



Notas de lançamento do Aurora MySQL

Amazon Aurora



Amazon Aurora: Notas de lançamento do Aurora MySQL

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

As marcas comerciais e imagens de marcas da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestige a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

Notas de lançamento do Aurora MySQL	1
Calendários de lançamento do Aurora MySQL	2
Calendário de lançamento das versões principais do Aurora MySQL	2
Calendário de lançamento das versões secundárias do Aurora MySQL	4
Aurora MySQL versão 3	6
Atualizações do Aurora MySQL: 2026-02-17 (versão 3.12.0, compatível com o MySQL 8.0.44)	8
Melhorias	9
Integração de correções de bug do MySQL Community Edition	12
Atualizações do Aurora MySQL: 2025-12-16 (versão 3.11.1, compatível com o MySQL 8.0.43)	13
Melhorias	13
Atualizações do Aurora MySQL: 2025-11-13 (versão 3.11.0, compatível com o MySQL 8.0.43)	14
Melhorias	15
Integração de correções de bug do MySQL Community Edition	19
Atualizações do Aurora MySQL: 2026-01-02 (versão 3.10.3, compatível com o MySQL 8.0.42)	19
Melhorias	20
Atualizações do Aurora MySQL: 2025-11-20 (versão 3.10.2, compatível com o MySQL 8.0.42)	20
Melhorias	21
Atualizações do Aurora MySQL: 2025-09-30 (versão 3.10.1, compatível com MySQL 8.0.42)	22
Melhorias	23
Integração de correções de bug do MySQL Community Edition	25
Atualizações do Aurora MySQL: 2025-07-31 (versão 3.10.0, compatível com MySQL 8.0.42)	25
Novos atributos	27
Melhorias	27
Integração de correções de bug do MySQL Community Edition	30
Atualizações do Aurora MySQL: 2025-05-14 (versão 3.09.0, compatível com MySQL 8.0.40)	30
Destaques do lançamento	31
Melhorias	31
Integração de correções de bug do MySQL Community Edition	35
Atualizações do Aurora MySQL: 2025-04-07 (versão 3.08.2, compatível com MySQL 8.0.39)	35

Melhorias	36
Integração de correções de bug do MySQL Community Edition	37
Atualizações do Aurora MySQL: 2025-02-17 (versão 3.08.1, compatível com MySQL 8.0.39)	37
Melhorias	36
Integração de correções de bug do MySQL Community Edition	37
Atualizações do Aurora MySQL: 2024-11-18 (versão 3.08.0, compatível com MySQL 8.0.39)	39
Novos atributos	39
Melhorias	41
Integração de correções de bug do MySQL Community Edition	47
Atualizações do Aurora MySQL: 2024-07-23 (versão 3.07.1) (Obsoleto)	47
Melhorias	48
Integração de correções de bug do MySQL Community Edition	48
Atualizações do Aurora MySQL: 2024-06-04 (versão 3.07.0) (Obsoleto)	48
Melhorias	49
Integração de correções de bug do MySQL Community Edition	53
Atualizações do Aurora MySQL: 2024-06-26 (versão 3.06.1) (Obsoleto)	54
Melhorias	55
Integração de correções de bug do MySQL Community Edition	57
Atualizações do Aurora MySQL: 2024-03-07 (versão 3.06.0) (Obsoleto)	57
Novos recursos	58
Melhorias	55
Integração de correções de bug do MySQL Community Edition	57
Atualizações do Aurora MySQL: 2024-01-31 (versão 3.05.2) (Obsoleto)	63
Melhorias	64
Integração de correções de bug do MySQL Community Edition	65
Atualizações do Aurora MySQL: 2023-11-21 (versão 3.05.1) (Obsoleto)	66
Melhorias	66
Integração de correções de bug do MySQL Community Edition	67
Atualizações do Aurora MySQL: 2023-10-30 (versão 3.05.0.1) (Obsoleto)	67
Melhorias	68
Atualizações do Aurora MySQL: 2023-10-25 (versão 3.05.0) (Obsoleto)	68
Melhorias	69
Integração de correções de bug do MySQL Community Edition	73
Atualizações do Aurora MySQL: 2026-01-02 (versão 3.04.6, compatível com o MySQL 8.0.28)	73
Melhorias	74

Integração de correções de bug do MySQL Community Edition	75
Atualizações do Aurora MySQL: 2025-05-05 (versão 3.04.4, compatível com MySQL 8.0.28)	75
Melhorias	76
Integração de correções de bug do MySQL Community Edition	78
Atualizações do Aurora MySQL: 2024-06-26 (versão 3.04.3, compatível com o MySQL 8.0.28)	78
Melhorias	79
Integração de correções de bug do MySQL Community Edition	81
Atualizações do Aurora MySQL: 2024-03-15 (versão 3.04.2, compatível com o MySQL 8.0.28)	82
Melhorias	83
Integração de correções de bug do MySQL Community Edition	85
Atualizações do Aurora MySQL: 2023-11-13 (versão 3.04.1, compatível com o MySQL 8.0.28)	85
Melhorias	87
Integração de correções de bug do MySQL Community Edition	88
Atualizações do Aurora MySQL: 2023-07-31 (versão 3.04.0, compatível com o MySQL 8.0.28)	89
Melhorias	90
Integração de correções de bug do MySQL Community Edition	96
Atualizações do Aurora MySQL: 2023-12-08 (versão 3.03.3) (obsoleta)	97
Melhorias	97
Integração de correções de bug do MySQL Community Edition	99
Atualizações do Aurora MySQL: 2023-08-29 (versão 3.03.2) (obsoleta)	99
Melhorias	100
Integração de correções de bug do MySQL Community Edition	102
Atualizações do Aurora MySQL: 2023-05-11 (versão 3.03.1) (obsoleta)	102
Melhorias	103
Integração de correções de bug do MySQL Community Edition	105
Atualizações do Aurora MySQL: 2023-03-01 (versão 3.03.0) (obsoleta)	105
Melhorias	106
Integração de correções de bug do MySQL Community Edition	109
Atualizações do Aurora MySQL: 2023-04-17 (versão 3.02.3) (obsoleta)	110
Melhorias	111
Atualizações do Aurora MySQL: 2022-11-18 (versão 3.02.2) (obsoleta)	112
Melhorias	112

Integração de correções de bug do MySQL Community Edition	109
Atualizações do Aurora MySQL: 2022-09-07 (versão 3.02.1) (obsoleta)	115
Melhorias	116
Atualizações do Aurora MySQL: 2022-04-20 (versão 3.02.0) (obsoleta)	117
Melhorias	118
Integração de correções de bug da edição MySQL community	120
Atualizações do Aurora MySQL: 2022-04-15 (versão 3.01.1) (obsoleta)	121
Melhorias	122
Integração de correções de bug da edição MySQL community	123
Atualizações do Aurora MySQL: 2021-11-18 (versão 3.01.0) (obsoleta)	124
Melhorias	125
Aurora MySQL versão 2	126
Atualizações do Aurora MySQL de 2025-04-09 (versão 2.12.5, compatível com MySQL 5.7.44): versão do Suporte estendido do RDS	130
Melhorias	131
Atualizações do Aurora MySQL de 2024-10-23 (versão 2.12.4, compatível com MySQL 5.7.44): versão do Suporte estendido do RDS	131
Melhorias	131
Integração de correções de bug do MySQL Community Edition	133
Recursos não compatíveis com o Aurora MySQL versão 2	133
Compatibilidade com o MySQL 5.7	133
Atualizações do Aurora MySQL de 2024-07-09 (versão 2.12.3, compatível com MySQL 5.7.44): versão do Suporte estendido do RDS	134
Melhorias	134
Integração de correções de bug do MySQL Community Edition	137
Recursos não compatíveis com o Aurora MySQL versão 2	137
Compatibilidade com o MySQL 5.7	138
Atualizações do Aurora MySQL de 2024-03-19 (versão 2.12.2, compatível com MySQL 5.7.44): versão do Suporte estendido do RDS	138
Melhorias	139
Integração de correções de bug do MySQL Community Edition	140
Recursos não compatíveis com o Aurora MySQL versão 2	140
Compatibilidade com o MySQL 5.7	141
Atualizações do Aurora MySQL de 2023-12-28 (versão 2.12.1, compatível com MySQL 5.7.40): versão do Suporte estendido do RDS	141
Melhorias	142

Integração de correções de bug do MySQL Community Edition	145
Recursos não compatíveis com o Aurora MySQL versão 2	145
Compatibilidade com o MySQL 5.7	145
Atualizações do Aurora MySQL de 2023-10-25 (versão 2.12.0.1, compatível com MySQL 5.7.40): versão do Suporte estendido do RDS (Beta)	146
Melhorias	146
Atualizações do Aurora MySQL de 2023-07-25 (versão 2.12.0, compatível com MySQL 5.7.40): versão do Suporte estendido do RDS	147
Melhorias	148
Integração de correções de bug do MySQL Community Edition	151
Recursos não compatíveis com o Aurora MySQL versão 2	151
Compatibilidade com o MySQL 5.7	151
Atualizações do Aurora MySQL de 2024-07-19 (versão 2.11.6, compatível com MySQL 5.7.12): versão do Suporte estendido do RDS	152
Melhorias	153
Integração de correções de bug do MySQL Community Edition	155
Recursos não compatíveis com o Aurora MySQL versão 2	155
Compatibilidade com o MySQL 5.7	155
Atualizações do Aurora MySQL: 2024-03-26 (versão 2.11.5, compatível com MySQL 5.7.12): versão do Suporte estendido do RDS (padrão)	156
Melhorias	157
Integração de correções de bug do MySQL Community Edition	158
Recursos não compatíveis com o Aurora MySQL versão 2	159
Compatibilidade com o MySQL 5.7	159
Atualizações do Aurora MySQL de 2023-10-17 (versão 2.11.4, compatível com MySQL 5.7.12): versão do Suporte estendido do RDS	159
Melhorias	161
Integração de correções de bug do MySQL Community Edition	163
Recursos não compatíveis com o Aurora MySQL versão 2	163
Compatibilidade com o MySQL 5.7	163
Atualizações do Aurora MySQL de 2023-06-09 (versão 2.11.3, compatível com MySQL 5.7.12): versão do Suporte estendido do RDS	164
Melhorias	165
Recursos não compatíveis com o Aurora MySQL versão 2	166
Compatibilidade com o MySQL 5.7	167

Atualizações do Aurora MySQL de 2023-03-24 (versão 2.11.2, compatível com MySQL 5.7.12):	
versão do Suporte estendido do RDS	167
Melhorias	168
Recursos não compatíveis com o Aurora MySQL versão 2	169
Compatibilidade com o MySQL 5.7	169
Atualizações do Aurora MySQL de 2023-02-14 (versão 2.11.1, compatível com MySQL 5.7.12):	
versão do Suporte estendido do RDS	170
Melhorias	171
Comparação com o Aurora MySQL versão 1	172
Compatibilidade com o MySQL 5.7	173
Atualizações do Aurora MySQL de 2022-10-25 (versão 2.11.0, compatível com MySQL 5.7.12):	
versão do Suporte estendido do RDS	173
Melhorias	174
Integração de correções de bug do MySQL Community Edition	178
Comparação com o Aurora MySQL versão 1	180
Compatibilidade com o MySQL 5.7	181
Atualizações do Aurora MySQL: 2022-11-01 (versão 2.10.3) (obsoleta)	
Melhorias	182
Integração de correções de bug do MySQL Community Edition	183
Comparação com o Aurora MySQL versão 1	184
Compatibilidade com o MySQL 5.7	184
Atualizações do Aurora MySQL: 2022-01-26 (versão 2.10.2) (obsoleta)	
Melhorias	186
Integração de correções de bug do MySQL Community Edition	109
Comparação com o Aurora MySQL versão 1	190
Compatibilidade com o MySQL 5.7	190
Atualizações do Aurora MySQL: 2021-10-21 (versão 2.10.1) (obsoleta)	
Melhorias	191
Integração de correções de bug da edição MySQL community	193
Comparação com o Aurora MySQL versão 1	193
Compatibilidade com o MySQL 5.7	193
Atualizações do Aurora MySQL: 2021-05-25 (versão 2.10.0) (obsoleta)	
Melhorias	195
Integração de correções de bug da edição MySQL community	198
Comparação com o Aurora MySQL versão 1	202
Compatibilidade com o MySQL 5.7	202

Atualizações do Aurora MySQL: 2021-11-12 (versão 2.09.3) (obsoleta)	203
Melhorias	204
Integração de correções de bug da edição MySQL community	206
Comparação com o Aurora MySQL versão 1	207
Compatibilidade com o MySQL 5.7	207
Atualizações do Aurora MySQL: 2021-02-26 (versão 2.09.2) (obsoleta)	208
Melhorias	209
Comparação com o Aurora MySQL versão 1	210
Compatibilidade com o MySQL 5.7	211
Atualizações do Aurora MySQL: 2020-12-11 (versão 2.09.1) (obsoleta)	211
Melhorias	212
Integração de correções de bug da edição MySQL community	213
Comparação com o Aurora MySQL versão 1	214
Compatibilidade com o MySQL 5.7	214
Atualizações do Aurora MySQL: 2020-09-17 (versão 2.09.0) (obsoleta)	215
Melhorias	216
Integração de correções de bug da edição MySQL community	221
Comparação com o Aurora MySQL versão 1	222
Compatibilidade com o MySQL 5.7	223
Atualizações do Aurora MySQL: 2022-01-06 (versão 2.08.4) (obsoleta)	224
Melhorias	224
Comparação com o Aurora MySQL versão 1	225
Compatibilidade com o MySQL 5.7	225
Atualizações do Aurora MySQL: 2020-11-12 (versão 2.08.3) (obsoleta)	226
Melhorias	227
Integração de correções de bug da edição MySQL community	228
Comparação com o Aurora MySQL versão 1	228
Compatibilidade com o MySQL 5.7	228
Atualizações do Aurora MySQL: 2020-08-28 (versão 2.08.2) (obsoleta)	229
Melhorias	230
Comparação com o Aurora MySQL versão 1	230
Compatibilidade com o MySQL 5.7	231
Atualizações do Aurora MySQL: 2020-06-18 (versão 2.08.1) (obsoleta)	231
Melhorias	232
Comparação com o Aurora MySQL versão 1	233
Compatibilidade com o MySQL 5.7	233

Atualizações do Aurora MySQL: 2020-06-02 (versão 2.08.0) (obsoleta)	234
Melhorias	235
Integração de correções de bug da edição MySQL community	238
Comparação com o Aurora MySQL versão 1	238
Compatibilidade com o MySQL 5.7	239
Atualizações do Aurora MySQL: 2023-08-15 (versão 2.07.10) (Obsoleto)	239
Melhorias	240
Recursos não compatíveis com o Aurora MySQL versão 2	241
Compatibilidade com o MySQL 5.7	242
Atualizações do Aurora MySQL: 2023-05-04 (versão 2.07.9) (Obsoleto)	242
Melhorias	243
Recursos não compatíveis com o Aurora MySQL versão 2	244
Compatibilidade com o MySQL 5.7	244
Atualizações do Aurora MySQL: 2022-06-16 (versão 2.07.8) (obsoleta)	245
Melhorias	246
Integração de correções de bug da edição MySQL community	246
Comparação com o Aurora MySQL versão 1	246
Compatibilidade com o MySQL 5.7	247
Atualizações do Aurora MySQL: 2021-11-24 (versão 2.07.7) (obsoleta)	248
Melhorias	248
Comparação com o Aurora MySQL versão 1	249
Compatibilidade com o MySQL 5.7	250
Atualizações do Aurora MySQL: 2021-09-02 (versão 2.07.6) (obsoleta)	251
Integração de correções de bug da edição MySQL community	251
Comparação com o Aurora MySQL versão 1	251
Compatibilidade com o MySQL 5.7	252
Atualizações do Aurora MySQL: 2021-07-06 (versão 2.07.5) (obsoleta)	253
Melhorias	253
Comparação com o Aurora MySQL versão 1	254
Compatibilidade com o MySQL 5.7	254
Atualizações do Aurora MySQL: 2021-03-04 (versão 2.07.4) (obsoleta)	255
Melhorias	256
Integração de correções de bug da edição MySQL community	257
Comparação com o Aurora MySQL versão 1	257
Compatibilidade com o MySQL 5.7	257
Atualizações do Aurora MySQL: 2020-11-10 (versão 2.07.3) (obsoleta)	258

Melhorias	259
Integração de correções de bug da edição MySQL community	260
Comparação com o Aurora MySQL versão 1	262
Compatibilidade com o MySQL 5.7	262
Atualizações do Aurora MySQL: 2020-04-17 (versão 2.07.2) (obsoleta)	263
Melhorias	264
Integração de correções de bug da edição MySQL community	264
Comparação com o Aurora MySQL versão 1	265
Compatibilidade com o MySQL 5.7	265
Atualizações do Aurora MySQL: 2019-12-23 (versão 2.07.1) (obsoleta)	266
Melhorias	267
Comparação com o Aurora MySQL versão 1	267
Compatibilidade com o MySQL 5.7	268
Atualizações do Aurora MySQL: 2019-11-25 (versão 2.07.0) (obsoleta)	268
Melhorias	269
Integração de correções de bug da edição MySQL community	271
Comparação com o Aurora MySQL versão 1	271
Compatibilidade com o MySQL 5.7	271
Atualizações do Aurora MySQL: 2019-11-22 (versão 2.06.0) (obsoleta)	272
Melhorias	273
Comparação com o Aurora MySQL versão 1	276
Compatibilidade com o MySQL 5.7	276
Atualizações do Aurora MySQL: 2019-11-11 (versão 2.05.0) (obsoleta)	277
Melhorias	278
Integração de correções de bugs do MySQL	279
Comparação com o Aurora MySQL versão 1	279
Compatibilidade com o MySQL 5.7	279
Atualizações do Aurora MySQL: 2020-08-14 (versão 2.04.9) (obsoleta)	280
Melhorias	281
Integração de correções de bugs do MySQL	283
Comparação com o Aurora MySQL versão 1	284
Compatibilidade com o MySQL 5.7	285
Atualizações do Aurora MySQL: 2019-11-20 (versão 2.04.8) (obsoleta)	286
Melhorias	287
Comparação com o Aurora MySQL versão 1	287
Compatibilidade com o MySQL 5.7	288

Atualizações do Aurora MySQL: 2019-11-14 (versão 2.04.7) (obsoleta)	288
Melhorias	289
Comparação com o Aurora MySQL versão 1	290
Compatibilidade com o MySQL 5.7	291
Atualizações do Aurora MySQL: 2019-09-19 (versão 2.04.6) (obsoleta)	291
Melhorias	292
Integração de correções de bugs do MySQL	293
Comparação com o Aurora MySQL versão 1	293
Compatibilidade com o MySQL 5.7	293
Atualizações do Aurora MySQL: 2019-07-08 (versão 2.04.5) (obsoleta)	294
Melhorias	295
Comparação com o Aurora MySQL versão 1	296
Compatibilidade com o MySQL 5.7	296
Atualizações do Aurora MySQL: 2019-05-29 (versão 2.04.4) (obsoleta)	297
Melhorias	298
Comparação com o Aurora MySQL versão 1	298
Compatibilidade com o MySQL 5.7	299
Atualizações do Aurora MySQL: 2019-05-09 (versão 2.04.3) (obsoleta)	299
Melhorias	300
Comparação com o Aurora MySQL versão 1	301
Compatibilidade com o MySQL 5.7	301
Atualizações do Aurora MySQL: 2019-05-02 (versão 2.04.2) (obsoleta)	302
Melhorias	303
Integração de correções de bugs do MySQL	303
Comparação com o Aurora MySQL versão 1	303
Compatibilidade com o MySQL 5.7	304
Atualizações do Aurora MySQL: 2019-03-25 (versão 2.04.1) (obsoleta)	304
Melhorias	305
Comparação com o Aurora MySQL versão 1	306
Compatibilidade com o MySQL 5.7	306
Atualizações do Aurora MySQL: 2019-03-25 (versão 2.04.0) (obsoleta)	307
Melhorias	308
Integração de correções de bugs do MySQL	308
Comparação com o Aurora MySQL versão 1	308
Compatibilidade com o MySQL 5.7	309
Atualizações do Aurora MySQL: 2019-02-07 (versão 2.03.4) (obsoleta)	309

Melhorias	310
Comparação com o Aurora MySQL versão 1	310
Compatibilidade com o MySQL 5.7	311
Atualizações do Aurora MySQL: 2019-01-18 (versão 2.03.3) (obsoleta)	312
Melhorias	313
Integração de correções de bugs do MySQL	314
Comparação com o Aurora MySQL versão 1	314
Compatibilidade com o MySQL 5.7	315
Atualizações do Aurora MySQL: 2019-01-09 (versão 2.03.2) (obsoleta)	315
Melhorias	316
Comparação com o Aurora MySQL versão 1	317
Compatibilidade com o MySQL 5.7	317
Atualizações do Aurora MySQL: 2018-10-24 (versão 2.03.1) (obsoleta)	318
Melhorias	319
Comparação com o Aurora MySQL versão 1	319
Compatibilidade com o MySQL 5.7	319
Atualizações do Aurora MySQL: 2018-10-11 (versão 2.03) (obsoleta)	320
Melhorias	321
Integração de correções de bug da edição MySQL community	321
Comparação com o Aurora MySQL versão 1	322
Compatibilidade com o MySQL 5.7	322
Atualizações do Aurora MySQL: 2018-10-08 (versão 2.02.5) (obsoleta)	323
Melhorias	324
Comparação com o Aurora MySQL versão 1	324
Compatibilidade com o MySQL 5.7	324
Atualizações do Aurora MySQL: 2018-09-21 (versão 2.02.4) (obsoleta)	325
Melhorias	326
Integração de correções de bug da edição MySQL community	326
Comparação com o Aurora MySQL versão 1	327
Compatibilidade com o MySQL 5.7	327
Atualizações do Aurora MySQL: 2018-08-23 (versão 2.02.3) (obsoleta)	328
Comparação com o Aurora MySQL versão 1	329
Compatibilidade com o MySQL 5.7	330
Diferenças de CLI entre o Aurora MySQL 2.x e o Aurora MySQL 1.x	330
Melhorias	331
Atualizações do Aurora MySQL: 2018-06-04 (versão 2.02.2) (obsoleta)	331

Melhorias	332
Comparação com o Aurora MySQL 5.6	332
Compatibilidade com o MySQL 5.7	333
Diferenças de CLI entre o Aurora MySQL 2.x e o Aurora MySQL 1.x	330
Melhorias	332
Atualizações do Aurora MySQL: 2018-05-03 (versão 2.02) (obsoleta)	334
Comparação com o Aurora MySQL 5.6	335
Compatibilidade com o MySQL 5.7	336
Diferenças de CLI entre o Aurora MySQL 2.x e o Aurora MySQL 1.x	330
Melhorias	337
Integração de correções de bugs do MySQL	337
Atualizações do Aurora MySQL: 2018-03-13 (versão 2.01.1) (obsoleta)	337
Comparação com o Aurora MySQL 5.6	338
Compatibilidade com o MySQL 5.7	339
Diferenças de CLI entre o Aurora MySQL 2.x e o Aurora MySQL 1.x	339
Melhorias	340
Atualizações do Aurora MySQL: 2018-02-06 (versão 2.01) (obsoleta)	340
Comparação com o Aurora MySQL 5.6	340
Compatibilidade com o MySQL 5.7	341
Diferenças de CLI entre o Aurora MySQL 2.x e o Aurora MySQL 1.x	330
Aurora MySQL versão 1 (obsoleta)	343
Atualizações do Aurora MySQL: 2021-09-30 (versão 1.23.4) (obsoleta)	346
Melhorias	347
Atualizações do Aurora MySQL: 2021-06-28 (versão 1.23.3) (obsoleta)	347
Melhorias	348
Atualizações do Aurora MySQL: 2021-03-18 (versão 1.23.2) (obsoleta)	348
Melhorias	349
Integração de correções de bug da edição MySQL community	350
Atualizações do Aurora MySQL: 2020-11-24 (versão 1.23.1) (obsoleta)	350
Melhorias	351
Atualizações do Aurora MySQL: 2020-09-02 (versão 1.23.0) (obsoleta)	352
Melhorias	353
Integração de correções de bug da edição MySQL community	356
Atualizações do Aurora MySQL: 2021-06-03 (versão 1.22.5) (obsoleta)	358
Melhorias	358
Atualizações do Aurora MySQL: 2021-03-04 (versão 1.22.4) (obsoleta)	359

Melhorias	360
Atualizações do Aurora MySQL: 2020-11-09 (versão 1.22.3) (obsoleta)	360
Melhorias	361
Integração de correções de bug da edição MySQL community	362
Atualizações do Aurora MySQL: 2020-03-05 (versão 1.22.2) (obsoleta)	363
Melhorias	364
Atualizações do Aurora MySQL: 2019-12-23 (versão 1.22.1) (obsoleta)	364
Melhorias	365
Atualizações do Aurora MySQL: 2019-11-25 (versão 1.22.0) (obsoleta)	366
Melhorias	367
Integração de correções de bug da edição MySQL community	370
Atualizações do Aurora MySQL: 2019-11-25 (versão 1.21.0) (obsoleta)	371
Melhorias	371
Integração de correções de bug da edição MySQL community	373
Atualizações do Aurora MySQL: 2020-03-05 (versão 1.20.1) (obsoleta)	373
Melhorias	374
Atualizações do Aurora MySQL: 2019-11-11 (versão 1.20.0) (obsoleta)	374
Melhorias	375
Integração de correções de bug da edição MySQL community	376
Atualizações do Aurora MySQL: 2020-03-05 (versão 1.19.6) (obsoleta)	377
Melhorias	377
Atualizações do Aurora MySQL: 2019-09-19 (versão 1.19.5) (obsoleta)	378
Melhorias	379
Integração de correções de bug da edição MySQL community	379
Atualizações do Aurora MySQL: 2019-06-05 (versão 1.19.2) (obsoleta)	380
Melhorias	381
Atualizações do Aurora MySQL: 2019-05-09 (versão 1.19.1) (obsoleta)	381
Melhorias	382
Atualizações do Aurora MySQL: 2019-02-07 (versão 1.19.0) (obsoleta)	382
Recursos	383
Melhorias	383
Integração de correções de bug da edição MySQL community	385
Atualizações do Aurora MySQL: 2018-09-20 (versão 1.18.0) (obsoleta)	385
Recursos	386
Atualizações do Aurora MySQL: 2020-03-05 (versão 1.17.9) (obsoleta)	387
Melhorias	387

Atualizações do Aurora MySQL: 2019-01-17 (versão 1.17.8) (obsoleta)	388
Melhorias	388
Integração de correções de bug da edição MySQL community	389
Atualizações do Aurora MySQL: 2018-10-08 (versão 1.17.7) (obsoleta)	389
Melhorias	389
Integração de correções de bug da edição MySQL community	390
Atualizações do Aurora MySQL: 2018-09-06 (versão 1.17.6) (obsoleta)	390
Melhorias	391
Integração de correções de bug da edição MySQL community	391
Atualizações do Aurora MySQL: 2018-08-14 (versão 1.17.5) (obsoleta)	391
Melhorias	392
Atualizações do Aurora MySQL: 2018-08-07 (versão 1.17.4) (obsoleta)	392
Melhorias	393
Atualizações do Aurora MySQL: 2018-06-05 (versão 1.17.3) (obsoleta)	394
Melhorias	395
Atualizações do Aurora MySQL: 2018-04-27 (versão 1.17.2) (obsoleta)	395
Melhorias	395
Atualizações do Aurora MySQL: 2018-03-23 (versão 1.17.1) (obsoleta)	396
Melhorias	397
Atualizações do Aurora MySQL: 2018-03-13 (versão 1.17) (obsoleta)	397
Aplicação de patches com tempo de inatividade zero	397
Novos recursos	398
Melhorias	398
Integração de correções de bugs do MySQL	398
Atualizações do Aurora MySQL: 2017-12-11 (versão 1.16) (obsoleta)	399
Aplicação de patches com tempo de inatividade zero	399
Novos recursos	399
Melhorias	400
Integração de correções de bugs do MySQL	400
Atualizações do Aurora MySQL: 2017-11-20 (versão 1.15.1) (obsoleta)	400
Aplicação de patches com tempo de inatividade zero	401
Melhorias	401
Integração de correções de bugs do MySQL	402
Atualizações do Aurora MySQL: 2017-10-24 (versão 1.15) (obsoleta)	402
Aplicação de patches com tempo de inatividade zero	403
Novos recursos	403

Melhorias	403
Integração de correções de bugs do MySQL	402
Atualizações do Aurora MySQL: 2018-03-13 (versão 1.14.4) (obsoleta)	405
Aplicação de patches com tempo de inatividade zero	405
Novos recursos	406
Melhorias	406
Integração de correções de bugs do MySQL	406
Atualizações do Aurora MySQL: 2017-09-22 (versão 1.14.1) (obsoleta)	406
Melhorias	407
Atualizações do Aurora MySQL: 2017-08-07 (versão 1.14) (obsoleta)	407
Aplicação de patches com tempo de inatividade zero	407
Melhorias	408
Integração de correções de bugs do MySQL	409
Atualizações do Aurora MySQL: 2017-05-15 (versão 1.13) (obsoleta)	409
Aplicação de patches com tempo de inatividade zero	410
Novos recursos	410
Melhorias	410
Integração de correções de bugs do MySQL	411
Atualizações do Aurora MySQL: 2017-04-05 (versão 1.12) (obsoleta)	412
Novos recursos	412
Melhorias	412
Integração de correções de bugs do MySQL	413
Atualizações do Aurora MySQL: 2017-02-23 (versão 1.11) (obsoleta)	414
Novos recursos	414
Melhorias	415
Integração de correções de bugs do MySQL	417
Atualizações do Aurora MySQL: 2017-01-12 (versão 1.10.1) (obsoleta)	417
Novos recursos	418
Melhorias	418
Atualizações do Aurora MySQL: 2016-12-14 (versão 1.10) (obsoleta)	418
Novos recursos	418
Melhorias	420
Integração de correções de bugs do MySQL	421
Atualizações do Aurora MySQL: 2016-11-10 (versão 1.9.0, 1.9.1) (obsoleta)	421
Novos recursos	421
Melhorias	422

Atualizações do Aurora MySQL: 2016-10-26 (versão 1.8.1) (obsoleta)	422
Melhorias	422
Integração de correções de bugs do MySQL	423
Atualizações do Aurora MySQL: 2016-10-18 (versão 1.8) (obsoleta)	423
Novos recursos	423
Melhorias	423
Integração de correções de bugs do MySQL	425
Atualizações do Aurora MySQL: 2016-09-20 (versão 1.7.1) (obsoleta)	425
Melhorias	425
Atualizações do Aurora MySQL: 2016-08-30 (versão 1.7.0) (obsoleta)	426
Novos recursos	426
Melhorias	426
Integração de correções de bugs do MySQL	427
Atualizações do Aurora MySQL: 2016-06-01 (versão 1.6.5) (obsoleta)	427
Novos recursos	427
Melhorias	428
Integração de correções de bugs do MySQL	428
Atualizações do Aurora MySQL: 2016-04-06 (versão 1.6) (obsoleta)	428
Novos recursos	428
Melhorias	430
Integração de correções de bugs do MySQL	430
Atualizações do Aurora MySQL: 2016-01-11 (versão 1.5) (obsoleta)	431
Melhorias	431
Integração de correções de bugs do MySQL	432
Atualizações do Aurora MySQL: 2015-12-03 (versão 1.4) (obsoleta)	432
Novos recursos	432
Melhorias	432
Integração de correções de bugs do MySQL	433
Atualizações do Aurora MySQL: 2015-10-16 (versão 1.2, 1.3) (obsoleta)	434
Correções	434
Melhorias	434
Integração de correções de bugs do MySQL	435
Atualizações do Aurora MySQL: 2015-08-24 (versão 1.1) (obsoleta)	438
Bugs do MySQL corrigidos por atualizações do Aurora MySQL	439
Bugs do MySQL corrigidos por atualizações do Aurora MySQL 3.x	439
Bugs do MySQL corrigidos por atualizações do Aurora MySQL 2.x	458

Bugs do MySQL corrigidos por atualizações do Aurora MySQL 1.x 483

Vulnerabilidades de segurança corrigidas no Aurora MySQL 504

Histórico do documento 513

..... dxxxiii

Notas de lançamento para o Amazon Aurora edição compatível com MySQL

As versões da edição do Amazon Aurora compatível com MySQL são atualizadas regularmente. Essas atualizações são aplicadas a clusters de bancos de dados do Aurora MySQL durante janelas de manutenção do sistema. O cronograma de aplicação das atualizações depende da configuração da Região da AWS e da janela de manutenção para o cluster de banco de dados, além do tipo de atualização.

As versões do Amazon Aurora MySQL são disponibilizadas para todas as Regiões da AWS ao longo de vários dias. Algumas regiões podem mostrar temporariamente uma versão do mecanismo que ainda não está disponível em uma região diferente. As notas de lançamento são publicadas depois que as novas versões estiverem disponíveis em todas as regiões.

As atualizações são aplicadas a todas as instâncias em um cluster de banco de dados simultaneamente. Uma atualização requer uma reinicialização do banco de dados em todas as instâncias em um cluster de banco de dados. Então, haverá um tempo de inatividade de 20 a 30 segundos, mas depois será possível retomar o uso dos clusters ou clusters de banco de dados. É possível visualizar ou alterar as configurações da janela de manutenção no [Console de gerenciamento da AWS](#).

Tópicos

- [Calendários de lançamento do Amazon Aurora MySQL](#)
- [Atualizações feitas no mecanismo de banco de dados do Amazon Aurora MySQL versão 3](#)
- [Atualizações feitas no mecanismo de banco de dados do Amazon Aurora MySQL versão 2](#)
- [Atualizações feitas no mecanismo de banco de dados do Amazon Aurora MySQL versão 1 \(obsoleta\)](#)
- [Bugs do MySQL corrigidos por atualizações do mecanismo de banco de dados Aurora MySQL](#)
- [Vulnerabilidades de segurança corrigidas no Aurora MySQL](#)

Calendários de lançamento do Amazon Aurora MySQL

Os calendários de lançamento nesta página podem ajudar você a planejar suas atualizações de versões principais e secundárias. Consulte mais informações sobre atualizações, versionamento e ciclo de vida do Amazon Aurora em [Versões do Amazon Aurora](#).

Tópicos

- [Calendário de lançamento das versões principais do Aurora MySQL](#)
- [Calendário de lançamento das versões secundárias do Aurora MySQL](#)

Calendário de lançamento das versões principais do Aurora MySQL

As versões principais do Aurora MySQL ficam disponíveis no suporte padrão pelo menos até o fim da vida útil da comunidade para a versão da comunidade correspondente. Você pode continuar executando uma versão principal após a data de término do suporte padrão do Aurora mediante o pagamento de uma taxa. Consulte mais informações em [Usar o Suporte estendido do Amazon RDS](#) e [Definição de preço do Amazon Aurora](#).

No momento, o Aurora MySQL é compatível com as versões principais a seguir.

Note

Você também pode ver informações sobre datas de suporte para as principais versões do mecanismo usando a API AWS CLI ou RDS. Para obter mais informações, consulte [Visualizar datas de suporte de versões de mecanismo no Suporte estendido do Amazon RDS](#).

Versão principal da comunidade	Versão principal do Aurora	Data do fim da vida útil para a comunidade	Data de término do suporte padrão para o Aurora	Data de definição de preço do início do suporte estendido para o RDS no primeiro ano	Data de definição de preço do início do suporte estendido para o RDS no terceiro ano	Data de término do suporte estendido para o RDS	Versões secundárias elegíveis para suporte estendido do RDS
MySQL 8.0	Aurora MySQL versão 3	Abril de 2026	30 de abril de 2028	1 de maio de 2028	Não aplicável	31 de julho de 2029	A ser determinado
MySQL 5.7	Aurora MySQL versão 2	Outubro de 2023	31 de outubro de 2024	1.º de dezembro de 2024	Não aplicável	28 de fevereiro de 2027	Aurora MySQL 2.11 e 2.12
MySQL 5.6 (obsoleto)	Aurora MySQL versão 1 (obsoleta)	5 de fevereiro de 2021	28 de fevereiro de 2023	Não aplicável	Não aplicável	Não aplicável	Não aplicável

Note

O Suporte estendido do Amazon RDS para o Aurora MySQL versão 2 inicia-se em 1.º de novembro de 2024, mas só será cobrado a partir de 1.º de dezembro de 2024. Entre 1.º e 30 de novembro de 2024, todos os clusters de banco de dados do Aurora MySQL versão 2 serão cobertos pelo Suporte estendido do Amazon RDS.

Calendário de lançamento das versões secundárias do Aurora MySQL

No momento, o Aurora MySQL é compatível com as versões secundárias a seguir.

Em geral, as versões secundárias do Aurora são lançadas trimestralmente. O cronograma de lançamento pode variar para coletar mais recursos ou correções.

As versões secundárias podem chegar ao fim do suporte padrão antes das versões principais correspondentes. Por exemplo, a versão 3.07 atingirá a data de fim do suporte padrão em agosto de 2025, enquanto a versão principal 3 chegará ao fim do suporte padrão em 30 de abril de 2028. O Amazon RDS será compatível com mais versões secundárias 3.* lançadas entre essas datas.

As cobranças do Suporte estendido do Amazon RDS se aplicam somente a determinadas versões secundárias após a versão principal se tornar elegível para o Suporte estendido. Consulte mais informações sobre as versões principais elegíveis para o Suporte estendido em [Calendário de lançamento das versões principais do Aurora MySQL](#).

Aurora MySQL versão	Notas de lançamento do Aurora MySQL	Data de término do suporte padrão para o Aurora MySQL
3.12 (Compatível com o Community MySQL 8.0.44)	17 de fevereiro de 2026	17 de fevereiro de 2027
3.11 (Compatível com o Community MySQL 8.0.43)	13 de novembro de 2025	13 de novembro de 2026
3.10 (compatível com o MySQL Community 8.0.42) (LTS)	31 de julho de 2025	30 de abril de 2028
3.09 (compatível com o MySQL Community 8.0.40)	14 de maio de 2025	31 de maio de 2026
3.08 (compatível com o MySQL Community 8.0.39)	18 de novembro de 2024	31 de maio de 2026

Aurora MySQL versão	Notas de lançamento do Aurora MySQL	Data de término do suporte padrão para o Aurora MySQL
3.04 (compatível com o MySQL Community 8.0.28) (LTS)	31 de julho de 2023	31 de outubro de 2026
2.12 ¹ (compatível com o MySQL Community 5.7.40 ou 5.7.44 ²)	25 de julho de 2023	31 de outubro de 2024
2.11 ¹ (compatível com o MySQL Community 5.7.12)	25 de outubro de 2022	31 de outubro de 2024

LTS: versões com suporte de longo prazo (LTS) do Aurora MySQL. Consulte mais informações em [Suporte de longo prazo \(LTS\) e versões beta do Amazon Aurora MySQL](#).

