni 1.* di Aurora MySQL sono compatibili con MySQL 5.6, mentre le versioni 2.* di Aurora MySQL sono compatibili con MySQL 5.7.

Le versioni Aurora MySQL attualmente supportate sono 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 1.20.*, 2.01.*, 2.02.*, 2.03.* e 2.04.*. Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite, the o l'API RDS Console di gestione AWS. AWS CLI È possibile eseguire l'aggiornamento dei cluster database di Aurora MySQL 1.* esistenti fino alla versione 1.19.5 a Aurora MySQL 1.20.0.

> **Note**
>
> Questa versione non è attualmente disponibile nelle seguenti AWS regioni: AWS GovCloud (Stati Uniti orientali) [us-gov-east-1], (Stati Uniti occidentali) [-1], Cina AWS GovCloud (Ningxia) [us-gov-westcn-northwest-1], Asia Pacifico (Hong Kong) [ap-east-1], Europa (Stoccolma) [eu-north-1] e Medio Oriente (Bahrein) [me-south-1]. Non appena disponibile verrà comunicato l'annuncio.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> La procedura per aggiornare il cluster di database è cambiata. Per ulteriori informazioni, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Correzioni importanti:

- CVE-2018-0734
- CVE-2019-2534
- CVE-2018-2612
- CVE-2017-3599
- CVE-2018-2562
- CVE-2017-3329
- CVE-2018-2696
- CVE-2015-4737

Correzioni con alta priorità:

- Consigliamo vivamente ai clienti con dimensioni del database vicine a 64 TiB di eseguire l'aggiornamento a questa versione per evitare tempi di inattività a causa di bug di stabilità che incidono sui volumi vicini al limite di storage Aurora.

Correzioni per la stabilità generale:

- È stato risolto un errore di interruzione query parallela su istanze Aurora Reader durante l'esecuzione di un carico di lavoro di scrittura intenso sull'istanza Aurora Writer.
- È stato risolto un problema su istanze Aurora Reader che riduceva la quantità di memoria libera durante le transazioni di lunga durata in presenza di intenso traffico di commit della transazione sull'istanza writer.
- Il valore del parametro `aurora_disable_hash_join` viene ora mantenuto dopo il riavvio del database o la sostituzione dell'host.
- È stato risolto un problema correlato alla cache Full Text Search che causava l'esaurimento della memoria dell'istanza Aurora. I clienti che utilizzano la ricerca full-text devono eseguire l'aggiornamento.
- È stata migliorata la stabilità del database quando la caratteristica hash join è abilitata e l'istanza ha poca memoria. I clienti che utilizzano l'hash join devono eseguire l'aggiornamento.
- È stato risolto un problema nella cache delle query a causa del quale l'errore "Too many connections" (Troppe connessioni) poteva causare un riavvio.
- Risolto il problema relativo al calcolo della memoria libera nelle istanze T2 per includere lo spazio di memoria swap per evitare riavvii non necessari.

Integrazione delle correzioni di bug della community di MySQL

- Bug #19929406: HANDLE_FATAL_SIGNAL (SIG=11) IN __MEMMOVE_ _BACK FROM STRING: :COPY SSSE3
- Bug #17059925: per istruzioni UNION, il valore esaminato dalle righe è stato calcolato erroneamente. Questo bug si manifestava con valori troppo grandi per la colonna `ROWS_EXAMINED` delle tabelle di istruzioni schema di prestazioni (ad esempio events_statements_current).
- Bug #11827369: alcune query con sottoquery `SELECT ... FROM DUAL` nidificate generavano un'asserzione.
- Bug #16311231: restituiti risultati errati se una query conteneva una sottoquery in una clausola `IN` che conteneva un'operazione XOR nella clausola `WHERE`.

Aggiornamenti del motore di database Aurora MySQL 03/05/2020 (versione 1.19.6) (obsoleta)

Versione: 1.19.6

Aurora MySQL 1.19.6 è disponibile a livello generale. Le versioni 1.* di Aurora MySQL sono compatibili con MySQL 5.6, mentre le versioni 2.* di Aurora MySQL sono compatibili con MySQL 5.7.

Questa versione del motore è pianificata per essere resa obsoleta il 28 febbraio 2023. Per ulteriori informazioni, consulta Preparazione per la fine del ciclo di vita di Amazon Aurora edizione compatibile con MySQL versione 1.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

Puoi ripristinare lo snapshot di un database Aurora MySQL 1.* in Aurora MySQL 1.19.6.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite la console RDS, la AWS CLI o l'API Amazon RDS.

> **Note**
>
> Questa versione non è attualmente disponibile nelle seguenti regioni: AWS GovCloud (Stati Uniti orientali) [us-gov-east-1], AWS GovCloud (Stati Uniti occidentali) [-1]. us-gov-west Non appena disponibile verrà comunicato l'annuncio.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Correzioni con alta priorità:

- È stato risolto un problema di errori di connessione intermittenti dopo la rotazione del certificato.

Aggiornamenti del motore di database Aurora MySQL 19/09/2019 (versione 1.19.5) (obsoleta)

Versione: 1.19.5

Aurora MySQL 1.19.5 è disponibile a livello generale. Le versioni 1.* di Aurora MySQL sono compatibili con MySQL 5.6, mentre le versioni 2.* di Aurora MySQL sono compatibili con MySQL 5.7.

Questa versione del motore è pianificata per essere resa obsoleta il 28 febbraio 2023. Per ulteriori informazioni, consulta Preparazione per la fine del ciclo di vita di Amazon Aurora edizione compatibile con MySQL versione 1.

Le versioni di Aurora MySQL attualmente supportate sono 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.04.*, 2.07.*, 2.08.*, 2.09.*, 2.10.*, 3.01.* e 3.02.*.

È possibile eseguire l'aggiornamento dei cluster database di Aurora MySQL 1.19.5. Puoi ripristinare gli snapshot di Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.1 e 1.19.2 in Aurora MySQL 1.19.5.

Per utilizzare una versione precedente di Aurora MySQL, è possibile creare nuovi cluster di database specificando la versione del motore tramite l' Console di gestione AWS API RDS. AWS CLI

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Questa versione non è attualmente disponibile nelle seguenti AWS regioni: Europa (Londra) [eu-west-2] AWS GovCloud , (Stati Uniti orientali) [-1], (Stati Uniti occidentali) us-gov-east [-1]

AWS GovCloud , Cina (Ningxia) [us-gov-westcn-northwest-1] e Asia Pacifico (Hong Kong) [ap-east-1]. Non appena disponibile verrà comunicato l'annuncio.

> **Note**
>
> La procedura per aggiornare il cluster di database è cambiata. Per ulteriori informazioni, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

- È stato risolto un problema su istanze Aurora Reader che riduceva la quantità di memoria libera durante le transazioni di lunga durata in presenza di intenso traffico di commit della transazione sull'istanza writer.
- È stato risolto un errore di interruzione query parallela su istanze Aurora Reader durante l'esecuzione di un carico di lavoro di scrittura intenso sull'istanza Aurora Writer.
- Il valore del parametro `aurora_disable_hash_join` viene ora mantenuto dopo il riavvio del database o la sostituzione dell'host.
- È stato risolto un problema correlato alla cache Full Text Search che causava l'esaurimento della memoria dell'istanza Aurora.
- È stata migliorata la stabilità del database quando le dimensioni del volume sono prossime al limite di 64 tebibyte (TiB) prenotando 160 GB di spazio per il completamento del ripristino del flusso di lavoro senza un failover.
- È stata migliorata la stabilità del database quando la caratteristica hash join è abilitata e l'istanza ha poca memoria.
- È stato corretto il calcolo della memoria libera per includere spazio di memoria di scambio su istanze T2 che ne causava il riavvio prematuro.
- È stato risolto un problema nella cache delle query a causa del quale l'errore "Too many connections" (Troppe connessioni) poteva causare un riavvio.

Integrazione delle correzioni di bug della community di MySQL

- CVE-2018-2696

- CVE-2015-4737
- Bug #19929406: HANDLE_FATAL_SIGNAL (SIG=11) IN __MEMMOVE_ _BACK FROM STRING: :COPY SSSE3
- Bug #17059925: per istruzioni UNION, il valore esaminato dalle righe è stato calcolato erroneamente. Questo bug si manifestava con valori troppo grandi per la colonna `ROWS_EXAMINED` delle tabelle di istruzioni schema di prestazioni (ad esempio events_statements_current).
- Bug #11827369: alcune query con sottoquery `SELECT ... FROM DUAL` nidificate generavano un'asserzione.
- Bug #16311231: restituiti risultati errati se una query conteneva una sottoquery in una clausola `IN` che conteneva un'operazione XOR nella clausola `WHERE`.

Aggiornamenti del motore di database Aurora MySQL 05/06/2019 (versione 1.19.2) (obsoleta)

Versione: 1.19.2

Aurora MySQL 1.19.2 è disponibile a livello generale. Tutti i nuovi cluster di database Aurora MySQL compatibili con MySQL 5.6, compresi quelli ripristinati da snapshot, verranno creati in nella versione 1.17.8, 1.19.0, 1.19.1 o 1.19.2. È possibile, ma non necessario, eseguire l'aggiornamento dei cluster di database esistenti a Aurora MySQL versione 1.19.2. Per usare una versione più vecchia puoi creare nuovi cluster di database in Aurora MySQL 1.14.4, Aurora MySQL 1.15.1, Aurora MySQL 1.16, Aurora MySQL 1.17.8 o Aurora MySQL 1,18. Puoi farlo utilizzando AWS CLI o l'API Amazon RDS e specificando la versione del motore.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Questa versione non è attualmente disponibile nelle regioni AWS GovCloud (Stati Uniti occidentali) [us-gov-west-1], Europa (Stoccolma) [eu-north-1], Cina (Ningxia) [cn-northwest-1] e Asia Pacifico (Hong Kong) [ap-east-1]. AWS Non appena disponibile verrà comunicato l'annuncio.

> **Note**
>
> La procedura per aggiornare il cluster di database è cambiata. Per ulteriori informazioni, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

- È stato risolto un problema che poteva causare errori durante il caricamento di dati in Aurora da Amazon S3.
- È stato risolto un problema che poteva causare errori durante il caricamento di dati Aurora in Amazon S3.
- È stato risolto un problema che creava sessioni zombie lasciate in uno stato terminato.
- È stato risolto un problema che causava l'interruzione delle connessioni durante la gestione di un errore nella gestione del protocollo di rete.
- È stato risolto un problema che poteva causare un arresto anomalo durante la gestione di tabelle partizionate.
- È stato risolto un problema correlato alla replica del binlog di creazione di trigger.

Aggiornamenti del motore di database Aurora MySQL 09/05/2019 (versione 1.19.1) (obsoleta)

Versione: 1.19.1

Aurora MySQL 1.19.1 è disponibile a livello generale. Tutti i nuovi cluster di database Aurora MySQL compatibili con MySQL 5.6, compresi quelli ripristinati da snapshot, verranno creati in 1.17.8, 1.19.0. o 1.19.1. È possibile, ma non necessario, eseguire l'aggiornamento dei cluster di database esistenti a Aurora MySQL versione 1.19.1. Per usare una versione più vecchia puoi creare nuovi cluster di database in Aurora MySQL 1.14.4, Aurora MySQL 1.15.1, Aurora MySQL 1.16, Aurora MySQL 1.17.8 o Aurora MySQL 1,18. Puoi farlo utilizzando AWS CLI o l'API Amazon RDS e specificando la versione del motore.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Questa versione non è attualmente disponibile nelle regioni AWS GovCloud (Stati Uniti occidentali) [us-gov-west-1] e Cina (Pechino) [cn-north-1]. Non appena disponibile verrà comunicato l'annuncio.

> **Note**
>
> La procedura per aggiornare il cluster di database è cambiata. Per ulteriori informazioni, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

- Corretto un bug nella replica binlog che può causare un problema sulle istanze Aurora configurate come slave binlog.
- Corretto un errore nella gestione di determinati tipi di comandi `ALTER TABLE`.
- Corretto un problema di connessioni interrotte per un errore di gestione del protocollo di rete.

Aggiornamenti del motore di database Aurora MySQL 07/02/2019 (versione 1.19.0) (obsoleta)

Versione: 1.19.0

Aurora MySQL 1.19.0 è disponibile a livello generale. Tutti i nuovi cluster di database Aurora MySQL compatibili con MySQL 5.6, compresi quelli ripristinati da snapshot, verranno creati nella versione 1.17.8 o 1.19.0. È possibile, ma non necessario, eseguire l'aggiornamento dei cluster di database esistenti a Aurora MySQL versione 1.19.0. Per usare una versione più vecchia puoi creare nuovi cluster di database in Aurora MySQL 1.14.4, Aurora MySQL 1.15.1, Aurora MySQL 1.16, Aurora MySQL 1.17.8 o Aurora MySQL 1.18.0. Puoi farlo utilizzando AWS CLI o l'API Amazon RDS e specificando la versione del motore.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Questa versione non è attualmente disponibile nelle regioni AWS GovCloud (Stati Uniti occidentali) [us-gov-west-1] e Cina (Pechino) [cn-north-1]. Non appena disponibile verrà comunicato l'annuncio.

> **Note**
>
> La procedura per aggiornare il cluster di database è cambiata. Per ulteriori informazioni, consulta Aggiornamento della versione secondaria o del livello di patch di un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Funzionalità

- Selezione della versione Aurora – A partire dalla Aurora MySQL 1.19.0, è possibile scegliere tra tre versioni di MySQL 5.6 compatibile con Aurora sulla console Amazon RDS. Per ulteriori informazioni, consulta Verifica o specifica delle versioni del motore Aurora MySQL tramite AWS nella Guida per l'utente di Amazon Aurora.

Miglioramenti

- Correzione di un problema di stabilità correlato a una query `CHECK TABLE` su una replica Aurora.
- Introduzione di una nuova variabile utente globale `aurora_disable_hash_join` per disabilitare Hash Join.
- Correzione di un problema di stabilità durante la creazione della riga di output durante più hash join di tabella.
- Correzione di un problema che ha restituito un risultato errato a causa di una modifica di piano durante il controllo dell'applicabilità Hash Join.

- L'applicazione delle patch senza tempi di inattività è supportata con transazioni a esecuzione lunga. Questo miglioramento diverrà effettivo durante l'aggiornamento dalla versione 1.19 a una superiore.
- L'applicazione delle patch senza tempi di inattività è ora supportata se binlog è abilitato. Questo miglioramento diverrà effettivo durante l'aggiornamento dalla versione 1.19 a una superiore.
- Correzione di un problema che ha causato un picco di utilizzo della CPU sulla replica Aurora non correlato al carico di lavoro.
- Correzione di una race condition nella gestione dei blocchi che provocava il riavvio del database.
- Correzione di una race condition nel componente di gestione dei blocchi per migliorare la stabilità delle istanze Aurora.
- Miglioramento della stabilità con il rilevatore di deadlock all'interno del componente di gestione dei blocchi.
- `INSERT`L'operazione sulla tabella è vietata se InnoDB rileva che l'indice è corrotto.
- Correzione di un problema di stabilità in Fast DDL.
- Miglioramento della stabilità di Aurora mediante la riduzione del consumo di memoria nella scansione del batching di una query secondaria a riga singola.
- Correzione di un problema di stabilità che si è verificato dopo che una chiave esterna è stata rilasciata quando la variabile del sistema `foreign_key_checks` era impostata su 0.
- Correzione di un problema nella caratteristica Out Of Memory Avoidance che ha sostituito erroneamente le modifiche al valore `table_definition_cache` apportate dall'utente.
- Correzione dei problemi di stabilità nella caratteristica Out Of Memory Avoidance.
- Correzione di un problema che ha impostato `query_time` e `lock_time` in `slow_query_log` su valori inutili.
- Correzione di un problema di stabilità della query parallela generato dalla gestione impropria di una collation interna di stringa.
- Correzione di un problema di stabilità della query parallela generato da una ricerca di indice secondaria.
- Correzione di un problema di stabilità della query parallela generato da un aggiornamento multi tabella.

Integrazione delle correzioni di bug della community di MySQL

- BUG #32917: DETECT ORPHAN TEMP-POOL FILES, AND HANDLE GRACEFULLY

- BUG #63144: CREATE TABLE IF NOT EXISTS - IL BLOCCO DEI METADATI È TROPPO RESTRITTIVO

Aggiornamenti del motore di database Aurora MySQL 20/09/2018 (versione 1.18.0) (obsoleta)

Versione: 1.18.0

Aurora MySQL 1.18.0 è disponibile a livello generale. Tutti i nuovi cluster di query parallele Aurora MySQL compatibili con MySQL 5.6, compresi quelli ripristinati da snapshot, verranno creati in Aurora MySQL; versione 1.18.0. È possibile, ma non necessario, eseguire l'aggiornamento dei cluster di query parallele esistenti a Aurora MySQL versione 1.18.0. È possibile creare nuovi cluster DB in Aurora MySQL 1.14.4, Aurora MySQL 1.15.1, Aurora MySQL 1.16 o Aurora MySQL 1.17.6. Puoi farlo utilizzando AWS CLI o l'API Amazon RDS e specificando la versione del motore.

Con la versione 1.18.0 di Aurora MySQL, viene utilizzato un modello di patch del cluster che consente di applicare le patch a tutti i nodi del cluster di database Aurora contemporaneamente.

> **Important**
>
> Aurora MySQL 1.18.0 si applica solo ai cluster di query Aurora parallele. Se si aggiorna un cluster 5.6.10a con provisioning, la versione risultante è 1.17.8. Se si aggiorna un cluster 5.6.10a query parallela, la versione risultante è 1.18.0.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Funzionalità

- La query parallela è disponibile con questa versione, per i nuovi cluster e gli snapshot ripristinati. La query parallela Aurora MySQL è un'ottimizzazione che parallelizza parte del calcolo coinvolto nell'elaborazione di query a uso intensivo di dati I/O . Il lavoro che viene parallelizzato include il recupero di righe dalla memoria, l'estrazione dei valori della colonna e la determinazione di quali righe corrispondono alle condizioni nelle clausole join e nella clausola `WHERE`. Questa attività che implicano grandi quantità di dati sono delegate (in termini di ottimizzazione del database, trasferite)

a più nodi nel livello di storage distribuito di Aurora. Senza query parallela, ogni query porta tutti i dati scansionati a un singolo nodo all'interno del cluster Aurora MySQL (il nodo head) ed esegue l'elaborazione in quel nodo.

- Quando la funzionalità di query in parallelo è abilitata, il motore Aurora MySQL determina automaticamente quando utilizzarla senza richiedere modifiche SQL come hint o attributi di tabella.

Per ulteriori informazioni, consulta Utilizzo di query in parallelo per Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

- OOM Avoidance: questa funzionalità monitora la memoria di sistema e tiene traccia del consumo di memoria da parte dei vari componenti del database. Quando la memoria del sistema si riduce, esegue una serie di azioni per liberare tale memoria da alcuni componenti tracciati nel tentativo di evitare l'esaurimento della memoria (Out of Memory, OOM) e il riavvio del database. Questa funzionalità migliorativa è abilitata per impostazione predefinita per le istanze t2 e può essere abilitata su altri classi di istanze attraverso un nuovo parametro di istanza denominato `aurora_oom_response`. Il parametro a livello di istanza acquisisce una stringa di azioni separate da virgola che un'istanza deve eseguire quando la memoria è insufficiente. Azioni valide sono "print", "tune", "decline", "kill_query" o qualsiasi combinazione di queste. Una stringa vuota significa che non ci sono azioni da eseguire e di fatto disabilita la funzionalità. Tieni presente che le azioni predefinite per la funzionalità sono "print, tune". Esempio di utilizzo:
  - "print": –stampa solo le query che richiedono una grande quantità di memoria.
  - "tune":– ottimizza le cache delle tabelle interne per restituire un po' di memoria al sistema.
  - "decline":– rifiuta le nuove query quando l'istanza ha poca memoria.
  - "kill_query":– termina le query in ordine discendente relativamente al consumo di memoria fino a che la memoria dell'istanza non supera la soglia minima. Le istruzioni DDL (Data Definition Language) non vengono eliminate.
  - "print, tune": –esegue le azioni descritte per "print" e "tune".
  - "tune, decline, kill_query":– esegue le azioni descritte per "tune", "decline" e "kill_query".

  Per informazioni sulle out-of-memory condizioni di gestione e altri consigli per la risoluzione dei problemi, consulta i problemi di memoria esaurita di Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Aggiornamenti del motore di database Aurora MySQL 03/05/2020 (versione 1.17.9) (obsoleta)

Versione: 1.17.9

Aurora MySQL 1.17.9 è disponibile a livello generale. Le versioni 1.* di Aurora MySQL sono compatibili con MySQL 5.6, mentre le versioni 2.* di Aurora MySQL sono compatibili con MySQL 5.7.

Le versioni Aurora MySQL attualmente supportate sono 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.* e 2.07.*. Puoi ripristinare lo snapshot di un database Aurora MySQL 1.* in Aurora MySQL 1.17.9.

Per creare un cluster con una versione precedente di Aurora MySQL, specifica la versione del motore tramite la console RDS, la AWS CLI o l'API Amazon RDS.

> **Note**
>
> Questa versione non è attualmente disponibile nelle seguenti regioni: AWS GovCloud (Stati Uniti orientali) [us-gov-east-1], AWS GovCloud (Stati Uniti occidentali) [-1]. us-gov-west Non appena disponibile verrà comunicato l'annuncio.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

Correzioni con alta priorità:

- È stato risolto un problema di errori di connessione intermittenti dopo la rotazione del certificato.

Aggiornamenti del motore di database Aurora MySQL 17/01/2019 (versione 1.17.8) (obsoleta)

Versione: 1.17.8

Aurora MySQL 1.17.8 è disponibile a livello generale. Tutti i nuovi cluster di database Aurora MySQL compatibili con MySQL 5.6, compresi quelli ripristinati da snapshot, verranno creati in Aurora MySQL; versione 1.17.8. È possibile, ma non necessario, eseguire l'aggiornamento dei cluster di database esistenti a Aurora MySQL versione 1.17.8. Per usare una versione più vecchia puoi creare nuovi cluster di database in Aurora MySQL 1.14.4, 1.15.1, 1.16 o 1.17.7. Puoi farlo utilizzando AWS CLI o l'API Amazon RDS e specificando la versione del motore.

Con la versione 1.17.8 di Aurora MySQL, viene utilizzato un modello di patch del cluster che consente di applicare le patch a tutti i nodi del cluster di database Aurora contemporaneamente.

> **Note**
>
> Questa versione non è attualmente disponibile nelle regioni AWS GovCloud (Stati Uniti occidentali) [us-gov-west-1] e Cina (Pechino) [cn-north-1]. Non appena disponibile verrà comunicato l'annuncio.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

- Risolto un problema di prestazioni che aumentava l'utilizzo della CPU su una replica di Aurora dopo un riavvio.
- È stato risolto un problema di stabilità per le query `SELECT` che utilizzano l'hash join.

Integrazione delle correzioni di bug della community di MySQL

- BUG #13418638: CREATE TABLE IF NOT EXISTS - IL BLOCCO DEI METADATI È TROPPO RESTRITTIVO

Aggiornamenti del motore di database Aurora MySQL 08/10/2018 (versione 1.17.7) (obsoleta)

Versione: 1.17.7

Aurora MySQL 1.17.7 è disponibile a livello generale. Tutti i nuovi cluster di database Aurora MySQL compatibili con MySQL 5.6, compresi quelli ripristinati da snapshot, verranno creati in Aurora MySQL; versione 1.17.7. È possibile, ma non necessario, eseguire l'aggiornamento dei cluster di database esistenti a Aurora MySQL versione 1.17.7. Per usare una versione più vecchia puoi creare nuovi cluster di database in Aurora MySQL 1.14.4, 1.15.1, 1.16 o 1.17.6. Puoi farlo utilizzando AWS CLI o l'API Amazon RDS e specificando la versione del motore.

Con la versione 1.17.7 di Aurora MySQL, viene utilizzato un modello di patch del cluster che consente di applicare le patch a tutti i nodi del cluster di database Aurora contemporaneamente.

> **Note**
>
> Questa versione non è attualmente disponibile nelle regioni AWS GovCloud (Stati Uniti occidentali) [us-gov-west-1] e Cina (Pechino) [cn-north-1]. Non appena disponibile verrà comunicato l'annuncio.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

- La variabile di stato InnoDB `innodb_buffer_pool_size` è stata resa visibile pubblicamente per la modifica da parte dei clienti.
- È stato risolto un problema di stabilità sul cluster Aurora che si verificava durante i failover.
- Disponibilità del cluster migliorata grazie alla risoluzione di un problema di ripristino DDL che si verificava dopo un'operazione `TRUNCATE` non riuscita.
- È stato risolto un problema di stabilità relativo all'aggiornamento della tabella `mysql.innodb_table_stats`, innescato dalle operazioni DDL.
- Sono stati risolti i problemi di stabilità delle repliche di Aurora attivate durante l'invalidazione della cache della query dopo un'operazione DDL.
- È stato risolto un problema di stabilità attivato dall'accesso alla memoria non valida durante l'espulsione periodica della cache del dizionario in background.

Integrazione delle correzioni di bug della community di MySQL

- Bug 16208542: drop index su una colonna chiave esterna porta a una tabella mancante.
- Bug 76349: perdita di memoria in add_derived_key().
- Bug 16862316: per le tabelle partizionate le query potrebbero restituire risultati differenti a seconda del fatto che l'unione dell'indice sia stato usato o meno.
- Bug 17588348: le query che usano l'ottimizzazione dell'unione dell'indice (consulta Ottimizzazione dell'unione dell'indice) possono restituire risultati non validi quando eseguite su tabelle partizionate da HASH.

Aggiornamenti del motore di database Aurora MySQL 06/09/2018 (versione 1.17.6) (obsoleta)

Versione: 1.17.6

Aurora MySQL 1.17.6 è disponibile a livello generale. Tutti i nuovi cluster di database Aurora MySQL compatibili con MySQL 5.6, compresi quelli ripristinati da snapshot, verranno creati in Aurora MySQL; versione 1.17.6. È possibile, ma non necessario, eseguire l'aggiornamento dei cluster di database esistenti a Aurora MySQL versione 1.17.6. Per usare una versione più vecchia puoi creare nuovi cluster di database in Aurora MySQL 1.14.4, 1.15.1, 1.16 o 1.17.5. Puoi farlo utilizzando AWS CLI o l'API Amazon RDS e specificando la versione del motore.

Con la versione 1.17.6 di Aurora MySQL, viene utilizzato un modello di patch del cluster che consente di applicare le patch a tutti i nodi del cluster di database Aurora contemporaneamente.

> **Note**
>
> Questa versione non è attualmente disponibile nelle regioni AWS GovCloud (Stati Uniti occidentali) [us-gov-west-1] e Cina (Pechino) [cn-north-1]. Non appena disponibile verrà comunicato l'annuncio.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

- È stato risolto un problema di stabilità relativo ad Aurora Reader per query `SELECT` mentre Aurora Writer esegue operazioni DDL sulla stessa tabella.
- È stato risolto un problema di stabilità causato dalla creazione e dall'eliminazione dei log DDL per le tabelle temporanee che utilizzano Heap/Memory il motore.
- È stato risolto un problema di stabilità sullo slave Binlog durante la replica di istruzioni DDL con connessione al master Binlog instabile.
- È stato risolto un problema di stabilità verificatosi durante la scrittura al log delle query lente.
- È stato risolto un problema con la tabella dello stato della replica che mostrava informazioni di ritardo di Aurora Reader errati.

Integrazione delle correzioni di bug della community di MySQL

- Per un'istruzione ALTER TABLE che ha rinominato o modificato il valore predefinito di una colonna BINARY, la modifica è stata fatta utilizzando una copia della tabella e non in ordine. (Bug 67141, 14735373, 69580, 17024290)
- Un outer join tra una tabella regolare e una derivata che è implicitamente gruppi può causare un'uscita dal server. (Bug 16177639)

Aggiornamenti del motore di database Aurora MySQL 14/08/2018 (versione 1.17.5) (obsoleta)

Versione: 1.17.5

Aurora MySQL 1.17.5 è disponibile a livello generale. Tutti i nuovi cluster di database Aurora MySQL compatibili con MySQL 5.6, compresi quelli ripristinati da snapshot, verranno creati in Aurora MySQL; versione 1.17.5. È possibile, ma non necessario, eseguire l'aggiornamento dei cluster di database esistenti a Aurora MySQL versione 1.17.5. Per usare una versione più vecchia puoi creare nuovi cluster di database in Aurora MySQL 1.14.4, 1.15.1, 1.16 o 1.17.4. Puoi farlo utilizzando AWS CLI o l'API Amazon RDS e specificando la versione del motore.