¹ Essa versão secundária continuará disponível quando a versão principal estiver no Amazon RDS Extended Support.

² As versões 2.12 a 2.12.1 do Aurora MySQL são compatíveis com a versão 5.7.40 do Aurora MySQL, e as versões 2.12.2 e superiores são compatíveis com a versão 5.7.44 do MySQL.

Atualizações feitas no mecanismo de banco de dados do Amazon Aurora MySQL versão 3

Veja a seguir as atualizações do mecanismo de banco de dados para o Amazon Aurora MySQL versão 3.

Tópicos

- [Atualizações do mecanismo de banco de dados Aurora MySQL 2026-02-17 \(versão 3.12.0, compatível com o MySQL 8.0.44\)](#)
- [Atualizações do mecanismo de banco de dados Aurora MySQL 2025-12-16 \(versão 3.11.1, compatível com o MySQL 8.0.43\)](#)
- [Atualizações do mecanismo de banco de dados Aurora MySQL 2025-11-13 \(versão 3.11.0, compatível com o MySQL 8.0.43\)](#)
- [Atualizações do mecanismo de banco de dados Aurora MySQL 2026-01-02 \(versão 3.10.3, compatível com o MySQL 8.0.42\)](#)
- [Atualizações do mecanismo de banco de dados Aurora MySQL 2025-11-20 \(versão 3.10.2, compatível com o MySQL 8.0.42\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2025-09-30 \(versão 3.10.1, compatível com MySQL 8.0.42\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2025-07-31 \(versão 3.10.0, compatível com MySQL 8.0.42\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2025-05-14 \(versão 3.09.0, compatível com MySQL 8.0.40\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2025-04-07 \(versão 3.08.2, compatível com MySQL 8.0.39\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2025-02-17 \(versão 3.08.1, compatível com MySQL 8.0.39\)](#)
- [Atualizações do mecanismo de banco de dados Aurora MySQL em 2024-11-18 \(versão 3.08.0, compatível com MySQL 8.0.39\)](#)
- [Atualizações do mecanismo de banco de dados Aurora MySQL 2024-07-23 \(versão 3.07.1\) \(Obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados Aurora MySQL 2024-06-04 \(versão 3.07.0\) \(Obsoleto\)](#)

- [Atualizações do mecanismo de banco de dados Aurora MySQL 2024-06-26 \(versão 3.06.1\) \(Obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados Aurora MySQL 2024-03-07 \(versão 3.06.0\) \(Obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados Aurora MySQL 2024-01-31 \(versão 3.05.2\) \(Obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados Aurora MySQL 2023-11-21 \(versão 3.05.1\) \(Obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados Aurora MySQL 2023-10-30 \(versão 3.05.0.1\) \(Obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados Aurora MySQL 2023-10-25 \(versão 3.05.0\) \(Obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados Aurora MySQL 2026-01-02 \(versão 3.04.6, compatível com o MySQL 8.0.28\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2025-05-05 \(versão 3.04.4, compatível com MySQL 8.0.28\)](#)
- [Atualizações do mecanismo de banco de dados Aurora MySQL 2024-06-26 \(versão 3.04.3, compatível com MySQL 8.0.28\)](#)
- [Atualizações do mecanismo de banco de dados Aurora MySQL 2024-03-15 \(versão 3.04.2, compatível com o MySQL 8.0.28\)](#)
- [Atualizações do mecanismo de banco de dados Aurora MySQL 2023-11-13 \(versão 3.04.1, compatível com MySQL 8.0.28\)](#)
- [Atualizações do mecanismo de banco de dados Aurora MySQL 2023-07-31 \(versão 3.04.0\), compatível com o MySQL 8.0.28\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2023-12-08 \(versão 3.03.3\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2023-08-29 \(versão 3.03.2\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2023-05-11 \(versão 3.03.1\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2023-03-01 \(versão 3.03.0\) \(obsoleta\)](#)

- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2023-04-17 \(versão 3.02.3\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2022-11-18 \(versão 3.02.2\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2022-09-07 \(versão 3.02.1\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2022-04-20 \(versão 3.02.0\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2022-04-15 \(versão 3.01.1\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-11-18 \(versão 3.01.0\) \(obsoleta\)](#)

Atualizações do mecanismo de banco de dados Aurora MySQL 2026-02-17 (versão 3.12.0, compatível com o MySQL 8.0.44)

Versão: 3.12.0

O Aurora MySQL 3.12.0 está disponível ao público em geral. As versões 3.12 do Aurora MySQL são compatíveis com o MySQL 8.0.44. Para obter mais informações sobre as mudanças que ocorreram na comunidade, consulte as [Notas de lançamento do MySQL 8.0](#).

Para obter detalhes sobre novos recursos no Aurora MySQL versão 3, consulte [Aurora MySQL versão 3 compatível com o MySQL 8.0](#). Consulte as diferenças entre o Aurora MySQL versão 3 e o Aurora MySQL versão 2 em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão 3](#). Consulte uma comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition em [Comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition](#) no Guia do usuário do Amazon Aurora.

Você pode realizar uma atualização no local [usando o Zero Downtime Patching \(ZDP\)](#), restaurar um snapshot ou iniciar uma atualização gerenciada blue/green usando o Amazon [RDS Deployments de Blue/Green qualquer cluster do Aurora MySQL versão 2 atualmente compatível para um cluster do Aurora MySQL versão 3.12.0](#).

Consulte informações sobre como planejar um upgrade para o Aurora MySQL versão 3 em [Planejar um upgrade da versão principal de um cluster do Aurora MySQL](#). Para obter informações gerais

sobre os upgrades do Aurora MySQL, consulte [Como atualizar os clusters de banco de dados de Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Consulte informações sobre solução de problemas em [Solução de problemas para atualização no local de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Consulte mais informações em [Manutenção de um cluster de banco de dados do Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correções de segurança

- Corrigido um problema que pode fazer com que algumas instruções SQL não sejam registradas no registro de auditoria.
- O plug-in caching_sha2_password foi corrigido para garantir que as instâncias de leitura e gravação do Aurora MySQL validem as senhas atualizadas de forma consistente.

Médio CVEs:

- [CVE-2025-53040](#)
- [CVE-2025-53042](#)
- [CVE-2025-53044](#)
- [CVE-2025-53045](#)
- [CVE-2025-53053](#)
- [CVE-2025-53054](#)
- [CVE-2025-53062](#)
- [CVE-2025-53069](#)

Melhorias de disponibilidade:

- Corrigido um problema que pode causar a reinicialização de uma instância de banco de dados quando a tabela de bloqueio está cheia durante uma leitura semiconsistente.
- Foi corrigido um problema na recuperação do log binário que poderia fazer com que a instância da réplica do log binário fosse reiniciada ao ser usada `aurora_in_memory_relaylog` com a replicação multiencadeada.

- Corrigido um problema que poderia fazer com que a instância do gravador fosse reiniciada quando o encaminhamento de gravação global ou o encaminhamento de gravação local estavam sendo desativados.
- Corrigido um problema em que configurações `thread_stack` excessivamente grandes podiam impedir a inicialização do servidor Aurora MySQL durante reinicializações ou atualizações. O servidor Aurora MySQL agora redefine automaticamente o `thread_stack` para o valor padrão do mecanismo (1 MB) quando excede a memória do sistema, evitando falhas na inicialização.
- Corrigido um problema que poderia causar a reinicialização do mecanismo ao ser executado `KILL` após a execução de uma consulta paralela `EXPLAIN FOR CONNECTION` em execução.
- Foi corrigido um problema que pode forçar uma consulta paralela em uma consulta aleatória e causar a reinicialização do mecanismo se essa consulta for encerrada à força.
- Correção de um problema que impedia que usuários com privilégios `CONNECTION_ADMIN` ou `SUPER` realizassem uma conexão adicional além do limite `max_connections`, conforme compatível com o MySQL Community Edition.
- Correção de um problema que podia causar a reinicialização de uma réplica de log binário (binlog) ao processar um grande número de arquivos de log de retransmissão durante a [recuperação do log de retransmissão](#).
- Correção de um problema que fazia com que a instância de banco de dados do gravador fosse reiniciada ao executar uma consulta paralela em uma instância de banco de dados de leitor.
- Correção de um problema que, em raras condições, podia desativar o log binário quando ocorria um erro durante a confirmação de uma transação grande.
- Corrigido um problema que pode fazer com que as réplicas de leitura sejam reiniciadas no caso de determinados pedidos raros de confirmação de transações na instância de banco de dados do gravador.
- Correção de um problema que pode levar à reinicialização do banco de dados quando [eventos programados](#) são abortados durante a execução em instâncias com o log binário aprimorado ativado.
- Corrigido um problema em que instâncias de banco de dados que usam tamanhos de armazenamento de vários TiB podem sofrer maior tempo de inatividade durante a reinicialização devido a falhas na validação do bufferpool do InnoDB.

Melhorias gerais:

- Excluiu uma alteração da comunidade introduzida no MySQL 8.0.44, em que tentativas de conexão malsucedidas adicionavam inesperadamente linhas às tabelas `performance_schema`, causando consumo excessivo de memória. [Bug #119766](#) da comunidade Ref
- Corrigido um problema que afetava os procedimentos armazenados de replicação específicos do Aurora ao configurar a replicação de log binário em instâncias com configurações de agrupamento personalizadas.
- Corrigido um problema para reduzir a sobrecarga da CPU ao estabelecer a criptografia em trânsito entre o mecanismo do banco de dados e a camada de armazenamento.
- Melhor desempenho de IOPS de gravação quando a variável do sistema `innodb_flush_log_at_trx_commit` é definida como 0.
- Desative automaticamente as ações `aurora_oom_response` (exceto imprimir, se configurado) quando `aurora_oom_response` falhar em resolver a pressão da memória após um tempo limite (na ordem de alguns minutos).
- Resolução de uma condição de corrida que poderia causar leituras incorretas de páginas do buffer pool durante as operações de redução da escala verticalmente do Aurora Sem Servidor ou durante a remoção de páginas do buffer pool. Ref. [Bug n.º 116.305 da comunidade](#).
- Corrigido um problema out-of-memory (OOM) que poderia fazer com que o leitor fosse reiniciado ao executar comandos relacionados a privilégios no gravador.
- Os seguintes privilégios foram adicionados `aords_superuser_role:FLUSH_OPTIMIZER_COSTS, FLUSH_STATUS, FLUSH_TABLES, FLUSH_USER_RESOURCES`. Para obter informações sobre `aords_superuser_role`, consulte a documentação do [Amazon Master User Accounts with Aurora](#). Consulte mais informações sobre esses privilégios dinâmicos na documentação do [MySQL](#).
- Corrigido um problema que causava o rastreamento impreciso de solicitações de consultas paralelas durante a execução de `EXPLAIN ANALYZE` instruções em que o `Aurora_pq_request_in_progress` contador não era atualizado com precisão.
- Corrigido um problema em que uma conexão preservada era tratada incorretamente durante a correção de tempo de inatividade zero (ZDP) /reinicialização com tempo de inatividade zero (ZDR), o que pode fazer com que o cliente espere indefinidamente pela conclusão de uma consulta.
- Corrija o problema em que a linha se torna ilegível por meio do índice espacial durante uma atualização.
- Corrigido um problema em que uma consulta contendo uma dica de otimizador que foi abortada durante uma operação de reinicialização com tempo de inatividade zero (ZDR) ou correção com tempo de inatividade zero (ZDP) pode ser tratada incorretamente.

- Corrija o problema em que a latência de confirmação não é medida quando `innodb_flush_log_at_trx_commit` é definido como 0.
- Introdução de otimizações para reduzir o uso de memória durante a recuperação do [dicionário de dados](#) lógico quando há um grande número de tabelas.
- Correção de um problema que fazia com que o comando `SHOW BINARY LOGS` demorasse mais para ser executado em um cluster em que o log binário aprimorado estava habilitado ou já havia estado habilitado. Esse problema também poderia causar maior latência de confirmação se vários comandos `SHOW BINARY LOGS` estivessem sendo executados simultaneamente.
- Corrija o problema de memória associado às funções padrão do definidor de visualização.
- Corrigido um problema que pode causar falha na conclusão do processo de desativação do recurso “encaminhamento de gravação”.
- Foi corrigido um problema que fazia com que uma instância do banco de dados do gravador fosse reiniciada quando uma instância do leitor usando o encaminhamento de gravação executava uma instrução DML que contém um valor de timestamp e o parâmetro do banco de dados `time_zone` é definido como “UTC”.
- Foi corrigido um problema que, em casos raros, causava indisponibilidade intermitente de uma réplica de leitura do Aurora ou inconsistências na definição da tabela com o erro “A tabela não existe” na réplica devido a consultas de leitura simultâneas na réplica e operações de DDL no gravador.
- Correção de um problema que pode causar um conjunto de resultados incompleto ao executar consultas envolvendo operações `LEFT-` ou `RIGHT-JOIN` usando o algoritmo `hash-join` com consulta paralela.

Integração de correções de bug do MySQL Community Edition

Esta versão inclui todas as correções de bugs da comunidade até 8.0.44, inclusive. Para obter mais informações, consulte [Bugs do MySQL corrigidos por atualizações do mecanismo de banco de dados do Aurora MySQL 3.x](#).

- Corrigido um problema em que uma consulta do formulário `SELECT 1 FROM t WHERE CAST(a AS UNSIGNED INTEGER) = 1 AND a = (SELECT 1 FROM t)` levava a uma falha de afirmação no `item_func.cc`. (Correção de bug da comunidade #36128964)
- Corrigido um problema que resolve o impasse quando `FLUSH STATUS`, `COM_CHANGE_USER` e `SHOW PROCESS LIST` são executados simultaneamente. (Bug #35218030)

Atualizações do mecanismo de banco de dados Aurora MySQL

2025-12-16 (versão 3.11.1, compatível com o MySQL 8.0.43)

Versão: 3.11.1

O Aurora MySQL 3.11.1 está disponível ao público em geral. As versões 3.11 do Aurora MySQL são compatíveis com o MySQL 8.0.43. Para obter mais informações sobre as mudanças que ocorreram na comunidade, consulte as [Notas de lançamento do MySQL 8.0](#).

Para obter detalhes sobre novos recursos no Aurora MySQL versão 3, consulte [Aurora MySQL versão 3 compatível com o MySQL 8.0](#). Consulte as diferenças entre o Aurora MySQL versão 3 e o Aurora MySQL versão 2 em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão 3](#). Consulte uma comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition em [Comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition](#) no Guia do usuário do Amazon Aurora.

Você pode realizar uma atualização no local [usando o Zero Downtime Patching \(ZDP\)](#), restaurar um snapshot ou iniciar uma atualização gerenciada blue/green usando o Amazon [RDS Deployments de Blue/Green qualquer cluster do Aurora MySQL versão 2 atualmente compatível para um cluster do Aurora MySQL versão 3.11.1](#).

Consulte informações sobre como planejar um upgrade para o Aurora MySQL versão 3 em [Planejar um upgrade da versão principal de um cluster do Aurora MySQL](#). Para obter informações gerais sobre as atualizações do Aurora MySQL, consulte Atualização de clusters de banco de dados Aurora MySQL no Guia do usuário do Amazon Aurora.

Consulte informações sobre solução de problemas em [Solução de problemas para atualização no local de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Se você tiver alguma dúvida ou preocupação, Suporte está disponível nos fóruns da comunidade e por meio de [Suporte](#). Consulte mais informações em [Manutenção de um cluster de banco de dados do Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Melhorias gerais

- Foi corrigido um problema que fazia com que as criações ou restaurações de clusters de banco de dados demorassem muito para serem concluídas.

- Reduziu a sobrecarga de uso da CPU ao estabelecer a criptografia em trânsito entre a instância do banco de dados e a camada de armazenamento.
- Corrigido um problema que pode fazer com que algumas instruções SQL não sejam registradas no registro de auditoria.

Atualizações do mecanismo de banco de dados Aurora MySQL 2025-11-13 (versão 3.11.0, compatível com o MySQL 8.0.43)

Versão: 3.11.0

O Aurora MySQL 3.11.0 está disponível ao público em geral. As versões 3.11 do Aurora MySQL são compatíveis com o MySQL 8.0.43. Para obter mais informações sobre as mudanças que ocorreram na comunidade, consulte as [Notas de lançamento do MySQL 8.0](#).

Para obter detalhes sobre novos recursos no Aurora MySQL versão 3, consulte [Aurora MySQL versão 3 compatível com o MySQL 8.0](#). Consulte as diferenças entre o Aurora MySQL versão 3 e o Aurora MySQL versão 2 em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão 3](#). Consulte uma comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition em [Comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition](#) no Guia do usuário do Amazon Aurora.

Você pode realizar um upgrade no local usando o ZDP (zero-downtime patching), restaurar um snapshot ou iniciar um upgrade gerenciado usando o blue/green Amazon RDS [Deployments de Blue/Green qualquer cluster do Aurora MySQL versão 2 atualmente compatível para um cluster do](#) Aurora MySQL versão 3.11.0.

Consulte informações sobre como planejar um upgrade para o Aurora MySQL versão 3 em [Planejar um upgrade da versão principal de um cluster do Aurora MySQL](#). Para obter informações gerais sobre as atualizações do Aurora MySQL, consulte Atualização de clusters de banco de dados Aurora MySQL no Guia do usuário do Amazon Aurora.

Consulte informações sobre solução de problemas em [Solução de problemas para atualização no local de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Se você tiver alguma dúvida ou preocupação, Suporte está disponível nos fóruns da comunidade e por meio de [Suporte](#). Consulte mais informações em [Manutenção de um cluster de banco de dados do Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Problemas de segurança corrigidos e CVEs

Médio 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](#)

Baixo CVEs:

- [CVE-2025-50098](#)
- [CVE-2025-50100](#)
- [CVE-2025-50104](#)

Melhorias na disponibilidade

- Corrigido um problema na recuperação lógica com a `aurora_enhanced_binlog` opção ativada que poderia impedir a reinicialização do banco de dados em caso de transações abortadas.
- Corrigido um problema que poderia fazer com que a instância do gravador fosse reiniciada quando o encaminhamento de gravação global ou o encaminhamento de gravação local estavam sendo desativados.
- Corrigido um problema que poderia fazer com que o gravador parasse de responder quando o encaminhamento de gravação estava sendo desativado enquanto duas ou mais réplicas estavam encaminhando solicitações.
- Correção de um problema que poderia causar a reinicialização do mecanismo ao ser executado `KILL <query-id>` após a execução de uma consulta paralela `EXPLAIN FOR CONNECTION <query-id>` em execução.
- Corrigido um problema em que a evitação out-of-memory (OOM) não persistia no valor do parâmetro de banco de `aurora_oom_response` dados configurado após a reinicialização do banco de dados.
- Foi corrigido um problema que impedia que usuários com `CONNECTION_ADMIN` `SUPER` privilégios realizassem uma conexão adicional além do `max_connections` limite, conforme suportado no MySQL Community Edition.
- Correção de um problema com a escalabilidade do Aurora Sem Servidor V2 que resultava na reinicialização da instância de banco de dados, impedindo que páginas de memória críticas fossem trocadas.
- Correção de um problema que poderia fazer com que as instâncias v2 do Aurora MySQL Serverless reiniciassem quando o `innodb_purge_threads` parâmetro fosse configurado manualmente com um valor diferente do padrão. Agora, o `innodb_purge_threads` parâmetro é gerenciado automaticamente para instâncias v2 do Aurora Serverless e não pode ser modificado.
- Correção de um problema que podia fazer com que a reinicialização da instância do leitor falhasse se o gravador estivesse executando muitas operações de DDL.
- Corrigido um problema que poderia fazer com que a instância do gravador parasse de responder se as instâncias do leitor reiniciassem usando o encaminhamento de gravação global ou o encaminhamento de gravação local.
- Corrigido um problema que poderia causar a reinicialização do mecanismo durante a correção de tempo de inatividade zero (ZDP) /reinicialização com tempo de inatividade zero (ZDR), ao preservar conexões com transações ativas. SSL/TLS

- Foi corrigido um problema que, em raras condições, fazia com que alterações na instância do gravador fossem enviadas para a instância do leitor em uma ordem incompatível com os encadeamentos de processamento de consultas. Isso pode levar a um impasse entre o encadeamento de processamento de consultas e o encadeamento de aplicação de replicação, o que, por sua vez, pode fazer com que a instância do leitor seja reiniciada.
- Foi corrigido um problema para evitar reinicializações desnecessárias do servidor de banco de dados que ocorriam devido à avaliação de integridade incorreta do agente de monitoramento.

Melhorias gerais

- Corrigido um problema em que as gravações no banco de dados podem parar durante a execução de uma transação de longa duração, levando à reinicialização do banco de dados ou à falha de uma atualização da versão principal.
- Corrigido um problema em que as réplicas eram reiniciadas incorretamente ao ingressar no gravador.
- Desempenho otimizado do banco de dados para instâncias de banco de dados com grandes volumes de armazenamento Aurora.
- Corrigido um problema no otimizador de intervalo em que expressões constantes não eram reconhecidas ao calcular limites de intervalo. (Bug da comunidade #112737)
- Corrigido um problema em que a exportação de um DB snapshot para o Amazon S3 tem um desempenho mais lento ao exportar tabelas contendo colunas JSON com valores nulos.
- Corrigido um problema em que os arquivos de log binários temporários não estavam sendo limpos adequadamente após as reversões de transações ao usar o registro binário. Essa correção evita o consumo de armazenamento de arquivos temporários retidos e, em certos casos, também pode evitar anomalias nos arquivos de log binários.
- Melhor desempenho de IOPS de gravação quando a variável do sistema `innodb_flush_log_at_trx_commit` está definida como `0`.
- Foi corrigido um problema em que a correção com tempo de inatividade zero (ZDP) e a reinicialização com tempo de inatividade zero (ZDR) podiam resultar na reinicialização da instância de banco de dados ao restaurar avisos com códigos de erro inválidos.
- Desative automaticamente `aurora_oom_response` as ações (exceto a impressão, se configurada) quando `aurora_oom_response` não conseguir resolver a pressão da memória após um tempo limite (na ordem de alguns minutos).

- Corrigido um problema em que uma coluna com atualizações parciais de JSON falhava na exportação paralela, causando um retorno interno para uma exportação muito mais lenta do Amazon RDS.
- Correção de um problema que pode causar problemas de gerenciamento de memória ao realizar operações de consulta paralelas na tabela com os campos blob.
- Corrigido um problema que poderia fazer com que o evento de Previous_gtids log binário excluísse determinados GTIDs com o Registro de binário aprimorado ativado e gtid_mode definido como ou. ON ON_PERMISSIVE
- Correção de um problema que causava um “erro de encaminhamento de gravação interno” inesperado nas instâncias de leitor quando o encaminhamento de gravação estava ativado.
- Resolução de uma condição de corrida que poderia causar leituras incorretas de páginas do buffer pool durante as operações de redução da escala verticalmente do Aurora Sem Servidor ou durante a remoção de páginas do buffer pool. Ref. [Bug n.º 116.305 da comunidade](#).
- Corrigido um problema em que a exportação de um DB snapshot para o Amazon S3 era mais lenta ao exportar tabelas contendo índices secundários com expressões geradas.
- Corrigido um problema que causava AbortedClients métricas imprecisas quando várias conexões terminavam inesperadamente.
- Foi adicionada uma nova variável global `aurora_lambda_request_timeout` para permitir que os usuários configurem os tempos limite de AWS Lambda solicitação (padrão: 10 segundos). Para obter mais informações sobre como invocar uma função Lambda de um cluster de banco de dados Aurora MySQL, [consulte Invocar uma função Lambda de um](#) cluster de banco de dados Aurora MySQL.
- Correção de um problema em que o encaminhamento de gravação local parava de funcionar após a reinicialização da instância do banco de dados com reinicialização sem tempo de inatividade.
- Correção de um problema que poderia fazer com que a operação de reinicialização da instância de banco de dados falhasse se estivesse `max_user_connections` definida com um valor baixo.
- Melhor desempenho de exportação paralela otimizando o processo de bootstrap para grandes volumes (>64 TB), reduzindo o tempo geral de operação de exportação.
- Correção de um problema que poderia fazer com que a instância do gravador fosse reiniciada quando executada ALTER TABLE em paralelo com consultas de leitura.
- Corrigido um problema que pode fazer com que as exportações de clusters de banco de dados demorem significativamente mais do que o esperado quando há tabelas maiores que 14 TB.

- Correção de um problema que causava o rastreamento impreciso de solicitações de consultas paralelas durante a execução de EXPLAIN ANALYZE instruções em que o `Aurora_pq_request_in_progress` contador não era atualizado com precisão.
- Correção de um problema que, em condições incomuns, pode fazer com que a instância do banco de dados seja reiniciada quando o volume do banco de dados estiver próximo do tamanho máximo permitido.
- Correção de um problema que pode causar a reinicialização de uma instância de banco de dados ao restaurar conexões durante a reinicialização com tempo de inatividade zero e a aplicação de patches com tempo de inatividade zero.

Integração de correções de bug do MySQL Community Edition

Esta versão inclui todas as correções de bugs da comunidade até 8.0.43, inclusive. Para obter mais informações, consulte [Bugs do MySQL corrigidos por atualizações do mecanismo de banco de dados do Aurora MySQL 3.x](#).

- Corrigido um problema em que uma consulta do formulário `SELECT 1 FROM t WHERE CAST(a AS UNSIGNED INTEGER) = 1 AND a = (SELECT 1 FROM t)` levava a uma falha de afirmação `noitem_func.cc`. (Correção de bug da comunidade #36128964)
- Uma consulta ao formulário `SELECT 1 FROM t WHERE CAST(a AS UNSIGNED INTEGER) = 1 AND a = (SELECT 1 FROM t)` levou a uma afirmação `emitem_func.cc`. (Bug #36128964)

Para obter mais informações sobre as mudanças na comunidade, consulte as notas de lançamento do [MySQL 8.0.43](#).

Atualizações do mecanismo de banco de dados Aurora MySQL 2026-01-02 (versão 3.10.3, compatível com o MySQL 8.0.42)

Versão: 3.10.3

O Aurora MySQL 3.10.3 está disponível ao público em geral. As versões do Aurora MySQL 3.10 são compatíveis com o MySQL 8.0.42. Para obter mais informações sobre as mudanças que ocorreram na comunidade, consulte as [Notas de lançamento do MySQL 8.0](#).

Para obter detalhes sobre novos recursos no Aurora MySQL versão 3, consulte [Aurora MySQL versão 3 compatível com o MySQL 8.0](#). Consulte as diferenças entre o Aurora MySQL versão 3 e o

Aurora MySQL versão 2 em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão 3](#). Consulte uma comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition em [Comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition](#) no Guia do usuário do Amazon Aurora.

Você pode realizar uma atualização no local usando o [Zero Downtime Patching \(ZDP\)](#), restaurar um snapshot ou iniciar uma atualização gerenciada blue/green usando o Amazon [RDS Deployments de Blue/Green qualquer cluster do Aurora MySQL versão 2 atualmente compatível para um cluster do Aurora MySQL versão 3.10.3](#).

Consulte informações sobre como planejar um upgrade para o Aurora MySQL versão 3 em [Planejar um upgrade da versão principal de um cluster do Aurora MySQL](#). Para obter informações gerais sobre as atualizações do Aurora MySQL, consulte Atualização de clusters de banco de dados Aurora MySQL no Guia do usuário do Amazon Aurora.

Consulte informações sobre solução de problemas em [Solução de problemas para atualização no local de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Se você tiver alguma dúvida ou preocupação, Suporte está disponível nos fóruns da comunidade e por meio de [Suporte](#). Consulte mais informações em [Manutenção de um cluster de banco de dados do Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Melhorias gerais:

- Reduziu a sobrecarga de uso da CPU ao estabelecer a criptografia em trânsito entre a instância do banco de dados e a camada de armazenamento.
- Corrigido um problema que pode fazer com que algumas instruções SQL não sejam registradas no registro de auditoria.

Atualizações do mecanismo de banco de dados Aurora MySQL 2025-11-20 (versão 3.10.2, compatível com o MySQL 8.0.42)

Versão: 3.10.2

O Aurora MySQL 3.10.2 está disponível ao público em geral. As versões do Aurora MySQL 3.10 são compatíveis com o MySQL 8.0.42. Para obter mais informações sobre as mudanças que ocorreram na comunidade, consulte as [Notas de lançamento do MySQL 8.0](#).

Para obter detalhes sobre novos recursos no Aurora MySQL versão 3, consulte [Aurora MySQL versão 3 compatível com o MySQL 8.0](#). Consulte as diferenças entre o Aurora MySQL versão 3 e o Aurora MySQL versão 2 em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão 3](#). Consulte uma comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition em [Comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition](#) no Guia do usuário do Amazon Aurora.

Você pode realizar uma atualização no local usando o [Zero Downtime Patching \(ZDP\)](#), restaurar um snapshot ou iniciar uma atualização gerenciada blue/green usando o Amazon [RDS Deployments de Blue/Green qualquer cluster do Aurora MySQL versão 2 atualmente compatível para um cluster do Aurora MySQL versão 3.10.2](#).

Consulte informações sobre como planejar um upgrade para o Aurora MySQL versão 3 em [Planejar um upgrade da versão principal de um cluster do Aurora MySQL](#). Para obter informações gerais sobre as atualizações do Aurora MySQL, consulte Atualização de clusters de banco de dados Aurora MySQL no Guia do usuário do Amazon Aurora.

Consulte informações sobre solução de problemas em [Solução de problemas para atualização no local de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Se você tiver alguma dúvida ou preocupação, Suporte está disponível nos fóruns da comunidade e por meio de [Suporte](#). Consulte mais informações em [Manutenção de um cluster de banco de dados do Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Melhorias de disponibilidade:

- Corrigido um problema que pode causar a reinicialização do mecanismo durante a correção de tempo de inatividade zero (ZDP) /reinicialização com tempo de inatividade zero (ZDR), ao preservar conexões com transações ativas. SSL/TLS
- Corrigido um problema que pode causar a reinicialização do mecanismo ao ser executado KILL <query-id> após a execução de uma consulta paralela EXPLAIN FOR CONNECTION <query-id> em execução.
- Corrigido um problema que poderia fazer com que o gravador parasse de responder quando o encaminhamento de gravação estava sendo desativado enquanto duas ou mais réplicas estavam encaminhando solicitações.

- Corrigido um problema que poderia fazer com que a instância do gravador fosse reiniciada quando o encaminhamento de gravação global ou o encaminhamento de gravação local estavam sendo desativados.

Melhorias gerais:

- Corrigido um problema que causava o rastreamento impreciso de solicitações de consulta paralela durante a execução de EXPLAIN ANALYZE instruções em que o `Aurora_pq_request_in_progress` contador não era atualizado com precisão.
- Desative automaticamente `aurora_oom_response` as ações (exceto a impressão, se configurada) quando `aurora_oom_response` não conseguir resolver a pressão da memória após um tempo limite (na ordem de alguns minutos).
- Melhor desempenho de IOPS de gravação quando a variável do sistema `innodb_flush_log_at_trx_commit` está definida como `0`.
- Corrigido um problema em que as réplicas eram reiniciadas incorretamente ao ingressar no gravador.
- Corrigido um problema em que as gravações no banco de dados podem parar durante a execução de uma transação de longa duração, levando à reinicialização do banco de dados ou à falha de uma atualização da versão principal.

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2025-09-30 (versão 3.10.1, compatível com MySQL 8.0.42)

Versão: 3.10.1

O Aurora MySQL 3.10.1 está disponível para o público. As versões do Aurora MySQL 3.10 são compatíveis com o MySQL 8.0.42. Para obter mais informações sobre as mudanças que ocorreram na comunidade, consulte as [Notas de lançamento do MySQL 8.0](#).

Para obter detalhes sobre novos recursos no Aurora MySQL versão 3, consulte [Aurora MySQL versão 3 compatível com o MySQL 8.0](#). Consulte as diferenças entre o Aurora MySQL versão 3 e o Aurora MySQL versão 2 em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão 3](#). Consulte uma comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition em [Comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition](#) no Guia do usuário do Amazon Aurora.

Você pode realizar um upgrade no local, [utilizando um patch de tempo de inatividade zero \(ZDP\)](#), restaurar um snapshot ou iniciar um upgrade gerenciado azul/verde usando [implantações azul/verde do Amazon RDS](#) de qualquer cluster do Aurora MySQL versão 2 atualmente compatível para um cluster do Aurora MySQL versão 3.10.1.

Consulte informações sobre como planejar um upgrade para o Aurora MySQL versão 3 em [Planejar um upgrade da versão principal de um cluster do Aurora MySQL](#). Para obter informações gerais sobre os upgrades do Aurora MySQL, consulte [Como atualizar os clusters de banco de dados de Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Consulte informações sobre solução de problemas em [Solução de problemas para atualização no local de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Consulte mais informações em [Manutenção de um cluster de banco de dados do Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Melhorias de disponibilidade:

- Correção de um problema com a ordem de bloqueio de página enviada às instâncias de leitor. Isso pode fazer com que a instância de leitor seja reiniciada devido a um impasse ao executar consultas SELECT em tabelas que estão sendo atualizadas pelo gravador.
- Correção de um problema que podia fazer com que a reinicialização da instância do leitor falhasse se o gravador estivesse executando muitas operações de DDL.
- Correção de um problema que poderia fazer com que as instâncias do Aurora MySQL Sem Servidor v2 reiniciassem quando o parâmetro `innodb_purge_threads` fosse configurado manualmente com um valor diferente do padrão. O parâmetro `innodb_purge_threads` agora é gerenciado automaticamente para instâncias do Aurora Sem Servidor v2 e não pode ser modificado.
- Correção de um problema com a escalabilidade do Aurora Sem Servidor V2 que resultava na reinicialização da instância de banco de dados, impedindo que páginas de memória críticas fossem trocadas.
- Correção de um problema em que o mecanismo de prevenção de falta de memória (OOM) do Aurora não mantinha o valor configurado do parâmetro de banco de dados `aurora_oom_response` após a reinicialização da instância.

- Correção de uma condição de corrida que podia causar problemas de disponibilidade após um failover em clusters Multi-AZ.
- Correção de um problema na recuperação lógica com `aurora_enhanced_binlog` ativado que podia impedir a reinicialização do banco de dados em caso de transações anuladas.
- Resolução de um problema em que as instâncias do banco de dados podiam parar de responder durante workloads de Consulta paralela (PQ).
- Correção de um problema que impedia que usuários com privilégios `CONNECTION_ADMIN` ou `SUPER` realizassem uma conexão adicional além do limite `max_connections`, conforme compatível com o MySQL Community Edition.

Melhorias gerais:

- Correção de um problema que podia causar reinicializações contínuas do servidor de banco de dados quando o volume tinha permissão de crescer até seu tamanho máximo devido ao fato de certos tipos de consultas serem permitidos por engano depois que o volume ultrapassava determinado limite.
- Correção de um problema que fazia com que a instância de gravador ficasse sem resposta caso as instâncias de leitor fossem reiniciadas ao usar o encaminhamento de gravação global ou o encaminhamento de gravação local.
- Correção de um problema que poderia fazer com que a instância de gravador fosse reiniciada ao executar `ALTER TABLE` em paralelo com consultas de leitura.
- Correção de um problema para melhorar a disponibilidade com a inicialização de grandes volumes (acima de 64 TB) durante a operação de exportação paralela.
- Correção de um problema que podia fazer com que a operação de reinicialização da instância de banco de dados falhasse caso `max_user_connections` estivesse definido como um valor baixo.
- Correção de um problema que gerava métricas imprecisas de `AbortedClients` quando várias conexões terminavam inesperadamente.
- Resolução de uma condição de corrida que poderia causar leituras incorretas de páginas do buffer pool durante as operações de redução da escala verticalmente do Aurora Sem Servidor ou durante a remoção de páginas do buffer pool. Ref. [Bug n.º 116.305 da comunidade](#).
- Correção de um problema que causava um “erro de encaminhamento de gravação interno” inesperado nas instâncias de leitor quando o encaminhamento de gravação estava ativado.

- Correção de um problema que fazia com que o evento de log binário `previous_gtids` excluísse determinados GTIDs com o log binário aprimorado habilitado e o `gtid_mode` definido como `ON` ou `ON_PERMISSIVE`.
- Correção de um problema que podia causar problemas de gerenciamento de memória em operações de consulta paralelas na tabela com os campos de blob.
- Correção de um problema em que uma coluna com atualizações parciais de JSON falharia na exportação paralela, causando um retorno interno para uma exportação do RDS muito mais lenta.
- Correção de um problema que podia causar reinicializações inesperadas da instância quando havia uma alta workload simultânea de encaminhamento de gravação.
- Correção de um problema em que a aplicação patches com tempo de inatividade zero (ZDP)/reinicialização com tempo de inatividade zero (ZDR) podia resultar na reinicialização da instância de banco de dados ao restaurar avisos com códigos de erro inválidos.
- Correção de um problema em que os arquivos de log binários temporários não eram limpos adequadamente após as reversões de transações ao usar o registro em log binário. Essa correção evita o consumo desnecessário de armazenamento de arquivos temporários retidos e, em certos casos, também pode evitar anomalias nos arquivos de log binários.

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bug da comunidade até a versão 8.0.42. Para obter mais informações, consulte [Bugs do MySQL corrigidos por atualizações do mecanismo de banco de dados do Aurora MySQL 3.x](#).

- Uma consulta no formato `SELECT 1 FROM t WHERE CAST(a AS UNSIGNED INTEGER) = 1 AND a = (SELECT 1 FROM t)` resultava em uma asserção em `item_func.cc`. (Bug #36128964)

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2025-07-31 (versão 3.10.0, compatível com MySQL 8.0.42)

Versão: 3.10.0

O Aurora MySQL 3.10.0 está disponível para o público. As versões do Aurora MySQL 3.10 são compatíveis com o MySQL 8.0.42. Para obter mais informações sobre as mudanças que ocorreram na comunidade, consulte as [Notas de lançamento do MySQL 8.0](#).