Con la versione 1.17.5 di Aurora MySQL, viene utilizzato un modello di patch del cluster che consente di applicare le patch a tutti i nodi del cluster di database Aurora contemporaneamente.

> **Note**
>
> Questa versione non è attualmente disponibile nelle regioni AWS GovCloud (Stati Uniti occidentali) [us-gov-west-1] e Cina (Pechino) [cn-north-1]. Non appena disponibile verrà comunicato l'annuncio.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

- È stato risolto un problema a causa del quale un Aurora Writer veniva riavviato dopo l'applicazione di patch a un cluster Aurora utilizzando la funzionalità per l'applicazione di patch senza tempi di inattività.

Aggiornamenti del motore di database Aurora MySQL 07/08/2018 (versione 1.17.4) (obsoleta)

Versione: 1.17.4

Aurora MySQL 1.17.4 è disponibile a livello generale. Tutti i nuovi cluster di database Aurora MySQL compatibili con MySQL 5.6, compresi quelli ripristinati da snapshot, verranno creati in Aurora MySQL; versione 1.17.4. È possibile, ma non necessario, eseguire l'aggiornamento dei cluster di database esistenti a Aurora MySQL versione 1.17.4. Per usare una versione più vecchia puoi creare nuovi cluster di database in Aurora MySQL 1.14.4, 1.15.1, 1.16 o 1.17.3. Puoi farlo utilizzando AWS CLI o l'API Amazon RDS e specificando la versione del motore.

Con la versione 1.17.4 di Aurora MySQL, viene utilizzato un modello di patch del cluster che consente di applicare le patch a tutti i nodi del cluster di database Aurora contemporaneamente.

> **Note**
>
> Questa versione non è attualmente disponibile nelle regioni AWS GovCloud (Stati Uniti occidentali) [us-gov-west-1] e Cina (Pechino) [cn-north-1]. Non appena disponibile verrà comunicato l'annuncio.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

- Miglioramenti di replica:
  - Traffico di rete ridotto grazie alla mancata trasmissione di record di binlog alle repliche cluster. Questo perfezionamento viene attivato per impostazione predefinita.
  - Traffico di rete ridotto grazie alla compressione dei messaggi di replica. Questo perfezionamento viene abilitato per impostazione predefinita per classi di istanze 8xlarge e 16xlarge. Le istanze di grandi dimensioni possono supportare un elevato volume di traffico in scrittura che si traduce in traffico di rete sostanziale per i messaggi di replica.
  - Correzioni per la cache della query di replica.
- È stato risolto un problema a causa del quale `ORDER BY LOWER(col_name)` poteva produrre un ordinamento sbagliato durante l'uso della regola di confronto `utf8_bin`.
- È stato risolto un problema a causa del quale i DDL (specialmente le istruzioni `TRUNCATE TABLE`) potevano causare problemi nelle repliche di Aurora, incluse instabilità o tabelle mancanti.
- È stato risolto un problema a causa del quale i socket venivano lasciati in stato semi aperto quando i nodi di storage venivano riavviati.
- Sono disponibili i seguenti nuovi parametri per cluster di database:
  - `aurora_enable_zdr`–: consente connessioni aperte su una replica Aurora per rimanere attivi al riavvio di replica.
  - `aurora_enable_replica_log_compression`–: abilita la compressione dei payload di replica per ottimizzare l'uso della larghezza di banda della rete tra il master e le repliche Aurora.
  - `aurora_enable_repl_bin_log_filtering`–: abilita il filtraggio dei record delle repliche che non possono essere utilizzati dalle repliche Aurora sul master.

Aggiornamenti del motore di database Aurora MySQL 05/06/2018 (versione 1.17.3) (obsoleta)

Versione: 1.17.3

Aurora MySQL 1.17.3 è disponibile a livello generale. Tutti i nuovi cluster di database Aurora MySQL compatibili con MySQL 5.6, compresi quelli ripristinati da snapshot, verranno creati in Aurora MySQL; versione 1.17.3. È possibile, ma non necessario, eseguire l'aggiornamento dei cluster di database esistenti a Aurora MySQL versione 1.17.3. È possibile creare nuovi cluster database in Aurora MySQL 1.14.4, Aurora MySQL 1.15.1, Aurora MySQL 1.16. Puoi farlo utilizzando AWS CLI o l'API Amazon RDS e specificando la versione del motore.

Con la versione 1.17.3 di Aurora MySQL, viene utilizzato un modello di patch del cluster che consente di applicare le patch a tutti i nodi del cluster di database Aurora contemporaneamente.

> **Note**
>
> Questa versione non è attualmente disponibile nelle regioni AWS GovCloud (Stati Uniti occidentali) [us-gov-west-1] e Cina (Pechino) [cn-north-1]. Non appena disponibile verrà comunicato l'annuncio.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

- È stato risolto un problema a causa del quale una replica Aurora si riavviava quando si usavano ripristini di cursore ottimistici durante la lettura dei record.
- È stato risolto un problema a causa del quale un Aurora Writer si riavviava quando si cercava di terminare una sessione MySQL (kill "«*<session id>*) con lo schema delle prestazioni abilitato.
- È stato risolto un problema a causa del quale Aurora Writer si riavviava durante l'elaborazione di una soglia per la garbage collection.
- È stato risolto un problema a causa del quale Aurora Writer si riavviava occasionalmente quando teneva traccia dell'avanzamento di una replica Aurora nell'applicazione log.
- È stato risolto un problema con la cache delle query quando il commit automatico era disattivato e questo poteva potenzialmente causare letture non aggiornate.

Aggiornamenti del motore di database Aurora MySQL 27/04/2018 (versione 1.17.2) (obsoleta)

Versione: 1.17.2

Aurora MySQL 1.17.2 è disponibile a livello generale. Tutti i nuovi cluster di database Aurora MySQL compatibili con MySQL 5.6, compresi quelli ripristinati da snapshot, verranno creati in Aurora MySQL; versione 1.17.2. È possibile, ma non necessario, eseguire l'aggiornamento dei cluster di database esistenti a Aurora MySQL versione 1.17.2. È possibile creare nuovi cluster database in Aurora MySQL 1.14.4, Aurora MySQL 1.15.1, Aurora MySQL 1.16. Puoi farlo utilizzando AWS CLI o l'API Amazon RDS e specificando la versione del motore.

Con la versione 1.17.2 di Aurora MySQL, viene utilizzato un modello di patch del cluster che consente di applicare le patch a tutti i nodi del cluster di database Aurora contemporaneamente.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

- È stato risolto un problema che provocava il riavvio nel corso di alcune operazioni di partizione DDL.
- È stato risolto un problema che causava la disabilitazione del supporto per l'invocazione di AWS Lambda funzioni tramite le funzioni native di Aurora MySQL.
- È stato risolto un problema di invalidazione della cache che provocava il riavvio nelle repliche di Aurora.
- È stato risolto un problema in gestione blocchi che provocava riavvii.

Aggiornamenti del motore di database Aurora MySQL 23/03/2018 (versione 1.17.1) (obsoleta)

Versione: 1.17.1

Aurora MySQL 1.17.1 è disponibile a livello generale. Tutti i nuovi cluster di database, compresi quelli ripristinati da snapshot, verranno creati in Aurora MySQL versione 1.17.1. È possibile, ma non

necessario, eseguire l'aggiornamento dei cluster di database esistenti a Aurora MySQL versione 1.17.1. È possibile creare nuovi cluster DB in Aurora MySQL 1.15.1, Aurora MySQL 1.16 o Aurora MySQL 1.17. Puoi farlo utilizzando AWS CLI o l'API Amazon RDS e specificando la versione del motore.

Con la versione 1.17.1 di Aurora MySQL, viene utilizzato un modello di patch del cluster che consente di applicare le patch a tutti i nodi del cluster di database Aurora contemporaneamente. In questa versione vengono risolti alcuni problemi relativi al motore e alle regressioni.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> È presente un problema nella versione più recente del motore di Aurora MySQL. Dopo aver eseguito l'aggiornamento alla versione 1.17.1, la versione del motore viene indicata erroneamente come `1.17`. Se si è eseguito l'aggiornamento alla versione 1.17.1, si può verificare l'aggiornamento nella colonna Maintenance (Manutenzione) del cluster di database in Console di gestione AWS. Se appare `none`, è stato eseguito l'aggiornamento del motore alla versione 1.17.1.

Miglioramenti

- È stato risolto un problema relativo al ripristino dei log binari che aveva come conseguenza tempi di ripristino più lunghi nel caso di file degli indici dei log binari di grandi dimensioni che si potevano verificare nel caso di rotazioni frequenti dei log binari.
- È stato risolto un problema relativo all'ottimizzatore di query che generava un piano delle query inefficiente per le tabelle partizionate.
- È stato risolto un problema relativo all'ottimizzatore di query a causa del quale una query in un intervallo provocava un riavvio del motore di database.

Aggiornamenti del motore di database Aurora MySQL 13/03/2018 (versione 1.17) (obsoleta)

Versione: 1.17

Aurora MySQL 1.17 è disponibile a livello generale. Le versioni Aurora MySQL 1.x sono compatibili solo con MySQL 5.6, e non con MySQL 5.7. Tutti i nuovi cluster di database compatibili con la versione 5.6, compresi quelli ripristinati da snapshot, verranno creati in Aurora versione 1.17. È possibile, ma non necessario, eseguire l'aggiornamento dei cluster di database esistenti a Aurora versione 1.17. È possibile creare nuovi cluster DB in Aurora 1.14.1, Aurora 1.15.1 o Aurora 1.16. Puoi farlo utilizzando la AWS CLI o l'API Amazon RDS e specificando la versione del motore.

Con la versione 1.17 di Aurora, viene utilizzato un modello di patch del cluster che consente di applicare le patch a tutti i nodi del cluster di database Aurora contemporaneamente. Supportiamo l'applicazione di patch senza tempi di inattività, che funziona sulla base del miglior tentativo per preservare le connessioni client attraverso il processo di applicazione di patch. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support.

Applicazione di patch senza tempi di inattività

I tentativi dell'applicazione di patch senza tempi di inattività funzionano sulla base del miglior tentativo per preservare le connessioni client attraverso le patch del motore. Per ulteriori informazioni su ZDP, consulta Applicazione di patch senza tempi di inattività (ZDP) nella Guida per l'utente di Amazon Aurora.

Nuove funzionalità

- Ora Aurora MySQL supporta la compressione dei blocchi, che ottimizza l'utilizzo di memoria consumata dalla gestione dei blocchi. A partire dalla versione 1.17, puoi utilizzare questa funzionalità senza dover abilitare la modalità di laboratorio.

Miglioramenti

- È stato risolto un problema verificatosi prevalentemente nelle istanze con meno core, in cui un singolo core utilizzava la CPU al 100% anche se il database era inattivo.
- Sono state migliorate le prestazioni del recupero dei log binari dai cluster di Aurora.
- È stato risolto un problema a causa del quale le repliche di Aurora cercavano di scrivere le statistiche delle tabelle nello storage persistente con un conseguente arresto anomalo.

- È stato risolto un problema a causa del quale la cache delle query non funzionava come previsto nelle repliche di Aurora.
- È stata risolta una race condition nella gestione dei blocchi che provocava il riavvio del motore.
- È stato risolto un problema a causa del quale i blocchi delle transazioni di sola lettura e di auto-commit provocavano un riavvio del motore.
- È stato risolto un problema a causa del quale alcune query non venivano scritte nei log di audit.
- È stato risolto un problema relativo al ripristino di alcune operazioni di manutenzione delle partizioni su failover.

Integrazione delle correzioni di bug di MySQL.

- Replica incorretta di LAST_INSERT_ID nel caso di utilizzo di filtri per la replica (bug 69861)
- La query restituisce risultati diversi a seconda dell'impostazione INDEX_MERGE (bug 16862316)
- Nuova procedura query in routine archiviata, piano delle query inefficiente (bug 16346367)
- INNODB FTS: Assert in FTS_CACHE_APPEND_DELETED_DOC_IDS (bug 18079671)
- Assert RBT_EMPTY(INDEX_CACHE->WORDS) nella colonna di modifica ALTER TABLE (bug 17536995)
- La ricerca nel testo INNODB non trova record quando si utilizzano punti di salvataggio (bug 70333 e 17458835)

Aggiornamenti del motore di database Aurora MySQL 11/12/2017 (versione 1.16) (obsoleta)

Versione: 1.16

Aurora MySQL 1.16 è disponibile a livello generale. Tutti i nuovi cluster di database, compresi quelli ripristinati da snapshot, verranno creati in Aurora versione 1.1. È possibile, ma non necessario, eseguire l'aggiornamento dei cluster di database esistenti a Aurora versione 1.16. È possibile creare nuovi cluster DB in Aurora 1.14.1 o Aurora 1.15.1. Puoi farlo utilizzando AWS CLI o l'API Amazon RDS e specificando la versione del motore.

Con la versione 1.16 di Aurora, viene utilizzato un modello di patch del cluster che consente di applicare le patch a tutti i nodi del cluster di database Aurora contemporaneamente. È consentita

l'applicazione di patch senza tempi di inattività, che funziona sulla base del miglior tentativo per preservare le connessioni client attraverso il processo di applicazione di patch. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support.

Applicazione di patch senza tempi di inattività

I tentativi dell'applicazione di patch senza tempi di inattività funzionano sulla base del miglior tentativo per preservare le connessioni client attraverso le patch del motore. Per ulteriori informazioni su ZDP, consulta Applicazione di patch senza tempi di inattività (ZDP) nella Guida per l'utente di Amazon Aurora.

Nuove funzionalità

- Aurora MySQL ora supporta le chiamate sincrone AWS Lambda tramite la funzione nativa. `lambda_sync()` Inoltre, è disponibile la funzione nativa `lambda_async()`, che può essere utilizzata come alternativa alla procedura di archiviazione esistente per l'invocazione Lambda asincrona. Per ulteriori informazioni, consulta Chiamare una funzione Lambda da un cluster DB Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Aurora MySQL ora supporta hash join per velocizzare le query equijoin. L'ottimizzatore basato sui costi di Aurora è in grado di decidere automaticamente quando utilizzare gli hash join, ma puoi anche scegliere di imporne l'utilizzo in un piano di query. Per ulteriori informazioni, consulta Ottimizzazione di grandi query di join Aurora MySQL con hash join nella Guida per l'utente di Amazon Aurora.
- Ora Aurora MySQL supporta il raggruppamento delle scansioni per velocizzare in modo significativo le query in memoria e orientate alla scansione. L'elaborazione in batch, inoltre, migliora le prestazioni delle scansioni delle tabelle intere, degli indici interi e degli intervalli di indici.

Miglioramenti

- È stato risolto un problema a causa del quale si verificava un arresto anomalo delle repliche di lettura durante l'esecuzione di query nelle tabelle eliminate nel master.
- È stato risolto un problema a causa del quale il riavvio del writer in un cluster del database con un gran numero di indici `FULLTEXT` richiedeva un ripristino più lungo del previsto.

- È stato risolto un problema a causa del quale i log binari di svuotamento provocavano incidenti `LOST_EVENTS` negli eventi binlog.
- Sono stati risolti problemi di stabilità del pianificatore che si verificavano con l'attivazione dello schema di prestazioni.
- È stato risolto un problema a causa del quale una subquery che utilizzava tabelle temporanee restituiva risultati parziali.

Integrazione delle correzioni di bug di MySQL.

Nessuno

Aggiornamenti del motore di database Aurora MySQL 20/11/2017 (versione 1.15.1) (obsoleta)

Versione: 1.15.1

Aurora MySQL 1.15.1 è disponibile a livello generale. Tutti i nuovi cluster di database, compresi quelli ripristinati da snapshot, verranno creati in Aurora versione 1.15.1. È possibile, ma non necessario, eseguire l'aggiornamento dei cluster di database esistenti a Aurora versione 1.15.1. È possibile creare nuovi cluster DB in Aurora 1.14.1. Puoi farlo utilizzando AWS CLI o l'API Amazon RDS e specificando la versione del motore.

Con la versione 1.15.1 di Aurora, viene utilizzato un modello di patch del cluster che consente di applicare le patch a tutti i nodi del cluster di database Aurora contemporaneamente. È consentita l'applicazione di patch senza tempi di inattività, che funziona sulla base del miglior tentativo per preservare le connessioni client attraverso il processo di applicazione di patch. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Applicazione di patch senza tempi di inattività

I tentativi dell'applicazione di patch senza tempi di inattività funzionano sulla base del miglior tentativo per preservare le connessioni client attraverso le patch del motore. Per ulteriori informazioni su ZDP,

consulta Applicazione di patch senza tempi di inattività (ZDP) nella Guida per l'utente di Amazon Aurora.

Miglioramenti

- È stato risolto un problema relativo al selettore di segmenti adattivo con le richieste di lettura a causa del quale veniva scelto lo stesso segmento due volte con un conseguente picco nella latenza di lettura in determinate condizioni.
- È stato risolto un problema provocato da un'ottimizzazione in Aurora MySQL per il pianificatore di thread. Il problema si manifestava negli errori spuri durante la scrittura nel log lento, con un'esecuzione corretta delle query associate.
- È stato risolto un errore di stabilità delle repliche di lettura su volumi elevati (più di 5 TB).
- È stato risolto un errore a causa del quale il numero di thread lavoratori aumentava continuamente dovuto a un conteggio di connessioni in corso falso.
- È stato risolto un problema a causa del quale i blocchi delle tabelle provocavano lunghe attese semaforo durante gli inserimenti dei carichi di lavoro.
- Sono state ripristinate le seguenti correzioni dei bug di MySQL comprese in Aurora MySQL versione 1.15:
  - Stallo delle istanze MySQL durante l'indicizzazione SYNC (bug 73816)
  - Assert RBT_EMPTY(INDEX_CACHE->WORDS) nella colonna di modifica ALTER TABLE (bug 17536995)
  - La ricerca nel testo InnoDB non trova record quando si utilizzano punti di salvataggio (bug 70333)

Integrazione delle correzioni di bug di MySQL.

Nessuno

Aggiornamenti del motore di database Aurora MySQL 24/10/2017 (versione 1.15) (obsoleta)

Versione: 1.15

Aurora MySQL 1.15 è disponibile a livello generale. Tutti i nuovi cluster di database, compresi quelli ripristinati da snapshot, verranno creati in Aurora versione 1.15. È possibile, ma non necessario,

eseguire l'aggiornamento dei cluster di database esistenti a Aurora versione 1.15. È possibile creare nuovi cluster DB in Aurora 1.14.1. Puoi farlo utilizzando AWS CLI o l'API Amazon RDS e specificando la versione del motore.

Con la versione 1.15 di Aurora, viene utilizzato un modello di patch del cluster che consente di applicare le patch a tutti i nodi del cluster di database Aurora contemporaneamente. Gli aggiornamenti richiedono un riavvio del database, pertanto ci sarà un tempo di inattività di 20-30 secondi, al termine del quale si potranno ricominciare a usare i cluster del DB. Se i cluster di database attualmente eseguono Aurora 1.14 o Aurora versione 1.14.1, l'applicazione di patch senza tempi di inattività in Aurora MySQL consente la conservazione delle connessioni client all'istanza primaria di Aurora MySQL durante l'aggiornamento, a seconda del carico di lavoro.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Applicazione di patch senza tempi di inattività

I tentativi dell'applicazione di patch senza tempi di inattività funzionano sulla base del miglior tentativo per preservare le connessioni client attraverso le patch del motore. Per ulteriori informazioni su ZDP, consulta Applicazione di patch senza tempi di inattività (ZDP) nella Guida per l'utente di Amazon Aurora.

Nuove funzionalità

- Prefetch asincrono delle chiavi:– il prefetch asincrono delle chiavi (Asynchronous key prefetch AKP) ha come obiettivo il miglioramento delle prestazioni dei join degli indici non memorizzati nella cache, attraverso il prefetch delle chiavi in memoria prima che diventino necessarie. Il caso d'uso principale di questa funzione è un join degli indici tra una tabella interna di grandi dimensioni e una esterna di dimensioni inferiori, in cui l'indice è altamente selettivo nella tabella più grande. Inoltre, quando si attiva l'interfaccia MRR (Multi-Range Read), il prefetch asincrono delle chiavi verrà utilizzato per una ricerca nell'indice da secondario a primario. In alcuni casi, le istanze più piccole con limiti di memoria potrebbero utilizzare il prefetch asincrono delle chiavi, vista la cardinalità corretta delle chiavi. Per ulteriori informazioni, consulta Ottimizzazione delle query di join indicizzate Aurora MySQL con prefetch asincrono delle chiavi nella Guida per l'utente di Amazon Aurora.
- Fast DDL:– la funzione resa disponibile con Aurora versione 1.1 è stata estesa alle operazioni che comprendono valori predefiniti. Pertanto, ora Fast DDL è applicabile alle operazioni che

consentono di aggiungere una colonna nullable alla fine di una tabella, con o senza valori predefiniti. Questa funzione resta attiva nella modalità di laboratorio di Aurora. Per ulteriori informazioni, consulta Alterazione delle tabelle in Amazon Aurora mediante Fast DDL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

- È stato corretto un errore di calcolo durante l'ottimizzazione delle interrogazioni WITHIN/CONTAINS spaziali che in precedenza produceva un set di risultati vuoto.
- È stato corretto il comando `SHOW VARIABLE` in modo che venga mostrato il valore del parametro `innodb_buffer_pool_size` aggiornato ogni volta che viene modificato nel gruppo di parametri.
- È stata migliorata la stabilità dell'istanza primaria durante l'inserimento in blocco in una tabella modificata con Fast DDL quando l'indicizzazione dell'hash adattiva è disattivata e il record da inserire è il primo di una pagina.
- È stata migliorata la stabilità di Aurora quando l'utente cerca di impostare il valore del parametro del cluster di database server_audit_events su **`default`**.
- È stato risolto un problema a causa del quale una modifica del set di caratteri del database per un'istruzione ALTER TABLE eseguita nell'istanza primaria di Aurora veniva replicata nelle repliche di Aurora solo dopo il riavvio.
- È stata migliorata la stabilità attraverso la correzione di una race condition nell'istanza primaria che precedentemente consentiva la registrazione di una replica di Aurora anche se l'istanza primaria aveva chiuso il proprio volume.
- Sono state migliorate le prestazioni dell'istanza primaria durante la creazione dell'indice in una tabella di grandi dimensioni attraverso la modifica del protocollo di blocco per attivare le istruzioni DML (Data Manipulation Language) simultanee durante la creazione dell'indice.
- È stata risolta un'incoerenza tra i metadati InnoDB durante la query ALTER TABLE RENAME che migliora la stabilità. Esempio: quando le colonne della tabella t1(c1, c2) vengono rinominate in ogni ciclo in t1(c2,c3) nell'ambito della stessa istruzione ALTER.
- È stata migliorata la stabilità delle repliche di Aurora per i casi in cui una replica di Aurora non dispone di un carico di lavoro attivo e l'istanza primaria non risponde.
- È stata migliorata la disponibilità delle repliche di Aurora per i casi in cui la replica di Aurora presenta un blocco esplicito su una tabella e impedisce al thread di replica di applicare le modifiche DDL ricevute dall'istanza primaria.

- È stata migliorata la stabilità dell'istanza primaria quando una colonna e una chiave esterna vengono aggiunte alla tabella durante l'esecuzione in simultanea di due sessioni distinte e con Fast DDL attivato.
- È stata migliorata la stabilità del thread di eliminazione dell'istanza primaria durante un carico di lavoro di scrittura elevata attraverso il blocco del troncamento dei record di annullamento fino alla loro eliminazione.
- È stata migliorata la stabilità attraverso la correzione dell'ordine di rilascio del blocco durante il commit delle transazioni che eliminano le tabelle.
- È stato corretto un difetto delle repliche di Aurora a causa del quale l'istanza database non poteva completare l'avvio e segnalava l'uso della porta 3306.
- È stata corretta la race condition a causa della quale la query SELECT eseguita in determinate tabelle di schemi di informazioni (innodb_trx, innodb_lock, innodb_lock_waits) aumentava l'instabilità del cluster.

Integrazione delle correzioni di bug di MySQL.

- CREATE USER accetta un hash di password e plugin, ma ignora l'hash di password (bug 78033)
- Il motore per il partizionamento aggiunge campi al bit letto impostato per restituire dati ordinati da un indice partizionato. Ne consegue che il buffer del join prova a leggere i campi non necessari. Il problema è stato corretto evitando di aggiungere tutti i campi partizionati a read_set e ordinando solo i campi prefisso già impostati in read_set. È stato aggiungo DBUG_ASSERT che nel caso di utilizzo di key_cmp, consente di leggere almeno il primo campo (bug 16367691).
- Stallo delle istanze MySQL durante l'indicizzazione SYNC (bug 73816)
- Assert RBT_EMPTY(INDEX_CACHE->WORDS) nella colonna di modifica ALTER TABLE (bug 17536995)
- La ricerca nel testo InnoDB non trova record quando si utilizzano punti di salvataggio (bug 70333)

Aggiornamenti del motore di database Aurora MySQL 13/03/2018 (versione 1.14.4) (obsoleta)

Versione: 1.14.4

Aurora MySQL 1.14.4 è disponibile a livello generale. Puoi creare nuovi cluster DB in Aurora 1.14.4, utilizzando la AWS CLI o l'API Amazon RDS e specificando la versione del motore. È possibile, se

lo si desidera, eseguire l'aggiornamento ai cluster del database 1.14.x esistenti in Aurora versione 1.14.4.

Con la versione 1.14.4 di Aurora, viene utilizzato un modello di patch del cluster che consente di applicare le patch a tutti i nodi del cluster di database Aurora contemporaneamente. Supportiamo l'applicazione di patch senza tempi di inattività, che funziona sulla base del miglior tentativo per preservare le connessioni client attraverso il processo di applicazione di patch. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Applicazione di patch senza tempi di inattività

I tentativi dell'applicazione di patch senza tempi di inattività funzionano sulla base del miglior tentativo per preservare le connessioni client attraverso le patch del motore. Per ulteriori informazioni su ZDP, consulta Applicazione di patch senza tempi di inattività (ZDP) nella Guida per l'utente di Amazon Aurora.

Nuove funzionalità

- Aurora MySQL ora supporta le classi di istanze db.r4.

Miglioramenti

- È stato risolto un problema a causa del quale venivano generati `LOST_EVENTS` durante la scrittura di eventi binlog di grandi dimensioni.

Integrazione delle correzioni di bug di MySQL.

- Gli eventi ignorabili non funzionano e non vengono sottoposti a test (bug 74683)
- NUOVO->VECCHIO ERRORE ASSERT 'GTID_MODE > 0' (bug 20436436)

Aggiornamenti del motore di database Aurora MySQL 22/09/2017 (versione 1.14.1) (obsoleta)

Versione: 1.14.1

Aurora MySQL 1.14.1 è disponibile a livello generale. Tutti i nuovi cluster di database, compresi quelli ripristinati da snapshot, verranno creati in Aurora MySQL versione 1.14.1. Inoltre, Aurora MySQL versione 1.14.1 è un aggiornamento obbligatorio per i cluster del database di Aurora MySQL esistenti. Per ulteriori informazioni, consulta Annuncio: estensione del programma di aggiornamento obbligatorio per Amazon Aurora sul sito AWS Web dei forum per sviluppatori.

Con la versione 1.14.1 di Aurora MySQL, viene utilizzato un modello di patch del cluster che consente di applicare le patch a tutti i nodi del cluster di database Aurora MySQL contemporaneamente. Gli aggiornamenti richiedono un riavvio del database, pertanto ci sarà un tempo di inattività di 20-30 secondi, al termine del quale si potranno ricominciare a usare i cluster del DB. Se i cluster di database eseguono la versione 1.13 o una versione successiva, l'applicazione di patch senza tempi di inattività di Aurora MySQL; consente la conservazione delle connessioni client all'istanza primaria di Aurora MySQL; durante l'aggiornamento, a seconda del carico di lavoro.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support.

Miglioramenti

- Sono state corrette le race condition associate agli inserimenti e alle eliminazioni per migliorare la stabilità della funzione Fast DLL, che resta disponibile nella modalità di laboratorio di Aurora MySQL.

Aggiornamenti del motore di database Aurora MySQL 07/08/2017 (versione 1.14) (obsoleta)

Versione: 1.14

Aurora MySQL 1.14 è disponibile a livello generale. Tutti i nuovi cluster di database, compresi quelli ripristinati da snapshot, verranno creati in Aurora MySQL versione 1.14. Inoltre, Aurora MySQL versione 1.14 è un aggiornamento obbligatorio per i cluster del database di Aurora MySQL esistenti.

Verrà inviato un annuncio distinto con le tempistiche per l'eliminazione delle versioni precedenti di Aurora MySQL.