 Note

Esta versão é designada como uma versão de suporte de longo prazo (LTS). Para obter mais informações, consulte [Versões de suporte de longo prazo \(LTS\) do Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Recomendamos que você não defina o parâmetro `AutoMinorVersionUpgrade` como `true` (nem ative Upgrade automático de versões secundárias no Console de gerenciamento da AWS) para versões LTS. Fazer isso poderia levar o cluster de banco de dados a ser atualizado para a próxima versão-alvo da campanha de Atualização automática de versão secundária, que pode não ser uma versão LTS.

Para obter detalhes sobre novos recursos no Aurora MySQL versão 3, consulte [Aurora MySQL versão 3 compatível com o MySQL 8.0](#). Consulte as diferenças entre o Aurora MySQL versão 3 e o Aurora MySQL versão 2 em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão 3](#). Consulte uma comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition em [Comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition](#) no Guia do usuário do Amazon Aurora.

Você pode realizar um upgrade no local, [utilizando um patch de tempo de inatividade zero \(ZDP\)](#), restaurar um snapshot ou iniciar um upgrade gerenciado azul/verde usando [implantações azul/verde do Amazon RDS](#) de qualquer cluster do Aurora MySQL versão 2 atualmente compatível para um cluster do Aurora MySQL versão 3.10.0.

Consulte informações sobre como planejar um upgrade para o Aurora MySQL versão 3 em [Planejar um upgrade da versão principal de um cluster do Aurora MySQL](#). Para obter informações gerais sobre os upgrades do Aurora MySQL, consulte [Como atualizar os clusters de banco de dados de Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Consulte informações sobre solução de problemas em [Solução de problemas para atualização no local de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Consulte mais informações em [Manutenção de um cluster de banco de dados do Aurora](#) no Guia do usuário do Amazon Aurora.

Novos atributos

- O Aurora MySQL versão 3.10 amplia o suporte do cache de log de retransmissão na memória para réplicas de log binário. Esse recurso, introduzido pela primeira vez na [versão 3.05](#), pode melhorar o throughput da replicação de log binário em até 40%. O cache de log de retransmissão na memória é habilitado por padrão para replicação de log binário de thread único, replicação de várias threads com [posicionamento automático GTID](#) habilitado e, a partir da versão 3.10, também é habilitado para replicação de várias threads com [replica_preserve_commit_order = ON](#) (mesmo sem GTIDs). É possível controlar esse recurso usando um novo parâmetro em 3.10, [aurora_in_memory_relaylog](#). Consulte mais informações em [Otimizações de log binário no Aurora MySQL](#).
- O Amazon Aurora dobrou sua capacidade máxima de armazenamento de 128 TiB para 256 TiB, permitindo workloads maiores em um único cluster de banco de dados. Para usar o limite de armazenamento ampliado com o Aurora MySQL, atualize o cluster para a versão 3.10 (compatível com o MySQL 8.0.42) ou posterior. Após o upgrade, o armazenamento do Aurora escala automaticamente para 256 TiB com base na quantidade de dados no volume do cluster.

Melhorias

Correções de segurança

CVEs médias:

- [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](#)

CVEs baixas:

- [CVE-2025-21520](#)
- [CVE-2025-21546](#)
- [CVE-2025-30703](#)
- [CVE-2025-30681](#)

Melhorias de disponibilidade:

- Correção de um problema que causava reinicializações desnecessárias do servidor de banco de dados devido a uma avaliação incorreta do andamento da recuperação.

Melhorias gerais:

- Redução da latência de confirmação quando a otimização de E/S está habilitada.
- Correção de um problema que fazia com que as instâncias de leitor não gerassem logs de erros quando o encaminhamento de gravação estava habilitado e o parâmetro `aurora_replica_read_consistency` era modificado.
- Correção de um problema que podia fazer com que uma instância de leitor fosse reiniciada devido à interação entre a thread de replicação e uma consulta acessando tabelas que não estavam presentes no cache do buffer.
- Correção de um problema em que o encaminhamento de gravação local parava de funcionar após a reinicialização da instância do banco de dados com reinicialização sem tempo de inatividade.
- Correção de um problema no encaminhamento de leitura em que as consultas encaminhadas podiam se comportar incorretamente para declarações contendo opções definidas.
- Correção de um problema de estabilidade em que a inserção de metadados em um espaço de tabela undo acionava uma reinicialização inesperada do banco de dados.
- Correção de um problema que podia fazer com que as exportações de clusters de banco de dados demorassem significativamente além do esperado quando havia tabelas maiores que 14 TB.
- Correção de um problema que podia fazer com que o relatório da variável de status `Innodb_buffer_pool_pages_misc` ficasse incorreto.
- Adição de suporte para preservar `LAST_INSERT_ID` durante a aplicação de patches com tempo de inatividade zero (ZDP) ou a reinicialização com tempo de inatividade zero (ZDR).
- O Aurora MySQL usa valores de 8 bits para IDs de índice virtual a fim de evitar problemas no formato undo do MySQL, já que ultrapassar esse limite pode causar indisponibilidade do cluster. Ao se aproximar desse limite, o sistema agora grava mensagens de aviso no log de erros do

MySQL. Se o limite for atingido, as tentativas de adicionar um novo índice retornarão um erro. Consulte mais informações sobre as práticas recomendadas de índice virtual em [Erros de estouro do ID de índice virtual](#) na documentação do Aurora MySQL.

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bug da comunidade até a versão 8.0.42. Para obter mais informações, consulte [Bugs do MySQL corrigidos por atualizações do mecanismo de banco de dados do Aurora MySQL 3.x](#).

- Uma saída do servidor pode gerar tentativas simultâneas de várias threads para registrar e cancelar o registro de objetos de metadados do Performance Schema, ou para adquirir e liberar bloqueios de metadados. (Erro nº 26502135)

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2025-05-14 (versão 3.09.0, compatível com MySQL 8.0.40)

Versão: 3.09.0

Aurora MySQL 3.09.0 está disponível para o público em geral. As versões do Aurora MySQL 3.09 são compatíveis com o MySQL 8.0.40. Para obter mais informações sobre as mudanças que ocorreram na comunidade, consulte as [Notas de lançamento do MySQL 8.0](#).

Para obter detalhes sobre novos recursos no Aurora MySQL versão 3, consulte [Aurora MySQL versão 3 compatível com o MySQL 8.0](#). Consulte as diferenças entre o Aurora MySQL versão 3 e o Aurora MySQL versão 2 em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão 3](#). Consulte uma comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition em [Comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition](#) no Guia do usuário do Amazon Aurora.

Você pode realizar um upgrade no local, que utiliza um [patch de tempo de inatividade zero](#), restaurar um snapshot ou iniciar um upgrade gerenciado azul/verde usando [implantações azul/verde do Amazon RDS](#) de qualquer cluster do Aurora MySQL versão 2 atualmente compatível para um cluster do Aurora MySQL versão 3.09.0.

Consulte informações sobre como planejar um upgrade para o Aurora MySQL versão 3 em [Planejar um upgrade da versão principal de um cluster do Aurora MySQL](#). Para obter informações gerais

sobre os upgrades do Aurora MySQL, consulte [Como atualizar os clusters de banco de dados de Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Consulte informações sobre solução de problemas em [Solução de problemas para atualização no local de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Consulte mais informações em [Manutenção de um cluster de banco de dados](#) no Guia do usuário do Amazon Aurora.

Destaques do lançamento

- Bancos de dados globais aprimorados do Aurora MySQL para permitir que instâncias secundárias de leitura concluam a inicialização e atendam às solicitações de leitura durante eventos não planejados (falhas de hardware, interrupções na rede). Anteriormente, as instâncias secundárias do leitor não podiam ser reiniciadas durante esses eventos. Consulte mais informações em [Resiliência entre regiões para clusters secundários do Global Database](#) no Guia do usuário do Amazon Aurora.
- Reduziu o tempo de inatividade do gravador durante transições entre regiões do Global Database do Aurora MySQL para normalmente menos de um minuto, minimizando o tempo de inatividade durante as transições regionais planejadas.

Melhorias

Correções de segurança

CVEs críticas:

- [CVE-2024-11053](#)
- [CVE-2024-37371](#)

CVEs médias:

- [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](#)

Melhorias de disponibilidade:

- Correção de um problema em que consultas com várias declarações encaminhadas de instâncias de leitor para gravador podiam travar quando `innodb_flush_log_at_trx_commit` estava definido como 0 no gravador e era diferente de zero no leitor, evitando possíveis falhas no encaminhamento de gravação.
- Correção de um problema de deadlock com o log binário avançado ativado, que poderia causar reinicializações do banco de dados ao executar `SHOW BINARY LOGS` simultaneamente com a confirmação de transações nos mecanismos [BLACKHOLE](#) ou a execução de declarações XA `PREPARE`, evitando possíveis problemas de gravação travada e disponibilidade de instâncias.

- Corrigidas as condições de corrida no encaminhamento de gravação que poderiam causar a reinicialização da instância de gravador do Aurora, impedindo que novas solicitações fossem aceitas antes que as anteriores fossem totalmente concluídas, melhorando a estabilidade das operações de encaminhamento de gravação.
- Correção de um problema na réplica em que uma interrupção na rede podia impedir que a conexão com o gravador fosse estabelecida corretamente, fazendo a replicação travar e podendo levar à reinicialização da instância.
- A resposta de fora de memória (OOM) do Aurora MySQL agora implementa o redimensionamento em fases do pool de buffers que reduz gradualmente o uso da memória com base no estado da memória do sistema (BAIXO/RESERVADO) quando ativada por meio do parâmetro de banco de dados `aurora_oom_response`, proporcionando melhor gerenciamento de memória em situações de pressão.
- Melhoria no tempo de recuperação de arquivos Binlog durante a reinicialização do banco de dados, otimizando o processo para que ele leve um tempo constante, independentemente do tamanho do arquivo Binlog. Anteriormente, em alguns casos, o tempo de recuperação era proporcional ao tamanho do último arquivo Binlog.
- Correção de um problema que poderia causar reinicializações inesperadas do servidor MySQL ao executar operações simultâneas de truncamento de tabela do InnoDB durante consultas em `performance_schema.data_lock_waits`.
- Correção de um problema que podia fazer com que uma instância de banco de dados fosse reiniciada ao confirmar grandes eventos de log binário durante condições de baixo armazenamento.
- Correção de um problema em que as operações de redimensionamento do buffer pool acionadas para evitar a falta de memória (OOM) podiam parar de responder em cenários de alta workload, levando a uma possível reinicialização do banco de dados.
- Correção de um problema que podia causar um loop de reinicialização do banco de dados ao criar um gatilho. O problema também podia ocorrer quando um novo arquivo Binlog ou Relaylog é adicionado ou quando esses arquivos eram alternados.
- Correção de um problema que podia fazer com que a instância do Aurora Reader fosse reiniciada ao usar o encaminhamento de gravação com consultas de confirmação implícitas ou com várias declarações.

Melhorias gerais:

- Correção de um problema em que as operações `ALTER TABLE ... REBUILD / OPTIMIZE TABLE` podiam consumir memória excessiva alocando `innodb_ddl_buffer_size` bytes por thread de DDL em vez de dividir o tamanho do buffer entre as threads, evitando um possível esgotamento de memória durante as operações de DDL.
- Alteração do valor padrão para `aurora_oom_response`, em todas as classes de instância de banco de dados com mais de 4 GiB de memória, de `print` para `print,decline,kill_connect`. Consulte mais informações em [Problemas de falta de memória do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Os seguintes privilégios foram adicionados a `rds_superuser_role`: `FLUSH_OPTIMIZER_COSTS`, `FLUSH_STATUS`, `FLUSH_TABLES` e `FLUSH_USER_RESOURCES`. Consulte informações sobre `rds_superuser_role` na documentação de [Contas de usuário principal da Amazon com o Amazon Aurora](#). Consulte mais informações sobre esses privilégios dinâmicos na documentação do [MySQL](#).
- A partir dessa versão do Aurora MySQL, a otimização de inserção rápida não é mais habilitada. Para obter mais informações, consulte [Melhorias de performance do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Correção de um problema de violação incorreta do limite `max_user_connections`, que resultava em erros de conexão para alguns usuários. Isso ocorria em alguns casos de borda, como quando as conexões eram criadas e encerradas quase imediatamente.
- Correção de um problema de registro em log de auditoria que causava alta utilização da CPU, fazendo com que uma instância do servidor de banco de dados não respondesse.
- Correção de um problema de gerenciamento de memória ao usar transações XA, impedindo possíveis reinicializações da instância quando o log binário aprimorado estava habilitado.
- Correção de um problema em que o desempenho da consulta diminuía quando o otimizador fazia estimativas de custo incorretas devido à atualização incorreta das estatísticas do índice do Bufferpool após a reinicialização do servidor do banco de dados.
- Correção de um problema que impedia que os clientes desativassem a funcionalidade local de encaminhamento de gravação devido a uma thread de operador travada.
- Correção de um problema que fazia com que o comando `SHOW BINARY LOGS` demorasse mais para ser executado em um cluster em que o log binário aprimorado estava habilitado ou já havia estado habilitado. Esse problema também poderia causar maior latência de confirmação se vários comandos `SHOW BINARY LOGS` estivessem sendo executados simultaneamente.

Upgrades e migrações:

- Correção de um problema em que a aplicação de patches com tempo de inatividade zero (ZDP) podia falhar ao tentar preservar a conexão pertencente a um usuário que havia sido excluído. Mais informações sobre o comando DROP USER e seu efeito nas conexões ativas podem ser encontradas na documentação do [MySQL](#).

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bug da comunidade até a versão 8.0.40. Para obter mais informações, consulte [Bugs do MySQL corrigidos por atualizações do mecanismo de banco de dados do Aurora MySQL 3.x](#).

- Enquanto transações grandes estavam sendo recebidas e aplicadas, e uma solicitação para parar o canal de replicação era feita usando STOP REPLICA, o MySQL não executava essa ação corretamente e, em seguida, deixava de processar quaisquer comandos do canal. Além disso, o processo de desligamento do servidor não era concluído de forma adequada, exigindo que o processo do MySQL fosse encerrado à força ou que o sistema host fosse reiniciado. (Bug nº 115966, Bug nº 37008345)

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2025-04-07 (versão 3.08.2, compatível com MySQL 8.0.39)

Versão: 3.08.2

O Aurora MySQL 3.08.2 está disponível para o público. As versões do Aurora MySQL 3.08 são compatíveis com o MySQL 8.0.39. Para obter mais informações sobre as mudanças que ocorreram na comunidade, consulte as [Notas de lançamento do MySQL 8.0](#).

Para obter detalhes sobre novos recursos no Aurora MySQL versão 3, consulte [Aurora MySQL versão 3 compatível com o MySQL 8.0](#). Consulte as diferenças entre o Aurora MySQL versão 3 e o Aurora MySQL versão 2 em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão 3](#). Consulte uma comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition em [Comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition](#) no Guia do usuário do Amazon Aurora.

As versões atualmente compatíveis do Aurora MySQL são 2.11.*, 2.12.*, 3.04.*, 3.05.*, 3.06.*, 3.07.* e 3.08.*.

Você pode realizar um upgrade no local, restaurar um snapshot ou iniciar um upgrade gerenciado azul/verde usando [implantações azul/verde do Amazon RDS](#) de qualquer cluster do Aurora MySQL versão 2 atualmente compatível para um cluster do Aurora MySQL versão 3.08.2.

Consulte informações sobre como planejar um upgrade para o Aurora MySQL versão 3 em [Planejar um upgrade da versão principal de um cluster do Aurora MySQL](#). Para obter informações gerais sobre os upgrades do Aurora MySQL, consulte [Como atualizar os clusters de banco de dados de Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Consulte informações sobre solução de problemas em [Solução de problemas para atualização no local de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correções de segurança:

- [CVE-2024-11053](#)

Melhorias de disponibilidade:

- Correção de um problema na réplica em que uma interrupção na rede podia impedir que a conexão com o gravador.
- Correção de um problema em que uma consulta encaminhada de várias declarações podia falhar caso `innodb_flush_log_at_trx_commit` estivesse definido como 0 na instância de gravador.
- Correção de um problema que fazia com que uma instância de banco de dados de gravador do Aurora fosse reiniciada quando o encaminhamento de gravação estava habilitado.
- Correção de um problema que podia fazer com que uma instância de banco de dados de leitor do Aurora fosse reiniciada ao usar o encaminhamento de gravação com consultas de confirmação implícitas ou de várias declarações.

Melhorias gerais

- A partir dessa versão do Aurora MySQL, a otimização de inserção rápida não é mais habilitada. Para obter mais informações, consulte [Melhorias de performance do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Correção de um problema de violação incorreta do limite `max_user_connections`, que resultava em erros de conexão para alguns usuários. Isso ocorria em alguns casos de borda, como quando as conexões eram criadas e encerradas quase imediatamente.

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bugs da comunidade até 8.0.39, incluindo as indicadas a seguir. Para obter mais informações, consulte [MySQL bugs fixed by Aurora MySQL 3.x database engine updates](#).

- Enquanto transações grandes estavam sendo recebidas e aplicadas, e uma solicitação para parar o canal de replicação era feita usando `STOP REPLICA`, o MySQL não executava essa ação corretamente e, em seguida, deixava de processar quaisquer comandos do canal. Além disso, o processo de desligamento do servidor não era concluído de forma adequada, exigindo que o processo do MySQL fosse encerrado à força ou que o sistema host fosse reiniciado. (Correção de bugs n.º 115966 e n.º 37008345 da comunidade)

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2025-02-17 (versão 3.08.1, compatível com MySQL 8.0.39)

Versão: 3.08.1

O Aurora MySQL 3.08.1 está disponível para o público em geral. As versões do Aurora MySQL 3.08 são compatíveis com o MySQL 8.0.39. Para obter mais informações sobre as mudanças que ocorreram na comunidade, consulte as [Notas de lançamento do MySQL 8.0](#).

Para obter detalhes sobre novos recursos no Aurora MySQL versão 3, consulte [Aurora MySQL versão 3 compatível com o MySQL 8.0](#). Consulte as diferenças entre o Aurora MySQL versão 3 e o Aurora MySQL versão 2 em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão 3](#). Consulte uma comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition em [Comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition](#) no Guia do usuário do Amazon Aurora.

As versões atualmente compatíveis do Aurora MySQL são 2.11.*, 2.12.*, 3.04.*, 3.05.*, 3.06.*, 3.07.* e 3.08.*.

Você pode realizar um upgrade no local, restaurar um snapshot ou iniciar um upgrade gerenciado azul/verde usando [implantações azul/verde do Amazon RDS](#) de qualquer cluster do Aurora MySQL versão 2 atualmente compatível para um cluster do Aurora MySQL versão 3.08.1.

Consulte informações sobre como planejar um upgrade para o Aurora MySQL versão 3 em [Planejar um upgrade da versão principal de um cluster do Aurora MySQL](#). Para obter informações gerais sobre os upgrades do Aurora MySQL, consulte [Como atualizar os clusters de banco de dados de Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Consulte informações sobre solução de problemas em [Solução de problemas para atualização no local de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Melhorias de disponibilidade:

- Fizemos melhorias que reduzem o tempo de reinicialização da instância de banco de dados durante reinicializações do banco de dados, atualizações de patches ou de versões secundárias e failovers.

Upgrades e migrações

- Correção de um problema que podia causar a falha na aplicação de patches com tempo de inatividade zero (ZDP) ao tentar restaurar os perfis e as concessões de usuários removidos. Consulte mais informações sobre a operação DROP USER em [DROP USER Statement](#) na documentação do MySQL.

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bugs da comunidade até 8.0.39, incluindo as indicadas a seguir. Para obter mais informações, consulte [MySQL bugs fixed by Aurora MySQL 3.x database engine updates](#).

Atualizações do mecanismo de banco de dados Aurora MySQL em 2024-11-18 (versão 3.08.0, compatível com MySQL 8.0.39)

Versão: 3.08.0

O Aurora MySQL 3.08.0 está disponível para o público. As versões do Aurora MySQL 3.08 são compatíveis com o MySQL 8.0.39. Para obter mais informações sobre as mudanças que ocorreram na comunidade, consulte as [Notas de lançamento do MySQL 8.0](#).

Para obter detalhes sobre novos recursos no Aurora MySQL versão 3, consulte [Aurora MySQL versão 3 compatível com o MySQL 8.0](#). Consulte as diferenças entre o Aurora MySQL versão 3 e o Aurora MySQL versão 2 em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão 3](#). Consulte uma comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition em [Comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition](#) no Guia do usuário do Amazon Aurora.

As versões atualmente compatíveis do Aurora MySQL são 2.11.*, 2.12.*, 3.04.*, 3.05.*, 3.06.*, 3.07.* e 3.08.*.

Você pode realizar um upgrade no local, restaurar um snapshot ou iniciar um upgrade gerenciado azul/verde usando [implantações azul/verde do Amazon RDS](#) de qualquer cluster do Aurora MySQL versão 2 atualmente compatível para um cluster do Aurora MySQL versão 3.08.0.

Consulte informações sobre como planejar um upgrade para o Aurora MySQL versão 3 em [Planejar um upgrade da versão principal de um cluster do Aurora MySQL](#). Para obter informações gerais sobre os upgrades do Aurora MySQL, consulte [Como atualizar os clusters de banco de dados de Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Consulte informações sobre solução de problemas em [Solução de problemas para atualização no local de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Novos atributos

- Adição de três novas métricas do Amazon CloudWatch para permitir que os usuários monitorem o processo de limpeza do InnoDB:

- `PurgeBoundary`
- `PurgeFinishedPoint`
- `TruncateFinishedPoint`

Consulte mais informações em [Métricas do Amazon CloudWatch para o Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

- Adição de uma nova métrica `TransactionAgeMaximum` do CloudWatch para permitir que os usuários identifiquem transações de longa duração que possam estar atrasando o limite de limpeza. Consulte mais informações em [Métricas do Amazon CloudWatch para o Amazon Aurora](#) no Guia do usuário do Amazon Aurora.
- Adição de três novas métricas do CloudWatch para evitar a falta de memória (OOM):
 - `AuroraMillisecondsSpentInOomRecovery`
 - `AuroraNumOomRecoverySuccessful`
 - `AuroraNumOomRecoveryTriggered`

Consulte mais informações em [Métricas do Amazon CloudWatch para o Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

- Alteração de três métricas do CloudWatch para evitar a falta de memória (OOM), passando de valores totais acumulados para contadores incrementais:
 - `AuroraMemoryNumDeclinedSqlTotal`
 - `AuroraMemoryNumKillConnTotal`
 - `AuroraMemoryNumKillQueryTotal`

Consulte mais informações em [Métricas do Amazon CloudWatch para o Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

- Adição de duas variáveis de status globais para mostrar a quantidade de memória usada pelas [tabelas temporárias internas](#): `aurora_temptable_ram_allocation` e `aurora_temptable_max_ram_allocation`. Essas variáveis de status globais aumentam a observabilidade e ajudam a diagnosticar problemas relacionados ao uso interno da memória temporária da tabela.

Consulte mais informações em [Variáveis de status globais do Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

- Introdução da nova variável de sistema `aurora_optimizer_trace_print_before_purge` para imprimir [rastreamentos do otimizador](#) no log de erros antes que o servidor limpe os

rastreamentos da memória. É possível acionar uma limpeza com base nos limites definidos pelas variáveis do sistema [optimizer_trace_offset](#) e [optimizer_trace_limit](#).

- Adição de suporte para as seguintes classes de instância de banco de dados:
 - db.r7i
 - db.r8g

Para obter mais informações, consulte [Mecanismos de banco de dados compatíveis para classes de instância de banco de dados](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correção de problemas de segurança e CVEs:

- Introdução de um novo usuário para replicação de log binário (binlog), `rdsrepladmin_priv_checks_user`. Consulte mais informações em [Verificações de privilégios do usuário para replicação de log binário](#) no Guia do usuário do Amazon Aurora.
- Correção de um problema em que os parâmetros de entrada para [procedimentos armazenados do Aurora MySQL](#) podiam ser gerenciados incorretamente.

Essa versão inclui todas as correções de CVE da comunidade, incluindo o MySQL 8.0.39. As seguintes correções de CVE estão incluídas:

- [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](#)

Melhorias de disponibilidade:

- Correção de que podia fazer o servidor reiniciar inesperadamente após executar comandos de linguagem de manipulação de dados (DML) em uma tabela que já tinha colunas removidas instantaneamente, quando a contagem de colunas removidas e não removidas ultrapassava 1.017.
- Correção de um problema que podia fazer com que os arquivos de log do banco de dados não fossem rotacionados corretamente, resultando em maior [uso de espaço de armazenamento local](#) em uma instância de banco de dados.
- Correção de um problema que podia fazer com que a instância de banco de dados deixasse de encerrar conexões em condições de pouca memória quando [aurora_oom_response](#) estava ativado, levando a uma reinicialização por falta de memória que poderia ser evitada.
- Correção um problema que podia fazer com que uma instância de banco de dados de leitor fosse reiniciada ao liberar memória usada para a aplicação de log.
- Correção de um problema no cálculo de métricas internas para índices de pesquisa de texto completo (FTS) que podia causar a reinicialização do banco de dados.

- Correção de um problema da comunidade que, às vezes, podia gerar resultados incorretos de consultas quando uma cláusula LIMIT era usada em subconsultas junto com a redução de condição de índice.
- Correção de um problema que causava uma violação de acesso à memória que levava à liberação de um objeto mutex que não pertencia mais à thread.
- Correção de um problema que podia causar a reinicialização de uma réplica de log binário (binlog) ao processar um grande número de arquivos de log de retransmissão durante a [recuperação do log de retransmissão](#).
- Correção de um problema que fazia com que uma instância de banco de dados de leitor do Aurora fosse reiniciada quando o encaminhamento de gravação estava habilitado.
- Correção de um problema em que uma consulta contendo várias cláusulas UNION podia alocar uma grande quantidade de memória, levando à reinicialização da instância de banco de dados.
- Correção de um problema que fazia com que a instância de banco de dados do gravador fosse reiniciada ao executar uma consulta paralela em uma instância de banco de dados de leitor.
- Correção de um problema em que a replicação do log binário travava na réplica devido a um deadlock encontrado no encadeamento de E/S da réplica quando o comando FLUSH RELAY LOGS era executado.
- Correção de um problema que causava a reinicialização de uma instância de banco de dados ao lidar com grandes conjuntos de GTID em um cluster de banco de dados com o log binário aprimorado habilitado.
- Correção de um problema que podia causar a reinicialização de uma réplica de log binário quando o cache de log de retransmissão na memória estava habilitado. O cache de log de retransmissão na memória é ativado nas réplicas de log binário gerenciadas pelo Aurora MySQL ao usar a replicação de log binário de thread única ou a replicação multithread com o [posicionamento automático GTID](#) habilitado.
- Correção de um problema que podia fazer com que uma instância de réplica de log binário fosse reiniciada ao aplicar instruções de linguagem de controle de dados (DCL) durante a inicialização do mecanismo de banco de dados.
- Correção de um problema que, em raras ocasiões, podia fazer com que uma instância de banco de dados de leitor fosse reiniciada devido a um deadlatch ao executar consultas SELECT em tabelas que estavam sendo atualizadas por sua instância de banco de dados de gravador.
- Correção de um problema que fazia com que uma instância de leitor do Aurora Global Database fosse reiniciada com uma sessão ativa de encaminhamento de gravação.

- Correção de um problema que podia fazer com que as réplicas de leitura do Aurora fossem reiniciadas no caso de determinados pedidos raros de confirmação de transações na instância de banco de dados de gravador.
- Correção de um problema que podia fazer com que um banco de dados fosse reiniciado quando [eventos programados](#) eram cancelados durante a execução em instâncias de banco de dados com o log binário aprimorado habilitado.

Melhorias gerais:

- Correção de um problema em que a conexão de um cliente podia ficar travada durante a aplicação de patches com tempo de inatividade zero (ZDP) ou a reinicialização com tempo de inatividade zero (ZDR).
- Correção de um problema que, em casos raros, fazia com que a instância de banco de dados fosse reiniciada devido a um problema de gerenciamento de memória que podia ocorrer ao lidar com uma falha de abertura de tabela.
- Correção de um problema que fazia com que as declarações SQL apresentassem erros ou avisos inesperados de violação de chave primária em algumas linhas ao executar declarações INSERT simultâneas em uma tabela que tem uma coluna de chave primária AUTO_INCREMENT e uma coluna de chave exclusiva, e quando uma declaração INSERT tem violações de chave exclusivas em linhas diferentes.
- O instrumento `memory/sql/sp_head::main_mem_root` do Performance Schema agora é controlado. Consequentemente, a memória alocada para análise e representação de programas armazenados agora passa a contar com relação aos limites de memória por conexão.
- Correção de um problema em que uma linha ficava ilegível por meio do índice espacial durante uma atualização.
- Correção de um problema que impedia que os usuários desativassem o encaminhamento de gravação local.
- Correção de um problema em que uma consulta `SELECT COUNT` podia retornar o resultado errado ao usar o encaminhamento de gravação.
- Correção de um problema que podia gerar resultados de consulta incorretos quando a ZDP restaurava incorretamente as variáveis de sessão definidas como dicas nas consultas.
- Correção de um problema com o truncamento automático de espaços de tabela undo quando eles eram maiores do que o limite [innodb_max_undo_log_size](#) em cenários de atualização.

- Correção de um problema no qual a latência e o throughput de confirmação não eram medidos quando `innodb_flush_log_at_trx_commit` era definido como 0.
- Correção de um problema que fazia com que o atraso na réplica do Aurora fosse relatado incorretamente.
- Correção de um problema que fornecia um valor incorreto para a variável `threads_running` ao usar o Aurora Global Database.
- Correção de um problema em que uma réplica do log binário do Aurora MySQL com a [otimização do índice secundário paralelo](#) ativada era reiniciada ao aplicar alterações de replicação em tabelas com chaves estrangeiras.
- Correção de um problema de reinicialização causado pelo redimensionamento prolongado da tabela de hash de bloqueio durante eventos de aumento ou redução da escala verticalmente.
- Correção de um problema que podia causar a reinicialização de uma instância de banco de dados após a desativação do log binário, quando o log binário aprimorado estava habilitado anteriormente.
- Introdução de otimizações para reduzir o uso de memória durante a recuperação do [dicionário de dados](#) lógico quando há um grande número de tabelas.
- Correção de um problema em que um usuário poderia enfrentar um erro `ERROR 1377 (HY000): Fatal error during log purge` ao executar o procedimento armazenado [mysql.rds_set_external_source](#) em uma réplica de log binário, quando a replicação de log binário já estava configurada.
- Correção de uma falha que impedia a persistência dos privilégios de perfil do usuário após a [ZDR](#).
- Correção de um problema de memória associado aos perfis padrão do definidor de visualização.
- Correção de um problema que podia fazer com que uma instância de banco de dados fosse reiniciada quando `SHOW VOLUME STATUS` era executado.
- Correção de um problema de reinicialização causado pelo redimensionamento prolongado do buffer pool durante eventos de aumento ou redução da escala verticalmente.
- Correção de um problema de reinicialização causado pelo acesso da leitura lógica antecipada (LRA) às páginas liberadas devido ao redimensionamento do buffer durante um evento de redução da escala verticalmente.
- Correção de um problema que fazia consultas `SELECT` retornarem resultados incorretos quando a consulta utilizava `LEFT OUTER JOIN` com a [transformação de semijunção](#) que empregava materialização como estratégia.
- Correção de um problema que podia causar falha na conclusão do processo de desativação do encaminhamento de gravação.

- Correção de um problema em que as métricas `ActiveTransactions` e `BlockedTransactions` do CloudWatch relatavam valores mais baixos do que o esperado.
- Correção de um problema em que a replicação do log binário era interrompida quando a réplica processava uma declaração `DELETE` de várias tabelas que excluía explicitamente tanto uma tabela principal como uma tabela secundária.
- Correção de um problema que podia levar à reinicialização de uma instância de banco de dados ao processar um gatilho com funções definidas pelo usuário que retornam um tipo `enum`.
- Corrigido um problema em que a métrica `DMLLatency` do CloudWatch mostrava valores incorretos para uma instância de réplica de log binário ao replicar usando `binlog_format` definido como `row`.
- Correção de um problema em que consultas lentas `INSERT`, `DELETE` e `UPDATE` executadas pelo [Agendador de eventos](#) do MySQL não eram registradas no log de consultas lentas, a menos que fossem precedidas por uma consulta lenta `SELECT`.
- Correção de um problema que, em casos raros, causava indisponibilidade intermitente de uma réplica de leitura do Aurora ou inconsistências na definição da tabela, às vezes com o erro `Table does not exist`, na réplica. Isso ocorria devido às consultas de leitura simultâneas na réplica e às operações da linguagem de definição de dados (DDL) na instância de banco de dados de gravador.
- Correção de um problema que fazia o comando `SHOW BINARY LOGS` levar mais tempo para ser executado em um cluster de banco de dados em que o log binário aprimorado estava habilitado ou já havia estado habilitado. Esse problema também poderia causar maior latência de confirmação se vários comandos `SHOW BINARY LOGS` estivessem sendo executados simultaneamente.

Upgrades e migrações:

- Melhoria no desempenho de atualização de versões principais do Aurora MySQL versão 2 para a versão 3 para clusters de banco de dados com um grande número de objetos de banco de dados (como tabelas, gatilhos e rotinas).

Para classes maiores de instâncias de banco de dados, o processo de atualização do banco de dados atualiza os metadados de objetos tradicionais do MySQL para o novo dicionário de dados atômico do MySQL 8.0 em paralelo usando várias threads.

- Correção de um problema que fazia com que os upgrades ou as migrações falhassem quando a versão de destino do mecanismo de banco de dados Aurora MySQL era 3.04.0 ou posterior. Isso ocorria quando o parâmetro do cluster de banco de dados `lower_case_table_names`

era definido como 1, e a ordenação do banco de dados MySQL era incompatível com nomes de tabelas em letras minúsculas.

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bugs da comunidade até 8.0.39, incluindo as indicadas a seguir. Para obter mais informações, consulte [MySQL bugs fixed by Aurora MySQL 3.x database engine updates](#).

- Correção de um problema que fazia com que valores NULL fossem omitidos incorretamente do conjunto de resultados para determinadas consultas que têm operações JOIN e UNION. (Correção de bug nº 114301 da comunidade)

Atualizações do mecanismo de banco de dados Aurora MySQL 2024-07-23 (versão 3.07.1) (Obsoleto)

Versão: 3.07.1

O Aurora MySQL 3.07.1 está disponível ao público em geral. As versões 3.07 do Aurora MySQL são compatíveis com o MySQL 8.0.36. Para obter mais informações sobre as mudanças que ocorreram na comunidade, consulte as [Notas de lançamento do MySQL 8.0](#).

Para obter detalhes sobre novos recursos no Aurora MySQL versão 3, consulte [Aurora MySQL versão 3 compatível com o MySQL 8.0](#). Consulte as diferenças entre o Aurora MySQL versão 3 e o Aurora MySQL versão 2 em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão 3](#). Consulte uma comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition em [Comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition](#) no Guia do usuário do Amazon Aurora.

As versões atualmente compatíveis do Aurora MySQL são 2.11.*, 2.12.*, 3.03.*, 3.04.*, 3.05.*, 3.06.* e 3.07.*.

Você pode realizar uma atualização no local, restaurar um snapshot ou iniciar uma atualização gerenciada blue/green usando o [Amazon RDS Blue/Green Deployments](#) de qualquer cluster do Aurora MySQL versão 2 atualmente compatível para um cluster do Aurora MySQL versão 3.07.1.

Consulte informações sobre como planejar um upgrade para o Aurora MySQL versão 3 em [Planejar um upgrade da versão principal de um cluster do Aurora MySQL](#). Para obter informações gerais

sobre os upgrades do Aurora MySQL, consulte [Como atualizar os clusters de banco de dados de Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Consulte informações sobre solução de problemas em [Solução de problemas para atualização no local de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Problemas de segurança corrigidos e CVEs:

- Introdução de um novo usuário para replicação de log binário (binlog), `rdsrepladmin_priv_checks_user`. Consulte mais informações em [Verificações de privilégios do usuário para replicação de log binário](#) no Guia do usuário do Amazon Aurora.

Essa versão inclui todas as correções de CVE da comunidade, incluindo o MySQL 8.0.36.

Melhorias de disponibilidade:

- Correção um problema que podia fazer com que uma instância de banco de dados de leitor fosse reiniciada ao liberar memória usada para a aplicação de log.
- Correção de um problema no cálculo de métricas internas para índices de pesquisa de texto completo (FTS) que podia causar a reinicialização do banco de dados.
- Correção de um problema que podia desativar o log binário quando ocorre um erro ao confirmar uma transação grande.

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bug da comunidade até a versão 8.0.36. Para obter mais informações, consulte [Bugs do MySQL corrigidos por atualizações do mecanismo de banco de dados do Aurora MySQL 3.x](#).

Atualizações do mecanismo de banco de dados Aurora MySQL 2024-06-04 (versão 3.07.0) (Obsoleto)

Versão: 3.07.0

O Aurora MySQL 3.07.0 está disponível ao público em geral. As versões 3.07 do Aurora MySQL são compatíveis com o MySQL 8.0.36. Para obter mais informações sobre as mudanças que ocorreram na comunidade, consulte as [Notas de lançamento do MySQL 8.0](#).

Para obter detalhes sobre novos recursos no Aurora MySQL versão 3, consulte [Aurora MySQL versão 3 compatível com o MySQL 8.0](#). Consulte as diferenças entre o Aurora MySQL versão 3 e o Aurora MySQL versão 2 em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão 3](#). Consulte uma comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition em [Comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition](#) no Guia do usuário do Amazon Aurora.

As versões atualmente compatíveis do Aurora MySQL são 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.03.*, 3.04.*, 3.05.*, 3.06.* e 3.07.*.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Problemas de segurança corrigidos e CVEs:

- Suporte habilitado para criptografia validada por FIPS, uma implementação totalmente própria. AWS Consulte mais informações em [AWS-LC is now FIPS 140-3 certified](#) no [Blog de segurança da AWS](#).