Con la versione 1.14 di Aurora MySQL, viene utilizzato un modello di patch del cluster che consente di applicare le patch a tutti i nodi del cluster di database Aurora contemporaneamente. Gli aggiornamenti richiedono un riavvio del database, pertanto ci sarà un tempo di inattività di 20-30 secondi, al termine del quale si potranno ricominciare a usare i cluster del DB. Se i cluster di database eseguono la versione 1.13, l'applicazione di patch senza tempi di inattività di Aurora consente la conservazione delle connessioni client all'istanza primaria di Aurora durante l'aggiornamento, a seconda del carico di lavoro.

In caso di domande o dubbi, l' AWS assistenza è disponibile nei forum della community e tramite AWS Support.

Applicazione di patch senza tempi di inattività

I tentativi dell'applicazione di patch senza tempi di inattività funzionano sulla base del miglior tentativo per preservare le connessioni client attraverso le patch del motore. Per ulteriori informazioni su ZDP, consulta Applicazione di patch senza tempi di inattività (ZDP) nella Guida per l'utente di Amazon Aurora.

Miglioramenti

- È stato risolto un messaggio "record non trovato" che veniva visualizzato erroneamente quando un record veniva individuato nell'indice secondario ma non in quello primario.
- È stato risolto un problema di stabilità che si verificava a causa di un'asserzione difensiva (aggiunta nella versione 1.12) troppo evidente nel caso di una singola scrittura di più di 32 pagine. Questa situazione poteva verificarsi, ad esempio, con valori BLOB alti.
- È stato risolto un problema di stabilità provocato da incoerenze tra la cache degli spazi tabelle e la cache dei dizionari.
- È stato risolto un problema a causa del quale la replica di Aurora smetteva di rispondere dopo aver superato il numero massimo di tentativi di connessione all'istanza primaria. Ora una replica di Aurora si riavvia se il periodo di inattività è più lungo del periodo di timeout di heartbeat utilizzato per il controllo dello stato dall'istanza primaria.
- È stato risolto un livelock che si verificava in caso di alta simultaneità nel tentativo di una connessione di acquisire un blocco a livello di metadati (MDL) durante l'esecuzione di un comando, ad esempio `ALTER TABLE`.

- È stato risolto un problema di stabilità in una Read Read Replica di Aurora in presenza della funzione di logical/parallel lettura anticipata.
- Miglioramento di `LOAD FROM S3` in due modi:
  1. Migliore gestione degli errori di timeout di Amazon S3 attraverso lo schema di nuovi tentativi SDK oltre allo schema di nuovi tentativi esistente.
  2. Ottimizzazione delle prestazioni durante il caricamento di file di grandi dimensioni o di un numero elevato di file mediante il loro caching e il riutilizzo dello stato del client.
- Sono stati risolti i seguenti problemi di stabilità con Fast DDL per le operazioni `ALTER TABLE`:
  1. Quando l'istruzione `ALTER TABLE` prevede più comandi `ADD COLUMN` e i nomi delle colonne non sono in ordine ascendente.
  2. Quando la stringa del nome della colonna deve essere aggiornata e la stringa del nome corrispondente, prelevata dalla tabella del sistema interna, è diversa a causa di un carattere finale nullo (/0).
  3. Nel caso di alcune operazioni di suddivisione albero B
  4. Quando la tabella dispone di una chiave primaria di lunghezza variabile.
- È stato risolto un problema di stabilità con le repliche di Aurora nei casi in cui richiedeva troppo tempo rendere l'indice FTS coerente con quello dell'istanza primaria. Questo poteva verificarsi se un gran numero delle voci dell'indice FTS creato nell'istanza primaria non era stato incluso nel disco.
- È stato risolto un problema di stabilità che si verificava durante la creazione dell'indice.
- Nuova infrastruttura che tiene traccia del consumo di memoria per connessione e della telemetria associata che verrà utilizzata per sviluppare strategie di prevenzione Out-Of-Memory (OOM).
- È stato risolto un problema a causa del quale `ANALYZE TABLE` veniva consentito erroneamente nelle repliche Aurora. Questo codice è stato bloccato.
- È stato risolto un problema di stabilità provocato da un deadlock raro, conseguenza di una race condition tra la lettura anticipata logica e la rimozione.

Integrazione delle correzioni di bug di MySQL.

- Una ricerca full-text combinata con le tabelle derivate (sottoquery nella clausola `FROM`) ha provocato un'uscita dal server. Ora, se un'operazione full-text dipende da una tabella derivata, il server genera un errore in cui viene indicato che non è possibile eseguire una ricerca full-text in una tabella materializzata. (Bug 68751, 16539903)

Aggiornamenti del motore di database Aurora MySQL 15/05/2017 (versione 1.13) (obsoleta)

Versione: 1.13

> **Note**
>
> Abbiamo attivato una nuova funzione (SELECT INTO OUTFILE S3) nella versione 1.13 di Aurora MySQL dopo la versione iniziale e abbiamo aggiornato le note di rilascio per riflettere la modifica.

Aurora MySQL 1.13 è disponibile a livello generale. Tutti i nuovi cluster di database, compresi quelli ripristinati da snapshot, verranno creati in Aurora MySQL versione 1.1. È possibile, ma non necessario, eseguire l'aggiornamento dei cluster di database esistenti a Aurora MySQL versione 1.13. Con la versione 1.13 di Aurora, viene utilizzato un modello di patch del cluster che consente di applicare le patch a tutti i nodi del cluster di database Aurora contemporaneamente. È consentita l'applicazione di patch senza tempi di inattività, che funziona sulla base del miglior tentativo per preservare le connessioni client attraverso il processo di applicazione di patch. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Applicazione di patch senza tempi di inattività

I tentativi dell'applicazione di patch senza tempi di inattività funzionano sulla base del miglior tentativo per preservare le connessioni client attraverso le patch del motore. Per ulteriori informazioni su ZDP, consulta Applicazione di patch senza tempi di inattività (ZDP) nella Guida per l'utente di Amazon Aurora.

Nuove caratteristiche:

- SELECT INTO OUTFILE S3: –Aurora MySQL ora consente di caricare i risultati di una query in uno o più file in un bucket di Amazon S3. Per ulteriori informazioni, consulta Caricamento dei dati in un cluster DB Amazon Aurora MySQL da file di testo in un bucket Amazon S3 nella Guida per l'utente di Amazon Aurora.

Miglioramenti:

- È stato implementato il troncamento dei file di log in formato CSV all'avvio del motore per impedire tempi di ripristino lunghi. Ora le tabelle `general_log_backup`, `general_log`, `slow_log_backup` e `slow_log` devono essere salvate prima di un riavvio del database per non andare perse.
- È stato risolto un problema a causa del quale una migrazione di un database nominato **`test`** non avrebbe avuto un esito positivo.
- È stata migliorata la stabilità del garbage collector della gestione blocchi riutilizzando i segmenti di blocco corretti.
- È stata migliorata la stabilità della gestione blocchi attraverso la rimozione delle asserzioni non valide durante l'esecuzione dell'algoritmo di rilevamento del deadlock.
- È stata riattivata la replica asincrona ed è stato risolto un problema associato che provocavano una segnalazione di ritardo di replica errato in caso di assenza di carico o di carico di lavoro di sola lettura. I miglioramenti della pipeline di replica che sono stati introdotti nella versione 1.10. Questi miglioramenti sono stati introdotti per applicare gli aggiornamenti del flusso di log alla cache del buffer di una replica di Aurora. Questo aiuta a migliorare le prestazioni di lettura e la stabilità nelle repliche Aurora.
- È stato risolto un problema a causa del quale il commit automatico=OFF provocava il blocco degli eventi pianificati e il fatto che le transazioni lunghe restassero aperte fino al riavvio del server.
- È stato risolto un problema a causa del quale i log di query generali, di controllo e lente non potevano registrare le query gestite dal commit asincrono.
- Sono state migliorate le prestazioni della caratteristica di lettura anticipata logica di 2,5 volte. A tal fine, i prefetch non sono stati interrotti nelle pagine intermedie di un albero B.
- È stata aggiunta la convalida dei parametri per le variabili di controllo per eliminare gli spazi non necessari.
- È stata risolta una regressione, introdotta nella versione 1.11 di Aurora MySQL a causa della quale le query mostravano risultati errati quando si utilizzava l'opzione SQL_CALC_FOUND_ROWS e si invocava la funzione FOUND_ROWS().
- È stato risolto un problema di stabilità a causa del quale l'elenco di blocco a livello di metadati veniva creato in modo errato.
- È stata migliorata la stabilità con l'impostazione di sql_mode su PAD_CHAR_TO_FULL_LENGTH e con l'esecuzione del comando `SHOW FUNCTION STATUS WHERE Db='`*`string`*`'`.

- È stato risolto un caso raro in cui le istanze non venivano visualizzate dopo l'aggiornamento della versione di Aurora a causa di una verifica di consistenza dei volumi falsa.
- È stato risolto un problema di prestazioni, introdotto con la versione 1.12 di Aurora MySQL, a causa del quale le prestazioni del writer di Aurora risultavano ridotte nel caso di un numero elevato di tabelle degli utenti.
- È stato risolto un problema di stabilità che si verificava quando il writer di Aurora veniva configurato come slave binlog e il numero di connessioni si avvicinava a 16.000.
- È stato risolto un problema raro a causa del quale una replica di Aurora si riavviava in caso di connessione bloccata in attesa di un blocco a livello di metadati durante l'esecuzione di DDL sul master di Aurora.

Integrazione delle correzioni di bug di MySQL.

- Con una tabella InnoDB vuota, non è possibile ridurre il valore dell'incremento automatico utilizzando un'istruzione ALTER TABLE, anche se la tabella è vuota. (Bug 69882)
- Le query MATCH() ... AGAINST che utilizzano una stringa lunga come argomento per AGAINST() possono generare un errore durante l'esecuzione in una tabella InnoDB con un indice di ricerca full-text. (Bug 17640261)
- La gestione di SQL_CALC_FOUND_ROWS insieme a ORDER BY e LIMIT può portare alla visualizzazione di risultati errati per FOUND_ROWS(). (Bug 68458, 16383173)
- ALTER TABLE non consente di modificare il supporto dei valori Null della colonna qualora esista una chiave esterna. (Bug 77591)

Aggiornamenti del motore di database Aurora MySQL 04/05/2017 (versione 1.12) (obsoleta)

Versione: 1.12

Ora Aurora MySQL 1.12 è la versione preferita per la creazione di nuovi cluster di database, inclusi quelli ripristinati da snapshot.

Questo non è un aggiornamento obbligatorio per i cluster esistenti. Sarà possibile aggiornare i cluster esistenti alla versione 1.12 dopo che l'aggiornamento della patch del parco istanze alla versione 1.11 sarà completato (consulta le note di rilascio di Aurora 1.1 e il relativo annuncio nel forum). Con la versione 1.12 di Aurora, viene utilizzato un modello di patch del cluster che consente di applicare le

patch a tutti i nodi del cluster di database Aurora contemporaneamente. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Nuove funzionalità

- Fast DDL: –Aurora MySQL adesso consente l'esecuzione di un'operazione ALTER TABLE tbl_name ADD COLUMN col_name column_definition quasi istantaneamente. L'operazione si conclude senza che vi sia necessità di copiare la tabella e senza alcuna conseguenza materiale sulle altre istruzioni DML. Poiché non consuma storage temporaneo per la copia di una tabella, rende le istruzioni DDL pratiche anche nel caso di tabelle di grandi dimensioni su classi di istanze small. Al momento, l'operazione Fast DDL è supportata solo per l'aggiunta di una colonna nullable, senza un valore predefinito, alla fine della tabella. Al momento, questa caratteristica è disponibile nella modalità di laboratorio di Aurora. Per ulteriori informazioni, consulta Alterazione delle tabelle in Amazon Aurora mediante Fast DDL nella Guida per l'utente di Amazon Aurora.
- Visualizzazione dello stato del volume:– abbiamo aggiunto un nuovo comando per il monitoraggio, SHOW VOLUME STATUS, per visualizzare il numero di nodi e di dischi di un volume. Per ulteriori informazioni, consulta Visualizzazione dello stato del volume per un cluster DB Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

- Sono state implementate modifiche per bloccare la compressione ed evitare che venga ridotta ulteriormente la memoria per l'oggetto di blocco. Questo miglioramento è disponibile nella modalità di laboratorio.
- È stato risolto un problema a causa del quale il parametro `trx_active_transactions` decrementa rapidamente anche se il database è inattivo.
- È stato corretto un messaggio di errore non valido riguardante la sintassi della query fault injection durante la simulazione di errori relativi a dischi e nodi.
- Sono stati risolti vari errori relativi alle race condition e ai latch non attivi nella gestione dei blocchi.
- È stato risolto un errore che provocava un overflow del buffer nell'ottimizzatore di query.
- È stato risolto un problema di stabilità delle repliche di lettura di Aurora quando la memoria disponibile per i nodi di storage sottostanti è scarsa.
- È stato risolto un problema a causa del quale le connessioni inattive non venivano interrotte superato il tempo indicato nell'impostazione del parametro `wait_timeout`.

- È stato risolto un problema a causa del quale `query_cache_size` restituiva un valore inaspettato dopo il riavvio dell'istanza.
- È stato risolto un problema di prestazioni a causa del quale un thread di diagnostica sondava la rete con frequenza per verificare che le scritture fossero ricevute dallo storage.

Integrazione delle correzioni di bug di MySQL.

- Ricaricare una tabella rimossa mentre era vuota ha provocato il ripristino del valore di AUTO_INCREMENT. (Bug 21454472, 77743)
- Il record di un indice non è stato trovato durante il rollback a causa di incoerenze nella struttura purge_node_t. L'incoerenza ha generato avvisi e messaggi di errori quali "error in sec index entry update" (errore relativo all'aggiornamento delle voci dell'indice secondario), "unable to purge a record" (impossibile rimuovere un record) e "tried to purge sec index entry not marked for deletion" (tentativo di rimuovere una voce dell'indice secondario non contrassegnata per l'eliminazione). (Bug 19138298, 70214, 21126772, 21065746)
- Il calcolo errato della dimensione dello stack per l'operazione qsort ha creato un overflow dello stack. (Bug 73979)
- Il record non è stato trovato in un indice durante il rollback. (Bug 70214, 72419)
- Quando si aggiunge la colonna TIMESTAMP con ALTER TABLE, durante l'aggiornamento di CURRENT_TIMESTAMP non vengono inseriti dati (bug 17392)

Aggiornamenti del motore di database Aurora MySQL 23/02/2017 (versione 1.11) (obsoleta)

Versione: 1.11

A tutti i cluster di database Aurora MySQL verrà applicata la patch con la versione più recente poco dopo il rilascio della versione. Ai cluster del database vengono applicate patch utilizzando la procedura legacy con un tempo di inattività di 5-30 secondi.

L'applicazione delle patch avviene durante la finestra di manutenzione del sistema specificata per ogni istanza del database. È possibile visualizzare o modificare la finestra utilizzando Console di gestione AWS. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

In alternativa, è possibile applicare immediatamente la patch scegliendo un cluster DB, Console di gestione AWS scegliendo Cluster Actions e quindi scegliendo Aggiorna ora.

Con la versione 1.11 di Aurora MySQL, viene utilizzato un modello di patch del cluster che consente di applicare le patch a tutti i nodi del cluster di database Aurora contemporaneamente.

Nuove caratteristiche

- Opzione MANIFEST per LOAD DATA FROM S3:– il comando LOAD DATA FROM S3 è stato introdotto nella versione 1.8. Le opzioni per questo comando sono state ampliate e ora è possibile specificare un elenco di file che può essere caricato in un cluster di database di Aurora da Amazon S3 utilizzando un file manifest. Questo facilita il caricamento dei dati da file specifici in una o più sedi, rispetto al caricamento dei dati da un singolo file con l'opzione FILE o il caricamento di dati di più file con la stessa sede e lo stesso prefisso usando l'opzione PREFIX. Il formato del file manifest è uguale a quello usato da Amazon Redshift. Per ulteriori informazioni sull'uso di LOAD DATA FROM S3 con l'opzione MANIFEST, consulta Utilizzo di un manifesto per specificare i file di dati da caricare nella Guida per l'utente di Amazon Aurora.
- Indicizzazione spaziale attivata per impostazione predefinita: –questa caratteristica è stata resa disponibile in modalità di laboratorio nella versione 1.10 e ora viene attivata per impostazione predefinita. L'indicizzazione spaziale consente di migliorare le prestazioni delle query su set di dati di grandi dimensioni per le query che usano i dati spaziali. Per ulteriori informazioni sull'uso dell'indicizzazione spaziale, consulta Amazon Aurora MySQL e dati spaziali nella Guida per l'utente di Amazon Aurora.
- Modifica delle tempistiche dell'audit avanzato:– questa caratteristica è stata resa disponibile nella versione 1.10.1 per offrire una funzione dalle prestazioni elevate per il controllo dell'attività del database. In questa versione, la precisione dei timestamp dei log di audit è stata cambiata da un secondo a un microsecondo. Un timestamp più accurato consente di capire meglio quando si è verificato un evento di audit. Per ulteriori informazioni sull'audit, consulta Utilizzo dell'audit avanzato con un cluster di database Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

- È stato modificato il parametro `thread_handling` per impedire l'impostazione di un valore diverso da **`multiple-connections-per-thread`**, ovvero l'unico modello supportato con il pool di thread di Aurora.

- È stato risolto un problema provocato dall'impostazione del parametro `buffer_pool_size` o `query_cache_size` in modo da risultare più grande della memoria totale del cluster del database. In questo caso, Aurora imposta il valore predefinito per il parametro modificato, in modo che il cluster del database possa avviarsi senza che si verifichi un arresto anomalo.
- È stato risolto un problema relativo alla cache della query a causa del quale una transazione riceveva risultati di lettura obsoleti se la tabella era stata invalidata in un'altra transazione.
- È stato risolto un problema a causa del quale i file binlog contrassegnati per l'eliminazione venivano rimossi dopo un leggero ritardo e non immediatamente.
- È stato risolto un problema a causa del quale un database creato con il nome **`tmp`** veniva trattato come database di sistema archiviato nello storage temporaneo e non nello storage distribuito di Aurora.
- È stato modificato il comportamento di SHOW TABLES in modo da escludere alcune tabelle di sistema interne. In questo modo si riducono i failover non necessari provocati dal blocco da parte di mysqldump di tutti i file elencati in SHOW TABLES; blocco che impediva di scrivere nella tabella di sistema interna provocando il failover.
- È stato risolto un problema a causa del quale si riavviava erroneamente una replica di Aurora durante la creazione di una tabella temporanea in seguito a una query che invocava una funzione con un argomento corrispondente a una colonna della tabella InnoDB.
- È stato risolto un problema relativo al conflitto del blocco a livello di metadati in un nodo di replica di Aurora che provocava un ritardo della replica di Aurora rispetto al cluster di database primario e un possibile riavvio.
- È stato risolto un problema relativo al latch non attivo nella pipeline di replica dei nodi di lettura, che provocava un ritardo della replica di Aurora e un possibile riavvio.
- È stato risolto un problema relativo ai ritardi eccessivi di una replica di Aurora con i volumi crittografati più grandi di un terabyte (TB).
- È stato migliorato il rilevamento del latch non attivo della replica di Aurora attraverso l'uso di un metodo migliorato per leggere il tempo del sistema.
- È stato risolto un problema a causa del quale una replica di Aurora si riavviava due volte e non una in seguito all'annullamento della registrazione da parte dello scrittore.
- È stato risolto un problema di prestazioni scarse della query nelle repliche di Aurora che si verificava quando le statistiche transitorie provocavano una discrepanza nelle statistiche relative alle colonne dell'indice non uniche.

- È stato risolto un problema a causa del quale si verificava un arresto anomalo di una replica di Aurora quando un'istruzione DDL veniva replicata sulla replica di Aurora nello stesso momento in cui la replica di Aurora sta elaborando una query correlata.
- Sono stati modificati i miglioramenti della pipeline di replica che erano stati introdotti nella versione 1.10: per impostazione predefinita non vengono più attivati, bensì vengono disattivati. Questi miglioramenti sono stati introdotti per applicare gli aggiornamenti del flusso di log alla cache del buffer di una replica di Aurora e sebbene questa caratteristica contribuisca al miglioramento della stabilità e delle prestazioni di lettura nelle repliche di Aurora, aumenta il ritardo delle repliche in determinati carichi di lavoro.
- È stato risolto un problema a causa del quale occorrenze simultanee di un'istruzione DDL continua e una lettura anticipata parallela in attesa all'interno della stessa tabella provocavano un errore di asserzione durante la fase commit della transazione DDL.
- Sono stati migliorati i log di query generali e lente in modo da non andare perse con il riavvio del cluster del database.
- È stato risolto un out-of-memory problema relativo a determinate query di lunga durata riducendo il consumo di memoria nel modulo ACL.
- È stato risolto un problema di riavvio che si verificava quando una tabella disponeva di indici non spaziali, di predicati spaziali nella query e il pianificatore sceglieva di usare un indice non spaziale e trasmetteva erroneamente la condizione spaziale all'indice.
- È stato risolto un problema a causa del quale il cluster DB si riavviava in caso di eliminazione, aggiornamento o eliminazione di oggetti geospaziali molto grandi archiviati esternamente (come). LOBs
- Risolto un problema per il quale la simulazione dei guasti mediante ALTER SYSTEM SIMULATE ... FOR INTERVAL non funziona correttamente.
- È stato risolto un problema di stabilità provocato da un'asserzione non valida o da una invariabile errata nella gestione dei blocchi.
- Sono stati disattivati i seguenti due miglioramenti della ricerca full-text InnoDB che erano stati introdotti nella versione 1.10, poiché generavano problemi di stabilità per alcuni carichi di lavoro complessi:
  - L'aggiornamento della cache solo dopo una richiesta di lettura in una replica di Aurora per migliorare la velocità di replica della cache dell'indice di ricerca full-text.
  - Offload dell'attività di sincronizzazione della cache in un thread separato non appena la cache raggiunge il 10% della sua dimensione totale, per impedire lo stallo delle query di MySQL per

un periodo troppo lungo durante la sincronizzazione della cache FTS sul disco. (Bug 22516559, 73816).

Integrazione delle correzioni di bug di MySQL.

- L'esecuzione della chiave esterna DROP nella tabella ALTER contemporaneamente all'esecuzione di un'altra operazione DROP fa scomparire la tabella. (Bug 16095573)
- Alcune query INFORMATION_SCHEMA che utilizzano ORDER BY non hanno usato l'ottimizzazione filesort come in precedenza. (Bug 16423536)
- FOUND_ROWS () restituisce il numero sbagliato di righe di una tabella. (Bug 68458)
- Il server presenta dei problemi e non genera un errore quando si aprono troppe tabelle temporanee. (Bug 18948649)

Aggiornamenti del motore di database Aurora MySQL 12/01/2017 (versione 1.10.1) (obsoleta)

Versione: 1.10.1

La versione 1.10.1 di Aurora MySQL richiede l'attivazione e non viene utilizzata per applicare le patch alle istanze database. Può essere usata per creare nuove istanze di Aurora e per aggiornare le istanze esistenti. È possibile applicare la patch scegliendo un cluster nella console Amazon RDS, selezionando Cluster Actions (Azioni cluster), quindi Upgrade Now (Aggiorna ora). L'applicazione delle patch richiede un riavvio del database con un tempo di inattività di 5-30 secondi, al termine del quale si può riprendere a usare i cluster del database. Questa patch utilizza un modello di patch del cluster che consente di applicare le patch a tutti i nodi del cluster di Aurora contemporaneamente.

Nuove caratteristiche

- Audit avanzato:– Aurora MySQL offre una funzione di audit avanzato dalle prestazioni elevate che può essere usata per controllare l'attività del database. Per ulteriori informazioni sull'abilitazione e l'utilizzo dell'audit avanzato, consulta Utilizzo dell'audit avanzato con un cluster di database Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

- È stato risolto un problema relativo all'indicizzazione spaziale durante la creazione di una colonna e l'aggiunta di un indice nella stessa istruzione.
- È stato risolto un problema a causa del quale le statistiche spaziali non venivano conservate durante il riavvio del cluster del database.

Aggiornamenti del motore di database Aurora MySQL 14/12/2016 (versione 1.10) (obsoleta)

Versione: 1.10

Nuove caratteristiche

- Applicazione di patch senza tempi di inattività:– questa caratteristica consente di applicare patch a un'istanza del database senza tempi di inattività. Ciò significa che gli aggiornamenti del database vengono eseguiti senza scollegare le applicazioni client o senza riavviare il database. Questo approccio consente di aumentare la disponibilità dei cluster del database di Aurora durante la finestra di manutenzione. I dati temporanei come quelli dello schema di prestazioni vengono ripristinati durante l'aggiornamento. Questa caratteristica applica le patch fornite dal servizio e le patch avviate dall'utente durante la finestra di manutenzione.

  Quando si avvia una patch, il servizio garantisce che non vi siano blocchi, transazioni o tabelle temporanee aperti e attende un periodo di tempo appropriato durante il quale si possono applicare le patch e il database può essere riavviato. Le sessioni delle applicazioni vengono conservate, tuttavia si potrebbe registrare un peggioramento delle prestazioni durante l'applicazione della patch (circa 5 secondi). Se non è disponibile una finestra appropriata, per impostazione predefinita viene applicato il comportamento di applicazione delle patch standard.

  L'applicazione di patch senza tempi di inattività si svolge sulla base del miglior tentativo ed è soggetta alle seguenti limitazioni:

  - Al momento, questa caratteristica è valida solo per l'applicazione di patch ai cluster del database a nodo singolo o alle istanze writer dei cluster del database a più nodi.
  - Non è supportato l'uso delle connessioni SSL insieme a questa caratteristica. In caso di connessioni SSL attive, Amazon Aurora MySQL non esegue l'applicazione di patch senza tempi di inattività, ma riprova periodicamente fino a quando le connessioni SSL non saranno state

chiuse. In tal caso, si procederà con l'applicazione delle patch senza tempi di inattività. Se le connessioni SSL persistono dopo più di un paio di secondi, vengono applicate le patch standard con tempi di inattività.

  - Questa caratteristica è disponibile nella versione 1.10 e in quelle successive di Aurora. In futuro, verranno identificate le versioni o le patch che non possono essere applicate senza tempi di inattività.
  - Questa caratteristica non può essere usata se la replica basata su log binario è attiva.

- Indicizzazione spaziale:– l'indicizzazione spaziale consente di migliorare le prestazioni delle query su set di dati di grandi dimensioni per le query che usano i dati spaziali. Per ulteriori informazioni sull'uso dell'indicizzazione spaziale, consulta Amazon Aurora MySQL e dati spaziali nella Guida per l'utente di Amazon Aurora.

  Questa caratteristica è disattivata per impostazione predefinita ma può essere attivata nella modalità di laboratorio di Aurora. Per informazioni Aurora, consulta Modalità di laboratorio per Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Miglioramenti della pipeline di replica:– Aurora MySQL ora utilizza un meccanismo migliorato per applicare gli aggiornamenti del flusso di log alla cache del buffer di una replica di Aurora. Questa funzione migliora la stabilità e le prestazioni di lettura delle repliche di Aurora nel caso di un carico di scrittura elevato sul master, ma anche di un carico di lettura significativo sulla replica. Questa caratteristica viene attivata per impostazione predefinita.
- Miglioramento delle prestazioni per i carichi di lavoro con letture memorizzate nella cache:– Aurora MySQL ora utilizza un algoritmo simultaneo senza latch per implementare le viste delle letture, per ottenere un miglioramento delle prestazioni delle query di lettura gestite dalla cache del buffer. Grazie a questo e ad altri miglioramenti, Amazon Aurora MySQL può raggiungere un throughput fino a 625.000 letture al secondo rispetto alle 164.000 letture al secondo di MySQL 5.7 per un carico di lavoro solo Select. SysBench
- Miglioramento delle prestazioni per i carichi di lavoro con righe in conflitto:– Aurora MySQL ora utilizza un nuovo algoritmo per il rilascio dei blocchi che migliora le prestazioni, in particolare nel caso di conflitto nelle pagine (ovvero di molte transazioni che si contendono le righe di una stessa pagina). Nei test con il benchmark TPC-C, si può ottenere un miglioramento di 16 volte superiore delle transazioni al minuto relative a MySQL 5.7. Questa caratteristica è disattivata per impostazione predefinita ma può essere attivata nella modalità di laboratorio di Aurora. Per informazioni Aurora, consulta Modalità di laboratorio per Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

- La velocità della replica della cache dell'indice di ricerca full-text è stata migliorata attraverso l'aggiornamento della cache solo dopo una richiesta di lettura in una replica di Aurora. Questo approccio evita eventuali letture nel disco da parte del thread di replica.
- È stato risolto un problema a causa del quale l'invalidazione della cache del dizionario non funziona in una replica di Aurora per le tabelle contenenti un carattere speciale nel nome del database o della tabella.
- È stato risolto un problema relativo a `STUCK IO` che si verificava durante la migrazione dei dati e relativo ai nodi di storage distribuiti, quando era attivata la gestione dell'utilizzo dello storage.
- È stato risolto un problema relativo alla gestione dei blocchi a causa del quale un controllo delle asserzioni non andava a buon fine per il thread in attesa per il blocco delle transazioni durante la preparazione del rollback o del commit di una transazione.
- È stato risolto un problema relativo a una tabella di dizionario danneggiata tramite l'aggiornamento corretto del numero di riferimenti alle voci della tabella di dizionario.
- È stato corretto un bug in cui il punto di lettura minimo del cluster del database poteva essere conservato dalle repliche di Aurora lenta.
- È stato risolto un problema relativo a una potenziale perdita di memoria nella cache della query.
- È stato corretto un bug a causa del quale una replica di Aurora attivava un blocco a livello di riga in una tabella quando si utilizzava una query `IF` di una procedura archiviata.

Integrazione delle correzioni di bug di MySQL.

- L'unione delle tabelle derivate genera risultati errati con clausole "1=0/false". (Bug 69471)
- Si verifica un arresto anomalo del server in ITEM_FUNC_GROUP_CONCAT::FIX_FIELDS durante la seconda esecuzione della procedura archiviata. (Bug 20755389)
- Impedire lo stallo delle query di MySQL per un periodo troppo lungo durante la sincronizzazione della cache FTS sul disco tramite l'offload di un'attività di sincronizzazione in un thread separato non appena la cache raggiunge il 10% della sua dimensione totale. (Bug 22516559, 73816)

Aggiornamenti del motore di database Aurora MySQL: 10/11/2016 (versioni 1.9.0, 1.9.1) (obsolete)

Versione: 1.9.0, 1.9.1

Nuove caratteristiche

- Creazione migliorata degli indici:– l'implementazione per la creazione di indici secondari ora consente di creare indici dal basso verso l'alto e di eliminare così le divisioni di pagina non necessarie. In questo modo si riduce fino al 75% il tempo necessario alla creazione di un indice o alla rigenerazione di una tabella (in base alla classe di istanza database `db.r3.8xlarge`). Questa caratteristica è stata resa disponibile in modalità di laboratorio nella versione 1.7 di Aurora MySQL e viene attivata per impostazione predefinita nella versione 1.9 e in quelle successive di Aurora. Per informazioni Aurora, consulta Modalità di laboratorio per Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Compressione del blocco (modalità di laboratorio):– questa implementazione riduce in modo significativo (fino al 66%) la quantità di memoria consumata dalla gestione blocchi. Lock Manager out-of-memory può acquisire più blocchi di riga senza incontrare alcuna eccezione. Questa caratteristica è disattivata per impostazione predefinita ma può essere attivata nella modalità di laboratorio di Aurora. Per informazioni Aurora, consulta Modalità di laboratorio per Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Schema di prestazioni: –Aurora MySQL ora comprende il supporto dello schema di prestazioni con un impatto minimo sulle prestazioni. Nei nostri test SysBench, l'utilizzo dello schema di prestazioni potrebbe ridurre le prestazioni di MySQL fino al 60%.

  SysBench il test di un cluster Aurora DB ha mostrato un impatto sulle prestazioni 4 volte inferiore rispetto a MySQL. L'esecuzione della classe di istanze `db.r3.8xlarge` DB ha prodotto 100.000 letture SQL writes/sec e oltre 550.000 letture SQL al secondo, anche con lo schema delle prestazioni abilitato.
- Miglioramento dei conflitti delle righe: –questa caratteristica riduce l'utilizzo della CPU e migliora le prestazioni quando un numero elevato di connessioni accede a un numero ridotto di righe. Questa caratteristica elimina `error 188` in caso di miglioramento dei conflitti delle righe.
- out-of-memoryGestione migliorata: quando vengono eseguite istruzioni SQL non essenziali e bloccate e il pool di memoria riservato viene violato, Aurora impone il rollback di tali istruzioni SQL. Questa funzionalità libera memoria e previene i crash del motore dovuti a eccezioni. out-of-memory

- Selettore di lettura intelligente: questa implementazione migliora la latenza di lettura scegliendo il segmento di archiviazione ottimale tra diversi segmenti per ogni lettura, con conseguente miglioramento della velocità di lettura. SysBench i test hanno mostrato un aumento delle prestazioni fino al 27% per i carichi di lavoro di scrittura.

Miglioramenti

- È stato risolto un problema a causa del quale una replica di Aurora rilevava un blocco condiviso durante l'avvio del motore.
- È stato corretto un potenziale arresto anomalo in una replica di Aurora che sarebbe potuto verificare nel caso in cui il puntatore di lettura del sistema di rimozione fosse stato NULL.

Aggiornamenti del motore di database Aurora MySQL 26/10/2016 (versione 1.8.1) (obsoleta)

Versione: 1.8.1

Miglioramenti

- È stato risolto un problema a causa del quale gli inserimenti bulk che utilizzano trigger e invocano le procedure AWS Lambda non andavano a buon fine.
- È stato risolto un problema a causa del quale la migrazione del catalogo non andava a buon fine nei casi in cui il commit automatico non era attivo a livello globale.
- Risolto un errore di connessione ad Aurora quando si utilizzava SSL e migliorato il gruppo Diffie-Hellman per gestire gli attacchi. LogJam

Integrazione delle correzioni di bug di MySQL.

- OpenSSL ha modificato i parametri di lunghezza della chiave Diffie-Hellman a causa del problema. LogJam (Bug 18367167)

Aggiornamenti del motore di database Aurora MySQL 18/10/2016 (versione 1.8) (obsoleta)

Versione: 1.8

Nuove funzionalità

- AWS Lambda integrazione: ora puoi richiamare in modo asincrono una funzione AWS Lambda da un cluster Aurora DB utilizzando la procedura. `mysql.lambda_async` Per ulteriori informazioni, consulta Chiamare una funzione Lambda da un cluster DB Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Caricamento dei dati da Amazon S3:– ora è possibile caricare i file di testo o XML da un bucket Amazon S3 in un cluster di database Aurora utilizzando i comandi `LOAD DATA FROM S3` o `LOAD XML FROM S3`. Per ulteriori informazioni, consulta Caricamento dei dati in un cluster DB Amazon Aurora MySQL da file di testo in un bucket Amazon S3 nella Guida per l'utente di Amazon Aurora.
- Migrazione del catalogo:– Aurora ora salva i metadati del catalogo nel volume del cluster per supportare la funzione Versioni multiple. In questo modo si attiva una migrazione del catalogo tra versioni e ripristini.
- Applicazione di patch e manutenzione a livello di cluster: –Aurora ora gestisce gli aggiornamenti della manutenzione per un cluster di database intero. Per ulteriori informazioni, consulta Manutenzione di un cluster database Amazon Aurora nella Guida per l'utente di Amazon Aurora.

Miglioramenti

- È stato risolto un problema a causa del quale si verificava un arresto anomalo della replica di Aurora quando non si attuava un blocco a livello di metadati in una tabella DDL in transito.
- Sono consentite repliche di Aurora per modificare tabelle non InnoDB e facilitare la rotazione dei file di log CSV generali e lenti dove `log_output=TABLE`.
- È stato risolto un problema di rallentamento che si verificava durante l'aggiornamento delle statistiche da parte dell'istanza primaria in una replica di Aurora. Senza questa soluzione, le statistiche della replica di Aurora possono non risultare sincronizzate con le statistiche dell'istanza principale e generare un piano di query diverso (probabilmente con prestazioni scarse) in una replica di Aurora.
- È stata risolta una race condition che non permetteva alla replica di aurora di attivare blocchi.

- È stato risolto un problema raro a causa del quale una replica di Aurora che registrava o annullava la registrazione con l'istanza primaria non andava a buon fine.
- È stata risolta una race condition che portava a un deadlock nelle istanze `db.r3.large` all'apertura o chiusura di un volume.
- È stato risolto un out-of-memory problema che può verificarsi a causa di una combinazione di un carico di lavoro di scrittura di grandi dimensioni e di errori nel servizio Aurora Distributed Storage.
- È stato risolto un problema relativo a un consumo della CPU elevato a causa del thread di eliminazione che si avviava durante transazioni lunghe.
- È stato risolto un problema che si verificava durante l'esecuzione di query nello schema di informazioni per ottenere dati sui blocchi in caso di carichi elevati.
- È stato risolto un problema relativo alla procedura di diagnostica che in casi rari poteva provocare lo stallo e il riavvio o failover dei nodi di storage.
- È stato risolto un problema a causa del quale una tabella creata correttamente poteva essere eliminata durante il ripristino da arresto anomalo se questo si verificava durante la gestione di un'istruzione `CREATE TABLE [if not exists]`.
- È stato risolto un problema a causa del quale la procedura di rotazione del log veniva danneggiata quando i log generale e lento non venivano archiviati su disco attraverso la mitigazione del catalogo.
- È stato risolto un problema di arresto anomalo che si verificava quando un utente creava una tabella temporanea all'interno di una funzione definita dall'utente e successivamente tale funzione veniva utilizzata nell'elenco selezionato di query.
- È stato risolto un problema di arresto anomalo che si verificava durante la riproduzione di eventi GTID. GTID non è supportato da Aurora MySQL.

Integrazione delle correzioni di bug di MySQL.

- Quando si rilasciavano tutti gli indici in una colonna con più indici, InnoDB non riusciva a bloccare l'operazione DROP INDEX nei casi in cui una limitazione chiave esterna richiedeva un indice. (Bug 16896810)
- È stato risolto un problema di arresto anomalo in caso di aggiunta di limitazioni relative alla chiave esterna. (Bug 16413976)
- È stato risolto un problema di arresto anomalo relativo al prelievo di un cursore in una procedura archiviata e all'analisi o allo svuotamento della tabella in contemporanea. (Bug 18158639)

- È stato risolto un bug di incremento automatico che si verificava quando un utente modificava una tabella per cambiare il valore AUTO_INCREMENT e lo impostava su un valore inferiore a quello della colonna di incremento automatico massimo. (Bug 16310273)

Aggiornamenti del motore di database Aurora MySQL 20/09/2016 (versione 1.7.1) (obsoleta)

Versione: 1.7.1

Miglioramenti

- È stato risolto un problema a causa del quale si verificava un arresto anomalo della replica di Aurora se la cache della ricerca full-text InnoDB era piena.
- È stato risolto un problema di arresto anomalo del motore di database nel caso in cui un thread lavoratore del pool di thread restava in attesa di se stesso.
- È stato risolto un problema a causa del quale si verificava un arresto anomalo della replica di Aurora se il blocco a livello di metadati di una tabella provocava un deadlock.
- È stato risolto un problema di arresto anomalo del motore di database provocato da una race condition tra i thread lavoratori nel pool di thread.
- È stato risolto un problema relativo al verificarsi di un failover non necessario in caso di carichi elevati se l'agente di monitoraggio non rilevava l'avanzamento delle operazioni di scrittura nel sottosistema di storage distribuito.

Aggiornamenti del motore di database Aurora MySQL 30/08/2016 (versione 1.7.0) (obsoleta)

Versione: 1.7.0

Nuove caratteristiche

- Pianificatore compatibile con NUMA:– il pianificatore di attività per il motore Aurora MySQL ora è compatibile con NUMA (Non-Uniform Memory Access). In questo modo si riduce il conflitto dei socket tra CPU con un conseguente miglioramento delle prestazioni per la classe di istanza del database `db.r3.8xlarge`.

- La lettura anticipata parallela opera in modo asincrono in background: –la lettura anticipata parallela è stata aggiornata per migliorarne le prestazioni attraverso un thread dedicato con l'obiettivo di ridurre i conflitti tra i thread.
- Creazione migliorata degli indici (modalità di laboratorio):– l'implementazione per la creazione di indici secondari ora consente di creare indici dal basso verso l'alto e di eliminare così le divisioni di pagina non necessarie. In questo modo si riduce il tempo necessario alla creazione di un indice o alla rigenerazione di una tabella. Questa caratteristica è disattivata per impostazione predefinita ma può essere attivata nella modalità di laboratorio di Aurora. Per informazioni Aurora, consulta Modalità di laboratorio per Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Miglioramenti

- È stato risolto un problema che si verificava quando i tentativi di stabilire una connessione richiedevano troppo tempo a causa di una coda di richieste di connessioni a un'istanza.
- È stato risolto un problema di arresto anomalo se ALTER TABLE veniva eseguita su una tabella partizionata che non usava InnoDB.
- È stato risolto un problema a causa del quale un carico di lavoro di scrittura elevato poteva provocare un failover.
- È stato risolto un problema di asserzione errata che provocava un errore se RENAME TABLE veniva eseguita su una tabella partizionata.
- È stata migliorata la stabilità durante il rollback di una transazione nell'ambito di un carico di lavoro di inserimento gravoso.
- È stato risolto un problema a causa del quale gli indici di ricerca full-text non erano eseguibili in una replica di Aurora.

Integrazione delle correzioni di bug di MySQL.

- È stata migliorata la scalabilità attraverso la partizione del blocco LOCK_grant. (Porta WL 8355)
- L'apertura del cursore in SELECT in una procedura archiviata provocava segfault. (Bug 16499751)
- MySQL mostra risultati sbagliati nel caso di alcuni usi speciali. (Bug 11751794)
- Arresto anomalo di GET_SEL_ARG_FOR_KEYPART – Provocato dalla patch per il bug 11751794. (Bug 16208709)
- Risultati errati per una query semplici con GROUP BY. (Bug 17909656)

- Righe extra in una query semi-join con predicati in serie. (Bug 16221623)
- L'aggiunta di una clausola ORDER BY dopo una subquery IN poteva provocare la visualizzazione di righe duplicate. (Bug 16308085)
- Arresto anomalo con un Explain plan per una query con una scansione debole per GROUP BY, MyISAM. (Bug 16222245)
- La scansione degli indici debole con il predicato int tra virgolette restituisce dati casuali. (Bug 16394084)
- Se l'ottimizzatore utilizzava una scansione degli indici debole, poteva verificarsi un'uscita dal server durante il tentativo di creazione di una tabella temporanea. (Bug 16436567)
- COUNT(DISTINCT) non deve contare valori NULL, ma venivano comunque contati quando l'ottimizzatore utilizzava una scansione degli indici debole. (Bug 17222452)
- Se una query presentava MIN()/MAX() e aggregate_function(DISTINCT) (ad esempio SUM(DISTINCT)) e veniva eseguita utilizzando una scansione degli indici debole, i valori visualizzati relativi a MIN()/MAX() non erano corretti. (Bug 17217128)

Aggiornamenti del motore di database Aurora MySQL 01/06/2016 (versione 1.6.5) (obsoleta)

Versione: 1.6.5

Nuove caratteristiche

- Storage efficiente di log binari: –lo storage efficiente dei log binari è ora attivo per impostazione predefinita per tutti i cluster di database di Aurora MySQL e non configurabile. Lo storage efficiente dei log binari è stato introdotto con l'aggiornamento di aprile 2016. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 06/04/2016 (versione 1.6) (obsoleta).

Miglioramenti

- È stata migliorata la stabilità delle repliche di Aurora nei casi in cui l'istanza primaria rileva un carico di lavoro gravoso.
- È stata migliorata la stabilità delle repliche di Aurora durante l'esecuzione di query sulle tabelle partizionate e sulle tabelle contenenti caratteri speciali nel nome.

- Sono stati risolti i problemi di connessione che si verificavano in caso di utilizzo di connessioni sicure.

Integrazione delle correzioni di bug di MySQL.

- SLAVE CAN'T CONTINUE REPLICATION AFTER MASTER'S CRASH RECOVERY (Port Bug #17632285)

Aggiornamenti del motore di database Aurora MySQL 06/04/2016 (versione 1.6) (obsoleta)

Versione: 1.6

Questo aggiornamento include i seguenti miglioramenti:

Nuove caratteristiche

- Lettura anticipata parallela – la lettura anticipata parallela viene ora attivata per impostazione predefinita per tutti i cluster di database di Aurora MySQL e non è configurabile. La lettura anticipata parallela è stata introdotta con l'aggiornamento di dicembre 2015. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 03/12/2015 (versione 1.4) (obsoleta).

  Oltre ad attivare la lettura anticipata parallela per impostazione predefinita, questa versione include anche i seguenti miglioramenti per questa caratteristica:

  - È stata migliorata la logica per consentire alla lettura anticipata parallela di essere meno aggressiva, il che è un vantaggio quando il cluster del database rileva molti carichi di lavoro paralleli.
  - È stata migliorata la stabilità delle tabelle più piccole.
- Storage efficiente di log binari (modalità di laboratorio):– i file log binari di MySQL ora vengono archiviati in un modo più efficiente in Aurora MySQL. L'implementazione del nuovo storage consente di eliminare prima i file log binari e migliora le prestazioni del sistema per un'istanza in un cluster di database Aurora MySQL, che è un master di replica log binario.

  Per attivare uno storage efficiente dei log binari, occorre impostare il parametro `aurora_lab_mode` su 1 nel gruppo di parametri per l'istanza primaria o per la replica di Aurora.

`aurora_lab_mode` è un parametro a livello di istanza che viene impostato nel gruppo di parametri `default.aurora5.6` per impostazione predefinita. Per informazioni sulla modifica di un gruppo di parametri di database, consulta Modifica di parametri in un gruppo di parametri del database nella Guida per l'utente di Amazon Aurora. Per informazioni sui gruppi di parametri e su Aurora MySQL, consulta Parametri di configurazione Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Attivare lo storage efficiente dei log binari per le istanze in un cluster di database Aurora MySQL che sono istanze master di replica log binari di MySQL.

- Variabile di sistema AURORA_VERSION:– ora è possibile ottenere una versione di Aurora del cluster di database Aurora MySQL eseguendo una query per la variabile di sistema `AURORA_VERSION`.

  Per ottenere la versione di Aurora, è possibile utilizzare una delle seguenti query:

  ```
  select AURORA_VERSION();
  select @@aurora_version;
  show variables like '%version';
  ```

  Puoi anche vedere la versione di Aurora Console di gestione AWS quando modifichi un cluster DB o chiamando il describe-db-engine-versions AWS CLI comando o l'operazione API Descrivi DBEngine Versions.

- Metrica di utilizzo della memoria per la gestione dei blocchi – ora le informazioni sull'utilizzo della memoria per la gestione dei blocchi sono disponibili sotto forma di parametri.

  Per ottenere il parametro di utilizzo della memoria per la gestione dei blocchi, è possibile utilizzare una delle seguenti query:

  ```
  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');
  ```

Miglioramenti

- È stata migliorata la stabilità durante il ripristino delle transazioni XA e dei file binlog.

- È stato risolto un problema di memoria provocato da un numero elevato di connessioni.
- È stata migliorata l'accuratezza dei seguenti parametri: `Read Throughput`, `Read IOPS`, `Read Latency`, `Write Throughput`, `Write IOPS`, `Write Latency` e `Disk Queue Depth`.
- È stato risolto un problema di stabilità a causa del quale si verificava un riavvio lento delle istanze di grandi dimensioni dopo un arresto anomalo.
- È stata migliorata la simultaneità nel dizionario dati dei meccanismi di sincronizzazione e di espulsione della cache.
- Miglioramenti della stabilità e delle prestazioni per le repliche di Aurora:
  - È stato risolto un problema di stabilità delle repliche di Aurora nei casi di carichi di lavoro di scrittura per l'ottimizzazione o gravosi per l'istanza primaria.
  - È stato migliorato il ritardo di replica per le istanze di db.r3.4xlarge e db.r3.8xlarge.
  - Sono state migliorate le prestazioni tramite la riduzione dei conflitti tra applicazioni di record log e letture simultanee in una replica di Aurora.
  - È stato risolto un problema relativo all'aggiornamento delle statistiche nelle repliche di Aurora per le statistiche aggiornate o appena create.
  - È stata migliorata la stabilità delle repliche di Aurora nel caso di un numero elevato di transazioni nell'istanza primaria e di letture simultanee nelle repliche di Aurora relative agli stessi dati.
  - È stata migliorata la stabilità delle repliche di Aurora durante l'esecuzione delle istruzioni `UPDATE` e `DELETE` con le istruzioni `JOIN`.
  - È stata migliorata la stabilità delle repliche di Aurora durante l'esecuzione delle istruzioni `INSERT ... SELECT`.

Integrazione delle correzioni di bug di MySQL.

- Bug BACKPORT 18694052 SOLUZIONE PER ASSERZIONE ERRORE `! M_ORDERED_REC_BUFFER' IN 5.6 (bug 18305270)
- SEGV IN MEMCPY(), HA_PARTITION::POSITION (bug 18383840)
- RISULTATI ERRATI CON LA PARTIZIONE INDEX_MERGE E NO PK (bug 18167648)
- SVUOTAMENTO TABELLE PER ESPORTAZIONE: ASSERZIONE IN HA_PARTITION::EXTRA (bug 16943907)
- ARRESTO ANOMALE DEL SERVER IN GESTORE VIRTUALE HA_ROWS: :MULTI_RANGE_READ_INFO_CONST (bug 16164031)
- ARRESTO ANOMALO OTTIMIZZATORE SERIE IN SEL_ARG::RB_INSERT() (bug 16241773)

Aggiornamenti del motore di database Aurora MySQL 11/01/2016 (versione 1.5) (obsoleta)

Versione: 1.5

Questo aggiornamento include i seguenti miglioramenti:

Miglioramenti

- È stata eliminata una pausa di 10 secondi delle operazioni di scrittura per le istanze inattive durante le distribuzioni dello storage di Aurora.
- La lettura anticipata logica ora funziona quando `innodb_file_per_table` è impostato su `No`. Per ulteriori informazioni sulla lettura anticipata logica, consulta Aggiornamenti del motore di database Aurora MySQL 03/12/2015 (versione 1.4) (obsoleta).
- Sono stati risolti problemi relativi alla riconnessione delle repliche di Aurora all'istanza primaria. Questo miglioramento risolve anche un problema che si verificava nello specificare valori elevati per il parametro `quantity` durante il test degli errori della replica di Aurora attraverso le query fault-injection. Per ulteriori informazioni, consulta Test di un errore di replica Aurora nella Guida per l'utente di Amazon Aurora.
- È stato migliorato il monitoraggio delle repliche di Aurora in ritardo e che si riavviavano.
- È stato risolto un problema che provocava ritardi, annullamenti di registrazioni e riavvi della replica di Aurora.
- È stato risolto un problema che si verificava con l'esecuzione del comando `show innodb status` durante un deadlock.
- È stato risolto un problema relativo ai failover di istanza di grandi dimensioni in caso di velocità di lettura elevate.

Integrazione delle correzioni di bug di MySQL.

- È stata risolta una correzione incompleta relativa alla ricerca full-text in MySQL che interessava le tabelle con un nome del database che iniziava con un numero. (Bug 17607956)

Aggiornamenti del motore di database Aurora MySQL 03/12/2015 (versione 1.4) (obsoleta)

Versione: 1.4

Questo aggiornamento include i seguenti miglioramenti:

Nuove caratteristiche

- Inserimento rapido:– Accelera gli inserimenti paralleli ordinati in base alla chiave primaria. Per informazioni Aurora, consulta Miglioramenti alle prestazioni di Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.
- Prestazioni di lettura dei set di dati di grandi dimensioni: –Aurora MySQL rileva automaticamente un carico di lavoro I/O gravoso e lancia altri thread per migliorare le prestazioni del cluster di database. Il pianificatore di Aurora analizza l'attività I/O e decide di modificare in modo dinamico il numero ottimale di thread nel sistema, cambiando rapidamente i carichi di lavoro gravosi per CPU e attività I/O con un sovraccarico ridotto.
- Lettura anticipata parallela: –migliora le prestazioni delle scansioni dell'albero B troppo grandi per la memoria disponibile nell'istanza primaria o nella replica di Aurora (incluse le query in un intervallo). La lettura anticipata parallela rileva automaticamente i pattern di lettura della pagina ed esegue un prefetch delle pagine nella cache del buffer prima del necessario. La lettura anticipata parallela funziona su più tabelle contemporaneamente e nell'ambito della stessa transazione.

Miglioramenti:

- È stato risolto un problema di disponibilità breve del database Aurora durante le distribuzioni dello storage Aurora.
- È stato implementato correttamente il limite `max_connection`.
- È stato migliorato lo svuotamento del file binlog laddove Aurora è il master binlog e il database si riavvia dopo un carico di lavoro gravoso.
- È stato risolto un problema di gestione della memoria con la cache della tabella.
- È stato aggiunto il supporto per le pagine di grandi dimensioni nella cache del buffer della memoria condivisa per un ripristino più veloce.
- È stato risolto un problema relativo alla mancata inizializzazione dello storage locale dei thread.
- Consente 16.000 connessioni per impostazione predefinita.

- Pool di thread dinamici per i carichi di lavoro I/O gravosi.
- È stato risolto un problema di viste sull'invalidamento relative a UNION nella cache della query.
- È stato risolto un problema di stabilità relativo al thread delle statistiche del dizionario.
- È stato risolto un problema di perdita di memoria nel sottosistema del dizionario correlato all'espulsione della cache.
- È stato risolto un problema di latenza di lettura elevata nelle repliche di Aurora in cui è presente un carico di scrittura scarso sul master.
- Sono stati risolti problemi di stabilità delle repliche di Aurora durante l'esecuzione di operazioni sulle tabelle partizionate DDL, ad esempio ALTER TABLE ... REORGANIZE PARTITION sul master.
- Sono stati risolti problemi di stabilità relativi alle repliche di Aurora durante l'aumento del volume.
- È stato risolto un problema di prestazioni relativo alle scansioni degli indici non nel cluster nelle repliche di Aurora.
- È stato risolto un problema di stabilità che provocava un ritardo delle repliche di Aurora e possibilmente un annullamento della registrazione e un riavvio.

Integrazione delle correzioni di bug di MySQL.

- SEGV in FTSPARSE(). (Bug 16446108)
- Il dizionario dati InnoDB non viene aggiornato quando si ridenomina la colonna. (Bug 19465984)
- Arresto anomalo di FTS dopo aver modificato il nome della tabella in un database diverso. (Bug 16834860)
- Un errore nella preparazione del trigger per le tabelle troncate genera l'errore 1054. (Bug 18596756)
- Modifiche ai metadati potrebbero provocare problemi relativi all'esecuzione del trigger. (Bug 18684393)
- La materializzazione non viene scelta per un campo UTF8 VARCHAR lungo. (Bug 17566396)
- Piano con esecuzione scarsa quando si usa ORDER BY con un limite X. (Bug #16697792)
- Bug backport 11765744 nelle VERSIONI 5.1, 5.5 E 5.6. (Bug 17083851)
- Problema di mutex in SQL/SQL_SHOW.CC con conseguente. SIG6 Probabile origine: FILL_VARIABLES. (Bug 20788853)
- Bug backport 18008907 nelle versioni 5.5 e successive. (Bug 18903155)

- Adeguamento di una correzione per un errore dell'overflow dello stack in MySQL 5.7. (Bug 19678930)

Aggiornamenti del motore di database Aurora MySQL 16/10/2015 (versioni 1.2, 1.3) (obsolete)

Versioni: 1.2, 1.3

Questo aggiornamento include i seguenti miglioramenti:

Correzioni

- out-of-memoryProblema risolto nel nuovo gestore di blocchi con transazioni di lunga durata
- È stato risolto un problema di vulnerabilità della sicurezza durante la replica di database non RDS for MySQL
- È stato eseguito un aggiornamento per garantire che il quorum scriva i nuovi tentativi correttamente con gli errori di storage
- È stato eseguito un aggiornamento per segnalare in modo più accurato i ritardi delle repliche
- Sono state migliorate le prestazioni tramite la riduzione dei conflitti che si verificano quando più transazioni simultanee cercano di modificare la stessa riga
- È stato risolto un problema di invalidazione della cache della query per le viste create con l'unione di due tabelle
- È stata disattivata la cache della query per le transazioni con l'isolamento di `UNCOMMITTED_READ`

Miglioramenti

- Sono state migliorate le prestazioni per le query di catalogo lente nelle cache usate
- È stata migliorata la simultaneità nelle statistiche dei dizionari
- È stata migliorata la stabilità per la gestione delle risorse della cache delle query, la gestione delle estensioni, i file archiviati nello storage intelligente di Amazon Aurora e la scrittura in batch dei record dei log

Integrazione delle correzioni di bug di MySQL.