Essa versão inclui todas as correções de CVE da comunidade, incluindo o MySQL 8.0.36. As seguintes correções de CVE estão incluídas:

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

Melhorias de disponibilidade:

- Correção de um problema que podia fazer com que uma instância de banco de dados de leitor fosse reiniciada ao ler uma tabela que estava sendo alterada ou descartada na instância de banco de dados de gravador.
- Correção de um problema que fazia com que uma instância de banco de dados de gravador do Aurora MySQL fosse reiniciada quando uma sessão de encaminhamento de gravação fosse fechada durante a execução de uma consulta encaminhada.
- Correção de um problema que fazia com que uma instância de banco de dados fosse reiniciada ao lidar com grandes conjuntos de GTID em uma instância habilitada para log binário.
- Correção de um problema ao processar consultas INSERT em tabelas particionadas do InnoDB que podia causar um declínio gradual da memória livre na instância de banco de dados.
- Correção de um problema que, em raras condições, podia fazer com que as instâncias de banco de dados de leitor fossem reiniciadas.
- Correção de um problema que podia causar a reinicialização de uma instância de banco de dados ao executar as declarações [SHOW STATUS](#) e [PURGE BINARY LOGS](#) simultaneamente. PURGE BINARY LOGS é uma declaração gerenciada que é executada para honrar o período de retenção de logs binários configurado pelo usuário.
- Correção de um problema que podia fechar o servidor inesperadamente após a execução de declarações de linguagem de manipulação de dados (DML) em uma tabela cujas colunas não virtuais foram reordenadas com uma declaração MODIFY COLUMN ou CHANGE COLUMN.
- Correção de um problema que, durante a reinicialização de uma instância do banco de dados, podia causar uma reinicialização adicional.
- Correção de um problema que podia fazer com que uma instância de banco de dados de leitor que usa o encaminhamento de gravação fosse reiniciada quando uma [declaração de confirmação implícita](#) encaminhada encontrava um erro.
- Correção de um problema que, em raras condições, podia fazer com que uma instância de leitor fosse reiniciada ao realizar consultas SELECT em tabelas com uma restrição de chave estrangeira.
- Correção de um problema em que instâncias de banco de dados que utilizam volumes de cluster do Aurora de vários terabytes podiam apresentar maior tempo de inatividade durante a reinicialização devido a falhas na validação do buffer pool do InnoDB.
- Correção de um problema que podia fazer com que um banco de dados fosse reiniciado quando uma restrição de chave estrangeira UPDATE ou DELETE em cascata era definida em uma tabela em que uma coluna virtual estava envolvida como uma coluna na restrição de chave estrangeira ou como membro da tabela referenciada.

- Correção de um problema que podia interromper a recuperação do banco de dados durante a inicialização caso a reinicialização ocorra durante a execução de operações intensas de inserção envolvendo colunas AUTO_INCREMENT.
- Correção de um problema no Aurora Serverless v2 que podia fazer o banco de dados reiniciar durante a o aumento da escala verticalmente.

Melhorias gerais:

- I/O Uso reduzido e desempenho aprimorado para um subconjunto de consultas de varredura de intervalo de chaves primárias que empregam consultas paralelas.
- O [Aurora MySQL versão 3.06.0](#) adicionou suporte para a integração ao Amazon Bedrock. Como parte disso, novas palavras-chave reservadas (`accept`, `aws_bedrock_invoke_model`, `aws_sagemaker_invoke_endpoint`, `content_type` e `timeout_ms`) foram adicionadas. No Aurora MySQL versão 3.07.0, essas palavras-chave foram alteradas para palavras-chave não reservadas, que são permitidas como identificadores sem aspas. Consulte mais informações sobre como o MySQL lida com palavras-chave reservadas e não reservadas em [Keywords and reserved words](#) na documentação do MySQL.
- Corrigido um problema que não retornava claramente uma mensagem de erro ao cliente ao invocar o serviço Amazon Bedrock a partir de um cluster de banco de dados Aurora MySQL em um local em que a Região da AWS Amazon Bedrock ainda não estava disponível.
- Correção de um problema que podia causar consumo excessivo de memória ao consultar colunas BLOB usando a consulta paralela do Aurora.
- Adição de suporte para que os parâmetros `connection_memory_limit` e `connection_memory_chunk_size` possam ser definidos no nível da sessão, para ter um comportamento semelhante ao que apresenta no MySQL Community Edition. O `connection_memory_limit` é usado para definir a quantidade máxima de memória que pode ser usada por uma única conexão de usuário. O parâmetro `connection_memory_chunk_size` pode ser usado para definir o tamanho dos blocos usados para atualizar o [contador global de uso de memória](#).
- Correção de um problema em que o usuário não conseguia interromper nenhuma consulta ou definir tempos limite de sessão para consultas `performance_schema`.
- Correção de um problema em que a replicação de log binário (binlog) configurada para usar certificados SSL personalizados ([mysql.rds_import_binlog_ssl_material](#)) podia falhar quando a instância de replicação era substituída pelo host.

- Adição da variável de status global `Aurora_fts_cache_memory_used` para rastrear o uso da memória do sistema de pesquisa de texto completo em todas as tabelas. Consulte mais informações em [Variáveis de status globais do Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Corrigido um problema em que um cluster do Amazon Redshift configurado como um destino de zero ETL poderia sofrer um aumento temporário quando um cluster de banco de dados Amazon [IntegrationLag](#) Aurora MySQL era configurado como uma réplica de log binário, com a integração aprimorada de Binlog e Zero-ETL ativada.
- Correção de um problema relacionado ao gerenciamento de arquivos de log de auditoria que podia fazer com que os arquivos de log ficassem inacessíveis para download ou alternância e, em alguns casos, aumentava o uso da CPU.
- Recuperação otimizada de `AUTO_INCREMENT` chaves para reduzir o tempo de conclusão da restauração de instantâneos, da execução da point-in-time recuperação e da clonagem de clusters de banco de dados com um grande número de tabelas no banco de dados.
- [Corrigido um problema em que o evento `wait/io/redo\_log\_flush` não era exibido nas tabelas de resumo de eventos de espera do Performance Schema.](#)
- Correção de um problema que podia causar erros de chave duplicada em colunas `AUTO_INCREMENT` usando índices decrescentes após uma restauração de snapshot, retrocesso ou operação de clonagem de banco de dados.
- Correção de um problema que podia fazer uma instância de gravador ser reiniciada quando uma instância de leitor usando o encaminhamento de gravação executava uma declaração de linguagem de manipulação de dados (DML) que continha um valor de carimbo de data e hora e o parâmetro de banco de dados `time_zone` é definido como UTC.
- Correção de um problema em que uma consulta `SELECT` em uma instância de leitor do Aurora podia falhar com o erro a tabela não existe quando a tabela possuía pelo menos um índice de pesquisa de texto completo (FTS) e uma declaração `TRUNCATE` estava sendo executada na instância de banco de dados de gravador do Aurora.
- Correção de um problema que, em casos raros, fazia com que a aplicação de patches com tempo de inatividade zero (ZDP) falhassem.
- Correção de um problema que podia causar um conjunto de resultados incompleto ao executar consultas envolvendo operações `LEFT JOIN` ou `RIGHT JOIN` usando o algoritmo de junção de hash com o consulta paralela.

Upgrades e migrações:

- Correção de um problema que podia causar falhas de atualização do Aurora MySQL versão 2 para o Aurora MySQL versão 3 quando havia uma coluna FTS_DOC_ID definida pelo usuário presente no esquema da tabela.
- Correção de um problema que podia causar falhas de atualização do Aurora MySQL versão 2 para o Aurora MySQL versão 3 devido a um erro de sincronização ao processar espaços de tabela do InnoDB.
- Correção de um problema que podia fazer com que os upgrades de versões principais para o Aurora MySQL versão 3 falhassem devido à presença de entradas órfãs para espaços de tabela já excluídos nas tabelas do sistema InnoDB no Aurora MySQL versão 2.
- Corrigido um problema em que o valor [SERVER_ID](#) não era atualizado após uma transição de implantação do Amazon Blue/Green RDS. Isso gerou problemas em que drivers inteligentes, como o [driver JDBC da Amazon Web Services \(AWS\)](#), não conseguiram descobrir a topologia do cluster de banco de dados após uma transição. blue/green Com essa correção, os clusters de banco de dados Aurora renomeados como parte de uma Blue/Green implantação do RDS, que estão sendo executados no Aurora MySQL versão 3.07 e superior, terão o valor atualizado como parte da transição. SERVER_ID Para versões anteriores, as instâncias de banco de dados nos clusters azul e verde podem ser reinicializadas para atualizar o valor SERVER_ID.

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bugs da comunidade até 8.0.36, incluindo as indicadas a seguir. Para obter mais informações, consulte [MySQL bugs fixed by Aurora MySQL 3.x database engine updates](#).

- Correção de um problema em que o valor da linha de cache podia ser calculado incorretamente, causando uma falha durante a reinicialização do banco de dados em instâncias baseadas no Graviton. (Correção de bug da comunidade #35479763)
- Correção de um problema em que algumas instâncias de subconsultas em rotinas armazenadas não eram gerenciadas corretamente. (Correção de bug da comunidade #35377192)
- Correção um problema que podia causar maior uso da CPU devido à rotação do certificado TLS em segundo plano (Correção de bug n.º 34284186 da comunidade).
- Correção de um problema em que o InnoDB permitia a adição de colunas INSTANT às tabelas no esquema do sistema MySQL nas versões do Aurora MySQL anteriores à 3.05, o que podia fazer com que o servidor fechasse inesperadamente (reinicialização da instância do banco de

dados) após a atualização para o Aurora MySQL versão 3.05.0. (Correção de bug n.º 35625510 da comunidade).

Atualizações do mecanismo de banco de dados Aurora MySQL 2024-06-26 (versão 3.06.1) (Obsoleto)

Versão: 3.06.1

O Aurora MySQL 3.06.1 está disponível ao público em geral. As versões 3.06 do Aurora MySQL são compatíveis com o MySQL 8.0.34. Para obter mais informações sobre as mudanças que ocorreram na comunidade, consulte as [Notas de lançamento do MySQL 8.0](#).

Para obter detalhes sobre novos recursos no Aurora MySQL versão 3, consulte [Aurora MySQL versão 3 compatível com o MySQL 8.0](#). Consulte as diferenças entre o Aurora MySQL versão 3 e o Aurora MySQL versão 2 em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão 3](#). Consulte uma comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition em [Comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition](#) no Guia do usuário do Amazon Aurora.

As versões atualmente compatíveis do Aurora MySQL são 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.03.*, 3.04.*, 3.05.*, 3.06.* e 3.07.*.

Você pode realizar um upgrade no local, restaurar um snapshot ou iniciar um upgrade gerenciado blue/green usando o [Amazon RDS Blue/Green Deployments](#) de qualquer cluster Aurora MySQL versão 2 atualmente compatível para um cluster Aurora MySQL versão 3.06.1.

Consulte informações sobre como planejar um upgrade para o Aurora MySQL versão 3 em [Planejar um upgrade da versão principal de um cluster de banco de dados do Aurora MySQL](#). Para obter informações gerais sobre os upgrades do Aurora MySQL, consulte [Como atualizar os clusters de banco de dados de Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Consulte informações sobre solução de problemas em [Solução de problemas para atualização no local de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Problemas de segurança corrigidos e CVEs:

Essa versão inclui todas as correções de CVE da comunidade, incluindo o MySQL 8.0.34. As seguintes correções de CVE estão incluídas:

- [CVE-2023-44487](#)
- [CVE-2024-0853](#)

Melhorias de disponibilidade:

- Correção de um problema que fazia com que uma instância de banco de dados do Aurora MySQL fosse reiniciada ao executar uma consulta paralela.
- Correção de um problema que podia fazer com que uma instância de banco de dados de leitor fosse reiniciada ao ler uma tabela que estava sendo alterada ou descartada na instância de banco de dados de gravador.
- Correção de um problema que causava uma violação de acesso à memória que levava à liberação de um objeto mutex que não pertencia mais à thread.
- Correção de um problema que fazia com que uma instância de banco de dados de gravador do Aurora MySQL fosse reiniciada quando uma sessão de encaminhamento de gravação fosse fechada durante a execução de uma consulta encaminhada.
- Correção de um problema que fazia com que uma instância de banco de dados fosse reiniciada ao lidar com grandes conjuntos de GTID em uma instância habilitada para log binário.
- Correção de um problema que, em raras condições, podia fazer com que uma instância de leitor fosse reiniciada ao realizar consultas SELECT em tabelas com uma restrição de chave estrangeira.
- Correção de um problema que fazia com que uma instância de banco de dados fosse reiniciada ao tentar recuperar o dicionário de dados do InnoDB durante a recuperação do banco de dados.
- Correção de um problema no Aurora Serverless v2 que podia fazer o banco de dados reiniciar durante a o aumento da escala verticalmente.

Melhorias gerais:

- Correção de um problema no código de publicação de métricas em que a memória podia ser usada após ser liberada.

- Correção de um problema que causava reinicializações repetidas do mecanismo de banco de dados devido a um objeto de espaço de tabela undo inexistente.
- Correção de um problema com o truncamento automático de espaços de tabela undo quando eles eram maiores do que o limite [innodb_max_undo_log_size](#) em cenários de atualização.
- Correção de um problema que fornecia um valor incorreto para a variável de status `threads_running` ao usar o Aurora Global Database.
- Correção de um problema em que uma réplica de leitura do log binário (binlog) do Aurora MySQL com a [otimização do índice secundário paralelo](#) ativada era reiniciada ao aplicar alterações de replicação em tabelas com chaves estrangeiras.
- O [Aurora MySQL versão 3.06.0](#) adicionou suporte para a integração ao Amazon Bedrock. Como parte disso, novas palavras-chave reservadas (`accept`, `aws_bedrock_invoke_model`, `aws_sagemaker_invoke_endpoint`, `content_type` e `timeout_ms`) foram adicionadas. No Aurora MySQL versão 3.06.1, essas palavras-chave foram alteradas para palavras-chave não reservadas, que são permitidas como identificadores sem aspas. Consulte mais informações sobre como o MySQL lida com palavras-chave reservadas e não reservadas em [Keywords and reserved words](#) na documentação do MySQL.
- Corrigido um problema que não retornava claramente uma mensagem de erro ao cliente ao invocar o serviço Amazon Bedrock a partir de um cluster de banco de dados Aurora MySQL em um local em que a Região da AWS Amazon Bedrock ainda não estava disponível.
- Correção de um problema que fazia com que uma instância de banco de dados fosse reiniciada devido a informações imprecisas do detentor do bloqueio `rw_lock` ao usar leituras paralelas.
- Correção de um problema que podia fazer com que uma instância de banco de dados fosse reiniciada quando `SHOW VOLUME STATUS` era executado.
- Correção de um problema de gerenciamento de memória que causava uma diminuição na memória liberável ao longo do tempo ao executar consultas `SELECT ... INTO OUTFILE ...`.
- Adição de suporte para que os parâmetros `connection_memory_limit` e `connection_memory_chunk_size` possam ser definidos no nível da sessão, um comportamento semelhante à funcionalidade correspondente no MySQL Community Edition. O parâmetro `connection_memory_limit` define a quantidade máxima de memória que pode ser usada por uma única conexão de usuário. O parâmetro `connection_memory_chunk_size` define o tamanho dos blocos usados para atualizar o [contador global de uso de memória](#).
- Correção de um problema que podia fazer com que uma instância de banco de dados fosse reiniciada quando o armazenamento local na instância de banco de dados atingisse a capacidade total.

- Correção de um problema em que o Performance Schema não era ativado quando o gerenciamento automatizado do Insights de Performance estava ativado para instâncias de banco de dados `db.t4g.medium` e `db.t4g.large`.
- Correção de um problema que podia fazer uma instância de gravador ser reiniciada quando uma instância de leitor usando o encaminhamento de gravação executava uma declaração de linguagem de manipulação de dados (DML) que continha um valor de carimbo de data e hora e o parâmetro de banco de dados `time_zone` é definido como UTC.
- Correção de um problema durante a aplicação de patches com tempo de inatividade zero (ZDP) que impedia que uma instância de banco de dados fechasse as conexões do cliente ao atingir o valor mínimo configurado pelo cliente de `wait_timeout` ou `interactive_timeout`.

Upgrades e migrações:

- Correção de um problema que fazia com que os upgrades ou as migrações falhassem quando a versão de destino do mecanismo de banco de dados Aurora MySQL era 3.04.0 ou posterior. Isso ocorria quando o parâmetro do cluster de banco de dados `lower_case_table_names` era definido como 1, e a ordenação do banco de dados MySQL era incompatível com nomes de tabelas em letras minúsculas.

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bug da comunidade até a versão 8.0.34. Para obter mais informações, consulte [MySQL bugs fixed by Aurora MySQL 3.x database engine updates](#).

Atualizações do mecanismo de banco de dados Aurora MySQL 2024-03-07 (versão 3.06.0) (Obsoleto)

Versão: 3.06.0

O Aurora MySQL 3.06.0 está disponível ao público em geral. As versões 3.06 do Aurora MySQL são compatíveis com o MySQL 8.0.34. Para obter mais informações sobre as mudanças que ocorreram na comunidade, consulte as [Notas de lançamento do MySQL 8.0](#).

Para obter detalhes sobre novos recursos no Aurora MySQL versão 3, consulte [Aurora MySQL versão 3 compatível com o MySQL 8.0](#). Consulte as diferenças entre o Aurora MySQL versão 3 e o Aurora MySQL versão 2 em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão](#)

3. Consulte uma comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition em [Comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition](#) no Guia do usuário do Amazon Aurora.

As versões atualmente compatíveis do Aurora MySQL são 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.03.*, 3.04.*, 3.05.* e 3.06.*.

Você pode realizar um upgrade no local, restaurar um snapshot ou iniciar um upgrade gerenciado blue/green usando o [Amazon RDS Blue/Green Deployments](#) de qualquer cluster do Aurora MySQL versão 2 atualmente compatível para um cluster do Aurora MySQL versão 3.06.0.

Consulte informações sobre como planejar um upgrade para o Aurora MySQL versão 3 em [Planejar um upgrade da versão principal de um cluster de banco de dados do Aurora MySQL](#). Para obter informações gerais sobre os upgrades do Aurora MySQL, consulte [Como atualizar os clusters de banco de dados de Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Consulte informações sobre solução de problemas em [Solução de problemas para atualização no local de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Novos recursos

- O Aurora MySQL versão 3.06.0 é compatível com a integração com o Amazon Bedrock e apresenta as novas palavras-chave reservadas `accept`, `aws_bedrock_invoke_model`, `aws_sagemaker_invoke_endpoint`, `content_type` e `timeout_ms`. Verifique as definições do objeto quanto ao uso das novas palavras-chave reservadas antes de atualizar para a versão 3.06.0. Para mitigar o conflito com as novas palavras-chave reservadas, coloque entre aspas aquelas que foram usadas nas definições do objeto. Consulte mais informações sobre a integração com o Amazon Bedrock e como gerenciar as palavras-chave reservadas em [O que é o Amazon Bedrock?](#) no Guia do usuário do Amazon Aurora. Consulte mais informações em [Keywords and Reserved Words](#), [The INFORMATION_SCHEMA KEYWORDS Table](#) e [Schema Object Names](#) na documentação do MySQL.
- Melhoria no desempenho de réplicas de log binários ao replicar transações para tabelas grandes com mais de um índice secundário. Esse recurso introduz um grupo de threads para aplicar alterações de índice secundário em paralelo em uma réplica de log binário. O recurso é controlado

peelo parâmetro `aurora_binlog_replication_sec_index_parallel_workers` do cluster de banco de dados, que controla o número total de threads paralelos disponíveis para aplicar as alterações do índice secundário. Para obter mais informações, consulte [Otimização da replicação de log binário](#) no Guia do usuário do Amazon Aurora.

- Adição de um novo procedimento armazenado `mysql.rds_set_read_only` que permite alterar o valor da variável global do sistema em instâncias de banco de dados `read_only` no cluster do Aurora MySQL. Consulte mais informações em [Replicação](#) no Guia do usuário do Amazon Aurora.
- Adição de um novo procedimento armazenado `mysql.rds_set_binlog_source_ssl` que permite definir a criptografia em uma réplica de log binário especificando um valor para `SOURCE_SSL`. Consulte mais informações em [Replicação](#) no Guia do usuário do Amazon Aurora.
- [O Amazon Aurora Machine Learning](#) é uma integração otimizada entre o banco de dados Aurora MySQL e os serviços de aprendizado de AWS máquina (ML). O [Amazon Bedrock](#) agora é compatível, permitindo invocar modelos de machine learning no Amazon Bedrock diretamente do cluster de banco de dados do Aurora MySQL usando SQL. Consulte mais informações sobre como usar o Amazon Bedrock com o cluster de banco de dados do Aurora MySQL em [Usar o machine learning do Amazon Aurora com o Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- O Aurora MySQL versão 3.06 adiciona suporte para o [truncamento automático do espaço de tabela undo](#). Essa otimização permite recuperar espaço não utilizado em espaços de tabela de undo após a limpeza dos logs de undo.

Melhorias

Problemas de segurança corrigidos e CVEs:

As seguintes correções de CVE estão incluídas nesta versão:

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

Melhorias de disponibilidade:

- Correção de um problema em que uma instância de banco de dados de réplica de leitura não podia ser iniciada com êxito quando havia alta workload na instância de banco de dados de gravador.
- Correção de um problema em que uma instância de banco de dados de gravador do Aurora MySQL podia fazer failover devido a um defeito na comunicação com o armazenamento do Aurora. A falha ocorria como resultado de uma interrupção na comunicação entre a instância de banco de dados e o armazenamento subjacente após uma atualização de software da instância de armazenamento do Aurora.
- Correção de um problema ao processar consultas INSERT em tabelas particionadas do InnoDB que podia causar um declínio gradual da memória livre na instância de banco de dados.
- Correção de um problema que podia fazer uma instância de banco de dados do Aurora MySQL reiniciar ou fazer failover devido a uma diminuição na memória que pode ser liberada quando o hash join era usado durante a execução de consultas.
- Correção de um problema que podia causar a reinicialização de uma instância de banco de dados ao executar as declarações [SHOW STATUS](#) e [PURGE BINARY LOGS](#) simultaneamente. PURGE BINARY LOGS é uma declaração gerenciada que é executada para honrar o período de retenção de logs binários configurado pelo usuário.
- Correção de um problema que podia fechar o servidor inesperadamente após a execução de declarações de linguagem de manipulação de dados (DML) em uma tabela cujas colunas não virtuais foram reordenadas com uma declaração MODIFY COLUMN ou CHANGE COLUMN.
- Correção de um problema que, durante a reinicialização de uma instância do banco de dados, podia causar uma reinicialização adicional.
- Correção de um problema que podia fazer com que um banco de dados fosse reiniciado quando uma restrição de chave estrangeira UPDATE ou DELETE em cascata era definida em uma tabela em que uma coluna virtual estava envolvida como uma coluna na restrição de chave estrangeira ou como membro da tabela referenciada.
- No Aurora MySQL 2.10, adicionamos suporte para reinicializar um cluster de banco de dados do Aurora com disponibilidade de leitura. Esse recurso permite que as instâncias de banco de dados de leitor permaneçam online enquanto uma instância de banco de dados de gravador é reiniciada. Agora, esse recurso é compatível com Regiões da AWS secundárias nos bancos de dados globais do Aurora MySQL, garantindo que você ainda possa atender solicitações de leitura durante a reinicialização de uma instância de gravador no cluster primário. Anteriormente, quando uma instância de gravador era reiniciada, todas as instâncias de leitor em um cluster secundário do Aurora MySQL também eram reiniciadas. Com essa versão, as instâncias de leitor do cluster secundário continuam a atender solicitações de leitura durante a reinicialização da instância

de gravador, melhorando a disponibilidade de leitura no cluster. Consulte mais informações em [Reinicializar um cluster do Aurora com disponibilidade de leitura](#).

- Correção de um problema que podia interromper a recuperação do banco de dados durante a inicialização caso a reinicialização ocorra durante a execução de operações intensas de inserção envolvendo colunas AUTO_INCREMENT.

Melhorias gerais:

- Correção de um problema que podia fazer com que uma consulta paralela falhasse devido a problemas transitórios de rede durante a leitura de dados do volume do cluster do Aurora.
- Correção de um problema em que o usuário não conseguia interromper nenhuma consulta ou definir tempos limite de sessão para consultas performance_schema.
- Correção de um problema em que a replicação de log binário (binlog) configurada para usar certificados SSL personalizados ([mysql.rds_import_binlog_ssl_material](#)) podia falhar quando a instância de replicação era substituída pelo host.
- Instâncias de banco de dados pequenas, com 4 GiB de memória ou menos, agora encerram as conexões que mais consomem memória quando a instância de banco de dados está sob pressão de memória. Você também pode ajustar o buffer pool para reduzir seu tamanho. Para obter mais informações, consulte [out-of-memoryProblemas do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Alteração da resposta padrão para aurora_oom_response, em todas as classes de instância de banco de dados com mais de 4 GiB de memória, de vazio para print. Para obter mais informações, consulte [out-of-memoryProblemas do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Correção de um problema relacionado ao gerenciamento de arquivos de log de auditoria que podia fazer com que os arquivos de log ficassem inacessíveis para download ou alternância e, em alguns casos, aumentava o uso da CPU.
- Recuperação otimizada de AUTO_INCREMENT chaves para reduzir o tempo de conclusão da restauração de instantâneos, da execução da point-in-time recuperação e da clonagem de clusters de banco de dados com um grande número de tabelas no banco de dados.
- [Corrigido um problema em que o evento wait/io/redo_log_flush não era exibido nas tabelas de resumo de eventos de espera do Performance Schema.](#)
- Adição das métricas Aurora_lockmgr_memory_used e Aurora_lockmgr_buffer_pool_memory_used para rastrear o uso de memória do

gerenciador de bloqueio. Consulte mais informações em [Variáveis de status globais do Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

- Correção de um problema em que pequenas instâncias de réplica de leitura podem apresentar maior atraso de replicação após a atualização de versões do Aurora MySQL anteriores à 2.11.*.
- Correção de um problema que podia causar erros de chave duplicada em colunas AUTO_INCREMENT usando índices decrescentes após uma restauração de snapshot, retrocesso ou operação de clonagem de banco de dados.
- Correção de um problema em que uma consulta SELECT em uma instância de leitor do Aurora podia falhar com o erro a tabela não existe quando a tabela possuía pelo menos um índice de pesquisa de texto completo (FTS) e uma declaração TRUNCATE estava sendo executada na instância de banco de dados de gravador do Aurora.
- Correção de um problema que podia causar um conjunto de resultados incompleto ao executar consultas envolvendo operações LEFT JOIN ou RIGHT JOIN usando o algoritmo de junção de hash com o consulta paralela.

Upgrades e migrações:

- Correção de um problema que podia causar falhas de atualização da versão principal caso houvesse uma coluna FTS_DOC_ID definida pelo usuário presente no esquema da tabela.
- Corrigido um problema que pode causar falhas de atualização do Aurora MySQL versão 2 para o Aurora MySQL versão 3 devido a um problema de sincronização durante o processamento de espaços de tabela do InnoDB.
- Correção de um problema que podia fazer com que os upgrades de versões principais para o Aurora MySQL versão 3 falhassem devido à presença de entradas órfãs para espaços de tabela já excluídos nas tabelas do sistema InnoDB no Aurora MySQL versão 2.

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bugs da comunidade até 8.0.34, incluindo as indicadas a seguir. Para obter mais informações, consulte [MySQL bugs fixed by Aurora MySQL 3.x database engine updates](#).

- Correção de um problema em que o valor da linha de cache podia ser calculado incorretamente, causando uma falha durante a reinicialização do banco de dados em instâncias baseadas no Graviton. (Correção de bug da comunidade #35479763)

- Correção de um problema em que algumas instâncias de subconsultas em rotinas armazenadas nem sempre eram gerenciadas corretamente. (Correção de bug da comunidade #35377192)
- Correção um problema que podia causar maior uso da CPU devido à rotação do certificado TLS em segundo plano (Correção de bug n.º 34284186 da comunidade).
- Correção de um problema em que o InnoDB permitia a adição de colunas INSTANT às tabelas no esquema do sistema MySQL nas versões do Aurora MySQL anteriores à 3.05, o que podia fazer com que o servidor fechasse inesperadamente (reinicialização da instância do banco de dados) após a atualização para o Aurora MySQL versão 3.05.0. (Correção de bug n.º 35625510 da comunidade).

Atualizações do mecanismo de banco de dados Aurora MySQL 2024-01-31 (versão 3.05.2) (Obsoleto)

Versão: 3.05.2

O Aurora MySQL 3.05.2 está disponível ao público em geral. As versões do Aurora MySQL 3.05 são compatíveis com o MySQL 8.0.32. Para obter mais informações sobre as mudanças que ocorreram na comunidade, consulte as [Notas de lançamento do MySQL 8.0](#).

Para obter detalhes sobre novos recursos no Aurora MySQL versão 3, consulte [Aurora MySQL versão 3 compatível com o MySQL 8.0](#). Consulte as diferenças entre o Aurora MySQL versão 3 e o Aurora MySQL versão 2 em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão 3](#). Consulte uma comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition em [Comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition](#) no Guia do usuário do Amazon Aurora.

As versões atualmente compatíveis do Aurora MySQL são 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.03.*, 3.04.* e 3.05.*.

Você pode realizar uma atualização no local, restaurar um snapshot ou iniciar uma atualização gerenciada blue/green usando o [Amazon RDS Blue/Green Deployments](#) de qualquer cluster do Aurora MySQL versão 2 atualmente compatível para um cluster do Aurora MySQL versão 3.05.2.

Consulte informações sobre como planejar um upgrade para o Aurora MySQL versão 3 em [Planejamento de upgrade para o Aurora MySQL versão 3](#). Para obter informações gerais sobre os upgrades do Aurora MySQL, consulte [Como atualizar os clusters de banco de dados de Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Consulte informações sobre a solução de problemas em [Solução de problemas de upgrade com o Aurora MySQL versão 3](#) no Guia do usuário do Amazon Aurora.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Problemas de segurança corrigidos e CVEs:

As seguintes correções de CVE estão incluídas nesta versão:

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

Melhorias de disponibilidade:

- Correção de um problema em que o processamento de consultas INSERT em tabelas particionadas do InnoDB podia causar um declínio gradual da memória livre na instância.
- Correção de um problema que podia causar a reinicialização de uma instância de banco de dados ao executar as declarações [SHOW STATUS](#) e [PURGE BINARY LOGS](#) simultaneamente. PURGE BINARY LOGS é uma declaração gerenciada que é executada para honrar o período de retenção de logs binários configurado pelo usuário.
- Correção de um problema que podia fechar o servidor inesperadamente após a execução de declarações de linguagem de manipulação de dados (DML) em uma tabela cujas colunas não virtuais foram reordenadas com uma declaração MODIFY COLUMN ou CHANGE COLUMN.
- Correção de um problema que, durante a reinicialização de uma instância do banco de dados, podia causar uma reinicialização adicional.

Melhorias gerais:

- Correção de um problema em que o usuário não conseguia interromper nenhuma consulta ou definir tempos limite de sessão para consultas performance_schema.

- Correção de um problema em que a configuração de replicação de log binário (binlog) de usar certificados SSL personalizados ([mysql.rds_import_binlog_ssl_material](#)) podia falhar quando a instância de replicação era substituída pelo host.
- Correção de um problema relacionado ao gerenciamento de arquivos de log de auditoria que podia fazer com que os arquivos de log ficassem inacessíveis para download ou alternância e, em alguns casos, aumentava o uso da CPU.
- Nas versões do Aurora MySQL anteriores à 3.05.2, os usuários não conseguem recuperar a saída de `SHOW ENGINE INNODB STATUS` nas instâncias de banco de dados de leitor do Aurora MySQL. Isso se deve ao [comportamento padrão do InnoDB](#) quando `innodb_read_only` está habilitado.

No Aurora MySQL versão 3.05.2 e posterior, quando executado em uma instância de leitor, a saída `SHOW ENGINE INNODB STATUS` é gravada no log de erros do MySQL, facilitando a solução de problemas.

Consulte mais informações sobre como trabalhar com logs de erros do MySQL em [Logs de erros do Aurora MySQL](#). Consulte mais informações sobre a `SHOW ENGINE INNODB STATUS` em [SHOW ENGINE statement](#) na documentação do MySQL.

Upgrades e migrações:

- Correção de um problema que podia causar falhas de atualização do Aurora MySQL versão 2 para o Aurora MySQL versão 3 quando havia uma coluna `FTS_DOC_ID` definida pelo usuário presente no esquema da tabela.
- Correção de um problema que podia causar falhas de atualização do Aurora MySQL versão 2 para o Aurora MySQL versão 3 devido a um erro de sincronização ao processar espaços de tabela do InnoDB.
- Correção de um problema que podia fazer com que os upgrades de versões principais para o Aurora MySQL versão 3 falhassem devido à presença de entradas órfãs para espaços de tabela já excluídos nas tabelas do sistema InnoDB no Aurora MySQL versão 2.

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bugs da comunidade até 8.0.32, incluindo as indicadas a seguir. Para obter mais informações, consulte [MySQL bugs fixed by Aurora MySQL 3.x database engine updates](#).

- Correção de um problema em que `records_in_range` realizava um número excessivo de leituras de disco para operações INSERT, levando a um declínio gradual no desempenho. (Correção de bug da comunidade #34976138)

Atualizações do mecanismo de banco de dados Aurora MySQL 2023-11-21 (versão 3.05.1) (Obsoleto)

Versão: 3.05.1

O Aurora MySQL 3.05.1 está disponível ao público em geral. As versões do Aurora MySQL 3.05 são compatíveis com o MySQL 8.0.32. Consulte mais informações em [MySQL 8.0 Release Notes](#).

As versões atualmente compatíveis do Aurora MySQL são 2.07.*, 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.*, 3.04.* e 3.05.*.

Você pode atualizar um cluster de banco de dados Aurora MySQL 3.* existente para o Aurora MySQL 3.05.1. Você também pode restaurar um snapshot de qualquer versão atualmente compatível do Aurora MySQL para o Aurora MySQL 3.05.1.

Se você atualizar um banco de dados global do Aurora MySQL para uma versão 3.05.*, deverá atualizar os clusters de banco de dados primário e secundário para exatamente a mesma versão, inclusive o nível de patch. Para obter mais informações sobre como fazer upgrade da versão secundária de um banco de dados global do Aurora, consulte [Minor version upgrades](#).

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Problemas de segurança corrigidos e CVEs listados abaixo:

Essa versão inclui todas as CVEs correções da comunidade, incluindo o MySQL 8.0.32.

- [CVE-2023-38545](#)

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bugs da comunidade até a 8.0.32, incluindo as indicadas abaixo. Para obter mais informações, consulte [MySQL bugs fixed by Aurora MySQL 3.x database engine updates](#).

- Corrigido um problema no InnoDB quando, se uma tabela do MySQL em um esquema do sistema tivesse uma coluna adicionada `INSTANT ADD` entre as versões 3.01 do Aurora MySQL até as versões 3.04 do Aurora MySQL e após o upgrade do Aurora MySQL para a versão 3.05.0, essas tabelas resultariam no fechamento inesperado do servidor. DMLs (Correção de bug nº 35625510 da comunidade)

Atualizações do mecanismo de banco de dados Aurora MySQL 2023-10-30 (versão 3.05.0.1) (Obsoleto)

Versão: 3.05.0.1

O Aurora MySQL 3.05.0.1 geralmente está disponível nas seguintes regiões: Leste dos EUA (Norte da Virgínia), Leste dos EUA (Ohio), Oeste dos EUA (Norte da Califórnia), Oeste dos EUA (Oregon), (Leste dos EUA) e (Oeste dos EUA). AWS GovCloud AWS GovCloud Essa é uma versão inicial, somente com correções de segurança. Essas correções serão implantadas de forma mais ampla em todas as regiões com a próxima versão do patch, 3.05.1. As versões do Aurora MySQL 3.05 são compatíveis com o MySQL 8.0.32. Para obter mais informações sobre as mudanças que ocorreram na comunidade, consulte as [Notas de lançamento do MySQL 8.0](#).

As versões atualmente compatíveis do Aurora MySQL são 2.07.*, 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.*, 3.04.* e 3.05.*.

Você pode atualizar um cluster de banco de dados existente do Aurora MySQL 3.* para o Aurora MySQL 3.05.0.1. Também é possível restaurar um snapshot de qualquer versão atualmente compatível do Aurora MySQL para o Aurora MySQL 3.05.0.1.

Se você atualizar um banco de dados global do Aurora MySQL para uma versão 3.05.*, deverá atualizar os clusters de banco de dados primário e secundário para exatamente a mesma versão, inclusive o nível de patch. Para obter mais informações sobre como fazer upgrade da versão secundária de um banco de dados global do Aurora, consulte [Minor version upgrades](#).

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Problemas de segurança corrigidos e CVEs listados abaixo:

Essa versão inclui todas as CVEs correções da comunidade, incluindo o MySQL 8.0.32.

- [CVE-2023-38545](#)

Atualizações do mecanismo de banco de dados Aurora MySQL 2023-10-25 (versão 3.05.0) (Obsoleto)

Versão: 3.05.0

O Aurora MySQL 3.05.0 está disponível para o público. As versões do Aurora MySQL 3.05 são compatíveis com o MySQL 8.0.32. Para obter mais informações sobre as mudanças que ocorreram na comunidade, consulte as [Notas de lançamento do MySQL 8.0](#).

Para obter detalhes sobre novos recursos no Aurora MySQL versão 3, consulte [Aurora MySQL versão 3 compatível com o MySQL 8.0](#). Consulte as diferenças entre o Aurora MySQL versão 3 e o Aurora MySQL versão 2 em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão 3](#). Consulte uma comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition em [Comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition](#).

As versões atualmente compatíveis do Aurora MySQL são 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.03.*, 3.04.* e 3.05.*.

Você pode realizar uma atualização no local, restaurar um snapshot ou iniciar uma atualização gerenciada blue/green usando o [Amazon RDS Blue/Green Deployments](#) de qualquer cluster do Aurora MySQL versão 2 atualmente compatível para um cluster do Aurora MySQL versão 3.05.0.