- L'interruzione di una query in InnoDB ne può provocare un arresto anomalo con asserzione. (Bug 1608883)
- A causa di un errore nella creazione di un nuovo thread per il pianificatore di eventi, l'esecuzione di eventi o una nuova connessione, non è stato scritto alcun messaggio nel log di errore. (Bug 16865959)
- Se una connessione ha cambiato il suo database predefinito e in contemporanea un'altra connessione ha eseguito SHOW PROCESSLIST, la seconda connessione potrebbe accedere alla memoria non valida nel tentativo di mostrare la memoria del database predefinito della prima connessione. (Bug 11765252)
- LO SVUOTAMENTO DEI LOG BINARI in base al design non ha rimosso i file log binari in uso o attivi e non ha indicato quando ciò si è verificato. (Bug 13727933)
- Per alcune istruzioni, potrebbero verificarsi perdite di memoria durante la rimozione da parte dell'ottimizzatore di clausole delle subquery non necessarie. (Bug 15875919)
- Durante l'arresto, il server non ha potuto cercare di bloccare il mutex non inizializzato. (Bug 16016493)
- Un'istruzione preparata che utilizza GROUP_CONCAT() e una clausola ORDER BY che ha denominato più colonne potrebbe provocare un'uscita del server. (Bug 16075310)
- La strumentazione dello schema di prestazioni mancava per i thread lavoratore slave. (Bug 16083949)
- `STOP SLAVE` potrebbe causare un deadlock se emesso contemporaneamente a un'istruzione come SHOW STATUS che ha recuperato i valori per una o più variabili di stato `Slave_retried_transactions`, `Slave_heartbeat_period`, `Slave_received_heartbeats`, `Slave_last_heartbeat` o `Slave_running`. (Bug 16088188)
- Una query full-text che utilizza una modalità booleana potrebbe non mostrare risultati in alcuni casi, quando il termini di ricerca è una citazione. (Bug 16206253)
- Il tentativo dell'ottimizzatore di rimuovere le clausole di subquery ridondanti ha generato un'asserzione durante l'esecuzione di un'istruzione preparata con una subquery nella clausola ON di un join di una subquery. (Bug 16318585)
- GROUP_CONCAT instabile, arresto anomalo in ITEM_SUM::CLEAN_UP_AFTER_REMOVAL. (Bug 16347450)

- Il tentativo di sostituire l'elenco stopword di una ricerca full-text InnoDB predefinito tramite la creazione di una tabella InnoDB con la stessa struttura di INFORMATION_SCHEMA.INNODB_FT_DEFAULT_STOPWORD genera un errore. (Bug 16373868)
- Dopo che il thread del client in uno slave ha eseguito FLUSH TABLES WITH READ LOCK e dopo che sono stati eseguiti degli aggiornamenti nel master, lo slave resta in attesa durante l'esecuzione di `SHOW SLAVE STATUS`. (Bug 16387720)
- Quando si esegue l'analisi di una stringa di ricerca delimitata, ad esempio "abc-def" in una ricerca full-text, InnoDB utilizza gli stessi delimitatori della parole di MyISAM. (Bug 16419661)
- Arresto anomalo in FTS_AST_TERM_SET_WILDCARD. (Bug 16429306)
- SEGFAULT in FTS_AST_VISIT() per il test FTS RQG. (Bug 16435855)
- Per le build di debug, quanto l'ottimizzatore ha rimosso un elemento Item_ref che puntava a una subquery, ha provocato un'uscita del server. (Bug 16509874)
- Le ricerche full-text nelle tabelle InnoDB non hanno dato un esito positivo nel caso di ricerche di citazioni con gli operatori + o -. (Bug 16516193)
- `START SLAVE`fallito all'avvio del server con le opzioni -- master-info-repository =TABLE relay-log-info-repository =TABLE e con autocommit impostato su 0, insieme a. `--skip-slave-start` (Bug 16533802)
- I risultati di una ricerca full-text InnoDB estesa possono utilizzare una quantità di memoria eccessiva. (Bug 16625973)
- Per le build di debug, si può verificare un'asserzione in OPT_CHECK_ORDER_BY quando si utilizza un elemento binario in una stringa di ricerca, poiché l'elemento binario potrebbe includere byte NULL e altri caratteri non significativi. (Bug 16766016)
- Per alcune istruzioni, potrebbero verificarsi perdite di memoria durante la rimozione da parte dell'ottimizzatore di clausole delle subquery non necessarie. (Bug 16807641)
- È stato possibile generare un deadlock dopo aver utilizzato FLUSH TABLES WITH READ LOCK tramite `STOP SLAVE` in una nuova connessione allo slave e successivamente un `SHOW SLAVE STATUS` usando la connessione originale. (Bug 16856735)
- GROUP_CONCAT() con un separatore non valido può provocare un'uscita dal server. (Bug 16870783)
- Il server ha fatto un blocco eccessivo sui mutex LOCK_Active_mi e active_mi-> rli->data_lock per qualsiasi istruzione “pattern” SHOW STATUS LIKE, anche quando il pattern non corrispondeva alle variabili di stato che usano quei mutex (`Slave_heartbeat_period`, `Slave_last_heartbeat`,

`Slave_received_heartbeats`, `Slave_retried_transactions`, `Slave_running`). (Bug 16904035)

- Una ricerca full-text che utilizza un modificatore IN BOOLEAN MODE genera un errore di asserzione. (Bug 16927092)
- Una ricerca full-text nelle tabelle InnoDB non ha generato risultati quando si è usato l'operatore booleano +. (Bug 17280122)
- Deadlock in 4 modalità: zombie, svuotamento binlog, visualizzazione elenchi processi, visualizzazioni binlog. (Bug 17283409)
- Quando un thread SQL in attesa di un blocco commit viene interrotto e riavviato, la transazione viene ignorata nello slave. (Bug 17450876)
- Si verifica un errore relativo alla ricerca full-text InnoDB a causa di un token non completato. La stringa e la lunghezza della stringa devono essere trasmesse per consentire un confronto. (Bug 17659310)
- Un numero elevato di tabelle InnoDB partizionate può utilizzare molta più memoria se utilizzate in MySQL 5.6 o 5.7 rispetto alla memoria usata dalle stesse tabelle in versioni precedenti di MySQL Server. (Bug 17780517)
- Per le query full-text, un valore num_token inferiore a quello di max_proximity_item può generare un'asserzione. (Bug 18233051)
- Alcune query per le tabelle INFORMATION_SCHEMA TABLES e COLUMNS possono utilizzare troppa memoria nel caso di quantità elevate di tabelle InnoDB vuote. (Bug 18592390)
- Quando si esegue una transazione, viene usato un flag per controllare che sia stato creato un thread e non si controlla direttamente il thread, perché in tal modo si usano meno risorse, in particolare quando si esegue il server con master_info_repository=TABLE. (Bug 18684222)
- Se un thread del client in uno slave ha eseguito FLUSH TABLES WITH READ LOCK mentre il master ha eseguito un DML, l'esecuzione di `SHOW SLAVE STATUS` nello stesso client viene bloccato con un conseguente deadlock. (Bug 19843808)
- Ordinare in base al risultato GROUP_CONCAT() può provocare un'uscita dal server. (Bug 19880368)

Aggiornamenti del motore di database Aurora MySQL 24/08/2015 (versione 1.1) (obsoleta)

Versione: 1.1

Questo aggiornamento include i seguenti miglioramenti:

- Sono stati approvati miglioramenti alla stabilità delle repliche durante l'esecuzione delle stesse con un database MySQL (replica binlog). Per informazioni sulle repliche di Aurora MySQL con MySQL, consulta Replica con Amazon Aurora nella Guida per l'utente di Amazon Aurora.
- È stato aumentato il limite delle dimensioni dei log di inoltro e portato a 1 gigabyte (GB) per un cluster di database Aurora MySQL, che rappresenta uno slave di replica. Questo migliora la gestione dei file per i cluster del database di Aurora.
- È stata migliorata la stabilità di aree di lettura anticipata, relazioni chiave primaria su chiave esterna ricorsive e replica di Aurora.
- Integrazione delle correzioni di bug di MySQL.
  - I database InnoDB con nomi che iniziano con un numero generano un errore del parser per le ricerche full-text. (Bug 17607956)
  - Errore nelle ricerche full-text InnoDB nei database i cui nomi iniziano con un numero. (Bug 17161372)
  - Per i database InnoDB su Windows, l'ID oggetto della ricerca full-text non è nel formato esadecimale previsto. (Bug 16559254)
  - Una regressione del codice introdotta in MySQL 5.6 influisce negativamente sulle prestazioni di DROP TABLE e ALTER TABLE. Questo può provocare un calo delle prestazioni tra MySQL Server 5.5.x e 5.6.x. (Bug 16864741)
- La registrazione è stata semplificata per ridurre la dimensione dei file di log e la quantità di storage necessario.

Correzione dei bug di MySQL attraverso gli aggiornamenti del motore di database Aurora MySQL

Nella seguente sezione vengono identificati i bug di MySQL che sono stati corretti attraverso gli aggiornamenti del motore di database di Aurora MySQL

Argomenti

- Correzione dei bug di MySQL attraverso gli aggiornamenti del motore di database Aurora MySQL 3.x
- Correzione dei bug di MySQL attraverso gli aggiornamenti del motore di database Aurora MySQL 2.x
- Correzione dei bug di MySQL attraverso gli aggiornamenti del motore di database Aurora MySQL 1.x

Correzione dei bug di MySQL attraverso gli aggiornamenti del motore di database Aurora MySQL 3.x

La versione di Aurora compatibile con MySQL 8.0 contiene tutte le correzioni dei bug di MySQL attraverso la corrispondente versione di compatibilità MySQL. La seguente tabella identifica i bug di MySQL aggiuntivi che sono stati corretti attraverso gli aggiornamenti del motore di database di Aurora MySQL, nei quali sono stati risolti.

Aggiornamento del motore di database	Versione compatibile con MySQL	Versione	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 2024-11-1	8,0,39	3,08,0	• È stato risolto un problema a causa del quale `NULL` i valori venivano erroneamente omessi dal set di risultati per alcune query con entrambe le operazioni. `JOIN` `UNION`

Aggiornamento del motore di database	Versione compatibile con MySQL	Versione	Bug di MySQL corretti
8 (versione 3.08.0, compatibile con MySQL 8.0.39)			(Correzione di un bug della community #114301)

Aggiornamento del motore di database	Versione compatibile con MySQL	Versione	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 2024-06-04 (versione 3.07.0, compatibile con MySQL 8.0.36)	8.0.36	3,070	• È stato risolto un problema per cui il valore della riga della cache poteva essere calcolato in modo errato, causando un errore durante il riavvio del database su istanze basate su Graviton. (Correzione di bug della community #35479763) • È stato risolto un problema per cui alcune istanze di sottoquery all'interno delle routine archiviate non venivano sempre gestite correttamente. (Correzione di un bug della community #35377192) • È stato risolto un problema che poteva causare un maggiore utilizzo della CPU a causa della rotazione in background del certificato TLS (Community Bug Fix #34284186). • È stato risolto un problema in cui InnoDB consentiva l'aggiunta di `INSTANT` colonne alle tabelle nello schema del sistema MySQL nelle versioni di Aurora MySQL precedenti alla 3.05, il che poteva portare alla chiusura inaspettata del server (riavvio dell'istanza del database) dopo l'aggiornamento

Aggiornamento del motore di database	Versione compatibile con MySQL	Versione	Bug di MySQL corretti
			alla versione 3.05.0 di Aurora MySQL. (Correzione di bug della community #35625510).

Aggiornamento del motore di database	Versione compatibile con MySQL	Versione	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 2024-03-07 (versione 3.06.0, compatibile con MySQL 8.0.34)	8.0.34	3,06,0	• È stato risolto un problema per cui il valore della riga della cache poteva essere calcolato in modo errato, causando un errore durante il riavvio del database su un'istanza Graviton. (Correzione di bug della community #35479763) • È stato risolto un problema per cui alcune istanze di sottoquery all'interno delle routine archiviate non venivano sempre gestite correttamente. (Correzione di un bug della community #35377192) • È stato risolto un problema che poteva causare un maggiore utilizzo della CPU a causa della rotazione dei certificati TLS in background. (Correzione di un bug della community 34284186) • È stato risolto un problema in cui InnoDB consentiva l'aggiunta di `INSTANT` colonne alle tabelle nello schema del sistema MySQL nelle versioni di Aurora MySQL precedenti alla 3.05, il che poteva portare alla chiusura inaspettata del server (riavvio dell'istanza del database) dopo l'aggiornamento

Aggiornamento del motore di database	Versione compatibile con MySQL	Versione	Bug di MySQL corretti
			alla versione 3.05.0 di Aurora MySQL. (Correzione di un bug della community 35625510)
Aggiornamenti del motore di database Aurora MySQL 2024-01-31 (versione 3.05.2, compatibile con MySQL 8.0.32)	8.0.32	3,05-2	• È stato risolto un problema che `records_in_range` causava un numero eccessivo di letture del disco per `INSERT` le operazioni, con conseguente graduale calo delle prestazioni. (Correzione di bug #34976138 della community)

Aggiornamento del motore di database	Versione compatibile con MySQL	Versione	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 2023-11-21 (versione 3.05.1, compatibile con MySQL 8.0.32)	8.0.32	3,05,1	• È stato risolto un problema in InnoDB che, se a una tabella MySQL in uno schema di sistema veniva aggiunta una colonna `INSTANT ADD` tra Aurora MySQL versioni 3.01 e Aurora MySQL versioni 3.04 e dopo l'aggiornamento di Aurora MySQL alla versione 3.05.0, su queste tabelle il server si chiudeva inaspettatamente. DMLs (Correzione di un bug della community 35625510)
Aggiornamenti del motore del database di Aurora MySQL 25/10/2023 (versione 3.05.0, compatibile con MySQL 8.0.32)	8.0.32	3,05,0	• È stato risolto un problema che può causare un maggiore utilizzo della CPU a causa della rotazione dei certificati TLS in background (Correzione di un bug della community 34284186)

Aggiornamento del motore di database	Versione compatibile con MySQL	Versione	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 2024-03-15 (versione 3.04.2, compatibile con MySQL 8.0.28)	8,0,28	3,04,2	• È stato risolto un problema per cui il valore della riga della cache poteva essere calcolato in modo errato, causando un errore durante il riavvio del database su istanze basate su Graviton. (Correzione di bug della community #35479763) • L'esecuzione ripetuta di una routine memorizzata, utilizzando come sottoquery un'istruzione SELECT contenente più `XOR` condizioni o `AND OR`, ha portato a un consumo eccessivo e probabilmente all'esaurimento della memoria virtuale. (Correzione di bug della community #33852530)

Aggiornamento del motore di database	Versione compatibile con MySQL	Versione	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 2023-11-13 (versione 3.04.1, compatibile con MySQL 8.0.28)	8.0.28	3,04,1	• È stato risolto un problema che può causare un maggiore utilizzo della CPU a causa della rotazione dei certificati TLS in background (Correzione di un bug della community 34284186)

Aggiornamento del motore di database	Versione compatibile con MySQL	Versione	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 2023-07-31 (versione 3.04.0, compatibile con MySQL 8.0.28)	8,0,28	3,04,0	• È stato risolto un problema a causa del quale un blocco buffer contenente una pagina di tabella temporanea intrinsec a veniva riposizionato durante l'attraversamento delle pagine, causando un errore di asserzion e (Bug 33715694). • InnoDB: impedisce alle operazioni DDL online di accedere alla out-of-bounds memoria (bug # 34750489, bug # 108925) • È stato risolto un problema che a volte poteva produrre risultati di query errati durante l'elaborazione di istruzion i SQL complesse costituite da più espressioni di tabella comuni annidate () (Bug# 34572040, Bug# 34634469, Bug# 33856374) CTEs

Aggiornamento del motore di database	Versione compatibile con MySQL	Versione	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 2023-12-08 (versione 3.03.3) (obsoleta)	8.0.26	3,03,3	• È stato risolto un problema che può causare un maggiore utilizzo della CPU a causa della rotazione dei certificati TLS in background (Correzione di un bug della community 34284186)

Aggiornamento del motore di database	Versione compatibile con MySQL	Versione	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 2023-08-29 (versione 3.03.2) (obsoleto)	8,0,26	3,03,2	• È stato risolto un problema che a volte poteva produrre risultati di query errati durante l'elaborazione di istruzioni SQL complesse costituite da più espressioni di tabella comuni annidate (CTEs) (Bug #34572040, Bug #34634469, Bug #33856374) • InnoDB: una race condition tra thread che tentavano di de-inizializzare e inizializzare le statistiche per la stessa tabella ha generato un errore di asserzione (Bug 33135425) • InnoDB: impedisce alle operazioni DDL online di accedere alla out-of-bounds memoria (Bug #34750489, Bug #108925)

Aggiornamento del motore di database	Versione compatibile con MySQL	Versione	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 2023-05-11 (versione 3.03.1) (obsoleto)	8.0.26	3,03,1	• È stato risolto un problema a causa del quale un blocco buffer contenente una pagina di tabella temporanea di funzioni intrinsec he veniva riposizionato durante l'attraversamento delle pagine, causando un errore di asserzion e (Bug 33715694)

Aggiornamento del motore di database	Versione compatibile con MySQL	Versione	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL (versione 3.03.0) (obsoleta)	8,0,26	3,03,0	• È stato risolto un problema per cui alcuni tipi di colonne, tra cui JSON e TEXT, a volte esaurivano il buffer di ordinamento se la sua dimensione non era almeno 15 volte quella della riga più grande dell'ordinamento. Ora il buffer di ordinamento deve essere solo 15 volte più grande della chiave di ordinamento più grande. (Bug 103325, Bug 105532, Bug 32738705, Bug 33501541) • Risolto un problema per cui InnoDB non gestiva sempre correttamente alcuni nomi legali per le partizioni di tabella. (Bug 32208630) • È stato risolto un problema che, in determinate condizioni, poteva restituire risultati errati a causa di un calcolo impreciso della proprietà di nullabilità durante l'esecuzione di una query con una condizione OR. (Bug 34060289) • È stato risolto un problema che, in determinate condizioni, poteva restituire risultati errati quando venivano soddisfatte le due condizioni seguenti:

Aggiornamento del motore di database	Versione compatibile con MySQL	Versione	Bug di MySQL corretti
			• una tabella derivata viene unita al blocco di query esterno. • La query include un left join e una sottoquery IN. (Bug 34060289) • È stato risolto un problema a causa del quale venivano generati valori AUTO_INCREMENT errati quando veniva superato il valore massimo intero della colonna. L'errore era dovuto al fatto che il valore massimo della colonna non veniva considerato. In questo caso avrebbe dovuto essere restituito il precedente valore AUTO_INCREMENT valido, causando un errore di chiave duplicato. (Bug 87926, 26906787) • È stato risolto un problema per cui non era possibile revocare il privilegio DROP sul Performance Schema. (Bug 33578113) • È stato risolto un problema a causa del quale una procedura archiviata contenente un'istruzione IF che utilizzava EXISTS, che agiva su una o più

Aggiornamento del motore di database	Versione compatibile con MySQL	Versione	Bug di MySQL corretti
			tabelle eliminate e ricreate tra un'esecuzione e l'altra, non veniva eseguita correttamente per le invocazioni successive alla prima. (Bug 32855634). • Risolto un problema per cui una query che fa riferimento a una vista in una sottoquery e in un blocco di query esterno può causare un riavvio imprevisto (Bug 32324234)

Aggiornamento del motore di database	Versione compatibile con MySQL	Versione	Bug di MySQL corretti

Aggiornamenti del motore di database Aurora MySQL 2022-11-18 (versione 3.02.2) (obsoleto)	8,0,23	3,02,2	• È stato risolto un problema che, in determinate condizioni, poteva restituire risultati errati a causa di un calcolo impreciso della proprietà di nullabilità durante l'esecuzione di una query con una condizione OR. (Bug 34060289)
			• È stato risolto un problema che, in determinate condizioni, poteva restituire risultati errati quando venivano soddisfatte le due condizioni seguenti:
			  • Una tabella derivata viene unita nel blocco di query esterno.
			  • La query include un left join e una sottoquery IN. (Bug 34060289)
			• È stato risolto un problema per cui non era possibile revocare il privilegio DROP sul Performance Schema. (Bug 33578113)
			• È stato risolto un problema a causa del quale una procedura archiviata contenente un'istruzione IF che utilizzava EXISTS, che agiva su una o più tabelle eliminate e ricreate tra un'esecuzione e l'altra, non

Aggiornamento del motore di database	Versione compatibile con MySQL	Versione	Bug di MySQL corretti
			veniva eseguita correttamente per le invocazioni successive alla prima. (Bug MySQL 32855634). • Sono stati generati valori AUTO_INCREMENT errati quando è stato superato il valore massimo intero della colonna. L'errore era dovuto al fatto che il valore massimo della colonna non veniva considerato. In questo caso avrebbe dovuto essere restituito il precedente valore AUTO_INCREMENT valido, causando un errore di chiave duplicato. (Bug 87926, 26906787) • È stato risolto un problema che poteva causare un errore durante l'aggiornamento di un cluster di database Aurora MySQL versione 1 (compatibile con MySQL 5.6) contenente una tabella creata dall'utente con una determinata tabella. IDs L'assegnazione di queste tabelle IDs può causare conflitti nella tabella del dizionario dei dati IDs durante l'aggiornamento da Aurora MySQL versione 2 (compatibile con MySQL 5.7)

Aggiornamento del motore di database	Versione compatibile con MySQL	Versione	Bug di MySQL corretti
			ad Aurora MySQL versione 3 (compatibile con MySQL 8.0). (Bug 33919635)
Aggiornamenti del motore di database Aurora MySQL 2022-04-20 (versione 3.02.0) (obsoleto)	8.0.23	3,02,0	Risolta la gestione impropria delle tabelle temporanee utilizzate per i cursori all'interno delle procedure archiviate che poteva causare un comportamento imprevisto del server. (Bug 32416811)
Aggiornamenti del motore di database Aurora MySQL 2022-04-15 (versione 3.01.1) (obsoleto)	8,0,23	3,01,1	Risolta la gestione impropria delle tabelle temporanee utilizzate per i cursori all'interno delle procedure archiviate che poteva causare un comportamento imprevisto del server. (Bug 32416811)

Correzione dei bug di MySQL attraverso gli aggiornamenti del motore di database Aurora MySQL 2.x

La versione compatibile con MySQL 5.7 Aurora contiene tutte le correzioni di bug di MySQL tramite MySQL 5.7.44. La seguente tabella identifica i bug di MySQL aggiuntivi che sono stati corretti attraverso gli aggiornamenti del motore di database di Aurora MySQL, nei quali sono stati risolti.

Aggiornamento del motore di database	Version	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 2024-07-09 (versione 2.12.3, compatibile con MySQL 5.7.44) - Versione RDS Extended Support	2.12.3	• È stato risolto un problema per cui le tabelle temporane e associate ai trigger durante l'esecuzione delle istruzioni potevano causare un riavvio imprevisto del motore DB. • È stato risolto un difetto che poteva causare la chiusura del server quando una singola tabella `UPDATE` e `DELETE` le istruzioni che utilizzano espressioni indicizzate venivano eseguite come istruzioni preparate. (Bug #29257254)
Aggiornamenti del motore di database Aurora MySQL 2023-12-28 (versione 2.12.1, compatibile con MySQL 5.7.40) - Versione RDS Extended Support	2.12.1	• È stato risolto un problema che poteva causare il blocco delle connessioni remote esistenti e nuove se eseguite contempor aneamente all'istruzione `SHOW PROCESSLIST` (Bug della community 34857411) • Replica: alcuni log eventi binari non venivano sempre gestiti correttamente (Bug 34617506) • Corretta l'elaborazione di token a carattere singolo da parte di un plug-in parser di ricerca full-text (FTS) (Bug 35432973)
Aggiornamenti del motore di database Aurora MySQL 2023-07-25 (versione 2.12.0, compatibile con MySQL 5.7.40)	2.12.0	• È stato risolto un problema che può causare un maggiore utilizzo della CPU a causa della rotazione dei certificati TLS in background. (Correzione di un bug della community 34284186)

Aggiornamento del motore di database	Version	Bug di MySQL corretti
- Versione RDS Extended Support		
Aggiornamenti del motore di database Aurora MySQL 2023-10-17 (versione 2.11.4, compatibile con MySQL 5.7.12) - Versione RDS Extended Support	2.11.4	• Replica: alcuni eventi di log binari non venivano sempre gestiti correttamente. (Bug 34617506) • È stato risolto un problema che può causare un maggiore utilizzo della CPU a causa della rotazione dei certificati TLS in background. (Correzione di un bug della community 34284186) • Nelle istruzioni preparate, alcuni tipi di query secondarie potrebbero causare la chiusura del server. (Bug 33100586)

Aggiornamento del motore di database	Version	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 2022-10-25 (versione 2.11.0, compatibile con MySQL 5.7.12) - Versione RDS Extended Support	2.11.0	• È stato risolto un problema a causa del quale il codice per la lettura delle informazioni dalle tabelle degli eventi dell'istruzione Performance Schema (ad esempio, events_statements_current) non impediva la scrittura simultanea delle informazioni sul set di caratteri. Di conseguenza, il set di caratteri di testo della query SQL potrebbe non essere valido, con conseguente chiusura del server. Con questa correzione, un set di caratteri non valido causa il troncamento delle colonne SQL_TEXT e impedisce le chiusure del server. (Bug 23540008) • InnoDB: backport di correzioni dei bug della community 25189192, 84038. È stato risolto un problema per cui dopo un'operazione RENAME TABLE che spostava una tabella in uno schema diverso, InnoDB non riusciva ad aggiornare la tabella del dizionario dati INNODB_SYS_DATAFILES. Ciò ha provocato un errore al riavvio che indicava che non era possibile individuare il file di dati dello spazio di tabella. • InnoDB: risolto un problema per cui il server eliminava un indice di chiave esterna definito internamente durante l'aggiunta di un nuovo indice e tentava di utilizzare un indice secondario definito su una colonna generata virtuale come indice di chiave esterna, causando la chiusura del server. InnoDB ora consente a un vincolo di chiave esterna di fare riferimento a un indice secondario definito su una colonna generata virtuale. (Bug #23533396) • È stato risolto un problema che si verificava quando due sessioni eseguivano contemporaneamente un'istruzione INSERT ... L'operazione ON DUPLICATE KEY UPDATE ha generato un deadlock. Durante il rollback parziale di una tupla, un'altra sessione potrebbe aggiornarla. La correzione di questo bug ripristina le correzioni per il bug 11758237, il bug 17604730 e il bug 20040791. (Bug 25966845)

Aggiornamento del motore di database	Version	Bug di MySQL corretti
		• È stato risolto un problema per cui i privilegi EXECUTE e ALTER ROUTINE non venivano concessi correttamente ai creatori della routine anche con automatic_sp_privileges abilitato. (Bug 27407480) • Backport di una correzione per il bug della community 24671968: risolto un problema per cui una query poteva produrre risultati errati se la clausola WHERE conteneva una sottoquery dipendente, la tabella aveva un indice secondari o sulle colonne dell'elenco di selezione seguito dalle colonne nella sottoquery e `GROUP BY` o `DISTINCT` consentiva alla query di utilizzare una scansione degli indici debole. • È stato risolto un problema a causa del quale la replica si interrompeva se veniva emessa un'istruzione di eliminazione multi tabella su più tabelle con chiavi esterne. (Bug 80821) • È stato risolto un problema per cui in casi speciali alcuni errori slave non venivano ignorati anche con slave_skip_errors abilitato. Nei casi in cui l'apertura e il blocco di una tabella o le conversioni dei campi non sono riusciti su un server che esegue la replica basata su righe, l'errore viene considerato critico e lo stato slave_skip_errors viene ignorato. La correzion e garantisce che, con slave_skip_errors abilitato, tutti gli errori segnalati durante l'applicazione di una transazione vengano gestiti correttamente. (Bug 70640, Bug 17653275) • Risolto un problema per cui quando un'istruzione `SET PASSWORD` veniva replicata da un master MySQL 5.6 a uno slave MySQL 5.7 o da un master MySQL 5.7 con la variabile di sistema log_builtin_as_identified_by_password impostata su ON su uno slave MySQL 5.7, anche l'hash della password veniva sottoposto ad hash prima di essere archiviato sullo slave. Il problema è stato risolto e l'hash della password replicata viene archiviato come originariamente passato allo slave. (Bug 24687073)