Para obter informações sobre como planejar um upgrade para o Aurora MySQL versão 3, consulte [Fazer upgrade para o Aurora MySQL versão 3](#) no Guia do usuário do Amazon Aurora. Para obter informações gerais sobre os upgrades do Aurora MySQL, consulte [Como atualizar os clusters de banco de dados de Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Para obter informações sobre a solução de problemas, consulte [Solução de problemas de atualização com o Aurora MySQL versão 3](#).

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Novos recursos:

- Adicionado suporte para salvar dados de um cluster de banco de dados do Aurora MySQL em arquivos de texto em um bucket do Amazon S3 criptografado com uma chave do KMS (SSE-KMS). Para obter mais informações, consulte [Salvar dados a partir de um cluster de banco de dados do Amazon Aurora MySQL em arquivos de texto de um bucket do Amazon S3](#).
- Apresentada uma nova variável de status global `aurora_tmz_version` para indicar a versão atual das informações de fuso horário (TZ) usadas pelo mecanismo. Os valores seguem a versão do banco de dados de fuso horário da IANA e são formatados como "YYYYsuffix", por exemplo, 2022a e 2023c. Para obter mais informações, consulte [Variáveis de status globais do Aurora MySQL](#).

Problemas de segurança corrigidos e CVEs listados abaixo:

Correções e outras melhorias para ajustar o tratamento em um ambiente gerenciado. Veja as correções adicionais de CVE:

- [CVE-2022-37434](#)

Melhorias de disponibilidade:

- Corrigido um problema em que instâncias de banco de dados do Aurora MySQL que usam consultas paralelas podem sofrer uma reinicialização do banco de dados ao executar um grande número de consultas paralelas simultâneas.
- Corrigido um problema com a contenção de bloqueios causada por um thread de logs de auditoria que acabava levando à alta utilização da CPU e ao tempo limite da aplicação cliente.
- Corrigido um problema que pode fazer com que o conjunto GTID executado seja recuperado incorretamente em um cluster de réplica de log binário (binlog) com o log binário aprimorado habilitado quando qualquer origem de log binário tenha `gtid_mode` definido como ON ou

ON_PERMISSIVE. Esse problema pode fazer com que a instância de gravação do cluster de réplica seja reiniciada mais uma vez durante a recuperação ou levar a resultados incorretos ao consultar o conjunto de GTID executado.

- Corrigido um problema de gerenciamento de memória que pode causar a reinicialização ou o failover de uma instância do banco de dados do Aurora MySQL devido a uma diminuição na memória disponível quando o log binário aprimorado é ativado.
- Corrigido um problema que pode fazer com que a instância do banco de dados seja reiniciada ao tentar ler uma página do banco de dados que pertence a uma tabela descartada.
- Corrigido um problema que pode fazer com que a instância do leitor seja reiniciada quando a instância do gravador aumenta o volume do banco de dados para um múltiplo de 160 GB.
- Corrigido um problema em que uma instância do banco de dados do Aurora MySQL com o recurso aprimorado de log binário habilitado pode travar durante a inicialização da instância de banco de dados enquanto o processo de recuperação do log binário está sendo executado.
- Corrigido um problema em que uma instância do banco de dados do Aurora MySQL pode passar por várias reinicializações durante a inicialização da instância enquanto grandes segmentos de reversão são inicializados.
- Corrigido um problema durante a aplicação de patches sem tempo de inatividade que causa a reinicialização de uma instância fazendo com que as conexões do banco de dados sejam fechadas inesperadamente.
- Corrigido um problema que pode fazer com que a instância do banco de dados seja reiniciada devido a um impasse ao executar declarações [SHOW STATUS](#) e [PURGE BINARY LOGS](#) ao mesmo tempo. A limpeza de logs binários é uma declaração gerenciada que é executada para honrar o período de retenção de registros binários configurado pelo usuário.
- Corrigido um problema que pode causar indisponibilidade do cluster de banco de dados caso a instância do gravador seja reiniciada enquanto o banco de dados está criando ou descartando gatilhos nas tabelas internas do sistema.
- Corrigido um problema que pode fazer com que a instância do banco de dados seja reiniciada devido a uma longa espera de semáforo ao utilizar o recurso de log binário aprimorado em um cluster com uma réplica do Aurora.
- Corrigido um problema que pode fazer com que uma instância de banco de dados seja reiniciada ao executar uma consulta que faz referência a uma função agregada.
- Corrigido um problema que, em raras condições, pode fazer com que a instância de banco de dados seja reiniciada quando o Aurora Serverless v2 tenta incorretamente atualizar o cache da tabela durante a escalabilidade.

- Corrigido um problema em que métodos de acesso de verificação de índice incompatíveis eram considerados para expressões de tabela comuns (CTE) ao materializar tabelas temporárias intermediárias, o que pode levar a comportamentos indesejados, incluindo reinicializações de banco de dados ou resultados de consultas incorretos. Corrigimos esse problema evitando o uso desses métodos incompatíveis de acesso à varredura de índice em tabelas usando o mecanismo TempTable de armazenamento.

Melhorias gerais:

- Corrigido um problema que pode causar indisponibilidade do banco de dados quando o log binário aprimorado é ativado em um cluster de banco de dados do Aurora Serverless v2 executado no Aurora MySQL 3.04.0.
- Removidos os metadados de armazenamento não utilizados antes de gravar no armazenamento do Aurora quando o recurso aprimorado de log binário está habilitado. Isso evita certos cenários em que a reinicialização ou o failover do banco de dados podem ocorrer por conta do aumento da latência de gravação devido ao aumento de bytes transmitidos pela rede.
- Com a adição das tabelas `malloc_stats` e `malloc_stats_totals` no `performance_schema`, três variáveis avançadas do sistema foram adicionadas para controlar o comportamento do Jemalloc, um alocador de memória interna:
 - `aurora_jemalloc_background_thread`.
 - `aurora_jemalloc_dirty_decay_ms`.
 - `aurora_jemalloc_tcache_enabled`.
- Corrigido um problema em que as tabelas de esquema de desempenho específicas do Aurora não eram criadas após uma atualização ou migração.
- Adição de uma nova variável do sistema, `aurora_use_vector_instructions`. Quando esse parâmetro está ativado, o Aurora MySQL usa instruções otimizadas de processamento vetorial para melhorar o desempenho em cargas de trabalho pesadas. I/O Essa configuração é definida como ON por padrão no Aurora MySQL 3.05 e posterior. Para obter mais detalhes, consulte [Parâmetros de configuração do Aurora MySQL](#).
- Corrigido um problema que pode fazer com que as `NumBinaryLogFiles` métricas ativadas CloudWatch exibam resultados incorretos quando o log binário aprimorado está ativado.
- O tempo limite de solicitação das operações de [Machine Learning no Aurora MySQL](#) para o Amazon SageMaker aumentou de 3 para 30 segundos. Isso ajuda a resolver um problema em que os clientes podem ver um aumento no número de novas tentativas ou falhas nas solicitações do Machine Learning no Aurora MySQL para o Amazon Sagemaker ao usar lotes maiores.

- Adicionado suporte para tabelas `malloc_stats` e `malloc_stats_totals` no banco de dados `performance_schema`.
- Atualizada a palavra-chave `FROM` no comando `LOAD DATA FROM S3` para ser opcional. Para obter mais informações, consulte [Carregar dados em um cluster de banco de dados do Amazon Aurora MySQL a partir de arquivos de texto em um bucket do Amazon S3](#).
- Adicionado suporte para o parâmetro `innodb_aurora_instant_alter_column_allowed`, que controla se o algoritmo `INSTANT` pode ser usado para operações `ALTER COLUMN`. Para obter mais informações, consulte [Parâmetros no nível do cluster](#).
- Corrigido um problema que pode impedir o estabelecimento de novas conexões de clientes com o banco de dados quando o encaminhamento de gravação está ativado.
- Corrigido um problema que pode fazer com que a modificação do parâmetro `table_open_cache` do banco de dados não tenha efeito até que a instância de banco de dados seja reiniciada.
- Corrigido um problema que pode causar erros de chave duplicada em colunas `AUTO_INCREMENT` usando índices decrescentes após uma restauração de snapshot, retrocesso ou operação de clonagem de banco de dados.
- Corrigido um problema envolvendo verificações de índice em que um resultado impreciso poderia ser retornado ao executar uma consulta `SELECT` com a cláusula `GROUP BY` e o parâmetro `aurora_parallel_query` definido como `ON`.
- Corrigido um problema que pode causar o esgotamento da memória disponível ao executar consultas na tabela `INFORMATION_SCHEMA INNODB_TABLESPACES`.
- Corrigido um problema em que a instância do leitor não consegue abrir uma tabela, com `ERROR 1146`. Esse problema ocorre ao executar certos tipos de linguagem de definição de dados (DDL) online enquanto o algoritmo `INPLACE` está sendo usado na instância do gravador.
- Corrigido um problema para evitar a reinicialização da instância durante a escalabilidade do Aurora Serverless v2 quando o processo de monitoramento interno envia erroneamente solicitações de escalabilidade duplicadas.
- Foi corrigido um problema que pode causar a reinicialização do banco de dados quando consumidores conectados de log binário (binlog) estão usando um servidor de replicação de log binário duplicado. IDs
- Introdução de um cache de [log de retransmissão](#) na memória para réplicas de log binário gerenciadas pelo Aurora MySQL. Essa melhoria pode ajudar a alcançar um aumento de até 40% no throughput da replicação de log binário. Esse aprimoramento é habilitado automaticamente ao usar a replicação de log binário de um único thread ou ao usar a replicação de vários threads com [posicionamento automático de GTID](#) habilitado.

Upgrades e migrações:

- A atualização do MySQL 5.7 para o MySQL 8.0 com um número muito grande de tabelas em um único banco de dados fez com que o servidor consumisse memória em excesso. Descobriu-se que, durante o processo de verificar se as tabelas poderiam ser atualizadas, obtivemos todos os objetos `Table` do dicionário de dados antecipadamente, processando cada um e obtendo seu nome e, depois, realizamos a [verificação da compatibilidade de versões](#) na lista. Nesse caso, não era necessário obter todos os objetos com antecedência e isso contribuiu muito para o consumo de memória. Para corrigir esse problema, agora obtivemos um objeto `Table` por vez nesses casos, realizando todas as verificações necessárias, obtendo seu nome e liberando o objeto, antes de passar para o próximo. (Bug nº 34526001)
- Melhorou o desempenho das principais atualizações de versões do Aurora MySQL versão 2 para a versão 3 executando verificações de espaço de tabela em paralelo usando todos os v disponíveis na instância do banco de dados. CPUs

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bugs da comunidade até a 8.0.32, incluindo as indicadas abaixo. Para obter mais informações, consulte [MySQL bugs fixed by Aurora MySQL 3.x database engine updates](#).

- Corrigido um problema que pode causar maior utilização da CPU devido à rotação do certificado TLS em segundo plano. (Correção de bug nº 34284186 da comunidade)

Atualizações do mecanismo de banco de dados Aurora MySQL 2026-01-02 (versão 3.04.6, compatível com o MySQL 8.0.28)

Versão: 3.04.6

O Aurora MySQL 3.04.6 está disponível ao público em geral. As versões 3.04 do Aurora MySQL são compatíveis com o MySQL 8.0.28. Para obter mais informações sobre as mudanças que ocorreram na comunidade, consulte as [Notas de lançamento do MySQL 8.0](#).

Para obter detalhes sobre novos recursos no Aurora MySQL versão 3, consulte [Aurora MySQL versão 3 compatível com o MySQL 8.0](#). Consulte as diferenças entre o Aurora MySQL versão 3 e o Aurora MySQL versão 2 em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão 3](#). Consulte uma comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition em [3](#).

[Comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition](#) no Guia do usuário do Amazon Aurora.

Você pode realizar uma atualização local que aproveite um [zero-downtime-patch](#), restaure um snapshot ou inicie uma atualização gerenciada blue/green usando o [Amazon RDS Blue/Green Deployments de qualquer cluster do Aurora MySQL versão 2 atualmente compatível para um cluster do Aurora MySQL versão 3.04.6](#).

Consulte informações sobre como planejar um upgrade para o Aurora MySQL versão 3 em [Planejar um upgrade da versão principal de um cluster do Aurora MySQL](#). Para obter informações gerais sobre os upgrades do Aurora MySQL, consulte [Como atualizar os clusters de banco de dados de Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Consulte informações sobre solução de problemas em [Solução de problemas para atualização no local de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Melhorias na disponibilidade

- Corrigido um problema que poderia causar a reinicialização do mecanismo ao ser executado KILL <query-id> após a execução de uma consulta paralela EXPLAIN FOR CONNECTION <query-id> em execução.
- Problemas corrigidos que poderiam fazer com que a instância do gravador ficasse indisponível se o encaminhamento de gravação fosse desativado ou as instâncias do leitor fossem reiniciadas ao usar o encaminhamento de gravação global ou o encaminhamento de gravação local

Melhorias gerais

- Foi corrigido um problema que fazia com que as instâncias de leitura não gerassem registros de erros quando o encaminhamento de gravação estava ativado e o parâmetro “aurora_replica_read_consistency” era modificado.
- Corrigido um problema que pode fazer com que algumas instruções SQL não sejam registradas no registro de auditoria.

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bug da comunidade até a versão 8.0.28. Para obter mais informações, consulte [Bugs do MySQL corrigidos por atualizações do mecanismo de banco de dados do Aurora MySQL 3.x](#).

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2025-05-05 (versão 3.04.4, compatível com MySQL 8.0.28)

Versão: 3.04.4

O Aurora MySQL 3.04.4 está disponível para o público geral. As versões do Aurora MySQL 3.04 são compatíveis com o MySQL 8.0.28. Para obter mais informações sobre as mudanças que ocorreram na comunidade, consulte as [Notas de lançamento do MySQL 8.0](#).

Para obter detalhes sobre novos recursos no Aurora MySQL versão 3, consulte [Aurora MySQL versão 3 compatível com o MySQL 8.0](#). Consulte as diferenças entre o Aurora MySQL versão 3 e o Aurora MySQL versão 2 em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão 3](#). Consulte uma comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition em [Comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition](#) no Guia do usuário do Amazon Aurora.

Note

Esta versão é designada como uma versão de suporte de longo prazo (LTS). Para obter mais informações, consulte [Versões de suporte de longo prazo \(LTS\) do Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Recomendamos que você não defina o parâmetro `AutoMinorVersionUpgrade` como `true` (nem ative Upgrade automático de versões secundárias no Console de gerenciamento da AWS) para versões LTS. Fazer isso poderia levar o cluster de banco de dados a ser atualizado para a próxima versão-alvo da campanha de Atualização automática de versão secundária, que pode não ser uma versão LTS.

Você pode realizar um upgrade no local, que utiliza um [patch de tempo de inatividade zero](#), restaurar um snapshot ou iniciar um upgrade gerenciado azul/verde usando [implantações azul/verde do Amazon RDS](#) de qualquer cluster do Aurora MySQL versão 2 atualmente compatível para um cluster do Aurora MySQL versão 3.04.4.

Consulte informações sobre como planejar um upgrade para o Aurora MySQL versão 3 em [Planejar um upgrade da versão principal de um cluster do Aurora MySQL](#). Para obter informações gerais sobre os upgrades do Aurora MySQL, consulte [Como atualizar os clusters de banco de dados de Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Consulte informações sobre solução de problemas em [Solução de problemas para atualização no local de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correções de segurança

CVEs críticas:

- [CVE-2024-11053](#)
- [CVE-2024-37371](#)

Melhorias de disponibilidade

- Correção de um problema na réplica em que uma interrupção na rede podia impedir que a conexão com o gravador.
- Correção de um problema que podia causar a reinicialização de uma réplica de log binário (binlog) ao processar um grande número de arquivos de log de retransmissão durante a [recuperação do log de retransmissão](#).
- Correção de um problema que fazia com que uma instância de banco de dados de leitor fosse reiniciada ao executar uma consulta usando o recurso de consulta paralela.
- Correção de um problema que, em raras condições, podia desativar o log binário quando ocorria um erro durante a confirmação de uma transação grande.
- Correção de um problema que podia fazer com que as réplicas de leitura do Aurora fossem reiniciadas no caso de determinados pedidos raros de confirmação de transações na instância de banco de dados de gravador.
- Correção de um problema que podia causar a reinicialização do banco de dados quando [eventos programados](#) eram anulados durante a execução em instâncias com o log binário aprimorado ativado.

- Correção de um problema em que instâncias de banco de dados que utilizam volumes de cluster do Aurora de vários terabytes podiam apresentar maior tempo de inatividade durante a reinicialização devido a falhas na validação do buffer pool do InnoDB.

Melhorias gerais

- Os seguintes privilégios foram adicionados a `rds_superuser_role`:
`FLUSH_OPTIMIZER_COSTS`, `FLUSH_STATUS`, `FLUSH_TABLES` e `FLUSH_USER_RESOURCES`.
Consulte informações sobre `rds_superuser_role` em [Contas de usuário principal da Amazon com o Amazon Aurora](#) no Guia do usuário do Amazon Aurora. Consulte mais informações sobre esses privilégios dinâmicos na [documentação do MySQL](#).
- Correção de um problema em que uma conexão preservada era gerenciada incorretamente durante a aplicação de patches com tempo de inatividade zero (ZDP)/reinicialização com tempo de inatividade zero (ZDR), o que podia fazer com que o cliente esperasse indefinidamente pela conclusão de uma consulta.
- Correção de um problema em que a linha ficava ilegível por meio do índice espacial durante uma atualização.
- Correção de um problema em que uma consulta contendo uma dica do otimizador que fosse anulada durante uma operação de reinicialização com tempo de inatividade zero ou de aplicação de patch com tempo de inatividade zero podia ser gerenciada incorretamente.
- Correção de um problema no qual a latência de confirmação não era medida quando `innodb_flush_log_at_trx_commit` era definido como 0.
- Introdução de otimizações para reduzir o uso de memória durante a recuperação do [dicionário de dados](#) lógico quando há um grande número de tabelas.
- Correção de um problema que fazia com que o comando `SHOW BINARY LOGS` demorasse mais para ser executado em um cluster em que o log binário aprimorado estava habilitado ou já havia estado habilitado. Esse problema também poderia causar maior latência de confirmação se vários comandos `SHOW BINARY LOGS` estivessem sendo executados simultaneamente.
- Correção de um problema de memória associado aos perfis padrão do definidor de visualização.
- Correção de um problema que podia causar falha na conclusão do processo de desativação do recurso de encaminhamento de gravação.
- Correção de um problema que podia impedir o estabelecimento de novas conexões de clientes com o banco de dados quando o encaminhamento de gravação estava ativado.

- Correção de um problema que fazia com que uma instância de banco de dados de gravador fosse reiniciada quando uma instância de leitor usando o encaminhamento de gravação executava uma declaração de DML que continha um valor timestamp e o parâmetro de banco de dados `time_zone` é definido como "UTC".
- Correção de um problema que causava indisponibilidade intermitente de uma réplica de leitura do Aurora ou inconsistências na definição de tabela com erro 'Table does not exist' na réplica devido a consultas de leitura simultâneas na réplica e operações de DDL no gravador.
- Correção de um problema que podia causar um conjunto de resultados incompleto ao executar consultas envolvendo operações LEFT- ou RIGHT-JOIN usando o algoritmo de junção de hash com o Consulta paralela.
- A partir dessa versão do Aurora MySQL, a otimização de inserção rápida não é mais habilitada. Para obter mais informações, consulte [Melhorias de performance do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Upgrades e migrações

- Remoção de perfis padrão que tinham sido criados desnecessariamente durante a atualização do AMS2 para o AMS3.

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bug da comunidade até a versão 8.0.28. Para ter mais informações, consulte . [Bugs do MySQL corrigidos por atualizações do mecanismo de banco de dados do Aurora MySQL 3.x](#) .

- Correção de um problema que resolvia o deadlock quando `FLUSH STATUS`, `COM_CHANGE_USER` e `SHOW PROCESS LIST` eram executados simultaneamente. (Bug nº 35218030)

Atualizações do mecanismo de banco de dados Aurora MySQL 2024-06-26 (versão 3.04.3, compatível com MySQL 8.0.28)

Versão: 3.04.3

O Aurora MySQL 3.04.3 está disponível ao público em geral. As versões 3.04 do Aurora MySQL são compatíveis com o MySQL 8.0.28. Consulte mais informações sobre as mudanças que ocorreram na comunidade em [Notas de lançamento do MySQL 8.0](#).

Para obter detalhes sobre novos recursos no Aurora MySQL versão 3, consulte [Aurora MySQL versão 3 compatível com o MySQL 8.0](#). Consulte as diferenças entre o Aurora MySQL versão 3 e o Aurora MySQL versão 2 em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão 3](#). Consulte uma comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition em [Comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition](#).

Note

Esta versão é designada como uma versão de suporte de longo prazo (LTS). Para obter mais informações, consulte [Versões de suporte de longo prazo \(LTS\) do Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Recomendamos que você não defina o `AutoMinorVersionUpgrade` parâmetro como `true` (nem ative a atualização automática de versões secundárias no Console de gerenciamento da AWS) para versões LTS. Fazer isso poderia levar o cluster de banco de dados a ser atualizado para a próxima versão-alvo da campanha de Atualização automática de versão secundária, que pode não ser uma versão LTS.

As versões atualmente compatíveis do Aurora MySQL são 2.07.9, 2.7.10, 2.11.*, 2.12.*, 3.03.*, 3.04.*, 3.05.*, 3.06.* e 3.07.*.

Você pode realizar um upgrade no local, restaurar um snapshot ou iniciar um upgrade gerenciado blue/green usando o [Amazon RDS Blue/Green Deployments](#) de qualquer cluster do Aurora MySQL versão 2 atualmente disponível para um cluster do Aurora MySQL versão 3.04.3.

Consulte informações sobre como planejar um upgrade para o Aurora MySQL versão 3 em [Planejar um upgrade da versão principal de um cluster de banco de dados do Aurora MySQL](#). Para obter informações gerais sobre os upgrades do Aurora MySQL, consulte [Como atualizar os clusters de banco de dados de Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Consulte informações sobre solução de problemas em [Solução de problemas para atualização no local de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Problemas de segurança corrigidos e CVEs:

Essa versão inclui todas as correções de CVE da comunidade, incluindo o MySQL 8.0.28. As seguintes correções de CVE estão incluídas:

- [CVE-2024-0853](#)

Melhorias de disponibilidade:

- Correção de um problema que fazia com que uma instância de banco de dados do Aurora MySQL fosse reiniciada ao executar uma consulta paralela.
- Correção de um problema que podia fazer com que uma instância de banco de dados de leitor fosse reiniciada ao ler uma tabela que estava sendo alterada ou descartada na instância de banco de dados de gravador.
- Correção de um problema que causava uma violação de acesso à memória que levava à liberação de um objeto mutex que não pertencia mais à thread.
- Correção de um problema que fazia com que uma instância de banco de dados de gravador do Aurora MySQL fosse reiniciada quando uma sessão de encaminhamento de gravação fosse fechada durante a execução de uma consulta encaminhada.
- Correção de um problema que fazia com que uma instância de banco de dados fosse reiniciada ao lidar com grandes conjuntos de GTID em uma instância habilitada para log binário.
- Correção de um problema ao processar consultas INSERT em tabelas particionadas do InnoDB que podia causar um declínio gradual da memória livre na instância de banco de dados.
- Correção de um problema que, em raras condições, podia fazer com que uma instância de leitor fosse reiniciada ao realizar consultas SELECT em tabelas com uma restrição de chave estrangeira.
- Correção de um problema que podia fazer com que um banco de dados fosse reiniciado quando a recuperação do dicionário de dados do InnoDB demorava muito tempo durante a recuperação do banco de dados.
- Correção de um problema que podia fazer com que um banco de dados fosse reiniciado quando uma restrição de chave estrangeira UPDATE ou DELETE em cascata era definida em uma tabela em que uma coluna virtual estava envolvida como uma coluna na restrição de chave estrangeira ou como membro da tabela referenciada.
- Correção de um problema no Aurora Serverless v2 que podia fazer o banco de dados reiniciar durante a o aumento da escala verticalmente.

Melhorias gerais:

- Correção de um problema que fornecia um valor incorreto para a variável de status `threads_running` ao usar o Aurora Global Database.
- Correção de um problema que fazia com que uma instância de banco de dados fosse reiniciada devido a informações imprecisas do detentor do bloqueio `rw_lock` ao usar leituras paralelas.
- Correção de um problema de gerenciamento de memória que causava uma diminuição na memória liberável ao longo do tempo ao executar consultas `SELECT ... INTO OUTFILE ...`.
- Correção de um problema que podia fazer com que uma instância de banco de dados fosse reiniciada quando o armazenamento local na instância de banco de dados atingisse a capacidade total.
- Correção de um problema em que o Performance Schema não era ativado quando o gerenciamento automatizado do Insights de Performance estava ativado para instâncias de banco de dados `db.t4g.medium` e `db.t4g.large`.
- Correção de um problema durante a aplicação de patches com tempo de inatividade zero (ZDP) que impedia que uma instância de banco de dados fechasse as conexões do cliente ao acessar a configuração do cliente de `wait_timeout` ou `interactive_timeout`.
- Correção de um problema em que uma consulta `SELECT` em uma instância de leitor do Aurora podia falhar com o erro a tabela não existe quando a tabela possuía pelo menos um índice de pesquisa de texto completo (FTS) e uma declaração `TRUNCATE` estava sendo executada na instância de banco de dados de gravador do Aurora.

Upgrades e migrações:

- Correção de um problema que fazia com que os upgrades ou as migrações falhassem quando a versão de destino do mecanismo de banco de dados Aurora MySQL era 3.04.0 ou posterior. Isso ocorria quando o parâmetro do cluster de banco de dados `lower_case_table_names` era definido como 1, e a ordenação do banco de dados MySQL era incompatível com nomes de tabelas em letras minúsculas.

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bug da comunidade até a versão 8.0.28. Para obter mais informações, consulte [Bugs do MySQL corrigidos por atualizações do mecanismo de banco de dados do Aurora MySQL 3.x](#).

Atualizações do mecanismo de banco de dados Aurora MySQL

2024-03-15 (versão 3.04.2, compatível com o MySQL 8.0.28)

Versão: 3.04.2

O Aurora MySQL 3.04.2 está disponível ao público em geral. As versões 3.04 do Aurora MySQL são compatíveis com o MySQL 8.0.28. Consulte mais informações sobre as mudanças que ocorreram na comunidade em [Notas de lançamento do MySQL 8.0](#).

Para obter detalhes sobre novos recursos no Aurora MySQL versão 3, consulte [Aurora MySQL versão 3 compatível com o MySQL 8.0](#). Consulte as diferenças entre o Aurora MySQL versão 3 e o Aurora MySQL versão 2 em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão 3](#). Consulte uma comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition em [Comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition](#).

Note

Esta versão é designada como uma versão de suporte de longo prazo (LTS). Para obter mais informações, consulte [Versões de suporte de longo prazo \(LTS\) do Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Recomendamos que você não defina o `AutoMinorVersionUpgrade` parâmetro como `true` (nem ative a atualização automática de versões secundárias no Console de gerenciamento da AWS) para versões LTS. Fazer isso poderia levar o cluster de banco de dados a ser atualizado para a próxima versão-alvo da campanha de Atualização automática de versão secundária, que pode não ser uma versão LTS.

As versões atualmente compatíveis do Aurora MySQL são 2.07.9, 2.7.10, 2.11.*, 2.12.*, 3.03.*, 3.04.*, 3.05.* e 3.06.*.

Você pode realizar um upgrade no local, restaurar um snapshot ou iniciar um upgrade gerenciado blue/green usando o [Amazon RDS Blue/Green Deployments](#) de qualquer cluster do Aurora MySQL versão 2 atualmente disponível para um cluster do Aurora MySQL versão 3.04.2.

Para obter informações sobre como planejar um upgrade para o Aurora MySQL versão 3, consulte [Fazer upgrade para o Aurora MySQL versão 3](#) no Guia do usuário do Amazon Aurora. Para obter informações gerais sobre os upgrades do Aurora MySQL, consulte [Como atualizar os clusters de banco de dados de Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Para obter informações sobre a solução de problemas, consulte [Solução de problemas de atualização com o Aurora MySQL versão 3](#).

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Problemas de segurança corrigidos e CVEs:

As seguintes correções de CVE estão incluídas nesta versão:

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

Melhorias de disponibilidade:

- Correção de um problema em que uma instância de banco de dados de réplica de leitura não podia ser iniciada com êxito quando havia alta workload na instância de banco de dados de gravador.
- Correção de um problema em que uma instância de banco de dados de gravador do Aurora MySQL podia fazer failover devido a um defeito no componente que se comunica com o armazenamento do Aurora. O defeito ocorria como resultado de uma falha na comunicação entre a instância de banco de dados e o armazenamento subjacente após uma atualização de software.
- Correção de um problema que podia causar a reinicialização de uma instância de banco de dados ao executar as declarações [SHOW STATUS](#) e [PURGE BINARY LOGS](#) simultaneamente. PURGE BINARY LOGS é uma declaração gerenciada que é executada para honrar o período de retenção de logs binários configurado pelo usuário.
- Correção de um problema que, durante a reinicialização de uma instância do banco de dados, podia causar uma reinicialização adicional.
- Correção de um problema com a contenção de bloqueios causada por um thread de registo em log de auditoria que provoca alta utilização da CPU e tempos limite da aplicação cliente.

- Correção de um problema em que uma instância do banco de dados do Aurora MySQL podia passar por várias reinicializações durante a inicialização da instância enquanto grandes segmentos de reversão eram inicializados.
- Correção de um problema que podia fazer com que uma instância de banco de dados fosse reiniciada ao executar uma consulta que faz referência a uma função agregada.

Melhorias gerais:

- Correção de um problema que podia fazer com que uma consulta paralela falhasse devido a problemas transitórios de rede durante a leitura de dados do volume do cluster de banco de dados do Aurora.
- Correção de um problema em que o usuário não conseguia interromper nenhuma consulta ou definir tempos limite de sessão para consultas `performance_schema`.
- Correção de um problema em que a replicação de log binário (binlog) configurada para usar certificados SSL personalizados ([mysql.rds_import_binlog_ssl_material](#)) podia falhar quando a instância de replicação era substituída pelo host.
- Correção de um problema relacionado ao gerenciamento de arquivos de log de auditoria que podia fazer com que os arquivos de log ficassem inacessíveis para download ou alternância e, em alguns casos, aumentava o uso da CPU.
- Recuperação otimizada de `AUTO_INCREMENT` chaves para reduzir o tempo de conclusão da restauração de instantâneos, da execução da point-in-time recuperação e da clonagem de clusters de banco de dados com um grande número de tabelas no banco de dados.
- Correção de um problema em que as declarações SQL referentes a algumas tabelas `performance_schema` podiam retornar um erro devido à ausência dessas tabelas após a migração do MySQL Community para as versões 3.04.0 e 3.04.1 do Aurora MySQL.
- Correção de um problema em que pequenas instâncias de réplica de leitura podem apresentar maior atraso de replicação após a atualização de versões do Aurora MySQL anteriores à 2.11.*.
- Correção de um problema que podia causar erros de chave duplicada em colunas `AUTO_INCREMENT` usando índices decrescentes após uma restauração de snapshot, retrocesso ou operação de clonagem de banco de dados.
- Correção de um problema que podia fazer com que as modificações do parâmetro `table_open_cache` do banco de dados não tivesse efeito até que a instância de banco de dados fosse reiniciada.

- Correção de um problema em que a instância de banco de dados de leitor não consegue abrir uma tabela, com ERROR 1146. Esse problema ocorre ao executar certos tipos de declarações de linguagem de definição de dados (DDL) online enquanto o algoritmo INPLACE está sendo usado na instância de banco de dados de gravador.
- Corrigido um problema para evitar a reinicialização da instância durante a escalabilidade do Aurora Serverless v2 quando o processo de monitoramento interno envia erroneamente solicitações de escalabilidade duplicadas.
- Foi corrigido um problema que pode causar a reinicialização do banco de dados quando consumidores conectados de log binário (binlog) estão usando um servidor de replicação de log binário duplicado. IDs

Upgrades e migrações:

- Correção de um problema que podia fazer com que os upgrades de versões principais para o Aurora MySQL versão 3 falhassem devido à presença de entradas órfãs para espaços de tabela já excluídos nas tabelas do sistema InnoDB no Aurora MySQL versão 2.

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bugs da comunidade até 8.0.28, incluindo as indicadas a seguir. Para obter mais informações, consulte [MySQL bugs fixed by Aurora MySQL 3.x database engine updates](#).

- Correção de um problema em que o valor da linha de cache podia ser calculado incorretamente, causando uma falha durante a reinicialização do banco de dados em instâncias baseadas no Graviton. (Correção de bug da comunidade #35479763)
- A execução repetida de uma rotina armazenada, tendo como subconsulta uma declaração SELECT contendo várias condições AND, OR ou XOR levava ao consumo excessivo e a uma possível exaustão eventual da memória virtual. (Correção de bug da comunidade #33852530)

Atualizações do mecanismo de banco de dados Aurora MySQL 2023-11-13 (versão 3.04.1, compatível com MySQL 8.0.28)

Versão: 3.04.1

O Aurora MySQL 3.04.1 está disponível ao público em geral. As versões 3.04 do Aurora MySQL são compatíveis com o MySQL 8.0.28. Consulte mais informações sobre as mudanças que ocorreram na comunidade em [Notas de lançamento do MySQL 8.0](#).

 Note

Esta versão é designada como uma versão de suporte de longo prazo (LTS). Para obter mais informações, consulte [Versões de suporte de longo prazo \(LTS\) do Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Recomendamos que você não defina o `AutoMinorVersionUpgrade` parâmetro como `true` (nem ative a atualização automática de versões secundárias no Console de gerenciamento da AWS) para versões LTS. Fazer isso poderia levar o cluster de banco de dados a ser atualizado para a próxima versão-alvo da campanha de Atualização automática de versão secundária, que pode não ser uma versão LTS.

Para obter detalhes sobre novos recursos no Aurora MySQL versão 3, consulte [Aurora MySQL versão 3 compatível com o MySQL 8.0](#). Consulte as diferenças entre o Aurora MySQL versão 3 e o Aurora MySQL versão 2 em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão 3](#). Consulte uma comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition em [Comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition](#).

As versões atualmente compatíveis do Aurora MySQL são 2.07.9, 2.7.10, 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.*, 3.04.* e 3.05.*.

Você pode realizar um upgrade no local, restaurar um snapshot ou iniciar um upgrade gerenciado blue/green usando o [Amazon RDS Blue/Green Deployments](#) de qualquer cluster do Aurora MySQL versão 2 atualmente disponível para um cluster do Aurora MySQL versão 3.04.1.

Para obter informações sobre como planejar um upgrade para o Aurora MySQL versão 3, consulte [Fazer upgrade para o Aurora MySQL versão 3](#) no Guia do usuário do Amazon Aurora. Para obter informações gerais sobre os upgrades do Aurora MySQL, consulte [Como atualizar os clusters de banco de dados de Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Para obter informações sobre a solução de problemas, consulte [Solução de problemas de atualização com o Aurora MySQL versão 3](#).

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Melhorias de disponibilidade:

- Corrigido um problema em que instâncias de banco de dados do Aurora MySQL que usam consultas paralelas podem sofrer uma reinicialização do banco de dados ao executar um grande número de consultas paralelas simultâneas.
- Correção de um problema que podia fazer com que o conjunto GTID executado fosse recuperado incorretamente em um cluster de réplica de log binário (binlog) com o log binário aprimorado habilitado quando qualquer origem de log binário tinha `gtid_mode` definido como `ON` ou `ON_PERMISSIVE`. Esse problema pode fazer com que a instância de gravação do cluster de réplica seja reiniciada mais uma vez durante a recuperação ou levar a resultados incorretos ao consultar o conjunto de GTID executado.
- Corrigido um problema de gerenciamento de memória que pode causar a reinicialização ou o failover de uma instância do banco de dados do Aurora MySQL devido a uma diminuição na memória disponível quando o log binário aprimorado é ativado.
- Corrigido um problema que pode fazer com que a instância do leitor seja reiniciada quando a instância do gravador aumenta o volume do banco de dados para um múltiplo de 160 GB.
- Corrigido um problema em que uma instância do banco de dados do Aurora MySQL com o recurso aprimorado de log binário habilitado pode travar durante a inicialização da instância de banco de dados enquanto o processo de recuperação do log binário está sendo executado.
- Correção de um problema que pode fazer com que a instância do banco de dados seja reiniciada devido a um impasse ao executar declarações [SHOW STATUS](#) e [PURGE BINARY LOGS](#) ao mesmo tempo. A limpeza de logs binários é uma declaração gerenciada que é executada para honrar o período de retenção de registros binários configurado pelo usuário.
- Corrigido um problema que pode causar indisponibilidade do cluster de banco de dados caso a instância do gravador seja reiniciada enquanto o banco de dados está criando ou descartando gatilhos nas tabelas internas do sistema.
- Corrigido um problema que pode fazer com que a instância do banco de dados seja reiniciada devido a uma longa espera de semáforo ao utilizar o recurso de log binário aprimorado em um cluster com uma réplica do Aurora.

Melhorias gerais:

- Corrigido um problema que pode causar indisponibilidade do banco de dados quando o log binário aprimorado é ativado em um cluster de banco de dados do Aurora Serverless v2 executado no Aurora MySQL 3.04.0.
- Remoção dos metadados de armazenamento não utilizados antes de gravar no armazenamento do Aurora quando o recurso aprimorado de log binário está habilitado. Isso evita certos cenários em que a reinicialização ou o failover do banco de dados podem ocorrer por conta do aumento da latência de gravação devido ao aumento de bytes transmitidos pela rede.
- Corrigido um problema em que as tabelas de esquema de desempenho específicas do Aurora não eram criadas após uma atualização ou migração.
- Corrigido um problema que pode fazer com que as NumBinaryLogFiles métricas ativadas CloudWatch exibam resultados incorretos quando o log binário aprimorado está ativado.

Upgrades e migrações:

- A atualização do MySQL 5.7 para o MySQL 8.0 com um número muito grande de tabelas em um único banco de dados fez com que o servidor consumisse memória em excesso. Descobriu-se que, durante o processo de verificar se as tabelas poderiam ser atualizadas, obtivemos todos os objetos `Table` do dicionário de dados antecipadamente, processando cada um e obtendo seu nome e, depois, realizamos [CHECK TABLE ... FOR UPGRADE](#) na lista. Nesse caso, não era necessário obter todos os objetos com antecedência e isso contribuiu muito para o consumo de memória. Para corrigir esse problema, agora obtivemos um objeto `Table` por vez nesses casos, realizando todas as verificações necessárias, obtendo seu nome e liberando o objeto, antes de passar para o próximo. (Bug nº 34526001)

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bugs da comunidade até 8.0.28, incluindo as indicadas abaixo. Para obter mais informações, consulte [MySQL bugs fixed by Aurora MySQL 3.x database engine updates](#).

- Corrigido um problema que pode causar maior utilização da CPU devido à rotação do certificado TLS em segundo plano (Correção de bug nº 34284186 da comunidade).

Atualizações do mecanismo de banco de dados Aurora MySQL 2023-07-31 (versão 3.04.0), compatível com o MySQL 8.0.28)

Versão: 3.04.0

O Aurora MySQL 3.04.0 está disponível para o público. As versões do Aurora MySQL 3.04 são compatíveis com o MySQL 8.0.28, as versões do Aurora MySQL 3.03 são compatíveis com o MySQL 8.0.26, e as versões do Aurora MySQL 3.02 são compatíveis com o MySQL 8.0.23. Para obter mais informações sobre as alterações da comunidade que ocorreram de 8.0.23 para 8.0.28, consulte as [Notas de lançamento do MySQL 8.0](#).

Note

Esta versão é designada como uma versão de suporte de longo prazo (LTS). Para obter mais informações, consulte [Versões de suporte de longo prazo \(LTS\) do Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Recomendamos que você não defina o `AutoMinorVersionUpgrade` parâmetro como `true` (nem ative a atualização automática de versões secundárias no Console de gerenciamento da AWS) para versões LTS. Fazer isso poderia levar o cluster de banco de dados a ser atualizado para a próxima versão-alvo da campanha de Atualização automática de versão secundária, que pode não ser uma versão LTS.

Para obter detalhes sobre novos recursos no Aurora MySQL versão 3, consulte [Aurora MySQL versão 3 compatível com o MySQL 8.0](#). Consulte as diferenças entre o Aurora MySQL versão 3 e o Aurora MySQL versão 2 em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão 3](#). Consulte uma comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition em [Comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition](#).

As versões atualmente compatíveis do Aurora MySQL são 2.07.9, 2.11.1, 2.11.2, 3.01.*, 3.02.*, 3.03.* e 3.04.0.

Você pode realizar uma atualização no local, restaurar um snapshot ou iniciar uma atualização gerenciada blue/green usando o [Amazon RDS Blue/Green Deployments](#) de qualquer cluster do Aurora MySQL versão 2 atualmente compatível para um cluster do Aurora MySQL versão 3.04.0.

Para obter informações sobre como planejar um upgrade para o Aurora MySQL versão 3, consulte [Fazer upgrade para o Aurora MySQL versão 3](#) no Guia do usuário do Amazon Aurora. Para obter

informações gerais sobre os upgrades do Aurora MySQL, consulte [Como atualizar os clusters de banco de dados de Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Para obter informações sobre a solução de problemas, consulte [Solução de problemas de atualização com o Aurora MySQL versão 3](#).

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

O log binário aprimorado (binlog) do Aurora MySQL atualmente não é compatível com a instância de banco de dados do Aurora Sem Servidor v2 no Aurora MySQL versão 3.04.0. A habilitação desse recurso pode levar à indisponibilidade do banco de dados. Se você precisar usar o log binário aprimorado no Aurora MySQL versão 3.04.0, recomendamos usar uma [classe de instância de banco de dados que não usa a tecnologia servidor](#) ou definir a ACU mínima e máxima da instância de banco de dados do Serverless v2 com o mesmo valor.

Mais informações sobre o log binário aprimorado no Aurora MySQL estão disponíveis no [Guia do usuário do Aurora](#).

Melhorias

Novos recursos:

- Melhora no desempenho das consultas que usam índices de texto completo do InnoDB para pesquisar frases no [modo de linguagem natural](#). Para obter mais informações sobre pesquisas de texto completo no MySQL, consulte [Full-Text Search Functions](#).
- O Amazon Aurora MySQL é compatível com o encaminhamento de gravação local (no cluster). Agora você pode encaminhar operações de gravação de uma instância de banco de dados do leitor para uma instância de banco de dados do gravador em um cluster de banco de dados do Aurora MySQL. Para obter mais informações, consulte [Usar o encaminhamento de gravação local em um cluster de banco de dados do Amazon Aurora MySQL](#).
- Adicionada a capacidade de alterar o valor do parâmetro `aurora_replica_read_consistency` para o recurso [Como usar o encaminhamento](#)

[de gravação em um banco de dados global Amazon Aurora](#) em sessões com autocommit desabilitado. Para obter mais informações, consulte [Parâmetros de configuração para o encaminhamento de gravação](#).

- A partir do Aurora MySQL 3.04, para o recurso [encaminhamento de gravação em um banco de dados global](#), agora você pode definir o valor do parâmetro `aurora_replica_read_consistency` por meio do cluster de banco de dados e dos grupos de parâmetros da instância de banco de dados. Antes do Aurora MySQL versão 3.04, o valor desse parâmetro só podia ser configurado no nível da sessão.

Problemas de segurança corrigidos e CVEs:

- Alterou o SSL/TLS provedor de [AWS-LC](#) OpenSSL para. Isso traz uma série de mudanças, incluindo, mas não se limitando às seguintes:
 - As conexões de banco de dados usando SSL agora podem ser restauradas com reinício com tempo de inatividade zero e aplicação de patch com tempo de inatividade zero ao fazer upgrade do Aurora MySQL versão 3.04.0 para uma versão posterior.
 - Suporte para TLSv1 .3, que inclui suporte para cifras TLS_AES_128_GCM_, SHA256 TLS_AES_256_GCM_ e TLS_ _ _ SSL. SHA384 CHACHA20 POLY1305 SHA256
 - Remoção do suporte para cifras DHE-RSA-* menos seguras.

Para obter mais informações, consulte [Usar TLS com clusters de banco de dados do Aurora MySQL](#).

- Adicionado o privilégio dinâmico `SHOW_ROUTINE` ao `rds_superuser_role`, que permite o acesso às definições e propriedades de todas as rotinas armazenadas, como procedimentos e funções armazenados. Para obter mais detalhes, consulte [SHOW_ROUTINE](#).
- Corrigido um problema que pode fazer com que o log de auditoria perca eventos durante a rotação do arquivo de log de auditoria.
- Suporte habilitado para o protocolo Transport Layer Security (TLS) 1.3 seguro e de alto desempenho, mantendo a compatibilidade com a versão 1.2 do TLS.
- As versões TLS TLSv1 e TLSv1 .1 foram preteridas na comunidade MySQL 8.0.26 e, correspondentemente, no Aurora MySQL 3.03. Esses protocolos agora foram removidos no MySQL 8.0.28 da comunidade e, correspondentemente, no Aurora MySQL 3.04. Por padrão, qualquer cliente seguro que não possa se comunicar via TLS 1.2 ou posterior será rejeitado. Para obter mais informações sobre como se conectar às instâncias de banco de dados usando TLS, consulte [Segurança com o Amazon Aurora MySQL](#).

As seguintes correções de CVE estão incluídas nesta versão:

- [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](#)

Melhorias de disponibilidade:

- Correção de um problema que podia fazer com que o banco de dados fosse reiniciado durante a recuperação de transações longas.
- Correção de um problema com a criptografia de eventos de fluxos de atividades do banco de dados que pode causar reinicializações do banco de dados.
- Corrigido um problema de gerenciamento de memória devido a erros de falta de memória quando o grupo de buffer do InnoDB é inicializado durante a inicialização ou durante a escalabilidade no Aurora Sem Servidor v2. Esse problema pode ter causado a reinicialização da instância de banco de dados ou a degradação do desempenho, incluindo redução de throughput ou aumento da latência.
- Correção de um problema que pode fazer com que uma instância do leitor do Aurora MySQL seja reiniciada durante a execução de uma consulta que utiliza um plano de execução de consulta paralela do Aurora MySQL.
- Correção de um problema que, em determinadas situações, pode fazer com que as instâncias do leitor do Aurora sejam reiniciadas durante uma estimativa de intervalo.
- Correção de um problema que podia interromper a recuperação do banco de dados durante a inicialização caso a reinicialização ocorra durante a execução de operações intensas de inserção envolvendo colunas de incremento automático.
- Corrigido um problema com a Auditoria avançada do Aurora que causa o excesso de registro em log de mensagens informativas no log de erros do Aurora MySQL quando a variável do

servidor `server_audit_events` é definida como ALL ou QUERY. Esse problema podia causar a reinicialização da instância do banco de dados.

- Correção de um problema que podia fazer com que o banco de dados fosse reiniciado durante a reversão de uma declaração INSERT caso a consulta paralela esteja ativada.
- Correção de um problema que podia fazer com que a instância do banco de dados fosse reiniciada ao executar a ferramenta de criação de perfil EXPLAIN ANALYZE em uma consulta que retornava a saída `all select tables were optimized away` na coluna de informações EXTRA. Para obter mais informações, consulte a documentação do MySQL em [EXPLAIN Output Format](#).
- Correção de um problema que podia fazer com que uma instância de leitor da região secundária do banco de dados global do Aurora que usa o encaminhamento de gravação global fosse reiniciada quando uma [declaração de confirmação implícita](#) encaminhada encontrava um erro.
- Correção de um problema que podia fazer com que a instância de gravador em uma região primária do banco de dados global do Aurora fosse reiniciada quando uma consulta SELECT FOR UPDATE é executada usando o encaminhamento global de gravação de uma região secundária do banco de dados global do Aurora.

Melhorias gerais:

- Adicionado um novo procedimento armazenado, `mysql.rds_gtid_purged`, para permitir que os clientes definam a variável do sistema GTID_PURGED. Para obter mais informações, consulte [mysql.rds_gtid_purged](#).
- Adicionados dois novos procedimentos armazenados `mysql.rds_start_replication_until` e `mysql.rds_start_replication_until_gtid`, que permitem que os clientes configurem um local para interromper a replicação de log binário. Para obter mais informações sobre como configurar um local de parada para replicação de log binário no Aurora MySQL, consulte [mysql.rds_start_replication_until](#).
- Corrigido um problema que impede que os [procedimentos armazenados do controle de replicação do Aurora MySQL](#) modifiquem a variável `sql_log_bin` quando chamados de uma sessão com o modo de confirmação automática desabilitado.
- Adição de suporte à replicação lógica para as seguintes instruções de linguagem de controle de dados (DCL): GRANT/REVOKE e CREATE/DROP/ALTER/RENAME USER.
- Corrigido um problema para evitar que as estatísticas do InnoDB fiquem obsoletas, o que às vezes pode gerar um plano de execução de consulta abaixo do ideal que pode levar a um aumento no tempo de execução da consulta.

- Adicionadas duas novas visualizações do sistema `information_schema.aurora_global_db_instance_status` e `information_schema.aurora_global_db_status`. Essas visualizações podem ser usadas para exibir o status e a topologia dos recursos primários e secundários em um cluster de banco de dados global do Aurora MySQL. Os detalhes das duas visualizações do sistema podem ser encontrados aqui, [Tabelas information_schema específicas do Aurora MySQL](#).
- Corrigido um problema em que um usuário não consegue acessar o banco de dados com um caractere curinga no nome do banco de dados após executar a declaração `SET ROLE` com um caractere curinga de escape.
- Corrigido um problema em que os eventos relatados durante o processamento das rotações de log de auditoria podiam não ser gravados no log de auditoria.
- Corrigido um problema em que a criação de uma tabela temporária interna, por meio de uma execução de `TRIGGER`, pode fazer com que uma instância do banco de dados do gravador seja reiniciada.
- Adição de uma nova variável do sistema, `innodb_aurora_max_partitions_for_range`. Em alguns casos, em que estatísticas persistentes não estão disponíveis, esse parâmetro pode ser usado para aprimorar o tempo de execução das estimativas de contagem de linhas em tabelas particionadas. Mais informações podem ser encontradas na documentação, [Parâmetros de configuração do Aurora MySQL](#).
- Corrigido um problema que permite incorretamente que os clientes definam `ROW_FORMAT` como `COMPRESSED` ao criar tabelas particionadas. As tabelas serão convertidas implicitamente para o formato `COMPACT` com um aviso para informar que o Aurora MySQL não é compatível com tabelas compactadas.
- Correção de um problema que podia fazer a replicação de log binário de várias threads parar quando a variável `replica_parallel_type` era definida como `LOGICAL_CLOCK` e a variável `replica_preserve_commit_order` era definida como `ON`. Esse problema podia ocorrer quando uma transação maior que 500 MB era executada na origem.
- Corrigido um problema quando o recurso de [encaminhamento de gravação do banco de dados global](#) está habilitado, o que pode fazer com que as alterações na configuração `performance_schema` das instâncias do leitor nas regiões secundárias sejam encaminhadas involuntariamente para a instância do gravador na região primária.
- Corrigido um problema em que a variável de status do servidor `innodb_buffer_pool_reads` pode não ser atualizada após a leitura de uma página de dados no sistema de arquivos de armazenamento do Aurora.

- A consulta paralela do Aurora MySQL não é compatível ao escolher a configuração de cluster otimizada para E/S do Aurora. Para obter mais informações, consulte [Limitações](#).
- Corrigido um problema quando a consulta paralela é habilitada, o que faz com que o otimizador do plano de consulta escolha um plano de execução ineficiente para determinadas consultas SELECT que se beneficiam de um índice primário ou secundário.
- Atualizadas as definições de fuso horário para a versão 2023c de IANA.
- Introdução de otimizações de desempenho de gerenciamento de arquivos em réplicas de log binário para ajudar a reduzir a contenção ao gravar em arquivos de log de retransmissão.
- Corrigido um problema em que a RPO_LAG_IN_MILLISECONDS coluna na `information_schema.aurora_global_db_status` tabela e na AuroraGlobalDBRPOLag CloudWatch métrica sempre exibia zero, independentemente da carga de trabalho do usuário.
- Apresentação de um novo parâmetro `aurora_tmptable_enable_per_table_limit`. Quando esse parâmetro é ativado, a `tmp_table_size` variável define o tamanho máximo da tabela temporária interna individual na memória criada pelo mecanismo TempTable de armazenamento. Para obter mais informações, consulte [Mecanismo de armazenamento para tabelas temporárias internas \(implícitas\)](#).
- Corrigido um problema em que uma conexão adicional é criada quando o recurso de [encaminhamento de gravação do banco de dados global](#) está habilitado. O problema ocorre quando transações somente leitura em uma instância do leitor encaminham incorretamente uma confirmação implícita para o gravador.
- Corrigido um problema em que os campos `PROCESSLIST_USER` e `PROCESSLIST_HOST` na tabela `performance_schema.threads` não eram preenchidos no gravador na região primária para conexões usando o recurso de [encaminhamento de gravação do banco de dados global](#). Mais informações sobre essa tabela e o Performance Schema podem ser encontradas no Manual de referência do MySQL, [The threads Table](#), e na seção [Visão Geral do Performance Schema](#) do Guia do usuário do Amazon Aurora.
- Corrigido um problema em que a métrica `CommitLatency` do Cloudwatch exibe valores incorretos para instâncias do leitor em regiões secundárias quando o recurso de [encaminhamento de gravação do banco de dados global](#) é usado. Para monitorar a latência da declaração DML encaminhada em clusters de banco de dados secundários, é recomendável usar as métricas `ForwardingReplicaDMLLatency` e `ForwardingWriterDMLLatency`. A latência de confirmação também pode ser observada usando a métrica `CommitLatency` na instância do gravador da região primária. Mais informações estão disponíveis no Guia do usuário do Aurora, nas [CloudWatch métricas da Amazon para encaminhamento de gravação](#).

- Corrigido um problema em que os [procedimentos armazenados do controle de replicação do Aurora MySQL](#) usados para gerenciar e configurar a replicação de registros binários relatam erros incorretamente quando a replicação de logs binários de vários segmentos é configurada definindo o valor da variável [replica_parallel_workers](#) maior que 0.
- Correção de um problema que podia causar alto consumo de CPU quando várias sessões estavam tentando acessar uma página que não existe na memória.

Upgrades e migrações:

- Para executar um upgrade de uma versão secundária para um banco de dados global do Aurora MySQL versão 3.01, 3.02 ou 3.03 para o Aurora MySQL versão 3.04 ou posterior, consulte [Atualizar o Aurora MySQL modificando a versão do mecanismo](#).
- Corrigido um problema que pode causar falhas na verificação prévia do upgrade devido aos erros de inconsistência de esquema relatados para as tabelas `mysql.general_log_backup`, `mysql.general_log`, `mysql.slow_log_backup` e `mysql.slow_log` durante o upgrade do Aurora MySQL 2 para o Aurora MySQL 3. Para obter informações sobre a solução de problemas de upgrade, consulte [Solução de problemas de atualização com o Aurora MySQL versão 3](#).
- Corrigido um problema que pode causar grandes falhas no upgrade da versão durante o upgrade para o Aurora MySQL 3 quando uma definição de gatilho contém uma palavra-chave reservada que não está entre aspas.

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bugs da comunidade até 8.0.28, incluindo as indicadas abaixo. Para obter mais informações, consulte [MySQL bugs fixed by Aurora MySQL 3.x database engine updates](#).

- Corrigido um problema em que um bloco de buffer que contém uma página de tabela temporária intrínseca era realocado durante a passagem da página, causando uma falha de declaração (Bug nº 33715694)
- InnoDB: Impeça que operações online de DDL acessem a out-of-bounds memória (Bug# 34750489, Bug# 108925)
- Corrigido um problema que às vezes pode produzir resultados de consulta incorretos ao processar instruções SQL complexas que consistem em várias expressões de tabela comuns aninhadas (CTEs) (Bug# 34572040, Bug# 34634469, Bug# 33856374)

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2023-12-08 (versão 3.03.3) (obsoleta)

Versão: 3.03.3

O Aurora MySQL 3.03.3 está disponível para o público. As versões do Aurora MySQL 3.03 são compatíveis com o MySQL 8.0.26. Para obter mais informações sobre as alterações da comunidade que ocorreram de 8.0.23 para 8.0.28, consulte as [Notas de lançamento do MySQL 8.0](#).

Para obter detalhes sobre novos recursos no Aurora MySQL versão 3, consulte [Aurora MySQL versão 3 compatível com o MySQL 8.0](#). Consulte as diferenças entre o Aurora MySQL versão 3 e o Aurora MySQL versão 2 em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão 3](#). Consulte uma comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition em [Comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition](#).

As versões atualmente disponíveis do Aurora MySQL são 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.*, 3.04.* e 3.05.*.

Você pode realizar uma atualização no local, restaurar um snapshot ou iniciar um upgrade gerenciado azul/verde usando [implantações azul/verde do Amazon RDS](#) de qualquer cluster do Aurora MySQL versão 2 atualmente disponível para um cluster do Aurora MySQL versão 3.03.3.

Para obter informações sobre como planejar um upgrade para o Aurora MySQL versão 3, consulte [Fazer upgrade para o Aurora MySQL versão 3](#) no Guia do usuário do Amazon Aurora. Para obter informações gerais sobre os upgrades do Aurora MySQL, consulte [Como atualizar os clusters de banco de dados de Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Para obter informações sobre a solução de problemas, consulte [Solução de problemas de atualização com o Aurora MySQL versão 3](#).

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correção de problemas de segurança e CVEs listados abaixo:

Correções e outras melhorias para ajustar o tratamento em um ambiente gerenciado. Veja as correções adicionais de CVE:

- [CVE-2023-38545](#)

Melhorias de disponibilidade:

- Corrigido um problema em que instâncias de banco de dados do Aurora MySQL que usam consultas paralelas podem sofrer uma reinicialização do banco de dados ao executar um grande número de consultas paralelas simultâneas.
- Corrigido um problema que pode fazer com que o conjunto GTID executado seja recuperado incorretamente em um cluster de réplica de log binário (binlog) com o log binário aprimorado habilitado quando qualquer origem de log binário tenha `gtid_mode` definido como ON ou ON_PERMISSIVE. Esse problema pode fazer com que a instância de gravação do cluster de réplica seja reiniciada mais uma vez durante a recuperação ou levar a resultados incorretos ao consultar o conjunto de GTID executado.
- Correção de um problema de gerenciamento de memória que pode causar a reinicialização ou o failover de uma instância do banco de dados do Aurora MySQL devido a uma diminuição na memória disponível quando o log binário aprimorado está habilitado.
- Corrigido um problema que pode fazer com que a instância do leitor seja reiniciada quando a instância do gravador aumenta o volume do banco de dados para um múltiplo de 160 GB.
- Corrigido um problema em que uma instância do banco de dados do Aurora MySQL com o recurso aprimorado de log binário habilitado pode travar durante a inicialização da instância de banco de dados enquanto o processo de recuperação do log binário está sendo executado.
- Correção de um problema durante a aplicação de patches sem tempo de inatividade que causa a reinicialização de uma instância e faz com que as conexões do banco de dados sejam fechadas inesperadamente.
- Correção de um problema que pode fazer com que a instância do banco de dados seja reiniciada devido a um impasse ao executar declarações [SHOW STATUS](#) e [PURGE BINARY LOGS](#) ao mesmo tempo. A limpeza de logs binários é uma declaração gerenciada que é executada para honrar o período de retenção de registros binários configurado pelo usuário.
- Corrigido um problema que pode fazer com que a instância do banco de dados seja reiniciada devido a uma longa espera de semáforo ao utilizar o recurso de log binário aprimorado em um cluster com uma réplica do Aurora.

Melhorias gerais:

- Remoção dos metadados de armazenamento não utilizados antes de gravar no armazenamento do Aurora quando o recurso aprimorado de log binário está habilitado. Isso evita certos cenários em que a reinicialização ou o failover do banco de dados podem ocorrer por conta do aumento da latência de gravação devido ao aumento de bytes transmitidos pela rede.
- Corrigido um problema que pode fazer com que as métricas NumBinaryLogFiles no CloudWatch exibam resultados incorretos quando o log binário aprimorado é habilitado.
- Corrigido um problema que pode fazer com que a modificação do parâmetro `table_open_cache` do banco de dados não tenha efeito até que a instância de banco de dados seja reiniciada.
- Corrigido um problema que pode fazer com que o banco de dados seja reiniciado quando consumidores de log binário (binlog) conectados estão usando IDs de servidor de replicação de log binário (binlog) duplicados.

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bugs da comunidade até 8.0.26, incluindo as indicadas abaixo. Para obter mais informações, consulte [MySQL bugs fixed by Aurora MySQL 3.x database engine updates](#).

- Corrigido um problema que pode causar maior utilização da CPU devido à rotação do certificado TLS em segundo plano (Correção de bug nº 34284186 da comunidade).

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2023-08-29 (versão 3.03.2) (obsoleta)

Versão: 3.03.2

O Aurora MySQL 3.03.2 está disponível para o público. As versões do Aurora MySQL 3.04 são compatíveis com o MySQL 8.0.28, as versões do Aurora MySQL 3.03 são compatíveis com o MySQL 8.0.26, e as versões do Aurora MySQL 3.02 são compatíveis com o MySQL 8.0.23. Para obter mais informações sobre as alterações da comunidade que ocorreram de 8.0.23 para 8.0.28, consulte as [Notas de lançamento do MySQL 8.0](#).

Para obter detalhes sobre novos recursos no Aurora MySQL versão 3, consulte [Aurora MySQL versão 3 compatível com o MySQL 8.0](#). Consulte as diferenças entre o Aurora MySQL versão 3 e o Aurora MySQL versão 2 em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão](#)

[3. Consulte uma comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition em Comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition.](#)

As versões atualmente disponíveis do Aurora MySQL são 2.07.9, 2.07.10, 2.11.*, 3.01.*, 3.02.*, 3.03.* e 3.04.*.

Você pode realizar um upgrade no local, restaurar um snapshot ou iniciar um upgrade gerenciado azul/verde usando [implantações azul/verde do Amazon RDS](#) de qualquer cluster do Aurora MySQL versão 2 atualmente disponível para um cluster do Aurora MySQL versão 3.03.2.

Para obter informações sobre como planejar um upgrade para o Aurora MySQL versão 3, consulte [Fazer upgrade para o Aurora MySQL versão 3](#) no Guia do usuário do Amazon Aurora. Para obter informações gerais sobre os upgrades do Aurora MySQL, consulte [Como atualizar os clusters de banco de dados de Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Para obter informações sobre a solução de problemas, consulte [Solução de problemas de atualização com o Aurora MySQL versão 3](#).

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correção de problemas de segurança e CVEs:

- Correção de um problema que podia fazer com que o log de auditoria perdesse eventos durante a rotação do arquivo de log de auditoria.

As seguintes correções de CVE estão incluídas nesta versão:

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

Melhorias de disponibilidade:

- Correção de um problema que podia fazer com que o banco de dados fosse reiniciado durante a recuperação de transações longas.
- Correção de um problema que podia fazer com que o cluster do banco de dados ficasse indisponível quando a instância do gravador era reiniciada enquanto o banco de dados estava criando ou descartando acionadores nas tabelas internas do sistema.
- Correção de um problema que podia fazer com que uma instância de banco de dados fosse reiniciada ao executar uma consulta que faz referência a uma função agregada.
- Correção de um problema que podia fazer com que o banco de dados fosse reiniciado durante a reversão de uma declaração INSERT caso a consulta paralela esteja ativada.
- A inserção rápida só será habilitada para as tabelas regulares do InnoDB no Aurora MySQL versão 3.03.2 e posterior. Essa otimização não funciona para tabelas temporárias do InnoDB. Consulte mais informações sobre a otimização de inserção rápida em [Aprimoramentos de desempenho do Amazon Aurora MySQL](#).

Melhorias gerais:

- Corrigido um problema em que a instância do leitor não consegue abrir uma tabela, com ERROR 1146. Esse problema ocorre ao executar certos tipos de linguagem de definição de dados (DDL) online enquanto o algoritmo INPLACE está sendo usado na instância do gravador.
- Introdução de otimizações de desempenho de gerenciamento de arquivos em réplicas de log binário para ajudar a reduzir a contenção ao gravar em arquivos de log de retransmissão.
- Corrigido um problema quando a consulta paralela é habilitada, o que faz com que o otimizador do plano de consulta escolha um plano de execução ineficiente para determinadas consultas SELECT que se beneficiam de um índice primário ou secundário.
- Adição de suporte à replicação lógica para as seguintes instruções de linguagem de controle de dados (DCL): GRANT/REVOKE e CREATE/DROP/ALTER/RENAME USER.
- A consulta paralela para o Amazon Aurora MySQL não é compatível ao escolher a configuração de cluster otimizada para E/S do Aurora. Consulte [Limitações](#) para obter mais informações.

Upgrades e migrações:

- Para executar um upgrade de uma versão secundária para um banco de dados global do Aurora MySQL versão 3.01 ou 3.02 para o Aurora MySQL versão 3.03 ou posterior, consulte [Atualizar o Aurora MySQL modificando a versão do mecanismo](#).

- Corrigido um problema que pode causar grandes falhas no upgrade da versão durante o upgrade para o Aurora MySQL versão 3 quando uma definição de gatilho contém uma palavra-chave reservada que não está entre aspas.

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bugs da comunidade até 8.0.26, incluindo as indicadas abaixo. Para obter mais informações, consulte [MySQL bugs fixed by Aurora MySQL 3.x database engine updates](#).

- Corrigido um problema que às vezes pode produzir resultados de consulta incorretos ao processar instruções SQL complexas que consistem em várias expressões de tabela comuns (CTEs) aninhadas. (Bug nº 34572040, Bug nº 34634469, Bug nº 33856374)
- InnoDB: uma condição de corrida entre threads que tenta desinicializar e inicializar estatísticas para a mesma tabela gerou uma falha de declaração. (Bug nº 33135425)
- InnoDB: impede que operações de DDL online acessem memória fora dos limites. (Bug nº 34750489, Bug nº 108925)

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2023-05-11 (versão 3.03.1) (obsoleta)

Versão: 3.03.1

O Aurora MySQL 3.03.1 está disponível para o público. As versões do Aurora MySQL 3.03 são compatíveis com o MySQL 8.0.26, e as versões do Aurora MySQL 3.02 são compatíveis com o MySQL 8.0.23. Para obter mais informações sobre as alterações da comunidade que ocorreram de 8.0.23 para 8.0.26, consulte as [Notas de lançamento do MySQL 8.0](#).

Para obter detalhes sobre novos recursos no Aurora MySQL versão 3, consulte [Aurora MySQL versão 3 compatível com o MySQL 8.0](#). Consulte as diferenças entre o Aurora MySQL versão 3 e o Aurora MySQL versão 2 em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão 3](#). Consulte uma comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition em [Comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition](#).

As versões atualmente compatíveis do Aurora MySQL são 2.07.9, 2.11.1, 2.11.2, 3.01.*, 3.02.* e 3.03.*.

Você pode realizar um upgrade no local ou restaurar um snapshot de qualquer cluster do Aurora MySQL versão 2 compatível no Aurora MySQL 3.03.1.

Para obter informações sobre como planejar um upgrade para o Aurora MySQL versão 3, consulte [Fazer upgrade para o Aurora MySQL versão 3](#) no Guia do usuário do Amazon Aurora. Para obter informações gerais sobre os upgrades do Aurora MySQL, consulte [Como atualizar os clusters de banco de dados de Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Para obter informações sobre a solução de problemas, consulte [Solução de problemas de atualização com o Aurora MySQL versão 3](#).

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Novos atributos:

- A configuração de armazenamento de E/S do Aurora Optimized está disponível a partir da versão 3.03.1. Para obter mais informações, consulte [Configurações de armazenamento para clusters de banco de dados do Amazon Aurora](#).
- Adição de uma nova variável do sistema, `innodb_aurora_max_partitions_for_range`. Em alguns casos, em que estatísticas persistentes não estão disponíveis, esse parâmetro pode ser usado para aprimorar o tempo de execução das estimativas de contagem de linhas em tabelas particionadas. Mais informações podem ser encontradas na documentação, [Parâmetros de configuração do Aurora MySQL](#).

Melhorias de disponibilidade:

- Corrigido um problema que pode fazer com que a instância do banco de dados seja reiniciada devido ao acesso incorreto à memória inválida quando uma conexão é fechada imediatamente após a confirmação de uma transação.
- Corrigido um problema com a Auditoria avançada do Aurora que causa o excesso de registro em log de mensagens informativas no log de erros do Aurora MySQL quando a variável do servidor `server_audit_events` é definida como ALL ou QUERY. Esse problema pode causar a reinicialização da instância do banco de dados.

- Corrigido um problema que, em determinadas situações, pode fazer com que as instâncias do leitor do Aurora sejam reiniciadas ao tentar ler uma página que não está mais acessível durante uma estimativa de intervalo.
- Corrigido um problema que pode fazer com que uma instância do leitor do Aurora MySQL seja reiniciada durante a execução de uma consulta que utiliza um plano de execução de consulta paralela do Aurora.
- Corrigido um problema em que instâncias de banco de dados que usam replicação de log binário podem apresentar um aumento na utilização da CPU e falhas de conexão quando vários consumidores de replicação de log binário estão anexados.
- Corrigido um problema em que métodos de acesso de verificação de índice incompatíveis eram considerados para expressões de tabela comuns (CTE) ao materializar tabelas temporárias intermediárias, o que pode levar a comportamentos indesejados, incluindo reinicializações de banco de dados ou resultados de consultas incorretos. Corrigimos esse problema evitando o uso desses métodos de acesso de verificação de índice incompatíveis em tabelas usando o mecanismo de armazenamento TempTable.
- A inserção rápida não está habilitada nessa versão do Aurora MySQL, devido a um problema que pode causar inconsistências ao executar consultas como `INSERT INTO`, `SELECT` e `FROM`. Consulte mais informações sobre a otimização de inserção rápida em [Aprimoramentos de desempenho do Amazon Aurora MySQL](#).

Melhorias gerais:

- Corrigido um problema que pode causar tempos de execução maiores do que o esperado para a declaração `SHOW BINARY LOGS`. Isso pode levar a uma queda no throughput de confirmações do banco de dados.
- Corrigido um problema que pode fazer com que a exportação paralela de tabelas de usuários com colunas adicionadas usando a funcionalidade `Instant ADD COLUMN` falhe.
- Corrigido um problema em que os eventos relatados durante o processamento das rotações de log de auditoria podiam não ser gravados no log de auditoria.
- Corrigido um problema que pode causar o esgotamento da memória disponível ao executar consultas na tabela `INFORMATION_SCHEMA INNODB_TABLESPACES`.
- Corrigido um problema que permite incorretamente que os clientes definam `ROW_FORMAT` como `COMPRESSED` ao criar tabelas particionadas. As tabelas serão convertidas implicitamente para o formato `COMPACT` com um aviso para informar que o Aurora MySQL não é compatível com tabelas compactadas.

Upgrades e migrações:

- Para executar um upgrade de uma versão secundária para um banco de dados global do Aurora MySQL versão 3.01 ou 3.02 para o Aurora MySQL versão 3.03 ou posterior, consulte [Atualizar o Aurora MySQL modificando a versão do mecanismo](#).
- Corrigido um problema que pode causar falhas na verificação prévia do upgrade devido aos erros de inconsistência de esquema relatados para as tabelas `mysql.general_log_backup`, `mysql.general_log`, `mysql.slow_log_backup` e `mysql.slow_log` durante o upgrade do Aurora MySQL 2 para o Aurora MySQL 3. Para obter informações sobre a solução de problemas de upgrade, consulte [Solução de problemas de atualização com o Aurora MySQL versão 3](#).

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bugs da comunidade até 8.0.26, incluindo as indicadas abaixo. Para obter mais informações, consulte [MySQL bugs fixed by Aurora MySQL 3.x database engine updates](#).

- Corrigido um problema em que um bloco de buffer que contém uma página de tabela temporária intrínseca era realocado durante a passagem da página, causando uma falha de declaração. (Bug nº 33715694)

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2023-03-01 (versão 3.03.0) (obsoleta)

Versão: 3.03.0

O Aurora MySQL 3.03.0 está disponível para o público. As versões do Aurora MySQL 3.03 são compatíveis com o MySQL 8.0.26, e as versões do Aurora MySQL 3.02 são compatíveis com o MySQL 8.0.23. Para obter mais informações sobre as alterações da comunidade que ocorreram de 8.0.23 para 8.0.26, consulte as [Notas de lançamento do MySQL 8.0](#).

Para obter detalhes sobre novos recursos no Aurora MySQL versão 3, consulte [Aurora MySQL versão 3 compatível com o MySQL 8.0](#). Consulte as diferenças entre o Aurora MySQL versão 3 e o Aurora MySQL versão 2 em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão 3](#). Consulte uma comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition em [Comparação do Aurora MySQL versão 3 e do MySQL 8.0 Community Edition](#).

As versões atualmente compatíveis do Aurora MySQL são 2.07.*, 2.11.*, 3.01.*, 3.02.* e 3.03.*.

Você pode realizar um upgrade no local ou restaurar um snapshot de qualquer cluster do Aurora MySQL versão 2 compatível no Aurora MySQL 3.03.0.

Para obter informações sobre como planejar um upgrade para o Aurora MySQL versão 3, consulte [Fazer upgrade para o Aurora MySQL versão 3](#) no Guia do usuário do Amazon Aurora. Para obter informações gerais sobre os upgrades do Aurora MySQL, consulte [Como atualizar os clusters de banco de dados de Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Para obter informações sobre a solução de problemas, consulte [Solução de problemas de atualização com o Aurora MySQL versão 3](#).

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correção de problemas de segurança e CVEs listados abaixo:

Correções e outras melhorias para ajustar o tratamento em um ambiente gerenciado. Correções adicionais do CVE abaixo:

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

Melhorias de disponibilidade:

- Corrigido um problema em que classes de instâncias de banco de dados maiores podem ter problemas durante a reinicialização devido à inicialização do grupo de buffer demorar mais do que o esperado.
- Corrigido um problema em que a instância de banco de dados pode ser reiniciada durante o processo de recuperação do banco de dados quando log binário é habilitado.
- Corrigido um problema que pode causar falhas de conexão nas instâncias do leitor durante a execução de instruções da linguagem de controle de dados (DCL), por exemplo GRANT e REVOKE, ou ao estabelecer novas conexões na instância do gravador.

- Corrigido um problema em que a consulta paralela era usada incorretamente para operações de linguagem de manipulação de dados (DML), como as instruções DELETE e UPDATE, que não são compatíveis atualmente, o que levou à reinicialização da instância de banco de dados. Para obter mais informações sobre as operações compatíveis na consulta paralela, consulte [Limitações](#).
- Corrigido um problema que, em casos raros, pode fazer com que réplicas do Aurora sejam reiniciadas durante a execução simultânea de grandes operações de atualização ou workloads de linguagem de definição de dados (DDL) na instância do gravador e operações de leitura no mesmo conjunto de tabelas na réplica do Aurora.
- Corrigido um problema com a operação de redução de escala da instância do leitor do Aurora Sem Servidor v2 que pode fazer com que essa instância do leitor seja reiniciada e, em alguns casos raros, causar inconsistência de dados.
- Corrigido um problema que pode fazer com que a instância do banco de dados seja reiniciada devido ao acesso incorreto a um local de memória inválida quando uma conexão com a instância do banco de dados é fechada.
- Corrigido um problema que, em raras condições, pode fazer com que a instância de banco de dados seja reiniciada ao processar uma consulta com a cláusula GROUP BY que trunca uma coluna decimal para zero casas decimais.
- Corrigido um problema que pode causar a reinicialização de uma instância de banco de dados devido ao acesso incorreto a um registro ao executar uma consulta de intervalo usando o índice espacial.
- Corrigido um problema que pode causar a reinicialização de uma instância de banco de dados em instâncias de réplica do Aurora MySQL quando as tabelas temporárias internas excedem os valores de memória ou mmap padrão ou configurados pelo cliente.
- Corrigido um problema em que a rotação do log de auditoria avançada pode causar problemas de gerenciamento de memória.
- A inserção rápida não está habilitada nessa versão do Aurora MySQL, devido a um problema que pode causar inconsistências ao executar consultas como INSERT INTO, SELECT e FROM. Consulte mais informações sobre a otimização de inserção rápida em [Aprimoramentos de desempenho do Amazon Aurora MySQL](#).

Melhorias gerais:

- Melhora na latência da consulta de leitura das sessões de encaminhamento de gravação do banco de dados global usando a configuração GLOBAL de consistência de leitura.