Aggiornamento del motore di database	Version	Bug di MySQL corretti
		• È stato risolto un problema a causa del quale la serializz azione di un valore JSON costituito da un documento secondario di grandi dimensioni racchiuso in molti livelli di array JSON, oggetti o entrambi richiedeva a volte un tempo eccessivo per essere completata. (Bug 23031146) • Le istruzioni che non possono essere analizzate (a causa, ad esempio, di errori di sintassi) non vengono più scritte nel log delle query lente. (Bug 33732907)
Aggiornamenti del motore di database Aurora MySQL 01/11/2022 (versione 2.10.3) (obsoleta)	210.3	• È stato risolto un problema a causa del quale il codice per la lettura delle informazioni dalle tabelle degli eventi dell'istruzione Performance Schema (ad esempio, events_st atements_current) non impediva la scrittura simultanea delle informazioni sul set di caratteri. Di conseguenza, il set di caratteri di testo della query SQL potrebbe non essere valido, con conseguente chiusura del server. Con questa correzion e, un set di caratteri non valido causa il troncamento delle colonne SQL_TEXT e impedisce le chiusure del server. (Bug 23540008) • Risolto un problema per cui quando una istruzione UPDATE richiedeva una tabella temporanea con una chiave primaria più grande di 1024 byte e quella tabella veniva creata utilizzan do InnoDB, il server poteva chiudersi. (Bug 25153670)

Aggiornamento del motore di database	Version	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 26/01/2022 (versione 2.10.2) (obsoleta)	2.10.2	• È stato risolto un problema in InnoDB per cui un errore nel codice relativo alle statistiche delle tabelle generava un'asserzione nel file di origine dict0stats.cc (http://dict0stats.cc/). (Bug 24585978) • Un indice secondario su una colonna virtuale veniva danneggiato quando l'indice veniva creato online. Per le istruzioni UPDATE (https://dev.mysql.com/doc/refman/5.7/en/update.html), risolviamo questo problema come segue: se il valore della colonna virtuale del record dell'indice è impostato su NULL, generiamo questo valore dal record dell'indice del cluster. (Bug 30556595) • ASSERTION "!OTHER_LOCK" IN LOCK_REC_ADD_TO_QUEUE (Bug 29195848) • HANDLE_FATAL_SIGNAL (SIG=11) IN __STRCHR_ (Bug #28653104) SSE2 • È stato risolto un problema per cui un'interruzione della query durante un'attesa di blocco poteva causare un errore in InnoDB. (Bug 28068293) • Le transazioni interleave potevano talvolta bloccare l'applier replica quando il livello di isolamento delle transazioni era impostato su REPEATABLE READ. (Bug 25040331) • È stato risolto un problema che può causare l'arresto delle repliche binlog a causa del timeout di attesa. (Bug 27189701)
Aggiornamenti del motore di database Aurora MySQL 21/10/2021 (versione 2.10.1) (obsoleta)	2.10.1	CURRENT_TIMESTAMP PRODUCE ZERI NELL'ATTIVAZIONE. (Bug #25209512)

Aggiornamento del motore di database	Version	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 25/05/2021 (versione 2.10.0) (obsoleta)	2.10.0	• Le transazioni interleave potevano talvolta bloccare l'applier replica quando il livello di isolamento delle transazioni era impostato su REPEATABLE READ. (Bug 25040331) • Quando una procedure archiviata conteneva un'istruzione riferita a una vista che a sua volta faceva riferimento a un'altra vista, la procedura non poteva essere richiamata correttamente più di una volta. (Bug 87858, Bug 26864199) • Per le query con molte condizioni `OR`, l'ottimizzatore ora è più efficiente in termini di memoria e ha meno probabilità di superare il limite di memoria imposto dalla variabile di sistema range_optimizer_max_mem_size. Inoltre, il valore predefinito per tale variabile è stato incrementato da 1.536.000 a 8.388.608. (Bug 79450, Bug 22283790) • Replica: nella funzione `next_event()` , che viene chiamata dal thread SQL di una replica per leggere l'evento successivo dal log di relay, il thread SQL non ha rilasciato l'`relaylog.log_lock` acquisito quando si è verificato un errore (ad esempio, a causa di un log di relay chiuso), causando il blocco di tutti gli altri thread in attesa di acquisire un blocco sul log di relay. Con questa correzione, il blocco viene rilasciato prima che il thread SQL lasci la funzione nella situazione. (Bug 21697821) • Correzione di un danneggiamento della memoria per `ALTER TABLE` con colonna virtuale. (Bug 24961167, Bug 24960450) • Replica: non è stato possibile configurare le repliche multithread con dimensioni di coda ridotte utilizzando slave_pending_jobs_size_max se fosse stato necessario elaborare transazioni più grandi di quelle dimensioni. Qualsiasi pacchetto più grande di slave_pending_jobs_size_max è stato rifiutato con l'errore `ER_MTS_EVENT_BIGGER_PENDING_JOBS_SIZE_MAX` , anche se il pacchetto era più piccolo del limite impostato da slave_max_allowed_packet. Con

Aggiornamento del motore di database	Version	Bug di MySQL corretti
		questa correzione, slave_pending_jobs_size_max diventa un limite flessibile anziché un limite rigido. Se la dimensione di un pacchetto supera slave_pending_jobs_size_max ma è inferiore a slave_max_allowed_packet, la transazione viene trattenuta fino a quando tutti i dipendenti replica non hanno code vuote e quindi elaborate. Tutte le transazioni successive vengono mantenute fino al completamento della transazione di grandi dimensioni. Le dimensioni della coda per i dipendent i replica possono quindi essere limitate, pur consentendo transazioni occasionali più grandi. (Bug 21280753, Bug 77406) • Replica: quando si utilizzava una replica multithread, gli errori dell'applicatore visualizzavano i dati dell'ID dipendente che non erano coerenti con i dati esternalizzati nelle tabelle di replica dello schema di prestazioni. (Bug 25231367) • Replica: su una replica di replica basata su GTID in esecuzion e con -gtid-mode=on, -log-bin=off e che utilizza -, quando si verificava un errore che doveva essere ignorato non veniva aggiornato correttamenteslave-skip-errors, causando la perdita della sincronia con. `Exec_Master_Log_Pos` `Exec_Master_Log_Pos` `Read_master_log_pos` Se `GTID_NEXT` non era stato specificato, la replica non aggiornerebbe mai lo stato GTID durante il rollback da una singola transazione di rendiconto. `Exec_Master_Log_Po s` non verrebbe aggiornato perché anche se la transazione è stata completata, il suo stato GTID mostrerebbe altro. La correzione rimuove il vincolo dell'aggiornamento dello stato GTID quando una transazione viene ripristinata solo se viene specificato `GTID_NEXT` . (Bug 22268777) • Replica: un'istruzione parzialmente fallita non consumava correttamente un GTID generato automaticamente o specifica to quando la registrazione binaria è stata disattivata. La correzione assicura che un DROP TABLE parzialmente

Aggiornamento del motore di database	Version	Bug di MySQL corretti
		non riuscito, un DROP USER parzialmente non riuscito o un DROP VIEW parzialmente non riuscito consumino rispettivamente il GTID pertinente e lo salvino nella tabella `@@GLOBAL.GTID_EXECUTED` e `mysql.gtid_executed` quando la registrazione binaria è disabilitata. (Bug 21686749) • Replica: le repliche che eseguono MySQL 5.7 non sono in grado di connettersi a un'origine MySQL 5.5 a causa di un errore nel recupero del server_uuid, che non fa parte di MySQL 5.5. Ciò è stato causato da modifiche nel metodo di recupero del `server_uuid` . (Bug 22748612) • Replica Binlog: il meccanismo di salto delle transazioni GTID non funzionava correttamente per la transazione XA prima di questa correzione. Il server dispone di un meccanismo per saltare (silenziosamente) una transazione GTID se è già stata eseguita quella determinata transazione in passato. (BUG 25041920) • Le istruzioni XA ROLLBACK che non sono andate a buon fine a causa di un ID transazione errato, potrebbero essere registrate nel log binario con l'ID della transazione corretto e potrebbero quindi essere eseguite da repliche di replica. Viene ora effettuato un controllo della situazione di errore prima che si verifichi la registrazione binaria e le istruzioni XA `ROLLBACK` non riuscite non vengono registrate. (Bug 26618925) • Replica: se una replica è stata configurata utilizzando un'istruzione CHANGE MASTER TO che non specificava il nome del file di registro di origine e la posizione del registro di origine, veniva chiusa prima dell'emissione di START SLAVE, quindi riavviata con l'opzione - relay-log-recovery set, la replica non veniva avviata. Ciò accadeva perché il thread del ricevitore non era stato avviato prima del tentativo di recupero del log di inoltro, quindi nel log di inoltro non era disponibile alcun evento di rotazione del log per fornire il nome del file di log di

Aggiornamento del motore di database	Version	Bug di MySQL corretti
		origine e la posizione del log di origine. In questa situazione, la replica ora salta il recupero del log di inoltro e registra un avviso, quindi procede all'avvio della replica. (Bug 28996606, Bug 93397) • Replica: nella replica basata su riga, veniva restituito un messaggio che visualizzava in modo errato le lunghezze dei campi durante la replica da una tabella con una colonna `utf8mb3` a una tabella con la stessa definizione in cui la colonna era stata definita con un set di caratteri `utf8mb4`. (Bug 25135304, Bug 83918) • Replica: quando veniva emessa un'istruzione RESET SLAVE su una replica di replica GTIDs in uso, i file di log di inoltro esistenti venivano eliminati, ma il nuovo file di registro di inoltro sostitutivo veniva generato prima che il set di dati ricevuti GTIDs per il canale fosse cancellato. Il precedente set GTID è stato quindi scritto nel nuovo file di registro di inoltro come `PREVIOUS_GTIDS` evento, causando un errore fatale nella replica in cui si affermava che la replica aveva GTIDs più dell'origine, anche se il set gtid_executed per entrambi i server era vuoto. Ora, quando `RESET SLAVE` viene emesso, il set di dati ricevuti GTIDs viene cancellato prima della generazione del nuovo file di log di inoltro, in modo che questa situazione non si verifichi. (Bug 27411175) • Replica: utilizzata per la replica, le transazioni che includono le istruzioni che hanno causato un errore di analisi (ER_PARSE_ERROR) non possono essere ignorate manualmente con GTIDs il metodo consigliato di iniettare una transazione vuota o sostitutiva con lo stesso GTID. Questa azione dovrebbe avere come esito che la replica identific hi il GTID come già utilizzato e quindi salti la transazione indesiderata che ha condiviso il suo GTID. Tuttavia, nel caso di un errore di analisi, poiché l'istruzione è stata analizzat a prima che il GTID venisse controllato per verificare se

Aggiornamento del motore di database	Version	Bug di MySQL corretti

		dovesse essere saltato, il thread dell'applicatore di replica si è arrestato a causa dell'errore di analisi, anche se l'intenzi one era di saltare comunque la transazione. Con questa correzione, il thread dell'applicatore di replica ora ignora gli errori di analisi se la transazione interessata deve essere saltata perché il GTID era già stato utilizzato. Notare che questa modifica di comportamento non si applica nel caso di carichi di lavoro costituiti da output di log binario prodotto da `mysqlbinlog` . In tale situazione, esisterebbe il rischio che anche una transazione con un errore di analisi che segue immediatamente una transazione saltata venga ignorata in modo silenzioso, quando dovrebbe invece generare un errore. (Bug 27638268)
		• Replica: abilitare il thread SQL a GTID salta una transazione parziale. (Bug 25800025)
		• Replica: quando è stato fornito un parametro di timeout negativo o frazionario`WAIT_UNTIL_SQL_THREAD_AFTER _GTIDS()` , il server si è comportato in modi imprevisti. Con questa correzione:
		  • Un valore di timeout frazionario viene letto così com'è, senza arrotondamento.
		  • Un valore di timeout negativo viene rifiutato con un errore se il server è in modalità SQL rigorosa; se il server non è in una modalità SQL rigorosa, il valore restituisce `NULL` immediatamente la funzione senza alcuna attesa e quindi emette un avviso. (Bug 24976304, Bug 83537)
		• Replica: se la funzione `WAIT_FOR_EXECUTED_ GTID_SET()` è stata utilizzata con un valore di timeout che include una parte frazionaria (ad esempio 1,5), un errore nella logica di casting significava che il timeout è stato arrotonda to al secondo intero più vicino e a zero per valori inferiori a 1 secondo (ad esempio 0,1). La logica di casting è stata ora corretta in modo che il valore di timeout venga applicato come

Aggiornamento del motore di database	Version	Bug di MySQL corretti
		originariamente specificato senza arrotondamento. Grazie a Dirkjan Bussink per il contributo. (Bug 29324564, Bug 94247) • GTIDs Se abilitato, XA COMMIT su una transazione XA disconnessa all'interno di una transazione con più dichiaraz ioni generava un'asserzione. (Bug 22173903) • Replica: è stata generata un'asserzione nelle build di debug se è stata emessa un'istruzione ROLLBACK XA per un identificatore di transazione sconosciuto quando il valore gtid_next era stato impostato manualmente. Il server ora non tenta di aggiornare lo stato GTID se un'istruzione ROLLBACK XA non va a buon fine con un errore. (Bug 27928837, Bug 90640) • Risolve il problema dell'ordinamento errato quando vengono utilizzate più CASE funzioni nella clausola ORDER BY. (Bug 22810883) • Alcune query che utilizzavano l'ordinamento potevano accedere a una colonna non inizializzata durante l'ottimiz zazione e causare la chiusura del server. (Bug 27389294)
Aggiornamenti del motore di database Aurora MySQL 12/11/2021 (versione 2.09.3) (obsoleta)	2.09.3	• ASSERZIONE! M_PREBUILT >TRX->CHECK_FOREIGNS. (Bug #23533396) • Replica:* Un problema di blocco nella funzione WAIT_FOR_EXECUTED_GTID_SET() potrebbe causare il blocco del server in determinate circostanze. Il problema ora è stato corretto. (Bug #29550513)

Aggiornamento del motore di database	Version	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 11/12/2020 (versione 2.09.1) (obsoleta)	2.09.1	• Replica: le transazioni interleave potevano talvolta bloccare l'applier slave quando il livello di isolamento delle transazioni era impostato su REPEATABLE READ. (Bug 25040331) • Per una tabella con una colonna TIMESTAMP o DATETIME con un valore di default di CURRENT_TIMESTAMP, la colonna poteva essere inizializzata su `0000-00-00 00:00:00` se la tabella aveva un'attivazione `BEFORE INSERT`. (Bug 25209512, bug 84077) • Per un'istruzione INSERT per la quale l'elenco `VALUES` produceva valori per la seconda riga o successive utilizzan do una sottoquery contenente un join, il server poteva uscire dopo la mancata risoluzione dei privilegi richiesti. (Bug 23762382)
Aggiornamenti del motore di database Aurora MySQL 12/11/2020 (versione 2.08.3) (obsoleta)	2.08.3	• Bug 23762382 - INSERISCI VALORI QUERY CON JOIN IN SELECT CAUSA UN COMPORTAMENTO ERRATO. • Bug 25209512 - CURRENT_TIMESTAMP PRODUCE ZERI NEL TRIGGER.

Aggiornamento del motore di database	Version	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 02/06/2020 (versione 2.08.0) (obsoleta)	2.08.0	• Bug #25289359: un blocco della cache full-text eseguito quando i dati vengono sincronizzati non è stato rilasciato se la dimensione della cache full-text superava il limite di dimension e della cache full-text. • Bug #29138644: la modifica manuale dell'ora di sistema durante l'esecuzione del server MySQL ha causato ritardi nel thread del cleaner delle pagine. • Bug #25222337: un nome di campo colonna virtuale NULL in un indice virtuale ha causato l'uscita del server durante un confronto tra nomi di campo che si verifica durante il popolamento di colonne virtuali interessate da un vincolo di chiave esterna. • Bug #25053286: l'esecuzione di una procedura archiviata contenente una query che ha effettuato l'accesso a una vista potrebbe allocare memoria che non è stata liberata fino al termine della sessione. • Bug #25586773: l'esecuzione di una procedura archiviat a contenente un'istruzione che ha creato una tabella dal contenuto di alcune istruzioni SELECT potrebbe causare una perdita di memoria. • Bug #28834208: durante l'applicazione di log, dopo un'operaz ione OPTIMIZE TABLE, InnoDB non ha popolato le colonne virtuali prima di controllare gli aggiornamenti dell'indice delle colonne virtuali. • Bug #26666274: ciclo infinito nel container del buffer dello schema delle prestazioni a causa di un overflow di numeri interi non firmati a 32 bit.

Aggiornamento del motore di database	Version	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 16/06/2022 (versione 2.07.8) (obsoleta)	2,07,8	Quando una istruzione UPDATE richiedeva una tabella temporanea con una chiave primaria più grande di 1024 byte e quella tabella veniva creata utilizzando InnoDB, il server poteva chiudersi. (Bug 25153670)
Aggiornamenti del motore di database Aurora MySQL 02/09/2021 (versione 2.07.6) (obsoleta)	2,07,6	• L'INSERIMENTO DI REGISTRI DI DIMENSIONI 64K RICHIEDE TROPPO TEMPO. (Bug#23031146)
Aggiornamenti del motore di database Aurora MySQL 03/04/2021 (versione 2.07.4) (obsoleta)	2.07.4	• Risolto un problema nel parser ngram full-text quando si trattava di token contenenti " (spazio), '%' o ','. Se utilizzano il parser ngram, i clienti dovrebbero rigenerare i loro indici FTS. (Bug 25873310) • Risolto un problema che poteva causare il riavvio del motore durante l'esecuzione della query con viste SQL nidificate. (Bug 27214153, bug 26864199)

Aggiornamento del motore di database	Version	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 10/11/2020 (versione 2.07.3) (obsoleta)	2.07.3	• InnoDB: le transazioni XA simultanee che venivano eseguite correttamente allo stadio di preparazione XA sul master entravano in conflitto quando riprodotte sullo slave, causando un timeout di attesa di blocco nel thread dell'applier. Il conflitto era dovuto all'intervallo di blocco GAP che differiva quando le transazioni venivano riprodotte serialmente sullo slave. Per evitare questo tipo di conflitto, i blocchi GAP eseguiti dalle transazioni XA nel livello di isolamento READ COMMITTED vengono ora rilasciati (e non più ereditati) quando le transazioni XA raggiungono la fase di preparazione. (Bug 27189701, bug 25866046) • InnoDB: un blocco gap veniva preso inutilmente durante la convalida della chiave esterna durante l'utilizzo del livello di isolamento READ COMMITTED. (Bug 25082593) • Replica: quando si utilizzavano transazioni XA, se si verificava un timeout di attesa di blocco o un deadlock per il thread (SQL) dell'applier su uno slave di replica, il tentativo automatico non funzionava. La causa era che mentre il thread SQL eseguiva il rollback, non eseguiva il rollback della transazione XA. Ciò significa che quando la transazione veniva ritentata, il primo evento era XA START, che non era valido poiché la transazione XA era già in corso, causando un errore XAER_RMFAIL. (Bug 24764800) • Replica: le transazioni interleave potevano talvolta bloccare l'applier slave quando il livello di isolamento delle transazioni era impostato su REPEATABLE READ. (Bug 25040331) • Replica: il valore restituito da un'istruzione SHOW SLAVE STATUS per la dimensione totale combinata di tutti i file di log di inoltro esistenti (Relay_Log_Space) poteva diventare molto più grande dello spazio su disco effettivo utilizzato dai file di log di inoltro. Il I/O thread non bloccava la variabile mentre aggiornava il valore, quindi il thread SQL poteva eliminare automaticamente un file di log di inoltro e scrivere un valore

Aggiornamento del motore di database	Version	Bug di MySQL corretti
		ridotto prima che il I/O thread completasse l'aggiornamento del valore. Il thread I/O scriveva quindi il suo calcolo delle dimensioni originali, ignorando l'aggiornamento del thread SQL e aggiungendo pertanto lo spazio per il file eliminato. Il valore Relay_Log_Space è ora bloccato durante gli aggiornamenti per impedire aggiornamenti simultanei e garantire un calcolo accurato. (Bug 26997096, bug 87832) • Per un'istruzione INSERT per la quale l'elenco VALUES produceva valori per la seconda riga o successive utilizzando una sottoquery contenente un join, il server poteva uscire dopo la mancata risoluzione dei privilegi richiesti. (Bug 23762382) • Per una tabella con una colonna TIMESTAMP o DATETIME con un valore di default di CURRENT_TIMESTAMP, la colonna poteva essere inizializzata su `0000-00-00 00:00:00` se la tabella aveva un'attivazione `BEFORE INSERT`. (Bug 25209512, bug 84077) • Un'uscita dal server poteva essere causata da tentativi simultanei di più thread di registrare e annullare la registrazione degli oggetti dello schema delle prestazioni dei metadati. (Bug 26502135) • L'esecuzione di una procedura archiviata contenente un'istruzione che ha creato una tabella dal contenuto di alcune istruzioni SELECT potrebbe causare una perdita di memoria. (Bug 25586773) • L'esecuzione di una procedura archiviata contenente una query che ha effettuato l'accesso a una visualizzazione poteva allocare memoria che non veniva liberata fino al termine della sessione. (Bug 25053286) • Alcuni casi di materializzazione delle query secondarie potrebbero causare l'uscita del server. Queste query ora producono un errore che suggerisce che la materializzazione deve essere disabilitata. (Bug #26402045)

Aggiornamento del motore di database	Version	Bug di MySQL corretti
		• Le query con molti join rimanenti erano lente se veniva utilizzato il buffering di join (ad esempio, utilizzando l'algoritmo di loop nidificato del blocco). (Bug 18898433, bug 72854) • L'ottimizzatore saltava la seconda colonna in un indice composito durante l'esecuzione di un join interno con una clausola `LIKE` per la seconda colonna. (Bug 28086754)
Aggiornamenti del motore di database Aurora MySQL 17/04/2020 (versione 2.07.2) (obsoleta)	2.07.2	• Bug n. 23104498: Risolto un problema in Performance Schema nella segnalazione della memoria totale utilizzata. (https://github.com/mysql/mysql-server/commit/20b6840df5452f47313c6f9a6ca075bfbc00a96b) • Bug n. 22551677: Risolto un problema in Performance Schema che poteva causare il crash del motore del database nel tentativo di portarlo offline. (https://github.com/mysql/mysql-server/commit/05e2386eccd32b6b444b900c9f8a87a1d8d531e9) • Bug n. 23550835, Bug n. 23298025, Bug n. 81464: Risolto un problema in Performance Schema che causava un crash del motore di database dovuto al superamento della capacità di un buffer interno. (https://github.com/mysql/mysql-server/commit/b4287f93857bf2f99b18fd06f555bbe5b12debfc, https://github.com/mysql/mysql-server/commit/b4287f93857bf2f99b18fd06f555bbe5b12debfc)

Aggiornamento del motore di database	Version	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 25/11/2019 (versione 2.07.0) (obsoleta)	2.07.0	• Bug #26251621: INCORRECT BEHAVIOR WITH TRIGGER AND GCOL • Bug #22574695: ASSERTION `!TABLE || (!TABLE->READ_SET || BITMAP_IS_SET(TABLE->READ_SET, FIEL • Bug #25966845: INSERT ON DUPLICATE KEY GENERATE A DEADLOCK • Bug #23070734: CONCURRENT TRUNCATE TABLES CAUSE STALL • Bug #26191879: FOREIGN KEY CASCADES USE EXCESSIVE MEMORY • Bug #20989615: INNODB AUTO_INCREMENT PRODUCES SAME VALUE TWICE
Aggiornamenti del motore di database Aurora MySQL 11/11/2019 (versione 2.05.0) (obsoleta)	2.05.0	• Bug #23054591: PURGE BINARY LOGS TO sta leggendo l'intero file binlog e sta causando lo stallo MySql

Aggiornamento del motore di database	Version	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 14/08/2020 (versione 2.04.9) (obsoleta)	2.04.9	• Bug #23070734, Bug #80060: la combinazione di TRUNCATE causa stalli TABLEs • Bug #23103937: PS_TRUNCATE_ALL_TABLES () NON FUNZIONA IN MODALITÀ SUPER_READ • Bug #22551677: quando si disconnette il server, una race condition all'interno dello schema delle prestazioni potrebbe causare l'uscita del server. • Bug #27082268: sincronizzazione di sincronizzazione FTS non valida. • BUG #12589870: risolto un problema che causa un riavvio con l'istruzione multi-query quando la cache delle query è abilitata. • Bug #26402045: Alcuni casi di materializzazione delle query secondarie potrebbero causare l'uscita del server. Queste query ora producono un errore che suggerisce che la materializzazione deve essere disabilitata. • Bug #18898433: le query con molti join di sinistra erano lente se è stato utilizzato il buffering di join (ad esempio, utilizzando l'algoritmo di loop nidificato del blocco). • Bug #25222337: un nome di campo colonna virtuale NULL in un indice virtuale ha causato l'uscita del server durante un confronto tra nomi di campo che si verifica durante il popolamento di colonne virtuali interessate da un vincolo di chiave esterna. (https://github.com/mysql/mysql-server/commit/273d5c9d7072c63b6c47dbef6963d7dc491d5131) • Bug #25053286: l'esecuzione di una procedura archiviata contenente una query che ha effettuato l'accesso a una vista potrebbe allocare memoria che non è stata liberata fino al termine della sessione. (https://github.com/mysql/mysql-server/commit/d7b37d4d141a95f577916448650c429f0d6e193d) • Bug #25586773: L'esecuzione di una procedura memorizzata contenente un'istruzione che ha creato una tabella dal

Aggiornamento del motore di database	Version	Bug di MySQL corretti
		contenuto di determinate istruzioni SELECT (https://dev.mysql.com/doc/refman/5.7/en/select.html) potrebbe causare una perdita di memoria. () https://github.com/mysql/mysql-server/commit/88301e5adab65f6750f66af284be410c4369d0c1 • Bug #26666274: INFINITO LOOP IN PERFORMANCE SCHEMA BUFFER CONTAINER • Bug #23550835, Bug #23298025, Bug #81464: tabelle SELECT Performance Schema quando un buffer interno era pieno potrebbero causare l'uscita del server.
Aggiornamenti del motore di database Aurora MySQL 19/09/2019 (versione 2.04.6) (obsoleta)	2.04.6	• Bug #23054591: PURGE BINARY LOGS TO sta leggendo l'intero file binlog e causando lo stallo MySql
Aggiornamenti del motore di database Aurora MySQL 02/05/2019 (versione 2.04.2) (obsoleta)	2.04.2	Bug #24829050 - L'OTTIMIZZAZIONE DI INDEX_MERGE_INTERSECTION GENERA RISULTATI QUERY ERRATI
Aggiornamenti del motore di database Aurora MySQL 11/10/2018 (versione 2.03) (obsoleta)	2.03	• LA SCANSIONE INVERSA SU UNA TABELLA PARTIZIONATA ESEGUE ICP - ORDER BY DESC (bug 24929748). • L'OGGETTO JSON CREA CODICE JSON NON VALIDO (Bug#26867509). • L'INSERIMENTO DI GRANDI QUANTITÀ DI DATI JSON PRENDE UN'ECCESSIVA QUANTITÀ DI TEMPO (bug 22843444). • LE TABELLE PARTIZIONATE USANO PIÙ MEMORIA NELLA VERSIONE 5.7 CHE IN QUELLA 5.6 (bug 25080442).

Aggiornamento del motore di database	Version	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 21/09/2018 (versione 2.02.4) (obsoleta)	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(!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

Aggiornamento del motore di database	Version	Bug di MySQL corretti
		• 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
Aggiornamenti del motore di database Aurora MySQL 03/05/2018 (versione 2.02) (obsoleta)	2.02.0	Il join sinistro genera risultati errati nell'outer (bug 22833364)

Correzione dei bug di MySQL attraverso gli aggiornamenti del motore di database Aurora MySQL 1.x

La versione di Aurora compatibile con MySQL 5.6 contiene tutte le correzioni dei bug di MySQL attraverso MySQL 5.6.10. La seguente tabella identifica i bug di MySQL aggiuntivi che sono stati corretti attraverso gli aggiornamenti del motore di database di Aurora MySQL, nei quali sono stati risolti.