- Corrigido um problema em que o valor do parâmetro `wait_timeout` não era respeitado depois que uma sessão do cliente executava os comandos `reset_connection` ou `change_user`.
- Corrigido um problema em que as aplicações podem experimentar maior latência ao se conectar a uma instância de banco de dados quando a instância está passando por um aumento repentino nas conexões de entrada. Duas novas métricas do CloudWatch `AuroraSlowConnectionHandleCount` e `AuroraSlowHandshakeCount`, foram apresentadas para ajudar a solucionar problemas de atrasos no estabelecimento de conexão para instâncias de banco de dados do Aurora MySQL. Mais informações sobre essas métricas podem ser encontradas na documentação de definições de métricas do Aurora CloudWatch, [Métricas do Amazon CloudWatch para o Amazon Aurora](#).
- O parâmetro `temptable_use_mmap` foi desativado, e há previsões de que o suporte será removido em uma versão futura do MySQL. Para obter mais informações, consulte [Mecanismo de armazenamento para tabelas temporárias internas \(implícitas\)](#).
- Corrigido um problema que pode causar tempos de execução maiores do que o esperado para a declaração `SHOW BINARY LOGS`. Isso pode levar a uma queda no throughput de confirmações do banco de dados.

Upgrades e migrações:

- Para executar um upgrade de uma versão secundária para um banco de dados global do Aurora MySQL versão 3.01 ou 3.02 para o Aurora MySQL versão 3.03 ou posterior, consulte [Atualizar o Aurora MySQL modificando a versão do mecanismo](#).
- Corrigido um problema que pode fazer com que os upgrades de versões principais do Aurora MySQL versão 2 para o Aurora MySQL versão 3 falhem quando há um grande número de tabelas (mais de 750 mil) no cluster.
- Corrigido um problema que pode fazer com que os upgrades de versões principais do Aurora MySQL versão 2 para o Aurora MySQL versão 3 falhem porque a migração das tabelas `mysql.innodb_table_stats` e `mysql.innodb_index_stats` demora mais do que o esperado. Esse problema afetou principalmente clusters de banco de dados com milhões de tabelas.
- Corrigido um problema que pode causar falhas no upgrade do Aurora MySQL versão 2 para o Aurora MySQL versão 3 devido a erros de inconsistência de esquema. Esses erros são relatados pelo pré-verificador de atualização das tabelas `mysql.general_log_template` e `mysql.slow_log_template`. Para obter informações sobre a solução de problemas de upgrade, consulte [Solução de problemas de atualização com o Aurora MySQL versão 3](#).

- Corrigido um problema que pode causar falhas de atualização do Aurora MySQL versão 2 para o Aurora MySQL versão 3 devido ao erro `schemaInconsistencyCheck`. Esse erro é causado por inconsistências de esquema na tabela `mysql.table_migration_index_info`, conforme relatado pelo `upgrade-prechecks.log`. Para obter mais informações sobre como solucionar problemas de upgrade para o Aurora MySQL versão 3, consulte [Solução de problemas de atualização com o Aurora MySQL versão 3](#).

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bugs da comunidade até 8.0.26, incluindo as indicadas abaixo. Para obter mais informações, consulte [MySQL bugs fixed by Aurora MySQL 3.x database engine updates](#).

- Corrigido um problema em que alguns tipos de coluna, incluindo JSON e TEXT, às vezes, esgotavam o buffer de classificação se seu tamanho não fosse pelo menos 15 vezes maior do que a maior linha na classificação. Agora, o buffer de classificação precisa ser apenas 15 vezes maior que a maior chave de classificação. (Bug nº 103325, Bug nº 105532, Bug nº 32738705, Bug nº 33501541)
- Corrigido um problema em que o InnoDB nem sempre manipulava alguns nomes legais para partições de tabela corretamente. (Bug nº 32208630)
- Corrigido um problema que, em determinadas condições, pode retornar resultados incorretos devido a um cálculo impreciso da propriedade de nulidade ao executar uma consulta com uma condição OR. (Bug nº 34060289)
- Corrigido um problema que, em determinadas condições, pode retornar resultados incorretos quando as duas condições a seguir são atendidas:
 - uma tabela derivada é mesclada no bloco de consulta externo
 - a consulta inclui uma junção esquerda e uma subconsulta IN(Bug nº 34060289)
- Corrigido um problema em que valores de AUTO_INCREMENT incorretos eram gerados quando o valor máximo da coluna de número inteiro era excedido. O erro ocorreu porque o valor máximo da coluna não foi considerado. O valor anterior de AUTO_INCREMENT válido deveria ter sido retornado nesse caso, causando um erro de chave duplicada. (Bug nº 87926, Bug nº 26906787)
- Corrigido um problema em que não era possível revogar o privilégio DROP no Performance Schema. (Bug nº 33578113)

- Corrigido um problema em que um procedimento armazenado que contém uma declaração IF usando EXISTS, que atuava em uma ou mais tabelas que foram excluídas e recriadas entre as execuções, não era executado corretamente nas invocações subsequentes à primeira. (Bug nº 32855634).
- Corrigido um problema em que uma consulta que faz referência a uma visualização em uma subconsulta e em um bloco de consulta externo pode causar uma reinicialização inesperada. (Bug nº 32324234)

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2023-04-17 (versão 3.02.3) (obsoleta)

Versão: 3.02.3

O Aurora MySQL 3.02.3 está disponível para o público. As versões do Aurora MySQL 3.02 são compatíveis com as versões do MySQL 8.0.23, e as versões do Aurora MySQL 2.x são compatíveis com o MySQL 5.7.

Consulte detalhes sobre novos recursos no Aurora MySQL versão 3 e diferenças entre o Aurora MySQL versão 3 e o Aurora MySQL versão 2 ou o MySQL 8.0 da comunidade em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão 3](#) no Guia do usuário do Amazon Aurora.

As versões atualmente compatíveis do Aurora MySQL são 2.07.*, 2.11.1, 2.11.2, 3.01.*, 3.02.* e 3.03.*.

Você pode realizar um upgrade no local ou restaurar um snapshot de qualquer cluster do Aurora MySQL versão 2 compatível no Aurora MySQL 3.02.3.

Para obter informações sobre como planejar um upgrade para o Aurora MySQL versão 3, consulte [Fazer upgrade para o Aurora MySQL versão 3](#) no Guia do usuário do Amazon Aurora. Com relação ao procedimento de upgrade em si, consulte [Fazer upgrade para o Aurora MySQL versão 3](#) no Guia do usuário do Amazon Aurora. Para obter informações gerais sobre os upgrades do Aurora MySQL, consulte [Como atualizar os clusters de banco de dados de Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Para obter informações sobre a solução de problemas, consulte [Solução de problemas de atualização com o Aurora MySQL versão 3](#).

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Melhorias de disponibilidade:

- Corrigido um problema que pode fazer com que a instância do banco de dados seja reiniciada devido ao acesso incorreto à memória inválida quando uma conexão é fechada imediatamente após a confirmação de uma transação.
- A inserção rápida não está habilitada nessa versão do Aurora MySQL, devido a um problema que pode causar inconsistências ao executar consultas como INSERT INTO, SELECT e FROM. Consulte mais informações sobre a otimização de inserção rápida em [Aprimoramentos de desempenho do Amazon Aurora MySQL](#).

Melhorias gerais:

- Corrigido um problema em que métodos de acesso de verificação de índice incompatíveis eram considerados para expressões de tabela comuns (CTE) ao materializar tabelas temporárias intermediárias, o que pode levar a comportamentos indesejados, incluindo reinicializações de banco de dados ou resultados de consultas incorretos. Esse problema foi corrigido evitando o uso desses métodos de acesso de verificação de índice incompatíveis em tabelas usando o mecanismo de armazenamento TempTable.
- Corrigido um problema que, em casos raros, pode fazer com que uma instância de leitor do Aurora seja reiniciada ao acessar uma tabela que tem grandes atualizações ou operações de linguagem de definição de dados (DDL) em execução simultânea na instância do gravador do Aurora MySQL.
- Corrigido um problema que, em determinadas situações, pode fazer com que as instâncias do leitor do Aurora MySQL sejam reiniciadas ao tentar ler uma página que não está mais acessível durante uma estimativa de intervalo.
- Corrigido um problema em que instâncias de banco de dados que usam replicação de log binário podem apresentar um aumento na utilização da CPU e falhas de conexão quando vários consumidores de replicação de log binário estão anexados.
- Corrigido um problema que pode fazer com que uma instância do leitor do Aurora MySQL seja reiniciada durante a execução de uma consulta que utiliza um plano de execução de consulta paralela do Aurora.

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2022-11-18 (versão 3.02.2) (obsoleta)

Versão: 3.02.2

O Aurora MySQL 3.02.2 está disponível para o público. As versões do Aurora MySQL 3.02 são compatíveis com o MySQL 8.0.23, as versões do Aurora MySQL 2.x são compatíveis com o MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

Consulte detalhes sobre novos recursos no Aurora MySQL versão 3 e diferenças entre o Aurora MySQL versão 3 e o Aurora MySQL versão 2 ou o MySQL 8.0 da comunidade em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão 3](#) no Guia do usuário do Amazon Aurora.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Você pode restaurar um snapshot de qualquer cluster do Aurora MySQL versão 2 compatível no Aurora MySQL 3.02.2.

Para obter informações sobre como planejar um upgrade para o Aurora MySQL versão 3, consulte [Fazer upgrade para o Aurora MySQL versão 3](#) no Guia do usuário do Amazon Aurora. Com relação ao procedimento de upgrade em si, consulte [Fazer upgrade para o Aurora MySQL versão 3](#) no Guia do usuário do Amazon Aurora. Para obter informações gerais sobre os upgrades do Aurora MySQL, consulte [Como atualizar os clusters de banco de dados de Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Para obter informações sobre a solução de problemas, consulte [Solução de problemas de atualização com o Aurora MySQL versão 3](#).

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

O Aurora MySQL versão 3.02.2 está disponível para o público e tem compatibilidade geral com o MySQL 8.0.23 da comunidade.

Correção de problemas de segurança e CVEs listados abaixo:

Correções e outras melhorias para ajustar o tratamento em um ambiente gerenciado. Correções adicionais do CVE abaixo:

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

- [CVE-2022-21444](#)

Melhorias de disponibilidade:

- Corrigido um problema que pode fazer com que a instância do banco de dados seja reiniciada devido ao acesso incorreto à memória inválida quando uma conexão com a instância do banco de dados é fechada explícita ou implicitamente.
- Corrigido um problema que pode fazer com que a inicialização do banco de dados seja interrompida repetidamente em classes de instâncias maiores devido à inicialização do grupo de buffer demorar mais do que o esperado.
- Corrigido um problema que, em raras condições, pode fazer com que a instância de banco de dados seja reiniciada quando o Aurora Serverless v2 tenta incorretamente atualizar o cache da tabela durante a escalabilidade.
- Corrigido um problema que, em raras condições, poderia fazer com que o banco de dados fosse reiniciado ao processar uma consulta com a cláusula GROUP BY que trunca uma coluna decimal para zero casas decimais.
- A inserção rápida não está habilitada nessa versão do Aurora MySQL, devido a um problema que pode causar inconsistências ao executar consultas como INSERT INTO, SELECT e FROM. Consulte mais informações sobre a otimização de inserção rápida em [Aprimoramentos de desempenho do Amazon Aurora MySQL](#).

Melhorias gerais:

- Corrigido um problema que pode causar falhas de atualização do Aurora MySQL versão 2 (compatível com o MySQL 5.7) para o Aurora MySQL versão 3 (compatível com o MySQL 8.0) devido a uma inconsistência de metadados na tabela mysql.host.
- Adicionadas melhorias de desempenho para reduzir o tempo de atualização do Aurora MySQL versão 2 (compatível com MySQL 5.7) para o Aurora MySQL versão 3 (compatível com MySQL

8.0). Ao paralelizar determinadas etapas de atualização, o tempo é reduzido ainda mais ao utilizar classes de instâncias maiores, como db.r6g.16xlarge ou db.r5.24xlarge.

- Adicionado suporte para exibir todos os erros ao fazer upgrade do Aurora MySQL versão 2 (compatível com MySQL 5.7) para o Aurora MySQL versão 3 (compatível com MySQL 8.0) quando versões anteriores estavam limitadas a exibir apenas 50 erros.
- Corrigido um problema que, em raras condições, pode fazer com que contadores de incremento automático fiquem incorretos após o upgrade da versão principal do Aurora MySQL versão 2 (compatível com MySQL 5.7) para o Aurora MySQL versão 3 (compatível com MySQL 8.0).
- Corrigido um problema que pode fazer com que os upgrades de versões principais do Aurora MySQL versão 2 para o Aurora MySQL versão 3 falhem porque a migração das tabelas “mysql.innodb_table_stats” e “mysql.innodb_index_stats” demora mais do que o esperado. Esse problema afetou principalmente clusters de banco de dados com um grande número de tabelas (maior que 1,5 milhão).
- Corrigido um problema que pode fazer com que os principais upgrades de versões do Aurora MySQL versão 2 para o Aurora MySQL versão 3 falhem devido a um defeito no fluxo de trabalho de atualização do mecanismo AMS 8.0, que faz com que os registros de log sejam acumulados no volume do cluster de armazenamento do Aurora e interrompam as operações normais de gravação. Esse problema afetou principalmente clusters de banco de dados com um grande número de tabelas, aproximadamente mais de 750 mil.
- Corrigido um problema que impede que as instâncias ociosas do Aurora MySQL Sem Servidor v2 sejam reduzidas para 0,5 ACUs porque os threads de limpeza do MySQL eram mantidos ativos incorretamente.
- Corrigido um problema em que as aplicações podem apresentar maior latência ao se conectar a uma instância de banco de dados quando a instância está passando por um aumento repentino nas conexões de entrada.
- Apresentadas duas novas métricas do Amazon CloudWatch para ajudar a solucionar problemas de atrasos no estabelecimento de conexão para instâncias de banco de dados do Aurora MySQL. Mais informações sobre as métricas AuroraSlowHandshakeCount e AuroraSlowConnectionHandleCount podem ser encontradas nas [definições de métricas do Aurora CloudWatch](#).

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bugs da comunidade até 8.0.23, incluindo as indicadas abaixo. Para obter mais informações, consulte [MySQL bugs fixed by Aurora MySQL 3.x database engine updates](#).

- Corrigido um problema que, em determinadas condições, pode retornar resultados incorretos devido a um cálculo impreciso da propriedade de nulidade ao executar uma consulta com uma condição OR. (Bug nº 34060289)
- Corrigido um problema que, em determinadas condições, pode retornar resultados incorretos quando as duas condições a seguir são atendidas:
 - Uma tabela derivada é mesclada no bloco de consulta externo.
 - A consulta inclui uma junção esquerda e uma subconsulta IN. (Bug nº 34060289)
- Corrigido um problema em que não era possível revogar o privilégio DROP no Performance Schema. (Bug nº 33578113)
- Corrigido um problema em que um procedimento armazenado que contém uma instrução IF usando EXISTS, que atuava em uma ou mais tabelas que foram excluídas e recriadas entre as execuções, não era executado corretamente nas invocações subsequentes à primeira. (Bug nº 32855634 do MySQL).
- Valores de AUTO_INCREMENT incorretos foram gerados quando o valor máximo da coluna de número inteiro foi excedido. O erro ocorreu porque o valor máximo da coluna não foi considerado. O valor anterior de AUTO_INCREMENT válido deveria ter sido retornado nesse caso, causando um erro de chave duplicada. (Bug nº 87926, Bug nº 26906787)
- Corrigido um problema que pode causar uma falha durante o upgrade de um cluster de banco de dados do Aurora MySQL versão 1 (compatível com MySQL 5.6) que contém uma tabela criada pelo usuário com determinados IDs de tabela. A atribuição desses IDs de tabela pode resultar em IDs de tabela de dicionário de dados conflitantes durante o upgrade do Aurora MySQL versão 2 (compatível com MySQL 5.7) para o Aurora MySQL versão 3 (compatível com MySQL 8.0) (Bug nº 33919635)

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2022-09-07 (versão 3.02.1) (obsoleta)

Versão: 3.02.1

O Aurora MySQL 3.02.1 está disponível para o público. As versões do Aurora MySQL 3.02 são compatíveis com o MySQL 8.0.23, as versões do Aurora MySQL 2.x são compatíveis com o MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

Consulte detalhes sobre novos recursos no Aurora MySQL versão 3 e diferenças entre o Aurora MySQL versão 3 e o Aurora MySQL versão 2 ou o MySQL 8.0 da comunidade em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão 3](#) no Guia do usuário do Amazon Aurora.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Para obter informações sobre como planejar um upgrade para o Aurora MySQL versão 3, consulte [Fazer upgrade para o Aurora MySQL versão 3](#) no Guia do usuário do Amazon Aurora. Com relação ao procedimento de upgrade em si, consulte [Fazer upgrade para o Aurora MySQL versão 3](#) no Guia do usuário do Amazon Aurora. Para obter informações gerais sobre os upgrades do Aurora MySQL, consulte [Como atualizar os clusters de banco de dados de Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Para obter informações sobre a solução de problemas, consulte [Solução de problemas de atualização com o Aurora MySQL versão 3](#).

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

O Aurora MySQL versão 3.02.1 está disponível para o público e tem compatibilidade geral com o MySQL 8.0.23 da comunidade.

Correção de problemas de segurança e CVEs listados abaixo:

Correções e outras melhorias para ajustar o tratamento em um ambiente gerenciado. Correções adicionais do CVE abaixo:

- [CVE-2022-0778](#)

Melhorias de disponibilidade:

- Corrigido um problema que pode causar falha de conexão e alta latência quando várias réplicas de log binário (binlog) do MySQL são anexadas a um nó de gravação do Aurora ou quando há um

grande número de consultas simultâneas de longa execução em conjunto com um aumento nas novas solicitações de conexão.

- Corrigido um problema que faz com que um banco de dados seja reiniciado quando a auditoria avançada para eventos CONNECT é ativada.
- Corrigido um problema que pode causar a reinicialização do banco de dados em instâncias de réplica de leitura do Aurora MySQL quando tabelas temporárias internas esgotam o tamanho alocado na memória e os arquivos mmap definidos como um valor padrão ou configurado pelo cliente.
- Corrigido um problema que pode fazer com que uma réplica de leitura seja reiniciada repetidamente durante operações de DDL simultâneas em procedimentos armazenados.
- A inserção rápida não está habilitada nessa versão do Aurora MySQL, devido a um problema que pode causar inconsistências ao executar consultas como INSERT INTO, SELECT e FROM. Consulte mais informações sobre a otimização de inserção rápida em [Aprimoramentos de desempenho do Amazon Aurora MySQL](#).

Melhorias gerais:

- Adicionado suporte para instâncias R6i.

Informações adicionais:

- O Aurora MySQL versão 3.02.1 não contém suporte para atualizações de versões principais diretamente do Aurora MySQL versão 2 (compatível com MySQL 5.7). Para realizar um upgrade da versão principal para essa versão, primeiro faça upgrade da versão principal para o Aurora MySQL versão 3.02.0 e, depois, faça um upgrade local da versão secundária para a versão 3.02.1 do Aurora MySQL.

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2022-04-20 (versão 3.02.0) (obsoleta)

Versão: 3.02.0

O Aurora MySQL 3.02.0 está disponível para o público. As versões do Aurora MySQL 3.02 são compatíveis com o MySQL 8.0.23, as versões do Aurora MySQL 2.x são compatíveis com o MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

Consulte detalhes sobre novos recursos no Aurora MySQL versão 3 e diferenças entre o Aurora MySQL versão 3 e o Aurora MySQL versão 2 ou o MySQL 8.0 da comunidade em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão 3](#) no Guia do usuário do Amazon Aurora.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Você pode restaurar um snapshot de qualquer cluster do Aurora MySQL versão 2 compatível no Aurora MySQL 3.02.0.

Para obter informações sobre como planejar um upgrade para o Aurora MySQL versão 3, consulte [Fazer upgrade para o Aurora MySQL versão 3](#) no Guia do usuário do Amazon Aurora. Com relação ao procedimento de upgrade em si, consulte [Fazer upgrade para o Aurora MySQL versão 3](#) no Guia do usuário do Amazon Aurora. Para obter informações gerais sobre os upgrades do Aurora MySQL, consulte [Como atualizar os clusters de banco de dados de Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Para obter informações sobre a solução de problemas, consulte [Solução de problemas de atualização com o Aurora MySQL versão 3](#).

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

O Aurora MySQL versão 3.02.0 está disponível para o público e tem compatibilidade geral com o MySQL 8.0.23 da comunidade.

Correção de problemas de segurança e CVEs listados abaixo:

Correções e outras melhorias para ajustar o tratamento em um ambiente gerenciado. Correções adicionais do CVE abaixo:

- [CVE-2021-22946](#)

Novos atributos:

- Amazon Aurora Serverless v2O está disponível para o público em geral. Para obter mais informações, consulte a visão geral do [Amazon Aurora Sem Servidor](#), o [blog](#) e a documentação

[Usar o Aurora Serverless v2](#). Comece hoje mesmo criando um banco de dados do Aurora Serverless v2 usando apenas algumas etapas no Console de gerenciamento da AWS.

Melhorias de disponibilidade:

- Corrigido um problema que pode fazer com que o servidor entre em um loop de reinicialização e cause indisponibilidade ao excluir um registro ou descartar uma tabela contendo duas ou mais colunas de tamanho variável (tipos VARCHAR, VARBINARY, BLOB e TEXT). Para obter mais detalhes sobre os tipos de coluna, consulte [innodb-row-format](#).
- Corrigido um problema em que o tempo limite das conexões existentes e as novas conexões não podiam ser estabelecidas em um cluster com o log binário ativado e com pelo menos um consumidor de log binário anexado, o que resultava na contenção de recursos entre a aplicação e o(s) consumidor(es).
- A memória que pode ser liberada é indicada pela métrica FreeableMemory do CloudWatch. Para obter mais informações, consulte [Métricas do Amazon CloudWatch para o Amazon Aurora](#).
- Corrigido um problema que pode causar a reinicialização ou o failover de uma instância de banco de dados devido a uma diminuição na memória que pode ser liberada quando a replicação de log binário está habilitada.
- Corrigido um problema que pode causar a reinicialização ou o failover de uma instância de banco de dados devido a uma diminuição na memória que pode ser liberada ao definir variáveis de sessão.
- Corrigido um problema que pode causar a reinicialização ou o failover de uma instância de banco de dados devido a uma diminuição na memória que pode ser liberada quando o processo do banco de dados abre um arquivo existente.
- Correção de um problema que, em raras condições, pode causar um erro de entrada duplicada ao inserir novas linhas em uma tabela contendo uma coluna AUTO_INCREMENT em um cluster restaurado do snapshot.
- A inserção rápida não está habilitada nessa versão do Aurora MySQL, devido a um problema que pode causar inconsistências ao executar consultas como INSERT INTO, SELECT e FROM. Consulte mais informações sobre a otimização de inserção rápida em [Aprimoramentos de desempenho do Amazon Aurora MySQL](#).

Melhorias gerais:

- Corrigido um problema em que o status do volume não era exibido ao usar o comando `SHOW VOLUME STATUS`. Para obter mais informações, consulte [AuroraMySQL.Managing.VolumeStatus](#).
- Corrigido um problema que fazia com que as chamadas para [mysql_rds_import_binlog_ssl_material](#) falhassem com o [MySQL server ERROR 3512](#).
- Corrigido um problema em que o atraso na réplica do Aurora era relatado incorretamente em instâncias do leitor do Aurora excluídas.

Upgrades/migração:

- Corrigido um problema que pode causar falhas na migração dos bancos de dados do MySQL 8.0.x para o Aurora MySQL versão 3 devido a um problema na cópia de arquivos e espaços de tabela `ibdata` para o armazenamento do Aurora.
- Corrigido um problema que pode fazer com que os upgrades de clusters do Aurora MySQL versão 2 para o Aurora MySQL versão 3 falhem quando as tabelas de banco de dados possuem uma grande quantidade de dados.
- Correção de um problema que pode causar falhas ao restaurar clusters do Aurora MySQL versão 2 para o Aurora MySQL versão 3 devido a uma falha na criação de [informações de dicionário de dados serializados](#) (SDI) para uma tabela.
- Correção de um problema que pode causar falhas de atualização do Aurora MySQL versão 2 para o Aurora MySQL versão 3 devido a erros de inconsistência de esquema relatados por pré-verificações de upgrade para tabelas do sistema do RDS.
- Corrigido um problema que pode causar falhas ao migrar ou restaurar dos bancos de dados do RDS para MySQL 8.0 ou Aurora MySQL versão 2 para Aurora MySQL versão 3 devido à sintaxe inválida em um procedimento armazenado gerenciado pelo RDS.
- Correção de um problema que pode causar falhas de upgrade do Aurora MySQL 2 para o Aurora MySQL 3 devido a erros de inconsistência de esquema relatados pelas pré-verificações de upgrade das tabelas de [log geral](#) e [log lento](#).

Integração de correções de bug da edição MySQL community

Essa versão inclui todas as correções de bugs da comunidade até 8.0.23, incluindo as indicadas abaixo. Para obter mais informações, consulte [MySQL bugs fixed by Aurora MySQL 3.x database engine updates](#).

- Corrigido o manuseio incorreto de tabelas temporárias usadas para cursores em procedimentos armazenados que poderia resultar em um comportamento inesperado do servidor, [mysqld-8-0-24-bug](#). (Bug nº 32416811)

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2022-04-15 (versão 3.01.1) (obsoleta)

Versão: 3.01.1

O Aurora MySQL 3.01.1 está disponível para o público. As versões do Aurora MySQL 3.01 são compatíveis com o MySQL 8.0.23, as versões do Aurora MySQL 2.x são compatíveis com o MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

Consulte detalhes sobre novos recursos no Aurora MySQL versão 3 e diferenças entre o Aurora MySQL versão 3 e o Aurora MySQL versão 2 ou o MySQL 8.0 da comunidade em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão 3](#) no Guia do usuário do Amazon Aurora.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Você pode restaurar um snapshot de qualquer cluster do Aurora MySQL versão 2 compatível no Aurora MySQL 3.01.1.

Para obter informações sobre como planejar um upgrade para o Aurora MySQL versão 3, consulte [Fazer upgrade para o Aurora MySQL versão 3](#) no Guia do usuário do Amazon Aurora. Com relação ao procedimento de upgrade em si, consulte [Fazer upgrade para o Aurora MySQL versão 3](#) no Guia do usuário do Amazon Aurora. Para obter informações gerais sobre os upgrades do Aurora MySQL, consulte [Como atualizar os clusters de banco de dados de Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Para obter informações sobre a solução de problemas, consulte [Solução de problemas de atualização com o Aurora MySQL versão 3](#).

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

O Aurora MySQL versão 3.01.1 está disponível para o público e tem compatibilidade geral com o MySQL 8.0.23 da comunidade.

O Aurora MySQL versão 3.01.1 é recomendado para upgrades e migrações para um banco de dados do Aurora compatível com MySQL 8.0.

Correção de problemas de segurança e CVEs listados abaixo:

Correções e outras melhorias para ajustar o tratamento em um ambiente gerenciado. Correções adicionais do CVE abaixo:

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

Melhorias de disponibilidade:

- A memória que pode ser liberada é indicada pela métrica `FreeableMemory` do CloudWatch. Para obter mais informações, consulte [Métricas do Amazon CloudWatch para o Amazon Aurora](#).
- Corrigido um problema que pode causar a reinicialização ou o failover de uma instância de banco de dados devido a uma diminuição na memória que pode ser liberada quando a replicação de log binário está habilitada.
- Corrigido um problema que pode causar a reinicialização ou o failover de uma instância de banco de dados devido a uma diminuição na memória que pode ser liberada ao definir variáveis de sessão.
- Corrigido um problema que pode causar a reinicialização ou o failover de uma instância de banco de dados devido a uma diminuição na memória que pode ser liberada quando o processo do banco de dados abre um arquivo existente.
- Correção de um problema que, em raras condições, pode causar um erro de entrada duplicada ao inserir novas linhas em uma tabela contendo uma coluna `AUTO_INCREMENT` em um cluster restaurado do snapshot.
- A inserção rápida não está habilitada nessa versão do Aurora MySQL, devido a um problema que pode causar inconsistências ao executar consultas como `INSERT INTO`, `SELECT` e `FROM`. Consulte mais informações sobre a otimização de inserção rápida em [Aprimoramentos de desempenho do Amazon Aurora MySQL](#).

Melhorias gerais:

- Corrigido um problema em que o status do volume não era exibido ao usar o comando `SHOW VOLUME STATUS`. Para obter mais informações, consulte [AuroraMySQL.Managing.VolumeStatus](#).
- Corrigido um problema que fazia com que as chamadas para [mysql_rds_import_binlog_ssl_material](#) falhassem com o [MySQL server ERROR 3512](#).
- Corrigido um problema em que o atraso na réplica do Aurora era relatado incorretamente em instâncias do leitor do Aurora excluídas.

Upgrades/migração:

- Corrigido um problema que pode causar falhas na migração dos bancos de dados do MySQL 8.0.x para o Aurora MySQL versão 3 devido a um problema na cópia de arquivos e espaços de tabela `ibdata` para o armazenamento do Aurora.
- Corrigido um problema que pode fazer com que os upgrades de clusters do Aurora MySQL versão 2 para o Aurora MySQL versão 3 falhem quando as tabelas de banco de dados possuem uma grande quantidade de dados.
- Correção de um problema que pode causar falhas ao restaurar clusters do Aurora MySQL versão 2 para o Aurora MySQL versão 3 devido a uma falha na criação de [informações de dicionário de dados serializados](#) (SDI) para uma tabela.
- Correção de um problema que pode causar falhas de atualização do Aurora MySQL versão 2 para o Aurora MySQL versão 3 devido a erros de inconsistência de esquema relatados por pré-verificações de upgrade para tabelas do sistema do RDS.
- Corrigido um problema que pode causar falhas ao migrar ou restaurar dos bancos de dados do RDS para MySQL 8.0 ou Aurora MySQL versão 2 para Aurora MySQL versão 3 devido à sintaxe inválida em um procedimento armazenado gerenciado pelo RDS.
- Correção de um problema que pode causar falhas de upgrade do Aurora MySQL 2 para o Aurora MySQL 3 devido a erros de inconsistência de esquema relatados pelas pré-verificações de upgrade das tabelas de [log geral](#) e [log lento](#).

Integração de correções de bug da edição MySQL community

Essa versão inclui todas as correções de bugs da comunidade até 8.0.23, incluindo as indicadas abaixo. Para obter mais informações, consulte [MySQL bugs fixed by Aurora MySQL 3.x database engine updates](#).

- Corrigido o manuseio incorreto de tabelas temporárias usadas para cursores em procedimentos armazenados que poderia resultar em um comportamento inesperado do servidor, [mysqld-8-0-24-bug](#). (Bug nº 32416811)

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-11-18 (versão 3.01.0) (obsoleta)

Versão: 3.01.0

O Aurora MySQL 3.01.0 está disponível para o público. As versões do Aurora MySQL 3.01 são compatíveis com o MySQL 8.0.23, as versões do Aurora MySQL 2.x são compatíveis com o MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

Consulte detalhes sobre novos recursos no Aurora MySQL versão 3 e diferenças entre o Aurora MySQL versão 3 e o Aurora MySQL versão 2 ou o MySQL 8.0 da comunidade em [Comparação do Aurora MySQL versão 2 e do Aurora MySQL versão 3](#) no Guia do usuário do Amazon Aurora.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Você pode restaurar um snapshot de qualquer cluster do Aurora MySQL versão 2 compatível no Aurora MySQL 3.01.0.

Para obter informações sobre como planejar um upgrade para o Aurora MySQL versão 3, consulte [Fazer upgrade para o Aurora MySQL versão 3](#) no Guia do usuário do Amazon Aurora. Com relação ao procedimento de upgrade em si, consulte [Fazer upgrade para o Aurora MySQL versão 3](#) no Guia do usuário do Amazon Aurora. Para obter informações gerais sobre os upgrades do Aurora MySQL, consulte [Como atualizar os clusters de banco de dados de Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Para obter informações sobre a solução de problemas, consulte [Solução de problemas de atualização com o Aurora MySQL versão 3](#).

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

O Aurora MySQL versão 3.01.0 geralmente tem compatibilidade com o MySQL 8.0.23 edição da comunidade. Essa versão inclui as correções de segurança para problemas de Vulnerabilidades e exposições comuns (CVE) a partir do MySQL 8.0.23 edição da comunidade.

O Aurora MySQL versão 3.01.0 contém todas as correções de bugs específicas do Aurora por meio do Aurora MySQL versão 2.10.0.

Para obter detalhes sobre novos recursos no Aurora MySQL versão 3, consulte [Recursos do MySQL 8.0 Community Edition](#) e [Novas otimizações de consultas paralelas](#) no Guia do usuário do Amazon Aurora.

Melhorias de disponibilidade:

- A inserção rápida não está habilitada nessa versão do Aurora MySQL, devido a um problema que pode causar inconsistências ao executar consultas como INSERT INTO, SELECT e FROM. Consulte mais informações sobre a otimização de inserção rápida em [Aprimoramentos de desempenho do Amazon Aurora MySQL](#).

Atualizações feitas no mecanismo de banco de dados do Amazon Aurora MySQL versão 2

Important

O Aurora MySQL versão 2 atingiu o fim do suporte padrão em 31 de outubro de 2024. Para obter mais informações, consulte [Preparação para o fim do suporte padrão do Amazon Aurora, edição compatível com MySQL versão 2](#).

Veja a seguir as atualizações do mecanismo de banco de dados para o Amazon Aurora MySQL versão 2.

- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2025-04-09 \(versão 2.12.5, compatível com MySQL 5.7.44\): versão do Suporte estendido do RDS](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2024-10-23 \(versão 2.12.4, compatível com MySQL 5.7.44\): versão do Suporte estendido do RDS](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2024-07-09 \(versão 2.12.3, compatível com MySQL 5.7.44\): versão do Suporte estendido do RDS](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2024-03-19 \(versão 2.12.2, compatível com MySQL 5.7.44\): versão do Suporte estendido do RDS](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2023-12-28 \(versão 2.12.1, compatível com MySQL 5.7.40\): versão do Suporte estendido do RDS](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2023-10-25 \(versão 2.12.0.1, compatível com MySQL 5.7.40\): versão do Suporte estendido do RDS \(Beta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2023-07-25 \(versão 2.12.0, compatível com MySQL 5.7.40\): versão do Suporte estendido do RDS](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2024-07-19 \(versão 2.11.6, compatível com MySQL 5.7.12\): versão do Suporte estendido do RDS](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2024-03-26 \(versão 2.11.5, compatível com MySQL 5.7.12\): versão do Suporte estendido do RDS \(padrão\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2023-10-17 \(versão 2.11.4, compatível com MySQL 5.7.12\): versão do Suporte estendido do RDS](#)

- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2023-06-09 \(versão 2.11.3, compatível com MySQL 5.7.12\): versão do Suporte estendido do RDS](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2023-03-24 \(versão 2.11.2, compatível com MySQL 5.7.12\): versão do Suporte estendido do RDS](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2023-02-14 \(versão 2.11.1, compatível com MySQL 5.7.12\): versão do Suporte estendido do RDS](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2022-10-25 \(versão 2.11.0, compatível com MySQL 5.7.12\): versão do Suporte estendido do RDS](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2022-11-01 \(versão 2.10.3\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2022-01-26 \(versão 2.10.2\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-10-21 \(versão 2.10.1\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-05-25 \(versão 2.10.0\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-11-12 \(versão 2.09.3\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-02-26 \(versão 2.09.2\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-12-11 \(versão 2.09.1\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-09-17 \(versão 2.09.0\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2022-01-06 \(versão 2.08.4\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-11-12 \(versão 2.08.3\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-08-28 \(versão 2.08.2\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-06-18 \(versão 2.08.1\) \(obsoleta\)](#)

- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-06-02 \(versão 2.08.0\) \(obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados Aurora MySQL 2023-08-15 \(versão 2.07.10\) \(Obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados Aurora MySQL 2023-05-04 \(versão 2.07.9\) \(Obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2022-06-16 \(versão 2.07.8\) \(obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-11-24 \(versão 2.07.7\) \(obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-09-02 \(versão 2.07.6\) \(obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-07-06 \(versão 2.07.5\) \(obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-03-04 \(versão 2.07.4\) \(obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-11-10 \(versão 2.07.3\) \(obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-04-17 \(versão 2.07.2\) \(obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-12-23 \(versão 2.07.1\) \(obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-11-25 \(versão 2.07.0\) \(obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-11-22 \(versão 2.06.0\) \(obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-11-11 \(versão 2.05.0\) \(obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-08-14 \(versão 2.04.9\) \(obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-11-20 \(versão 2.04.8\) \(obsoleto\)](#)

- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-11-14 \(versão 2.04.7\) \(obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-09-19 \(versão 2.04.6\) \(obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-07-08 \(versão 2.04.5\) \(obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-05-29 \(versão 2.04.4\) \(obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-05-09 \(versão 2.04.3\) \(obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-05-02 \(versão 2.04.2\) \(obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-03-25 \(versão 2.04.1\) \(obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-03-25 \(versão 2.04.0\) \(obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-02-07 \(versão 2.03.4\) \(obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-01-18 \(versão 2.03.3\) \(obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-01-09 \(versão 2.03.2\) \(obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-10-24 \(versão 2.03.1\) \(obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-10-11 \(versão 2.03\) \(obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-10-08 \(versão 2.02.5\) \(obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-09-21 \(versão 2.02.4\) \(obsoleto\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-08-23 \(versão 2.02.3\) \(obsoleto\)](#)

- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-06-04 \(versão 2.02.2\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-05-03 \(versão 2.02\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-03-13 \(versão 2.01.1\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-02-06 \(versão 2.01\) \(obsoleta\)](#)

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2025-04-09 (versão 2.12.5, compatível com MySQL 5.7.44): versão do Suporte estendido do RDS

Versão: 2.12.5

O Aurora MySQL 2.12.5 está disponível para o público. As versões do Aurora MySQL 2.12 são compatíveis até o MySQL 5.7.44. Para obter mais informações sobre alterações da comunidade, consulte [Changes in MySQL 5.7.44 \(2022-10-11, General Availability\)](#).