Aggiornamento del motore di database	Version	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 18/03/2021 (versione 1.23.2) (obsoleta)	1.23.2	• Replica: durante l'esecuzione di un' `SHOW BINLOG EVENTS` istruzione, qualsiasi transazione parallela è stata bloccata. La correzione assicura che il `SHOW BINLOG EVENTS` processo acquisisca ora solo un blocco per la durata del calcolo della posizione finale del file, pertanto le transazioni parallele non vengono bloccate per lunghe durate. (Bug #76618, Bug #20928790)

Aggiornamento del motore di database	Version	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 02/09/2020 (versione 1.23.0) (obsoleta)	1.23.0	• Gli eventi Binlog con `ALTER TABLE ADD COLUMN ALGORITHM=QUICK` verranno riscritti come `ALGORITHM =DEFAULT` da essere compatibili con l'edizione della community. • BUG #22350047: IF CLIENT KILLED AFTER ROLLBACK TO SAVEPOINT PREVIOUS STMTS COMMITTED • Bug #29915479: RUNNING COM_REGISTER_SLAVE WITHOUT COM_BINLOG_DUMP CAN RESULTS IN SERVER EXIT • Bug #30441969: BUG #29723340: MYSQL SERVER CRASH AFTER SQL QUERY WITH DATA ?AST • Bug #30628268: OUT OF MEMORY CRASH • Bug #27081349: UNEXPECTED BEHAVIOUR WHEN DELETE WITH SPATIAL FUNCTION • Bug #27230859: UNEXPECTED BEHAVIOUR WHILE HANDLING INVALID POLYGON" • Bug #27081349: UNEXPECTED BEHAVIOUR WHEN DELETE WITH SPATIAL" • Bug #26935001: ALTER TABLE AUTO_INCREMENT TRIES TO READ INDEX FROM DISCARDED TABLESPACE • Bug #29770705: SERVER CRASHED WHILE EXECUTING SELECT WITH SPECIFIC WHERE CLAUSE • Bug #27659490: SELECT USING DYNAMIC RANGE AND INDEX MERGE USE TOO MUCH MEMORY(OOM) • Bug #24786290: REPLICATION BREAKS AFTER BUG #74145 HAPPENS IN MASTER • Bug #27703912: EXCESSIVE MEMORY USAGE WITH MANY PREPARE • Bug #20527363: TRONCA IL CRASH TEMPORANEO DELLA TABELLA:! DICT_ TF2 _FLAG_IS_SET (TABELLA, DICT_ _TEMPORANEO) TF2

Aggiornamento del motore di database	Version	Bug di MySQL corretti
		• Bug#23103937 PS_TRUNCATE_ALL_TABLES() DOES NOT WORK IN SUPER_READ_ONLY MODE • Bug #25053286: USE VIEW WITH CONDITION IN PROCEDURE CAUSES INCORRECT BEHAVIOR (risolto nella versione 5.6.36) • Bug #25586773: INCORRECT BEHAVIOR FOR CREATE TABLE SELECT IN A LOOP IN SP (fixed in 5.6.39) • Bug #27407480: AUTOMATIC_SP_PRIVILEGES REQUIRES NEED THE INSERT PRIVILEGES FOR MYSQL.USER TABLE • Bug #26997096: `relay_log_space` valore non viene aggiornato in modo sincronizzato in modo che il suo valore è talvolta molto più alto dello spazio su disco effettivo utilizzato dai log di inoltro. • Bug#15831300 SLAVE_TYPE_CONVERSIONS=ALL_ NON_LOSSY NOT WORKING AS EXPECTED • SSL Bug backport Bug #17087862, Bug #20551271 • Bug #16894092: PERFORMANCE REGRESSION IN 5.6.6+ FOR INSERT INTO ... SELECT ... FROM (risolto nella versione 5.6.15). • Porta una correzione di bug relativa a `SLAVE_TYP E_CONVERSIONS` .

Aggiornamento del motore di database	Version	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 11/09/2020 (versione 1.22.3) (obsoleta)	1.22.3	• Bug 26654685: Un ID indice danneggiato rilevato durante un controllo di chiave esterna generava un'asserzione • Bug 15831300: Per impostazione predefinita, quando si promuovono numeri interi da un tipo più piccolo sul master a un tipo più grande sullo slave (ad esempio, da una colonna SMALLINT sul master a una colonna BIGINT sullo slave), i valori promossi vengono trattati come se fossero firmati. Ora in questi casi è possibile modificare o sovrascrivere questo comportamento utilizzando uno o entrambi `ALL_SIGNED` , `ALL_UNSIGNED` nell'insieme di valori specificati per la variabile di sistema server slave_type_conversions . Per ulteriori informazioni, consultare Replica basata su righe: promozione e riduzione di livello degli attributi, nonché la descrizione della variabile. • Bug 17449901: Con `foreign_key_checks=0` , InnoDB consentiva l'eliminazione di un indice richiesto da un vincolo di chiave esterna, posizionando la tabella in uno stato incoerente e causando il fallimento del controllo della chiave esterna che si verifica al caricamento della tabella. Ora InnoDB impedisce di eliminare un indice richiesto da un vincolo di chiave esterna, anche con foreign_key_checks=0. Il vincolo di chiave esterna deve essere rimosso prima di eliminare l'indice di chiave esterna. • BUG n. 20768847: Una operazione ALTER TABLE ... DROP INDEX su una tabella con dipendenze da chiavi esterne ha generato un’asserzione.

Aggiornamento del motore di database	Version	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 25/11/2019 (versione 1.22.0) (obsoleta)	1.22.0	• Bug#16346241 - SERVER CRASH IN ITEM_PARAM::QUERY_VAL_STR • Bug#17733850 - NAME_CONST() CRASH IN ITEM_NAME_CONST::ITEM_NAME_CONST() • Bug #20989615 - INNODB AUTO_INCREMENT PRODUCES SAME VALUE TWICE • Bug #20181776 - ACCESS CONTROL DOESN'T MATCH MOST SPECIFIC HOST WHEN IT CONTAINS WILDCARD • Bug #27326796 - MYSQL CRASH WITH INNODB ASSERTION FAILURE IN FILE PARS0PARS.CC • Bug #20590013 - IF YOU HAVE A FULLTEXT INDEX AND DROP IT YOU CAN NO LONGER PERFORM ONLINE DDL
Aggiornamenti del motore di database Aurora MySQL 25/11/2019 (versione 1.21.0) (obsoleta)	1.21.0	• Bug #19929406: HANDLE_FATAL_SIGNAL (SIG=11) IN __MEMMOVE_ _INDIETRO DA STRING: :COPY SSSE3 • Bug #17059925: per istruzioni UNION, il valore esaminato dalle righe è stato calcolato erroneamente. Questo bug si manifestava con valori troppo grandi per la colonna `ROWS_EXAMINED` delle tabelle di istruzioni schema di prestazioni (ad esempio events_statements_current). • Bug #11827369: alcune query con sottoquery `SELECT ... FROM DUAL` nidificate generavano un'asserzione. • Bug #16311231: restituiti risultati errati se una query conteneva una sottoquery in una clausola `IN` che conteneva un'operazione XOR nella clausola `WHERE`.

Aggiornamento del motore di database	Version	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 11/11/2019 (versione 1.20.0) (obsoleta)	1.20.0	• Bug #19929406: HANDLE_FATAL_SIGNAL (SIG=11) IN __MEMMOVE_ _BACK FROM STRING: :COPY SSSE3 • Bug #17059925: per istruzioni UNION, il valore esaminato dalle righe è stato calcolato erroneamente. Questo bug si manifestava con valori troppo grandi per la colonna `ROWS_EXAMINED` delle tabelle di istruzioni schema di prestazioni (ad esempio events_statements_current). • Bug #11827369: alcune query con sottoquery `SELECT ... FROM DUAL` nidificate generavano un'asserzione. • Bug #16311231: restituiti risultati errati se una query conteneva una sottoquery in una clausola `IN` che conteneva un'operazione XOR nella clausola `WHERE`.
Aggiornamenti del motore di database Aurora MySQL 19/09/2019 (versione 1.19.5) (obsoleta)	1.19.5	• CVE-2018-2696 • CVE-2015-4737 • Bug #19929406: HANDLE_FATAL_SIGNAL (SIG=11) IN __MEMMOVE_ _BACK FROM STRING: :COPY SSSE3 • Bug #17059925: per istruzioni UNION, il valore esaminato dalle righe è stato calcolato erroneamente. Questo era manifestato come valori troppo grandi per la colonna `ROWS_EXAMINED` delle tabelle di istruzioni schema di prestazioni (ad esempio events_statements_current). • Bug #11827369: alcune query con sottoquery `SELECT ... FROM DUAL` nidificate generavano un'asserzione. • Bug #16311231: restituiti risultati errati se una query conteneva una sottoquery in una clausola `IN` che conteneva un'operazione XOR nella clausola `WHERE`.

Aggiornamento del motore di database	Version	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 07/02/2019 (versione 1.19.0) (obsoleta)	1.19.0	• BUG #32917: DETECT ORPHAN TEMP-POOL FILES, AND HANDLE GRACEFULLY • BUG #63144: CREATE TABLE IF NOT EXISTS - IL BLOCCO DEI METADATI È TROPPO RESTRITTIVO
Aggiornamenti del motore di database Aurora MySQL 17/01/2019 (versione 1.17.8) (obsoleta)	1.17.8	• BUG #13418638: CREATE TABLE IF NOT EXISTS - IL BLOCCO DEI METADATI È TROPPO RESTRITTIVO
Aggiornamenti del motore di database Aurora MySQL 08/10/2018 (versione 1.17.7) (obsoleta)	1.17.7	• Drop index su una colonna chiave esterna porta a una tabella mancante. (Bug 16208542) • Perdita di memoria in add_derived_key(). (Bug 76349) • Per le tabelle partizionate le query potrebbero restituire risultati differenti a seconda del fatto che l'unione dell'indice sia stato usato o meno. (Bug 16862316) • Le query che usano l'ottimizzazione dell'unione dell'indice (consulta Ottimizzazione dell'unione dell'indice) possono restituire risultati non validi quando eseguite su tabelle partizionate da HASH. (Bug 17588348)
Aggiornamenti del motore di database Aurora MySQL 06/09/2018 (versione 1.17.6) (obsoleta)	1.17.6	• Per un'istruzione ALTER TABLE che ha rinominato o modificat o il valore predefinito di una colonna BINARY, la modifica è stata fatta utilizzando una copia della tabella e non in ordine. (Bug 67141, 14735373, 69580, 17024290) • Un outer join tra una tabella regolare e una derivata che è implicitamente gruppi può causare un'uscita dal server. (Bug 16177639)

Aggiornamento del motore di database	Version	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 13/03/2018 (versione 1.17) (obsoleta)	1.17.0	• Replica incorretta di LAST_INSERT_ID nel caso di utilizzo di filtri per la replica (bug 69861) • La query restituisce risultati diversi a seconda dell'impo stazione INDEX_MERGE (bug 16862316) • Nuova procedura query in routine archiviata, piano delle query inefficiente (bug 16346367) • InnoDB FTS : Assert in FTS_CACHE_APPEND_D ELETED_DOC_IDS (Bug 18079671) • Assert RBT_EMPTY(INDEX_CACHE->WORDS) nella colonna di modifica ALTER TABLE (bug 17536995) • La ricerca nel testo InnoDB non trova record quando si utilizzano punti di salvataggio (bug 70333 e 17458835)
Aggiornamenti del motore di database Aurora MySQL 20/11/2017 (versione 1.15.1) (obsoleta)	1.15.1	• Ripristino — Stallo delle istanze MySQL durante l'indiciz zazione SYNC (bug 73816) • Ripristino — Assert RBT_EMPTY(INDEX_CACHE->WORDS) nella colonna di modifica ALTER TABLE (bug 17536995) • Ripristino — La ricerca nel testo InnoDB non trova record quando si utilizzano punti di salvataggio (bug 70333)

Aggiornamento del motore di database	Version	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 24/10/2017 (versione 1.15) (obsoleta)	1.15.0	• CREATE USER accetta un hash di password e plugin, ma ignora l'hash di password (bug 78033) • Il motore per il partizionamento aggiunge campi al bit letto impostato per restituire dati ordinati da un indice partizionato. Ne consegue che il buffer del join prova a leggere i campi non necessari. Il problema è stato corretto evitando di aggiunger e tutti i campi partizionati a read_set e ordinando solo i campi prefisso già impostati in read_set. È stato aggiungo DBUG_ASSERT che nel caso di utilizzo di key_cmp, consente di leggere almeno il primo campo (bug 16367691). • Stallo delle istanze MySQL durante l'indicizzazione SYNC (bug 73816) • Assert RBT_EMPTY(INDEX_CACHE->WORDS) nella colonna di modifica ALTER TABLE (bug 17536995) • La ricerca nel testo InnoDB non trova record quando si utilizzano punti di salvataggio (bug 70333)
Aggiornamenti del motore di database Aurora MySQL 13/03/2018 (versione 1.14.4) (obsoleta)	1.14.4	• Gli eventi ignorabili non funzionano e non vengono sottoposti a test (bug 74683) • NUOVO->VECCHIO ERRORE ASSERT 'GTID_MODE > 0' (bug 20436436)
Aggiornamenti del motore di database Aurora MySQL 07/08/2017 (versione 1.14) (obsoleta)	1.14.0	Una ricerca full-text combinata con le tabelle derivate (sottoquery nella clausola `FROM`) ha provocato un'uscita dal server. Ora, se un'operazione full-text dipende da una tabella derivata, il server genera un errore in cui viene indicato che non è possibile eseguire una ricerca full-text in una tabella materializzata. (Bug 68751, 16539903)

Aggiornamento del motore di database	Version	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 15/05/2017 (versione 1.13) (obsoleta)	1.13.0	• Ricaricare una tabella rimossa mentre era vuota ha provocato il ripristino del valore di AUTO_INCREMENT. (Bug 21454472, 77743) • Il record di un indice non è stato trovato durante il rollback a causa di incoerenze nella struttura purge_node_t. L'incoerenza ha generato avvisi e messaggi di errori quali "error in sec index entry update" (errore relativo all'aggiornamento delle voci dell'indice secondario), "unable to purge a record" (impossibile rimuovere un record) e "tried to purge sec index entry not marked for deletion" (tentativo di rimuovere una voce dell'indice secondario non contrassegnata per l'eliminazione). (Bug 19138298, 70214, 21126772, 21065746) • Il calcolo errato della dimensione dello stack per l'operazione qsort ha creato un overflow dello stack. (Bug 73979) • Il record non è stato trovato in un indice durante il rollback. (Bug 70214, 72419) • Quando si aggiunge la colonna TIMESTAMP con ALTER TABLE, durante l'aggiornamento di CURRENT_TIMESTAMP non vengono inseriti dati (bug 17392)

Aggiornamento del motore di database	Version	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 04/05/2017 (versione 1.12) (obsoleta)	1.12.0	• Ricaricare una tabella rimossa mentre era vuota ha provocato il ripristino del valore di AUTO_INCREMENT. (Bug 21454472, 77743) • Il record di un indice non è stato trovato durante il rollback a causa di incoerenze nella struttura purge_node_t. L'incoerenza ha generato avvisi e messaggi di errori quali "error in sec index entry update" (errore relativo all'aggiornamento delle voci dell'indice secondario), "unable to purge a record" (impossibile rimuovere un record) e "tried to purge sec index entry not marked for deletion" (tentativo di rimuovere una voce dell'indice secondario non contrassegnata per l'eliminazione). (Bug 19138298, 70214, 21126772, 21065746) • Il calcolo errato della dimensione dello stack per l'operazione qsort ha creato un overflow dello stack. (Bug 73979) • Il record non è stato trovato in un indice durante il rollback. (Bug 70214, 72419) • Quando si aggiunge la colonna TIMESTAMP con ALTER TABLE, durante l'aggiornamento di CURRENT_TIMESTAMP non vengono inseriti dati (bug 17392)
Aggiornamenti del motore di database Aurora MySQL 23/02/2017 (versione 1.11) (obsoleta)	1.11.0	• L'esecuzione della chiave esterna DROP nella tabella ALTER contemporaneamente all'esecuzione di un'altra operazione DROP fa scomparire la tabella. (Bug 16095573) • Alcune query INFORMATION_SCHEMA che utilizzano ORDER BY non hanno usato l'ottimizzazione filesort come in precedenza. (Bug 16423536) • FOUND_ROWS () restituisce il numero sbagliato di righe di una tabella. (Bug 68458) • Il server presenta dei problemi e non genera un errore quando si aprono troppe tabelle temporanee. (Bug 18948649)

Aggiornamento del motore di database	Version	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 14/12/2016 (versione 1.10) (obsoleta)	1.10.0	• L'unione delle tabelle derivate genera risultati errati con clausole "1=0/false". (Bug 69471) • Si verifica un arresto anomalo del server in ITEM_FUNC_GROUP_CONCAT::FIX_FIELDS durante la seconda esecuzione della procedura archiviata. (Bug 20755389) • Impedire lo stallo delle query di MySQL per un periodo troppo lungo durante la sincronizzazione della cache FTS sul disco tramite l'offload di un'attività di sincronizzazione in un thread separato non appena la cache raggiunge il 10% della sua dimensione totale. (Bug 22516559, 73816)
Aggiornamenti del motore di database Aurora MySQL 26/10/2016 (versione 1.8.1) (obsoleta)	1.8.1	• OpenSSL ha modificato i parametri di lunghezza della chiave Diffie-Hellman a causa del problema. LogJam (Bug 18367167)

Aggiornamento del motore di database	Version	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 18/10/2016 (versione 1.8) (obsoleta)	1.8.0	• Quando si rilasciavano tutti gli indici in una colonna con più indici, InnoDB non riusciva a bloccare l'operazione DROP INDEX nei casi in cui una limitazione chiave esterna richiedev a un indice. (Bug 16896810) • È stato risolto un problema di arresto anomalo in caso di aggiunta di limitazioni relative alla chiave esterna. (Bug 16413976) • È stato risolto un problema di arresto anomalo relativo al prelievo di un cursore in una procedura archiviata e all'anali si o allo svuotamento della tabella in contemporanea. (Bug 18158639) • È stato risolto un bug di incremento automatico che si verificav a quando un utente modificava una tabella per cambiare il valore AUTO_INCREMENT e lo impostava su un valore inferiore a quello della colonna di incremento automatico massimo. (Bug 16310273)

Aggiornamento del motore di database	Version	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 30/08/2016 (versione 1.7.0) (obsoleta)	1.7.0	• È stata migliorata la scalabilità attraverso la partizione del blocco LOCK_grant. (Porta WL 8355) • L'apertura del cursore in SELECT in una procedura archiviata provocava segfault. (Bug 16499751) • MySQL mostra risultati sbagliati nel caso di alcuni usi speciali. (Bug 11751794) • Arresto anomalo di GET_SEL_ARG_FOR_KEYPART – Provocato dalla patch per il bug 11751794. (Bug 16208709) • Risultati errati per una query semplici con GROUP BY. (Bug 17909656) • Righe extra in una query semi-join con predicati in serie. (Bug 16221623) • L'aggiunta di una clausola ORDER BY dopo una subquery IN poteva provocare la visualizzazione di righe duplicate. (Bug 16308085) • Arresto anomalo con un Explain plan per una query con una scansione debole per GROUP BY, MyISAM. (Bug 16222245) • La scansione degli indici debole con il predicato int tra virgolett e restituisce dati casuali. (Bug 16394084) • Se l'ottimizzatore utilizzava una scansione degli indici debole, poteva verificarsi un'uscita dal server durante il tentativo di creazione di una tabella temporanea. (Bug 16436567) • COUNT(DISTINCT) non deve contare valori NULL, ma venivano comunque contati quando l'ottimizzatore utilizzava una scansione degli indici debole. (Bug 17222452) • Se una query presentava MIN()/MAX() e aggregate_function (DISTINCT) (ad esempio SUM(DISTINCT)) e veniva eseguita utilizzando una scansione degli indici debole, i valori visualizz ati relativi a MIN()/MAX() non erano corretti. (Bug 17217128)

Aggiornamento del motore di database	Version	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 01/06/2016 (versione 1.6.5) (obsoleta)	1.6.5	• SLAVE CAN'T CONTINUE REPLICATION AFTER MASTER'S CRASH RECOVERY (Port Bug #17632285)
Aggiornamenti del motore di database Aurora MySQL 06/04/2016 (versione 1.6) (obsoleta)	1.6.0	• Bug BACKPORT 18694052 SOLUZIONE PER ASSERZION E ERRORE `!M_ORDERED_REC_BUFFER' IN 5.6 (bug 18305270) • SEGV IN MEMCPY(), HA_PARTITION::POSITION (bug 18383840) • RISULTATI ERRATI CON LA PARTIZIONE INDEX_MERGE E NO PK (bug 18167648) • SVUOTAMENTO TABELLE PER ESPORTAZIONE: ASSERZIONE IN HA_PARTITION::EXTRA (bug 16943907) • ARRESTO ANOMALE DEL SERVER IN GESTORE VIRTUALE HA_ROWS: :MULTI_RANGE_READ_INFO_CONST (bug 16164031) • ARRESTO ANOMALO OTTIMIZZATORE SERIE IN SEL_ARG::RB_INSERT() (bug 16241773)
Aggiornamenti del motore di database Aurora MySQL 11/01/2016 (versione 1.5) (obsoleta)	1.5.0	• È stata risolta una correzione incompleta relativa alla ricerca full-text in MySQL che interessava le tabelle con un nome del database che iniziava con un numero. (Bug 17607956)

Aggiornamento del motore di database	Version	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 03/12/2015 (versione 1.4) (obsoleta)	1.4	• SEGV in FTSPARSE(). (Bug 16446108) • Il dizionario dati InnoDB non viene aggiornato quando si ridenomina la colonna. (Bug 19465984) • Arresto anomalo di FTS dopo aver modificato il nome della tabella in un database diverso. (Bug 16834860) • Un errore nella preparazione del trigger per le tabelle troncate genera l'errore 1054. (Bug 18596756) • Modifiche ai metadati potrebbero provocare problemi relativi all'esecuzione del trigger. (Bug 18684393) • La materializzazione non è stata scelta per un campo VARCHAR lungo. UTF8 (Bug 17566396) • Piano con esecuzione scarsa quando si usa ORDER BY con un limite X. (Bug #16697792) • Bug backport 11765744 nelle VERSIONI 5.1, 5.5 E 5.6. (Bug 17083851) • Problema di mutex in SQL/SQL_SHOW.CC con conseguente. SIG6 Probabile origine: FILL_VARIABLES. (Bug 20788853) • Bug backport 18008907 nelle versioni 5.5 e successive. (Bug 18903155) • Adeguamento di una correzione per un errore dell'overflow dello stack in MySQL 5.7. (Bug 19678930)

Aggiornamento del motore di database	Version	Bug di MySQL corretti
Aggiornamenti del motore di database Aurora MySQL 16/10/2015 (versioni 1.2, 1.3) (obsolete)	1.2, 1.3	• L'interruzione di una query in InnoDB ne può provocare un arresto anomalo con asserzione. (Bug 1608883) • A causa di un errore nella creazione di un nuovo thread per il pianificatore di eventi, l'esecuzione di eventi o una nuova connessione, non è stato scritto alcun messaggio nel log di errore. (Bug 16865959) • Se una connessione ha cambiato il suo database predefinito e in contemporanea un'altra connessione ha eseguito SHOW PROCESSLIST, la seconda connessione potrebbe accedere alla memoria non valida nel tentativo di mostrare la memoria del database predefinito della prima connessione. (Bug 11765252) • LO SVUOTAMENTO DEI LOG BINARI in base al design non ha rimosso i file log binari in uso o attivi e non ha indicato quando ciò si è verificato. (Bug 13727933) • Per alcune istruzioni, potrebbero verificarsi perdite di memoria durante la rimozione da parte dell'ottimizzatore di clausole delle subquery non necessarie. (Bug 15875919) • Durante l'arresto, il server non ha potuto cercare di bloccare il mutex non inizializzato. (Bug 16016493) • Un'istruzione preparata che utilizza GROUP_CONCAT() e una clausola ORDER BY che ha denominato più colonne potrebbe provocare un'uscita del server. (Bug 16075310) • La strumentazione dello schema di prestazioni mancava per i thread lavoratore slave. (Bug 16083949) • `STOP SLAVE` potrebbe causare un deadlock se emesso contemporaneamente a un'istruzione come SHOW STATUS che ha recuperato i valori per una o più variabili di stato `Slave_retried_transactions`, `Slave_heartbeat_period`, `Slave_received_heartbeats`, `Slave_last_heartbeat` o `Slave_running`. (Bug 16088188)

Aggiornamento del motore di database	Version	Bug di MySQL corretti
		• Una query full-text che utilizza una modalità booleana potrebbe non mostrare risultati in alcuni casi, quando il termini di ricerca è una citazione. (Bug 16206253) • Il tentativo dell'ottimizzatore di rimuovere le clausole di subquery ridondanti ha generato un'asserzione durante l'esecuzione di un'istruzione preparata con una subquery nella clausola ON di un join di una subquery. (Bug 16318585) • GROUP_CONCAT instabile, arresto anomalo in ITEM_SUM::CLEAN_UP_AFTER_REMOVAL. (Bug 16347450) • Il tentativo di sostituire l'elenco stopword di una ricerca full-text InnoDB predefinito tramite la creazione di una tabella InnoDB con la stessa struttura di INFORMATION_SCHEMA.INNODB_FT_DEFAULT_STOPWORD genera un errore. (Bug 16373868) • Dopo che il thread del client in uno slave ha eseguito FLUSH TABLES WITH READ LOCK e dopo che sono stati eseguiti degli aggiornamenti nel master, lo slave resta in attesa durante l'esecuzione di `SHOW SLAVE STATUS`. (Bug 16387720) • Quando si esegue l'analisi di una stringa di ricerca delimitata, ad esempio "abc-def" in una ricerca full-text, InnoDB utilizza gli stessi delimitatori della parole di MyISAM. (Bug 16419661) • Arresto anomalo in FTS_AST_TERM_SET_WILDCARD. (Bug 16429306) • SEGFAULT in FTS_AST_VISIT() per il test FTS RQG. (Bug 16435855) • Per le build di debug, quanto l'ottimizzatore ha rimosso un elemento Item_ref che puntava a una subquery, ha provocato un'uscita del server. (Bug 16509874) • Le ricerche full-text nelle tabelle InnoDB non hanno dato un esito positivo nel caso di ricerche di citazioni con gli operatori + o -. (Bug 16516193)

Aggiornamento del motore di database	Version	Bug di MySQL corretti
		• `START SLAVE` fallito all'avvio del server con le opzioni `--master-info-repository=TABLE` relay-log-info-repository =TABLE e con autocommit impostato su 0, insieme a. `--skip-slave-start` (Bug 16533802) • I risultati di una ricerca full-text InnoDB estesa possono utilizzare una quantità di memoria eccessiva. (Bug 16625973) • Per le build di debug, si può verificare un'asserzione in OPT_CHECK_ORDER_BY quando si utilizza un elemento binario in una stringa di ricerca, poiché l'elemento binario potrebbe includere byte NULL e altri caratteri non significativi. (Bug 16766016) • Per alcune istruzioni, potrebbero verificarsi perdite di memoria durante la rimozione da parte dell'ottimizzatore di clausole delle subquery non necessarie. (Bug 16807641) • È stato possibile generare un deadlock dopo aver utilizzato FLUSH TABLES WITH READ LOCK tramite `STOP SLAVE` in una nuova connessione allo slave e successivamente un `SHOW SLAVE STATUS` usando la connessione originale. (Bug 16856735) • GROUP_CONCAT() con un separatore non valido può provocare un'uscita dal server. (Bug 16870783) • Il server ha fatto un blocco eccessivo sui mutex LOCK_active_mi e active_mi-> rli->data_lock per qualsiasi istruzione “pattern” SHOW STATUS LIKE, anche quando il pattern non corrispondeva alle variabili di stato che usano quei mutex (`Slave_heartbeat_period` , `Slave_last_heartbeat` , `Slave_received_heartbeats` , `Slave_retried_transactions` , `Slave_running`). (Bug 16904035) • Una ricerca full-text che utilizza un modificatore IN BOOLEAN MODE genera un errore di asserzione. (Bug 16927092)

Aggiornamento del motore di database	Version	Bug di MySQL corretti
		• Una ricerca full-text nelle tabelle InnoDB non ha generato risultati quando si è usato l'operatore booleano +. (Bug 17280122) • Deadlock in 4 modalità: zombie, svuotamento binlog, visualizzazione elenchi processi, visualizzazioni binlog. (Bug 17283409) • Quando un thread SQL in attesa di un blocco commit viene interrotto e riavviato, la transazione viene ignorata nello slave. (Bug 17450876) • Si verifica un errore relativo alla ricerca full-text InnoDB a causa di un token non completato. La stringa e la lunghezza della stringa devono essere trasmesse per consentire un confronto. (Bug 17659310) • Un numero elevato di tabelle InnoDB partizionate può utilizzar e molta più memoria se utilizzate in MySQL 5.6 o 5.7 rispetto alla memoria usata dalle stesse tabelle in versioni precedenti di MySQL Server. (Bug 17780517) • Per le query full-text, un valore num_token inferiore a quello di max_proximity_item può generare un'asserzione. (Bug 18233051) • Alcune query per le tabelle INFORMATION_SCHEMA TABLES e COLUMNS possono utilizzare troppa memoria nel caso di quantità elevate di tabelle InnoDB vuote. (Bug 18592390) • Quando si esegue una transazione, viene usato un flag per controllare che sia stato creato un thread e non si controlla direttamente il thread, perché in tal modo si usano meno risorse, in particolare quando si esegue il server con master_info_repository=TABLE. (Bug 18684222) • Se un thread del client in uno slave ha eseguito FLUSH TABLES WITH READ LOCK mentre il master ha eseguito un DML, l'esecuzione di `SHOW SLAVE STATUS` nello stesso

Aggiornamento del motore di database	Version	Bug di MySQL corretti
		client viene bloccato con un conseguente deadlock. (Bug 19843808) • Ordinare in base al risultato GROUP_CONCAT() può provocare un'uscita dal server. (Bug 19880368)
Aggiornamenti del motore di database Aurora MySQL 24/08/2015 (versione 1.1) (obsoleta)	1.1	• I database InnoDB con nomi che iniziano con un numero generano un errore del parser per le ricerche full-text. (Bug 17607956) • Errore nelle ricerche full-text InnoDB nei database i cui nomi iniziano con un numero. (Bug 17161372) • Per i database InnoDB su Windows, l'ID oggetto della ricerca full-text non è nel formato esadecimale previsto. (Bug 16559254) • Una regressione del codice introdotta in MySQL 5.6 influisce negativamente sulle prestazioni di DROP TABLE e ALTER TABLE. Questo può provocare un calo delle prestazioni tra MySQL Server 5.5.x e 5.6.x. (Bug 16864741)

Vulnerabilità di sicurezza risolte in Aurora MySQL

CVE (Common Vulnerabilities and Exposures - Vulnerabilità e rischi comuni) è un elenco di voci per vulnerabilità di sicurezza informatica pubblicamente note. Ogni voce contiene un numero di identificazione, una descrizione e almeno un riferimento pubblico.

In questa pagina è possibile trovare un elenco di vulnerabilità di sicurezza risolte in Aurora MySQL. Per informazioni generali sulla sicurezza di Aurora, consulta Sicurezza in Amazon Aurora nella Guida per l'utente di Amazon Aurora. Per ulteriori informazioni sulla sicurezza di Aurora MySQL, consulta Sicurezza con Amazon Aurora MySQL nella Guida per l'utente di Amazon Aurora.

Si consiglia di eseguire sempre l'aggiornamento alla versione Aurora più recente per essere protetti da vulnerabilità note. È possibile utilizzare questa pagina per verificare se una particolare versione di Aurora MySQL ha una correzione per una vulnerabilità di sicurezza specifica. Se il cluster non dispone della correzione per la protezione, è possibile visualizzare la versione Aurora MySQL a cui eseguire l'aggiornamento per tale correzione.

CVEs le correzioni corrette in Aurora MySQL versione 1, 2 e 3 sono elencate anche nelle note di rilascio per quella versione:

- Aggiornamenti del motore del database per Amazon Aurora MySQL versione 3
- Aggiornamenti del motore del database Amazon Aurora MySQL versione 2
- Aggiornamenti del motore del database per Amazon Aurora MySQL versione 1 (obsoleta)

> **Note**
>
> La versione iniziale di Aurora MySQL versione 3 include tutte le modifiche apportate alla community MySQL CVEs 8.0.23. Per le future CVEs che sono state risolte, cercale elencate qui e nelle note di rilascio di Aurora MySQL versione 3.

CVEs e versioni minime fisse di Aurora MySQL

- 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-2024-2105:2.12.3 , 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, 3,006,1 , 3,04,3, 2,12,6
- 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-2020-14812: 2.09.2, 2.07.4, 1.23.2, 1.22.4
- 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-2020-14567: 2.10.0, 2.09.1, 2.08.3, 2.07.3
- 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

Cronologia dei documenti delle note di rilascio di Aurora MySQL

La tabella seguente descrive i rilasci della documentazione per Aurora MySQL Release Notes.

Modifica	Descrizione	Data
Aurora MySQL versione 3.10.2, compatibile con MySQL 8.0.42	È disponibile la versione 3.10.2 di Aurora MySQL. Questa versione è compatibile con MySQL 8.0.42. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 2025-11-20 (versione 3.10.2, compatibile con MySQL 8.0.42).	20 novembre 2025
Aurora MySQL versione 3.11.0, compatibile con MySQL 8.0.43	È disponibile la versione 3.11.0 di Aurora MySQL. Questa versione è compatibile con MySQL 8.0.43. Per ulteriori informazioni, consulta Aggiornamenti del motore di database Aurora MySQL 2025-11-13 (versione 3.11.0, compatibile con MySQL 8.0.43).	13 novembre 2025
Aurora MySQL versione 3.10.1, compatibile con MySQL 8.0.42	È disponibile la versione 3.10.1 di Aurora MySQL. Questa versione è compatibile con MySQL 8.0.42. Per informazioni dettagliate, consultare Aurora MySQL	30 settembre 2025

	versione 3 compatibile con MySQL 8.0.	
Aurora MySQL versione 3.10.0, compatibile con MySQL 8.0.42	È disponibile la versione 3.10.0 di Aurora MySQL. Questa versione è compatibile con MySQL 8.0.42. Per informazioni dettagliate, consultare Aurora MySQL versione 3 compatibile con MySQL 8.0.	31 luglio 2025
Aurora MySQL versione 3.09.0, compatibile con MySQL 8.0.40	È disponibile la versione 3.09.0 di Aurora MySQL. Questa versione è compatibile con MySQL 8.0.40. Per informazioni dettagliate, consultare Aurora MySQL versione 3 compatibile con MySQL 8.0.	14 maggio 2025
Aurora MySQL versione 3.04.4, compatibile con MySQL 8.0.28	È disponibile la versione 3.04.4 di Aurora MySQL. Questa versione è compatibile con MySQL 8.0.28. Per informazioni dettagliate, consultare Aurora MySQL versione 3 compatibile con MySQL 8.0.	5 maggio 2025

Aurora MySQL versione 2.12.5, compatibile con MySQL 5.7.44	È disponibile la versione 2.12.5 di Aurora MySQL. Questa versione è compatibile con MySQL 5.7.44. Per informazioni dettagliate, consulta Aurora MySQL versione 2 compatibile con MySQL 5.7.	9 aprile 2025
Aurora MySQL versione 3.08.2, compatibile con MySQL 8.0.39	È disponibile la versione 3.08.2 di Aurora MySQL. Questa versione è compatibile con MySQL 8.0.39. Per informazioni dettagliate, consultare Aurora MySQL versione 3 compatibile con MySQL 8.0.	17 febbraio 2025
Aurora MySQL versione 3.08.1, compatibile con MySQL 8.0.39	È disponibile la versione 3.08.1 di Aurora MySQL. Questa versione è compatibile con MySQL 8.0.39. Per informazioni dettagliate, consultare Aurora MySQL versione 3 compatibile con MySQL 8.0.	17 febbraio 2025
Aurora MySQL versione 3.08.0, compatibile con MySQL 8.0.39	È disponibile la versione 3.08.0 di Aurora MySQL. Questa versione è compatibile con MySQL 8.0.39. Per informazioni dettagliate, consultare Aurora MySQL versione 3 compatibile con MySQL 8.0.	18 novembre 2024

Aurora MySQL versione 2.12.4, compatibile con MySQL 5.7.44	È disponibile la versione 2.12.4 di Aurora MySQL. Questa versione è compatibile con MySQL 5.7.44. Per informazioni dettagliate, consulta Aurora MySQL versione 2 compatibile con MySQL 5.7.	23 ottobre 2024
Aurora MySQL versione 3.07.1, compatibile con MySQL 8.0.36	È disponibile la versione 3.07.1 di Aurora MySQL. Questa versione è compatibile con MySQL 8.0.36. Per informazioni dettagliate, consultare Aurora MySQL versione 3 compatibile con MySQL 8.0.	23 luglio 2024
Aurora MySQL versione 2.11.6, compatibile con MySQL 5.7.12	È disponibile la versione 2.11.6 di Aurora MySQL. Questa versione è compatibile con MySQL 5.7.12. Per informazioni dettagliate, consulta Aurora MySQL versione 2 compatibile con MySQL 5.7.	19 luglio 2024
Aurora MySQL versione 2.12.3, compatibile con MySQL 5.7.44	È disponibile la versione 2.12.3 di Aurora MySQL. Questa versione è compatibile con MySQL 5.7.44. Per informazioni dettagliate, consulta Aurora MySQL versione 2 compatibile con MySQL 5.7.	9 luglio 2024

Aurora MySQL versione 3.06.1, compatibile con MySQL 8.0.34	È disponibile la versione 3.06.1 di Aurora MySQL. Questa versione è compatibile con MySQL 8.0.34. Per informazioni dettagliate, consultare Aurora MySQL versione 3 compatibile con MySQL 8.0.	26 giugno 2024
Aurora MySQL versione 3.04.3, compatibile con MySQL 8.0.28	È disponibile la versione 3.04.3 di Aurora MySQL. Questa versione è compatibile con MySQL 8.0.28. Per informazioni dettagliate, consultare Aurora MySQL versione 3 compatibile con MySQL 8.0.	26 giugno 2024
Aurora MySQL versione 3.07.0, compatibile con MySQL 8.0.36	È disponibile la versione 3.07.0 di Aurora MySQL. Questa versione è compatibile con MySQL 8.0.36. Per informazioni dettagliate, consultare Aurora MySQL versione 3 compatibile con MySQL 8.0.	4 giugno 2024
Aurora MySQL versione 2.11.5, compatibile con MySQL 5.7.12	È disponibile la versione 2.11.5 di Aurora MySQL. Questa versione è compatibile con MySQL 5.7.12. Per informazioni dettagliate, consulta Aurora MySQL versione 2 compatibile con MySQL 5.7.	26 marzo 2024

Aurora MySQL versione 2.12.2, compatibile con MySQL 5.7.44	È disponibile la versione 2.12.2 di Aurora MySQL. Questa versione è compatibile con MySQL 5.7.44. Per informazioni dettagliate, consulta Aurora MySQL versione 2 compatibile con MySQL 5.7.	19 marzo 2024
Aurora MySQL versione 3.04.2, compatibile con MySQL 8.0.28	È disponibile la versione 3.04.2 di Aurora MySQL. Questa versione è compatibile con MySQL 8.0.28. Per informazioni dettagliate, consultare Aurora MySQL versione 3 compatibile con MySQL 8.0.	15 marzo 2024
Aurora MySQL versione 3.06.0, compatibile con MySQL 8.0.34	È disponibile la versione 3.06.0 di Aurora MySQL. Questa versione è compatibile con MySQL 8.0.34. Per informazioni dettagliate, consultare Aurora MySQL versione 3 compatibile con MySQL 8.0.	7 marzo 2024
Aurora MySQL versione 3.05.2, compatibile con MySQL 8.0.32	È disponibile la versione 3.05.2 di Aurora MySQL. Questa versione è compatibile con MySQL 8.0.32. Per informazioni dettagliate, consultare Aurora MySQL versione 3 compatibile con MySQL 8.0.	31 gennaio 2024

Aurora MySQL versione 2.12.1, compatibile con MySQL 5.7.40	È disponibile Aurora MySQL versione 2.12.1. Questa versione è compatibile con MySQL 5.7.40. Per informazioni dettagliate, consulta Aurora MySQL versione 2 compatibile con MySQL 5.7.	28 dicembre 2023
Aurora MySQL versione 3.03.3, compatibile con MySQL 8.0.26	È disponibile Aurora MySQL versione 3.03.3. Questa versione è compatibile con MySQL 8.0.26. Per informazioni dettagliate, consultare Aurora MySQL versione 3 compatibile con MySQL 8.0.	8 dicembre 2023
Aurora MySQL versione 3.05.1, compatibile con MySQL 8.0.32	È disponibile Aurora MySQL versione 3.05.1. Questa versione è compatibile con MySQL 8.0.32. Per informazioni dettagliate, consultare Aurora MySQL versione 3 compatibile con MySQL 8.0.	21 novembre 2023
Aurora MySQL versione 3.04.1, compatibile con MySQL 8.0.28	È disponibile Aurora MySQL versione 3.04.1. Questa versione è compatibile con MySQL 8.0.28. Per informazioni dettagliate, consultare Aurora MySQL versione 3 compatibile con MySQL 8.0.	13 novembre 2023

Aurora MySQL versione 3.05.0.1, compatibile con MySQL 8.0.32, beta	È disponibile Aurora MySQL versione 3.05.0.1. Questa versione è compatibile con MySQL 8.0.32. Per informazioni dettagliate, consultare Aurora MySQL versione 3 compatibile con MySQL 8.0.	30 ottobre 2023
Aurora MySQL versione 3.05.0, compatibile con MySQL 8.0.32	È disponibile Aurora MySQL versione 3.05.0. Questa versione è compatibile con MySQL 8.0.32. Per informazioni dettagliate, consultare Aurora MySQL versione 3 compatibile con MySQL 8.0.	25 ottobre 2023
Aurora MySQL versione 2.12.0.1, compatibile con MySQL 5.7.40, beta	È disponibile Aurora MySQL versione 2.12.0.1 beta. Questa versione è compatibile con MySQL 5.7.40. Per informazioni dettagliate, consulta Aurora MySQL versione 2 compatibile con MySQL 5.7.	25 ottobre 2023
Aurora MySQL versione 2.11.4, compatibile con MySQL 5.7.12	È disponibile Aurora MySQL versione 2.11.4. Questa versione è compatibile con MySQL 5.7.12. Per informazioni dettagliate, consulta Aurora MySQL versione 2 compatibile con MySQL 5.7.	17 ottobre 2023

Aurora MySQL versione 3.03.2, compatibile con MySQL 8.0.26	È disponibile Aurora MySQL versione 3.03.2. Questa versione è compatibile con MySQL 8.0.26. Per informazioni dettagliate, consultare Aurora MySQL versione 3 compatibile con MySQL 8.0.	29 agosto 2023
Aurora MySQL versione 2.07.10, compatibile con MySQL 5.7.12	È disponibile Aurora MySQL versione 2.07.10. Questa versione è compatibile con MySQL 5.7.12. Per informazioni dettagliate, consulta Aurora MySQL versione 2 compatibile con MySQL 5.7.	15 agosto 2023
Aurora MySQL versione 3.04.0, compatibile con MySQL 8.0.28	È disponibile Aurora MySQL versione 3.04.0. Questa versione è compatibile con MySQL 8.0.28. Per informazioni dettagliate, consultare Aurora MySQL versione 3 compatibile con MySQL 8.0.	31 luglio 2023
Aurora MySQL versione 2.12.0, compatibile con MySQL 5.7.40	È disponibile Aurora MySQL versione 2.12.0. Questa versione è compatibile con MySQL 5.7.40. Per informazioni dettagliate, consulta Aurora MySQL versione 2 compatibile con MySQL 5.7.	25 luglio 2023

Aurora MySQL versione 2.11.3, compatibile con MySQL 5.7.12	È disponibile Aurora MySQL versione 2.11.3. Questa versione è compatibile con MySQL 5.7.12. Per informazioni dettagliate, consulta Aurora MySQL versione 2 compatibile con MySQL 5.7.	9 giugno 2023
Aurora MySQL versione 3.03.1, compatibile con MySQL 8.0.26	È disponibile Aurora MySQL versione 3.03.1. Questa versione è compatibile con MySQL 8.0.26. Per informazioni dettagliate, consultare Aurora MySQL versione 3 compatibile con MySQL 8.0.	11 maggio 2023
Aurora MySQL versione 2.07.9, compatibile con MySQL 5.7.12.	È disponibile Aurora MySQL versione 2.07.9. Questa versione è compatibile con MySQL 5.7.12. Per informazioni dettagliate, consulta Aurora MySQL versione 2 compatibile con MySQL 5.7.	4 maggio 2023
Aurora MySQL versione 3.02.3, compatibile con MySQL 8.0.23	È disponibile Aurora MySQL versione 3.02.3. Questa versione è compatibile con MySQL 8.0.23. Per informazioni dettagliate, consultare Aurora MySQL versione 3 compatibile con MySQL 8.0.	17 aprile 2023

Aurora MySQL versione 2.11.2, compatibile con MySQL 5.7.12	È disponibile Aurora MySQL versione 2.11.2. Questa versione è compatibile con MySQL 5.7.12. Per informazioni dettagliate, consulta Aurora MySQL versione 2 compatibile con MySQL 5.7.	24 marzo 2023
Aurora MySQL versione 3.03.0, compatibile con MySQL 8.0.26	È disponibile Aurora MySQL versione 3.03.0. Questa versione è compatibile con MySQL 8.0.26. Per informazioni dettagliate, consultare Aurora MySQL versione 3 compatibile con MySQL 8.0.	1 marzo 2023
Aurora MySQL versione 2.11.1, compatibile con MySQL 5.7.12.	È disponibile Aurora MySQL versione 2.11.1. Questa versione è compatibile con MySQL 5.7.12. Per informazioni dettagliate, consulta Aurora MySQL versione 2 compatibile con MySQL 5.7.	14 febbraio 2023
Aurora MySQL versione 3.02.2, compatibile con MySQL 8.0.23	È disponibile Aurora MySQL versione 3.02.2. Questa versione è compatibile con MySQL 8.0.23. Per informazioni dettagliate, consultare Aurora MySQL versione 3 compatibile con MySQL 8.0.	18 novembre 2022

Aurora MySQL versione 2.10.3, compatibile con MySQL 5.7	È disponibile Aurora MySQL versione 2.10.3. Questa versione è compatibile con MySQL 5.7. Per informazioni dettagliate, consulta Aurora MySQL versione 2 compatibile con MySQL 5.7.	1 novembre 2022
Aurora MySQL versione 2.11.0, compatibile con MySQL 5.7.12	È disponibile Aurora MySQL versione 2.11.0. Questa versione è compatibile con MySQL 5.7.12. Per informazioni dettagliate, consulta Aurora MySQL versione 2 compatibile con MySQL 5.7.	25 ottobre 2022
Aurora MySQL versione 3.02.1, compatibile con MySQL 8.0.23	È disponibile Aurora MySQL versione 3.02.1. Questa versione è compatibile con MySQL 8.0.23. Per informazioni dettagliate, consultare Aurora MySQL versione 3 compatibile con MySQL 8.0.	7 settembre 2022
Aggiornamento in loco per Aurora Serverless v1 compatibile con MySQL 5.6	È possibile eseguire un aggiornamento in loco per un cluster Aurora Serverless v1 compatibile con MySQL 5.6 per modificare un cluster esistente in un cluster Aurora Serverless v1 compatibile con MySQL 5.7. Per ulteriori informazioni, consulta Aurora MySQL Serverless 5.7 engine updates 2020-06-18 (version 2.07.1).	16 giugno 2022

Aggiornamento in loco per Aurora Serverless v1 compatibile con MySQL 5.6	È possibile eseguire un aggiornamento in loco per un cluster Aurora Serverless v1 compatibile con MySQL 5.6 per modificare un cluster esistente in un cluster Aurora Serverless v1 compatibile con MySQL 5.7. Per ulteriori informazioni, consulta Aurora MySQL Serverless 5.7 engine updates 2020-06-18 (version 2.07.1).	16 giugno 2022
Sono disponibili aggiornamenti del motore di database Aurora MySQL del 16/06/2022 (versione 2.07.8).	È disponibile Aurora MySQL versione 2.07.8.	16 giugno 2022
Aurora MySQL versione 3.02.0, compatibile con MySQL 8.0.23	È disponibile Aurora MySQL versione 3.02.0. Questa versione è compatibile con MySQL 8.0.23. Per informazioni dettagliate, consultare Aurora MySQL versione 3 compatibile con MySQL 8.0.	20 aprile 2022
Aurora MySQL versione 3.01.1, compatibile con MySQL 8.0.23	È disponibile Aurora MySQL versione 3.01.1. Questa versione è compatibile con MySQL 8.0.23. Per informazioni dettagliate, consultare Aurora MySQL versione 3 compatibile con MySQL 8.0.	15 aprile 2022
Versione iniziale	Versione iniziale di Aurora MySQL Release Notes.	22 marzo 2022

Aurora MySQL versione 2.10.2	Aurora MySQL versione 2.10.2 è disponibile.	26 gennaio 2022
Aurora MySQL versione 2.08.4	Aurora MySQL versione 2.08.4 è disponibile.	6 gennaio 2022
Aurora MySQL versione 2.07.7	Aurora MySQL versione 2.07.7 è disponibile.	24 novembre 2021
Aurora MySQL versione 3.01.0, compatibile con MySQL 8.0.23	Aurora MySQL versione 3.01.0 è disponibile. Questa versione è compatibile con MySQL 8.0.23. Per informazioni dettagliate, consultare Aurora MySQL versione 3 compatibile con MySQL 8.0.	18 novembre 2021
Aurora MySQL versione 2.09.3	Aurora MySQL versione 2.09.3 è disponibile.	12 novembre 2021
Aurora MySQL versione 2.10.1	Aurora MySQL versione 2.10.1 è disponibile.	21 ottobre 2021
Aurora MySQL versione 1.23.4	Aurora MySQL versione 1.23.4 è disponibile.	30 settembre 2021
Aurora MySQL versione 2.07.6	Aurora MySQL versione 2.07.6 è disponibile.	2 settembre 2021
Aurora MySQL versione 2.07.5	Aurora MySQL versione 2.07.5 è disponibile.	6 luglio 2021
Aurora MySQL versione 1.23.3	Aurora MySQL versione 1.23.3 è disponibile.	28 giugno 2021
Versione 1.22.5 Aurora MySQL	La versione 1.22.5 Aurora MySQL è disponibile.	3 giugno 2021

Aurora MySQL versione 2.10.0	Aurora MySQL versione 2.10.0 è disponibile. Alcuni dei punti salienti includono una maggiore disponibilità delle istanze di lettura durante il riavvio del writer, miglioramenti al patching senza interruzioni(ZDP), miglioramenti al riavvio senza interruzioni (ZDR) e l'ottimizzazione della cache binlog. I/O	25 maggio 2021
Aurora MySQL versione 1.23.2	Aurora MySQL versione 1.23.2 è ora disponibile.	18 marzo 2021
Aurora MySQL versione 2.07.4	Aurora MySQL versione 2.07.4 è ora disponibile.	4 marzo 2021
Aurora MySQL versione 1.22.4	Aurora MySQL versione 1.22.4 è ora disponibile.	04 marzo 2021
Aurora MySQL versione 2.09.2	Aurora MySQL versione 2.09.2 è ora disponibile.	26 febbraio 2021
Aurora MySQL versione 2.09.1	Aurora MySQL versione 2.09.1 è disponibile.	11 dicembre 2020
Aurora MySQL versione 1.23.1	Aurora MySQL versione 1.23.1 è disponibile.	24 novembre 2020
Aurora MySQL versione 2.08.3	Aurora MySQL versione 2.08.3 è disponibile.	12 Novembre 2020
Aurora MySQL versione 2.07.3	Aurora MySQL versione 2.07.3 è disponibile.	10 Novembre 2020
Aurora MySQL versione 1.22.3	Aurora MySQL versione 1.22.3 è disponibile.	9 novembre 2020

Aurora MySQL versione 2.09.0	Aurora MySQL versione 2.09.0 è disponibile.	17 settembre 2020
Aurora MySQL versione 1.23.0	Aurora MySQL versione 1.23.0 è disponibile.	2 settembre 2020
Aurora MySQL versione 2.08.2	Aurora MySQL versione 2.08.2 è disponibile.	28 agosto 2020
Aurora MySQL versione 2.04.9	Aurora MySQL versione 2.04.9 è disponibile.	14 agosto 2020
Aurora MySQL versione 2.08.1	Aurora MySQL versione 2.08.1 è disponibile.	18 giugno 2020
Aurora MySQL versione 1.22.2 per cluster di query paralleli	Aurora MySQL versione 1.22.2 è disponibile quando si crea un cluster di query paralleli.	18 giugno 2020
Aurora MySQL versione 1.20.1 per cluster di query paralleli	Aurora MySQL versione 1.20.1 è disponibile quando si crea un cluster di query paralleli.	11 giugno 2020
Versione Aurora MySQL 2.08.0	Versione Aurora MySQL 2.08.0 è disponibile.	2 giugno 2020
Aurora MySQL versione 1.19.6 per cluster di query paralleli	Aurora MySQL versione 1.19.6 è disponibile quando si crea un cluster di query paralleli.	2 giugno 2020
Aurora MySQL versione 2.07.2	Aurora MySQL versione 2.07.2 è disponibile.	17 aprile 2020
Aurora MySQL versione 1.22.2	Aurora MySQL versione 1.22.2 è disponibile.	5 marzo 2020
Aurora MySQL versione 1.20.1	Aurora MySQL versione 1.20.1 è disponibile.	5 marzo 2020

Aurora MySQL versione 1.19.6	Aurora MySQL versione 1.19.6 è disponibile.	5 marzo 2020
Aurora MySQL versione 1.17.9	Aurora MySQL versione 1.17.9 è disponibile.	5 marzo 2020
Aurora MySQL versione 2.07.1	Aurora MySQL versione 2.07.1 è disponibile.	23 dicembre 2019
Aurora MySQL versione 1.22.1	Aurora MySQL versione 1.22.1 è disponibile.	23 dicembre 2019
Aurora MySQL versione 2.07.0	Aurora MySQL versione 2.07.0 è disponibile.	25 novembre 2019
Aurora MySQL versione 1.22.0	Aurora MySQL versione 1.22.0 è disponibile.	25 novembre 2019
Aurora MySQL versione 1.21.0	Aurora MySQL versione 1.21.0 è disponibile.	25 novembre 2019
Aurora MySQL versione 2.06.0	Aurora MySQL versione 2.06.0 è disponibile.	22 novembre 2019
Aurora MySQL versione 2.04.8	Aurora MySQL versione 2.04.8 è disponibile.	20 novembre 2019
Aurora MySQL versione 2.04.7	Aurora MySQL versione 2.04.7 è disponibile.	14 novembre 2019
Aurora MySQL versione 2.05.0	Aurora MySQL versione 2.05.0 è disponibile.	11 novembre 2019
Aurora MySQL versione 1.20.0	Aurora MySQL versione 1.20.0 è disponibile.	11 novembre 2019
Aurora MySQL versione 2.04.6	Aurora MySQL versione 2.04.6 è disponibile.	19 settembre 2019

Aurora MySQL versione 1.19.5	Aurora MySQL versione 1.19.5 è disponibile.	19 settembre 2019
Aurora MySQL versione 2.04.5	Aurora MySQL versione 2.04.5 è disponibile.	8 luglio 2019
Aurora MySQL versione 1.19.2	Aurora MySQL versione 1.19.2 è disponibile.	5 giugno 2019
Aurora MySQL versione 2.04.4	Aurora MySQL versione 2.04.4 è disponibile.	29 maggio 2019
Aurora MySQL versione 2.04.3	Aurora MySQL versione 2.04.3 è disponibile.	9 maggio 2019
Aurora MySQL versione 1.19.1	Aurora MySQL versione 1.19.1 è disponibile.	9 maggio 2019
Aurora MySQL versione 2.04.2	Aurora MySQL versione 2.04.2 è disponibile.	2 maggio 2019
Aurora MySQL versione 2.04.1	Aurora MySQL versione 2.04.1 è disponibile.	25 marzo 2019
Aurora MySQL versione 2.04	Aurora MySQL versione 2.04 è disponibile.	25 marzo 2019
Aurora MySQL versione 2.03.4	Aurora MySQL versione 2.03.4 è disponibile.	7 febbraio 2019
Aurora MySQL versione 1.19.0	Aurora MySQL versione 1.19.0 è disponibile.	7 febbraio 2019
Aurora MySQL versione 2.03.3	Aurora MySQL versione 2.03.3 è disponibile.	18 gennaio 2019
Aurora MySQL versione 1.17.8	Aurora MySQL versione 1.17.8 è disponibile.	17 gennaio 2019

Aurora MySQL versione 2.03.2	Aurora MySQL versione 2.03.2 è disponibile.	9 gennaio 2019
Aurora MySQL versione 2.03.1	Aurora MySQL versione 2.03.1 è disponibile.	24 ottobre 2018
Aurora MySQL versione 2.03	Aurora MySQL versione 2.03 è disponibile.	11 ottobre 2018
Aurora MySQL versione 2.02.5	Aurora MySQL versione 2.02.5 è disponibile.	8 Ottobre 2018
Aurora MySQL versione 1.17.7	Aurora MySQL versione 1.17.7 è disponibile.	8 Ottobre 2018
Aurora MySQL versione 2.02.4	Aurora MySQL versione 2.02.4 è disponibile.	21 settembre 2018
Aurora MySQL versione 1.18.0	Aurora MySQL versione 1.18.0 è disponibile.	20 settembre 2018
Aurora MySQL versione 1.17.6	Aurora MySQL versione 1.17.6 è disponibile.	6 settembre 2018

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.