As versões atualmente compatíveis do Aurora MySQL são 2.11.*, 2.12.*, 3.04.*, 3.05.*, 3.06.*, 3.07.* e 3.08.*.

Você pode atualizar um cluster de banco de dados existente do Aurora MySQL 2.* para o Aurora MySQL 2.12.5. Também é possível restaurar um snapshot de qualquer versão atualmente compatível do Aurora MySQL para o Aurora MySQL 2.12.5.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correção de problemas de segurança e CVEs:

Essa versão inclui todas as correções de CVE da comunidade, incluindo o MySQL 5.7.44. As seguintes correções de CVE estão incluídas:

- [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](#)

Melhorias de disponibilidade:

- Correção de um problema na réplica em que uma interrupção na rede podia impedir que a conexão com o gravador.

**Atualizações do mecanismo de banco de dados do Aurora MySQL de 2024-10-23 (versão 2.12.4, compatível com MySQL 5.7.44):
versão do Suporte estendido do RDS**

Versão: 2.12.4

O Aurora MySQL 2.12.4 está disponível para o público. As versões do Aurora MySQL 2.12 são compatíveis até o MySQL 5.7.44. Para obter mais informações sobre alterações da comunidade, consulte [Changes in MySQL 5.7.44 \(2022-10-11, General Availability\)](#).

As versões atualmente compatíveis do Aurora MySQL são 2.11.*, 2.12.*, 3.04.*, 3.05.*, 3.06.* e 3.07.*.

Você pode atualizar um cluster de banco de dados existente do Aurora MySQL 2.* para o Aurora MySQL 2.12.4. Também é possível restaurar um snapshot de qualquer versão atualmente compatível do Aurora MySQL para o Aurora MySQL 2.12.4.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correção de problemas de segurança e CVEs:

Essa versão inclui todas as correções de CVE da comunidade, incluindo o MySQL 5.7.44. As seguintes correções de CVE estão incluídas:

- [CVE-2023-44487](#)
- [CVE-2024-21142](#)
- [CVE-2024-21177](#)
- [CVE-2024-25062](#)

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bug da comunidade até a versão 5.7.44. Para obter mais informações, consulte [Bugs do MySQL corrigidos por atualizações do mecanismo de banco de dados Aurora MySQL 2.x](#).

Recursos não compatíveis com o Aurora MySQL versão 2

No momento, os recursos a seguir não são compatíveis com o Aurora MySQL versão 2 (compatível com MySQL 5.7).

- Lotes de verificação

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- A declaração SQL `CREATE TABLESPACE`
- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- Protocolo X

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2024-07-09 (versão 2.12.3, compatível com MySQL 5.7.44): versão do Suporte estendido do RDS

Versão: 2.12.3

O Aurora MySQL 2.12.3 está disponível para o público. As versões do Aurora MySQL 2.12 são compatíveis até o MySQL 5.7.44. Para obter mais informações sobre alterações da comunidade, consulte [Changes in MySQL 5.7.44 \(2022-10-11, General Availability\)](#).

As versões atualmente compatíveis do Aurora MySQL são 2.11.*, 2.12.*, 3.03.*, 3.04.*, 3.05.*, 3.06.* e 3.07.*.

Você pode atualizar um cluster de banco de dados existente do Aurora MySQL 2.* para o Aurora MySQL 2.12.3. Também é possível restaurar um snapshot de qualquer versão atualmente compatível do Aurora MySQL para o Aurora MySQL 2.12.3.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correção de problemas de segurança e CVEs:

- Correção de um problema de segurança dos procedimentos armazenados do MySQL.

Essa versão inclui todas as correções de CVE da comunidade, incluindo o MySQL 5.7.44. As seguintes correções de CVE estão incluídas:

- [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](#)

Melhorias de disponibilidade:

- Correção de um problema que fazia com que uma instância de banco de dados do Aurora MySQL fosse reiniciada ao executar uma consulta paralela.
- Correção de um problema que podia fazer com que um servidor de banco de dados fosse reiniciado devido ao acesso simultâneo aos recursos de conexão durante a escalabilidade contínua, reinicialização com tempo de inatividade zero (ZDR) e aplicação de patch com tempo de inatividade zero (ZDP).
- Correção um problema que podia fazer com que uma instância de banco de dados de leitor fosse reiniciada ao liberar memória usada para a aplicação de log.
- Correção de um problema em um processo em segundo plano que causava a execução prolongada ou com falha de consultas enquanto a operação em segundo plano eliminava índices temporários.
- Correção de um problema com as rotinas de inicialização que podiam fazer com que as instâncias de banco de dados de gravador fossem reiniciadas devido à inconsistência dos metadados.

- Adição de um indicador para o progresso da recuperação da transação. Isso evita uma possível indisponibilidade, em raras situações, quando a recuperação da transação leva muito tempo para ser concluída.
- Correção de um problema que podia fazer com que uma instância de banco de dados de leitor fosse reiniciada ao ler uma tabela que estava sendo alterada ou descartada na instância de banco de dados de gravador.
- Correção de um problema em que um valor de parâmetro baixo `thread_stack` fazia com que o banco de dados fosse reiniciado repetidamente. O valor mínimo `thread_stack` permitido foi aumentado de 131.072 para 136.192 para garantir a inicialização bem-sucedida e evitar problemas de inicialização.
- Correção de um problema que fazia com que uma instância de banco de dados de leitor fosse reiniciada ao executar uma consulta paralela.
- Correção de um problema que podia fazer com que as réplicas de leitura do Aurora fossem reiniciadas no caso de determinados pedidos raros de confirmação de transações na instância de banco de dados de gravador.
- Correção de um problema que, em casos raros, podia causar a reinicialização de uma instância de banco de dados quando uma transação somente leitura obtinha bloqueios compartilhados.
- Correção de um problema que podia fazer com que uma instância de banco de dados de leitor que usa o encaminhamento de gravação fosse reiniciada quando uma [declaração de confirmação implícita](#) encaminhada encontrava um erro.

Melhorias gerais:

- Correção de um problema que fazia com que as declarações SQL apresentassem erros ou avisos inesperados de violação de chave primária em algumas linhas ao executar declarações INSERT simultâneas em uma tabela que tem uma coluna de chave primária AUTO_INCREMENT e uma coluna de chave exclusiva, e quando uma declaração INSERT tem violações de chave exclusivas em linhas diferentes.
- Correção de um problema que podia gerar resultados de consulta incorretos quando a ZDR restaurava incorretamente as variáveis de sessão definidas como dicas nas consultas.
- Correção de um problema na consulta paralela que fazia com que um conjunto de resultados incompleto fosse retornado ao usar as funções de string integradas LPAD e RPAD.
- Correção de um problema que fazia com que índices de chave estrangeira faltassem nas instâncias de banco de dados de leitor quando uma declaração ALTER TABLE RENAME COLUMN

era executada na instância de banco de dados de gravador em uma tabela com uma chave estrangeira.

- Correção de um problema que podia causar falha na conclusão do processo de desativação do encaminhamento de gravação.
- Correção de um problema durante o a escalabilidade do Aurora Serverless v1 que fazia com que a instância de banco de dados fosse reiniciada devido ao acesso incorreto a uma estrutura de dados interna ao encontrar um ponto de escalabilidade.
- Correção de um problema em que o Performance Schema não era ativado quando o gerenciamento automatizado do Insights de Performance estava ativado para instâncias de banco de dados db.t4g.medium e db.t4g.large.
- O tempo limite de solicitação das operações de [Machine learning do Aurora](#) para o Amazon SageMaker AI aumentou de 3 para 30 segundos. Isso ajuda a resolver um problema em que os clientes podiam perceber um aumento no número de novas tentativas ou falhas em solicitações ao Amazon SageMaker AI feitas a partir do machine learning do Aurora ao usar tamanhos de lote maiores.
- Correção de um problema em que consultas lentas INSERT, DELETE e UPDATE executadas pelo [Agendador de eventos](#) do MySQL não eram registradas no log de consultas lentas, a menos que fossem precedidas por uma consulta lenta SELECT.

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bug da comunidade até a versão 5.7.44. Para obter mais informações, consulte [Bugs do MySQL corrigidos por atualizações do mecanismo de banco de dados Aurora MySQL 2.x](#).

- Correção de um problema em que tabelas temporárias vinculadas a gatilhos durante a execução de declarações podiam causar uma reinicialização inesperada do mecanismo de banco de dados.
- Correção de um defeito que podia fazer com que o servidor saísse quando declarações UPDATE e DELETE de tabela única usando expressões indexadas eram executadas como declarações preparadas. ([Bug #29257254](#))

Recursos não compatíveis com o Aurora MySQL versão 2

No momento, os recursos a seguir não são compatíveis com o Aurora MySQL versão 2 (compatível com MySQL 5.7).

- Lotes de verificação

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- A declaração SQL CREATE TABLESPACE
- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- Protocolo X

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2024-03-19 (versão 2.12.2, compatível com MySQL 5.7.44): versão do Suporte estendido do RDS

Versão: 2.12.2

O Aurora MySQL 2.12.2 está disponível para o público. As versões do Aurora MySQL 2.12 são compatíveis até o MySQL 5.7.44. Para obter mais informações sobre alterações da comunidade, consulte [Changes in MySQL 5.7.44 \(2022-10-11, General Availability\)](#).

As versões atualmente compatíveis do Aurora MySQL são 2.11.*, 2.12.*, 3.03.*, 3.04.*, 3.05.* e 3.06.*.

Você pode atualizar um cluster de banco de dados existente do Aurora MySQL 2.* para o Aurora MySQL 2.12.2. Também é possível restaurar um snapshot de qualquer versão atualmente compatível do Aurora MySQL para o Aurora MySQL 2.12.2.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correção de problemas de segurança e CVEs listados abaixo:

Essa versão inclui todas as correções de CVE da comunidade, incluindo o MySQL 5.7.44. As seguintes correções de CVE estão incluídas:

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

Problemas de segurança:

- Adição de uma correção que garante que as réplicas de log binário usem SSL/TLS como padrão caso a origem seja compatível com conexões criptografadas, independentemente da configuração MASTER_SSL.

Melhorias de disponibilidade:

- Correção de um problema que impedia que uma instância de réplica de leitura fosse iniciada com êxito caso houvesse uma alta workload na instância de gravador.

- Correção de um problema que podia fazer com que uma instância de gravador de banco de dados do Aurora MySQL fizesse failover devido a uma falha no componente que se comunica com o armazenamento do Aurora. A falha ocorria como resultado de uma interrupção na comunicação entre a instância de banco de dados e o armazenamento subjacente após uma atualização de software da instância de armazenamento do Aurora.
- Correção de um problema que, em raras condições, podia fazer com que as instâncias de leitor fossem reiniciadas.
- Correção de um problema no qual um usuário privilegiado podia modificar os [limites de recursos](#) associados ao usuário [rdsadmin](#). Quando definidos incorretamente, esses limites de recursos podem impedir que o agente de monitoramento do RDS monitore a integridade da instância do banco de dados, tornando-o indisponível.

Upgrades e migrações:

- Correção de um problema que ocorria ao tentar iniciar a replicação de log binário para clusters do Aurora MySQL que tinham sido migrados do Amazon RDS MySQL 5.7 e que continham procedimentos armazenados incompatíveis.
- Desativação do agendador de eventos do banco de dados durante uma atualização da versão do Aurora MySQL versão 3. Essa atualização ajuda a evitar qualquer alteração no banco de dados por meio da execução de eventos enquanto a atualização da versão principal está em andamento.

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bug da comunidade até a versão 5.7.44. Para obter mais informações, consulte [MySQL bugs fixed by Aurora MySQL 2.x database engine updates](#).

Recursos não compatíveis com o Aurora MySQL versão 2

No momento, os recursos a seguir não são compatíveis com o Aurora MySQL versão 2 (compatível com MySQL 5.7).

- Lotes de verificação

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- A declaração SQL CREATE TABLESPACE
- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- Protocolo X

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2023-12-28 (versão 2.12.1, compatível com MySQL 5.7.40): versão do Suporte estendido do RDS

Versão: 2.12.1

O Aurora MySQL 2.12.1 está disponível para o público. As versões do Aurora MySQL 2.12 são compatíveis até o MySQL 5.7.40. Para obter mais informações sobre alterações da comunidade, consulte [Changes in MySQL 5.7.40 \(2022-10-11, General Availability\)](#).

As versões atualmente compatíveis do Aurora MySQL são 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.*, 3.04.* e 3.05.*.

Você pode atualizar um cluster de banco de dados existente do Aurora MySQL 2.* para o Aurora MySQL 2.12.1. Também é possível restaurar um snapshot de qualquer versão atualmente compatível do Aurora MySQL para o Aurora MySQL 2.12.1.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

 Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correção de problemas de segurança e CVEs listados abaixo:

Essa versão inclui todas as correções de CVEs da comunidade, incluindo o MySQL 5.7.44.

Correções e outras melhorias para ajustar o tratamento em um ambiente gerenciado. Veja as correções adicionais de CVE:

- [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](#)
- Correção do processamento de tokens de um único caractere por um plug-in de analisador de pesquisa de texto completo (FTS) (Bug n.º 35432973)
- Correção de um problema em que os eventos relatados durante o processamento de alternâncias de log de auditoria podiam não ser gravados no log de auditoria

Novos atributos:

- Compatibilidade adicionada com a replicação de log binário (binlog) de vários threads, em que o thread SQL na réplica do log binário aplica eventos de log binário em paralelo quando possível. Saiba mais sobre as opções de configuração para ajudar a ajustar a replicação de vários threads no [Guia do usuário do Aurora](#).

Melhorias de disponibilidade:

- Corrigido um problema em que instâncias de banco de dados do Aurora MySQL que usam consultas paralelas podem sofrer uma reinicialização do banco de dados ao executar um grande número de consultas paralelas simultâneas.
- Correção de um problema com a contenção de bloqueios causada por um thread de registo em log de auditoria que provoca alta utilização da CPU e tempos limite da aplicação cliente.
- Corrigido um problema que pode fazer com que a instância do banco de dados seja reiniciada ao tentar ler uma página do banco de dados que pertence a uma tabela descartada.
- Corrigido um problema que pode fazer com que a instância do leitor seja reiniciada quando a instância do gravador aumenta o volume do banco de dados para um múltiplo de 160 GB.
- Correção de um problema no gerenciador de bloqueio que podia causar a reinicialização ou o failover ao processar confirmações em duas fases com o nível de isolamento definido como READ_COMMITTED ou READ_UNCOMMITTED e quando eram usadas transações XA ou o log binário (binlog) estava habilitado.
- Corrigido um problema que pode causar indisponibilidade do cluster de banco de dados caso a instância do gravador seja reiniciada enquanto o banco de dados está criando ou descartando gatilhos nas tabelas internas do sistema.
- Correção de um problema que pode fazer com que a instância do banco de dados seja reiniciada quando o número de conexões do banco de dados se aproxima do valor definido pelo parâmetro `max_connections`.
- Correção de um problema que pode fazer com que uma instância do leitor do Aurora seja reiniciada ao executar consultas de linguagem de manipulação de dados (DML) em uma tabela contendo um índice de texto completo.
- A inserção rápida não está habilitada nessa versão do Aurora MySQL, devido a um problema que pode causar inconsistências ao executar consultas como INSERT INTO, SELECT e FROM. Consulte mais informações sobre a otimização de inserção rápida em [Aprimoramentos de desempenho do Amazon Aurora MySQL](#).

Melhorias gerais:

- Correção de um problema que pode fazer com que uma consulta paralela apresente falha devido a problemas transitórios de rede durante a leitura de dados do volume do cluster do Aurora.
- Correção de um problema relacionado ao gerenciamento de arquivos de log de auditoria que pode fazer com que os arquivos de log fiquem inacessíveis para download ou alternância e, em alguns casos, aumente a utilização da CPU.
- Correção de um problema em que pequenas instâncias de réplica de leitura podem apresentar maior atraso de replicação após a atualização de versões anteriores à 2.11.*.
- Correção de um problema que pode gerar mensagens de log excessivas ao consultar a [tabela de concessões procs_priv](#) para verificação de solicitações que envolvem rotinas armazenadas.
- Corrigido um problema de gerenciamento de memória que pode fazer com que a instância de banco de dados use memória excessiva ao executar consultas usando a otimização de junção de hash.
- Correção de um problema que pode produzir um valor incorreto da variável `Threads_running` nas tabelas de status `global information_schema` e `performance_schema` ao usar o encaminhamento de gravação.
- Corrigido um problema que causava a reinicialização do banco de dados ao executar declarações `SELECT` com tabelas particionadas (criadas em uma versão do MySQL compatível com o antigo manipulador de partições `ha_partition`) e a consulta paralela era escolhida pelo planejador de consultas.
- Corrigido um problema que pode impedir o estabelecimento de novas conexões de clientes com o banco de dados quando o encaminhamento de gravação está ativado.
- Redução do atraso na replicação do log binário (binlog) quando uma réplica do log binário do Aurora MySQL executa eventos `QUERY` gravados no arquivo de log binário da fonte sem um banco de dados padrão definido pelo comando `USE`.
- Corrigido um problema que pode fazer com que a métrica `CommitLatency` do CloudWatch seja relatada incorretamente quando o parâmetro `innodb_flush_log_at_trx_commit` não está definido como 1.
- Corrigido um problema que pode fazer com que as conexões de banco de dados sejam fechadas antes de serem estabelecidas. É mais provável que esse problema afete instâncias de banco de dados que abrem e fecham conexões em taxa alta.
- Corrigido um problema que pode fazer com que o banco de dados seja reiniciado quando consumidores de log binário (binlog) conectados estão usando IDs de servidor de replicação de log binário (binlog) duplicados.

- Correção de um problema que podia fazer a replicação de log binário de várias threads parar quando a variável `replica_parallel_type` era definida como `LOGICAL_CLOCK` e a variável `replica_preserve_commit_order` era definida como `ON`. Esse problema podia ocorrer quando uma transação maior que 500 MB era executada na origem.

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bugs da comunidade até a 5.7.40, incluindo as indicadas abaixo. Para obter mais informações, consulte [MySQL bugs fixed by Aurora MySQL 2.x database engine updates](#).

- Correção de um problema que pode fazer com que conexões remotas novas e existentes parem quando executadas simultaneamente com a instrução `SHOW PROCESSLIST` (Bug da comunidade n.º 34857411)
- Replicação: alguns eventos de logs binários nem sempre foram tratados corretamente (Bug n.º 34617506)

Recursos não compatíveis com o Aurora MySQL versão 2

No momento, os recursos a seguir não são compatíveis com o Aurora MySQL versão 2 (compatível com MySQL 5.7).

- Lotes de verificação

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- A declaração SQL `CREATE TABLESPACE`
- Plugin de replicação de grupo
- Maior tamanho de página

- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- Protocolo X

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2023-10-25 (versão 2.12.0.1, compatível com MySQL 5.7.40): versão do Suporte estendido do RDS (Beta)

Versão: 2.12.0.1

O Aurora MySQL 2.12.0.1 está disponível para o público nas seguintes regiões: Leste dos EUA (Norte da Virgínia), Leste dos EUA (Ohio), Oeste dos EUA (Norte da Califórnia), Oeste dos EUA (Oregon), AWS GovCloud (Leste dos EUA) e AWS GovCloud (Oeste dos EUA). Essa é uma versão inicial, somente com correções de segurança. Essas correções serão implantadas de forma mais ampla em todas as regiões com a próxima versão do patch, 2.12.1. As versões do Aurora MySQL 2.12 são compatíveis com o MySQL 5.7.40.

As versões atualmente compatíveis do Aurora MySQL são 2.07.*, 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.* e 3.04.*.

Você pode atualizar um cluster de banco de dados existente do Aurora MySQL 2.* para o Aurora MySQL 2.12.0.1. Também é possível restaurar um snapshot de qualquer versão atualmente compatível do Aurora MySQL para o Aurora MySQL 2.12.0.1.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correção de problemas de segurança e CVEs listados abaixo:

Essa versão inclui todas as correções de CVEs da comunidade, incluindo o MySQL 5.7.40.

- [CVE-2023-38545](#)

Melhorias de disponibilidade:

- A inserção rápida não está habilitada nessa versão do Aurora MySQL, devido a um problema que pode causar inconsistências ao executar consultas como INSERT INTO, SELECT e FROM. Consulte mais informações sobre a otimização de inserção rápida em [Aprimoramentos de desempenho do Amazon Aurora MySQL](#).

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2023-07-25 (versão 2.12.0, compatível com MySQL 5.7.40): versão do Suporte estendido do RDS

Versão: 2.12.0

O Aurora MySQL 2.12.0 está disponível para o público. As versões do Aurora MySQL 2.12 são compatíveis até o MySQL 5.7.40. Para obter mais informações sobre alterações da comunidade, consulte [Changes in MySQL 5.7.40 \(2022-10-11, General Availability\)](#).

As versões atualmente compatíveis do Aurora MySQL são 2.11.*, 2.12.*, 3.01.*, 3.02.* e 3.03.*.

Você pode atualizar um cluster de banco de dados existente do Aurora MySQL 2.* para o Aurora MySQL 2.12.0. Também é possível restaurar um snapshot de qualquer versão atualmente compatível do Aurora MySQL para o Aurora MySQL 2.12.0.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Problemas de segurança corrigidos e CVEs listados abaixo:

Esta versão inclui todas as CVEs correções da comunidade, incluindo o MySQL 5.7.40.

- [As cifras SSL padrão usadas pelo Aurora MySQL foram atualizadas para excluir os valores DES CBC3 - -SHA menos seguros do parâmetro do banco de dados SSL_CIPHER.](#) Se você encontrar problemas de conexão SSL devido à remoção da cifra DES- CBC3 -SHA, use uma cifra segura aplicável da lista a seguir, [Configurando conjuntos de criptografia para conexões com clusters de banco de dados Aurora MySQL](#). Mais informações sobre a [configuração de cifra de conexão](#) do cliente MySQL podem ser encontradas na documentação do 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](#)

Melhorias de disponibilidade:

- Corrigido um problema com a criptografia de eventos de fluxos de atividades do banco de dados que pode causar reinicializações do banco de dados
- Corrigidos dois problemas que podem fazer com que a reinicialização do banco de dados falhe caso ocorra durante a execução de uma consulta de linguagem de definição de dados (DDL).
- Corrigido um problema em que surtos de conexão podem causar maior latência de consulta ou reiniciar a instância de banco de dados.
- Corrigido um problema que, em casos raros, pode fazer com que uma réplica do Aurora seja reiniciada durante a execução simultânea de grandes operações de atualização ou workloads de linguagem de definição de dados (DDL) na instância do gravador e operações de leitura no mesmo conjunto de tabelas na réplica do Aurora.

- Corrigido um problema em que surtos de conexão poderiam fazer com que o processo de estabelecimento da conexão demorasse mais para ser concluído ou falhasse com erros de tempo limite.
- Corrigido um problema em que a rotação do log de auditoria avançada pode reduzir a memória liberável, o que poderia levar à reinicialização da instância de banco de dados.
- Corrigido um problema que pode fazer com que uma instância do leitor do Aurora MySQL seja reiniciada durante a execução de uma consulta que utiliza um plano de execução de consulta paralela do Aurora.
- Corrigido um problema que pode fazer com que a instância do gravador seja reiniciada ao executar a consulta `OPTIMIZE TABLE` em uma tabela com um índice de pesquisa de texto completo (FTS).
- Corrigido um problema que pode fazer com que a instância do gravador em uma AWS região primária do banco de dados global do Aurora fosse reiniciada quando uma `SELECT FOR UPDATE` consulta fosse executada usando o encaminhamento global de gravação de uma região secundária do banco de dados global do Aurora
- Corrigido um problema que pode fazer com que uma instância de leitor de AWS região secundária do banco de dados global Aurora usando encaminhamento de gravação global fosse reiniciada quando uma instrução de confirmação [implícita](#) encaminhada encontrasse um erro
- A inserção rápida não está habilitada nessa versão do Aurora MySQL, devido a um problema que pode causar inconsistências ao executar consultas como `INSERT INTO`, `SELECT` e `FROM`. Consulte mais informações sobre a otimização de inserção rápida em [Aprimoramentos de desempenho do Amazon Aurora MySQL](#).

Melhorias gerais:

- Introdução de otimizações de desempenho de gerenciamento de arquivos em réplicas de log binário para ajudar a reduzir a contenção ao gravar em arquivos de log de retransmissão.
- Corrigido um problema que pode fazer com que o contador `buffer_pool_read_requests` seja relatado incorretamente nas métricas `information_schema`
- Corrigido um problema que pode fazer com que o armazenamento local fique cheio durante a execução das operações `LOAD FROM S3` ou `SELECT INTO S3`. O problema também pode levar a uma maior utilização da CPU, à reinicialização do banco de dados devido à pouca memória e ao aumento da latência dessas consultas.

- Corrigido um problema em que instâncias de banco de dados que usam replicação de log binário podem apresentar um aumento na utilização da CPU e falhas de conexão quando vários consumidores de replicação de log binário estão anexados.
- Corrigido um problema em que as variáveis de status do servidor SSL não eram preenchidas.
- Corrigido um problema em que as instruções da linguagem de manipulação de dados (DML) que executam gravações duplicadas podiam levar ao registro em log excessivo de erros e ao aumento das latências de consulta.
- Atualizadas as definições de fuso horário para a versão 2023c de IANA.
- Adicionado suporte para habilitar e desabilitar o log binário no nível da sessão. Consulte [Stored Procedures - Replicating](#) no Guia do usuário do Amazon Aurora
- Adicionado suporte para definir o formato de log binário no nível da sessão. Consulte [Stored Procedures - Replicating](#) no Guia do usuário do Amazon Aurora
- Corrigido um problema em que definir o parâmetro `aurora_disable_hash_join` como 1 ou ON poderia não impedir que o otimizador usasse uma junção de hash.
- Corrigido um problema envolvendo verificações de índice em que um resultado impreciso poderia ser retornado ao executar uma consulta SELECT com a cláusula GROUP BY e o parâmetro `aurora_parallel_query` definido como ON.
- Corrigido um problema que, em casos raros, pode fazer com que uma instância de leitor do Amazon Aurora seja reiniciada ao acessar uma tabela que tem grandes atualizações ou operações de linguagem de definição de dados (DDL) em execução simultânea na instância do gravador.
- Corrigido um problema que pode fazer com que o contador `buffer_pool_read_requests` seja relatado incorretamente nas métricas `information_schema`
- Corrigido um problema que pode fazer com que uma réplica do log binário seja reiniciada caso a variável de sistema `server_uuid` da origem esteja ausente ou tenha um valor inválido.
- Corrigido um problema para evitar que as estatísticas do InnoDB fiquem obsoletas, o que às vezes pode gerar um plano de execução de consulta abaixo do ideal que pode levar a um aumento no tempo de execução da consulta.
- Corrigido um problema em que as AuroraGlobalDBRPO Lag CloudWatch métricas sempre exibiam zero, independentemente da carga de trabalho do usuário

Upgrades e migrações:

- Para executar um upgrade de uma versão secundária para um banco de dados global do Aurora MySQL versão 2.07 ou 2.11 para o Aurora MySQL versão 2.12 ou posterior, consulte [Atualizar o Aurora MySQL modificando a versão do mecanismo](#).

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bugs da comunidade até a 5.7.40, incluindo as indicadas abaixo. Para obter mais informações, consulte [MySQL bugs fixed by Aurora MySQL 2.x database engine updates](#).

- Corrigido um problema que pode causar maior utilização da CPU devido à rotação do certificado TLS em segundo plano (Correção de bug nº 34284186 da comunidade).

Recursos não compatíveis com o Aurora MySQL versão 2

No momento, os recursos a seguir não são compatíveis com o Aurora MySQL versão 2 (compatível com MySQL 5.7).

- Lote de varredura.

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- A declaração SQL CREATE TABLESPACE
- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens

- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- Protocolo X

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2024-07-19 (versão 2.11.6, compatível com MySQL 5.7.12): versão do Suporte estendido do RDS

Versão: 2.11.6

O Aurora MySQL 2.11.6 está disponível para o público. As versões do Aurora MySQL 2.11 são compatíveis com o MySQL 5.7.12. Para obter mais informações sobre alterações da comunidade, consulte [Changes in MySQL 5.7.12 \(2016-04-11, General Availability\)](#).

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Você pode atualizar um cluster de banco de dados existente do Aurora MySQL 2.* para o Aurora MySQL 2.11.6. Também é possível restaurar um snapshot de qualquer versão anterior atualmente compatível do Aurora MySQL versão 2 para o Aurora MySQL 2.11.6.

Se você atualizar um banco de dados global do Aurora MySQL para uma versão 2.11.*, deverá atualizar os clusters de banco de dados primário e secundário para exatamente a mesma versão, inclusive o nível de patch. Para obter mais informações sobre como fazer upgrade da versão secundária de um banco de dados global do Aurora, consulte [Minor version upgrades](#).

Imediatamente após o upgrade da versão do mecanismo no local para o Aurora MySQL 2.11.*, um upgrade do sistema operacional é aplicado automaticamente a todas as instâncias afetadas nas classes de instância de banco de dados db.r4, db.r5, db.t2 e db.t3, caso as instâncias estejam executando uma versão antiga do sistema operacional. As instâncias do leitor de um cluster de banco de dados multi-AZ aplicam a atualização do sistema operacional primeiro. Quando a atualização do sistema operacional na primeira instância do leitor é concluída, ocorre um failover e o upgrade da instância anterior do gravador.

 Note

O upgrade do sistema operacional não é aplicado automaticamente aos bancos de dados globais do Aurora durante os upgrades de versões principais.

 Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correção de problemas de segurança e CVEs:

- Correção de um problema de segurança nos procedimentos armazenados do MySQL.

Essa versão inclui todas as correções de CVE da comunidade, incluindo o MySQL 5.7.12. As seguintes correções de CVE estão incluídas nesta versão:

- [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](#)

Melhorias de disponibilidade:

- Correção de um problema que podia fazer com que um servidor de banco de dados fosse reiniciado devido ao acesso simultâneo aos recursos de conexão durante a escalabilidade contínua, reinicialização com tempo de inatividade zero (ZDR) e aplicação de patch com tempo de inatividade zero (ZDP).
- Correção um problema que podia fazer com que uma instância de banco de dados de leitor fosse reiniciada ao liberar memória usada para a aplicação de log.
- Correção de um problema que fazia com que uma instância de banco de dados de leitor fosse reiniciada ao executar uma consulta paralela.
- Correção de um problema que podia fazer com que uma instância de banco de dados de leitor que usa o encaminhamento de gravação fosse reiniciada quando uma [declaração de confirmação implícita](#) encaminhada encontrava um erro.

Melhorias gerais:

- Correção de um problema que fazia com que as declarações SQL apresentassem erros ou avisos inesperados de violação de chave primária em algumas linhas ao executar declarações INSERT simultâneas em uma tabela que tem uma coluna de chave primária AUTO_INCREMENT e uma coluna de chave exclusiva, e quando uma declaração INSERT tem violações de chave exclusivas em linhas diferentes.
- Correção de um problema que podia gerar resultados de consulta incorretos quando a ZDR restaurava incorretamente as variáveis de sessão definidas como dicas nas consultas.

- Correção de um problema durante o a escalabilidade do Aurora Serverless v1 que fazia com que a instância de banco de dados fosse reiniciada devido ao acesso incorreto a uma estrutura de dados interna ao encontrar um ponto de escalabilidade.
- Correção de um problema em que consultas lentas INSERT, DELETE e UPDATE executadas pelo [Agendador de eventos](#) do MySQL não eram registradas no log de consultas lentas, a menos que fossem precedidas por uma consulta lenta SELECT.

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bug da comunidade até a versão 5.7.12. Para obter mais informações, consulte [Bugs do MySQL corrigidos por atualizações do mecanismo de banco de dados Aurora MySQL 2.x](#).

Recursos não compatíveis com o Aurora MySQL versão 2

No momento, os recursos a seguir não são compatíveis com o Aurora MySQL versão 2 (compatível com MySQL 5.7).

- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens

- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL CREATE TABLESPACE

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2024-03-26 (versão 2.11.5, compatível com MySQL 5.7.12): versão do Suporte estendido do RDS (padrão)

Versão: 2.11.5

O Aurora MySQL 2.11.5 está disponível para o público. As versões do Aurora MySQL 2.11 são compatíveis com o MySQL 5.7.12. Para obter mais informações sobre alterações da comunidade, consulte [Changes in MySQL 5.7.12 \(2016-04-11, General Availability\)](#).

A versão 2.11.5 é a versão padrão atual do Aurora MySQL versão 2 quando você cria um cluster de banco de dados. Consulte mais informações em [Versões padrão do Amazon Aurora](#).

As versões atualmente compatíveis do Aurora MySQL são 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.*, 3.04.*, 3.05.* e 3.06.*.

Você pode atualizar um cluster de banco de dados existente do Aurora MySQL 2.* para o Aurora MySQL 2.11.5. Também é possível restaurar um snapshot de qualquer versão anterior atualmente compatível do Aurora MySQL versão 2 para o Aurora MySQL 2.11.5.

Se você atualizar um banco de dados global do Aurora MySQL para uma versão 2.11.*, deverá atualizar os clusters de banco de dados primário e secundário para exatamente a mesma versão, inclusive o nível de patch. Para obter mais informações sobre como fazer upgrade da versão secundária de um banco de dados global do Aurora, consulte [Minor version upgrades](#).

Imediatamente após o upgrade da versão do mecanismo no local para o Aurora MySQL 2.11.*, um upgrade do sistema operacional é aplicado automaticamente a todas as instâncias afetadas nas classes de instância de banco de dados db.r4, db.r5, db.t2 e db.t3, caso as instâncias estejam executando uma versão antiga do sistema operacional. As instâncias do leitor de um cluster de banco de dados multi-AZ aplicam a atualização do sistema operacional primeiro. Quando a

atualização do sistema operacional na primeira instância do leitor é concluída, ocorre um failover e o upgrade da instância anterior do gravador.

 Note

O upgrade do sistema operacional não é aplicado automaticamente aos bancos de dados globais do Aurora durante os upgrades de versões principais.

 Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correção de problemas de segurança e CVEs:

As seguintes correções de CVE estão incluídas nesta versão:

- [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](#)

Melhorias de disponibilidade:

- Correção de um problema em que uma instância de banco de dados de gravador do Aurora MySQL podia fazer failover devido a um defeito no componente que se comunica com o armazenamento do Aurora. O defeito ocorria como resultado de uma falha na comunicação entre a instância de banco de dados e o armazenamento subjacente após uma atualização de software.
- Correção de um problema que, em raras condições, podia fazer com que as instâncias de banco de dados de leitor fossem reiniciadas.
- Correção de um problema com a contenção de bloqueios causada por um thread de registo em log de auditoria que provoca alta utilização da CPU e tempos limite da aplicação cliente.

Melhorias gerais:

- Correção de um problema que podia fazer com que uma consulta paralela falhasse devido a problemas transitórios de rede durante a leitura de dados do volume do cluster de banco de dados do Aurora.
- Correção de um problema relacionado ao gerenciamento de arquivos de log de auditoria que podia fazer com que os arquivos de log ficassem inacessíveis para download ou alternância e, em alguns casos, aumentava o uso da CPU.
- Correção de um problema que podia produzir um valor incorreto da variável `Threads_running` nas tabelas de status `global information_schema` e `performance_schema` ao usar o encaminhamento de gravação.

Upgrades e migrações:

- Correção de um problema que impedia o início da replicação de log binário em clusters de banco de dados do Aurora MySQL migrados do RDS para MySQL 5.7.
- Desativação do agendador de eventos do banco de dados durante atualizações da versão do Aurora MySQL versão 3. Isso ajuda a evitar qualquer alteração no banco de dados por meio da execução de eventos enquanto a atualização da versão principal está em andamento.

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bugs da comunidade até a 5.7.12, incluindo as indicadas abaixo. Para obter mais informações, consulte [MySQL bugs fixed by Aurora MySQL 2.x database engine updates](#).

Recursos não compatíveis com o Aurora MySQL versão 2

No momento, os recursos a seguir não são compatíveis com o Aurora MySQL versão 2 (compatível com MySQL 5.7).

- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL `CREATE TABLESPACE`

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2023-10-17 (versão 2.11.4, compatível com MySQL 5.7.12): versão do Suporte estendido do RDS

Versão: 2.11.4

O Aurora MySQL 2.11.4 está disponível para o público. As versões do Aurora MySQL 2.11 são compatíveis com o MySQL 5.7.12. Para obter mais informações sobre alterações da comunidade, consulte [Changes in MySQL 5.7.12 \(2016-04-11, General Availability\)](#).

As versões atualmente compatíveis do Aurora MySQL são 2.07.9, 2.07.10, 2.11.*, 2.12.*, 3.01.*, 3.02.*, 3.03.* e 3.04.*.

Você pode atualizar um cluster de banco de dados existente do Aurora MySQL 2.* para o Aurora MySQL 2.11.4. Também é possível restaurar um snapshot de qualquer versão atualmente compatível do Aurora MySQL para o Aurora MySQL 2.11.4.

Se você atualizar um banco de dados global do Aurora MySQL para uma versão 2.11.*, deverá atualizar os clusters de banco de dados primário e secundário para exatamente a mesma versão, inclusive o nível de patch. Para obter mais informações sobre como fazer upgrade da versão secundária de um banco de dados global do Aurora, consulte [Minor version upgrades](#).

Imediatamente após o upgrade da versão do mecanismo no local para o Aurora MySQL 2.11.*, um upgrade do sistema operacional é aplicado automaticamente a todas as instâncias afetadas nas classes de instância de banco de dados db.r4, db.r5, db.t2 e db.t3, caso as instâncias estejam executando uma versão antiga do sistema operacional. As instâncias do leitor de um cluster de banco de dados multi-AZ aplicam a atualização do sistema operacional primeiro. Quando a atualização do sistema operacional na primeira instância do leitor é concluída, ocorre um failover e o upgrade da instância anterior do gravador.

 Note

O upgrade do sistema operacional não é aplicado automaticamente aos bancos de dados globais do Aurora durante os upgrades de versões principais.

 Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correção de problemas de segurança e CVEs listados abaixo:

- Corrigido um problema em que os eventos relatados durante o processamento das rotações do log de auditoria podiam não ser gravados no log de auditoria.
- [CVE-2022-24407](#)

Melhorias de disponibilidade:

- Corrigido um problema em que instâncias de banco de dados do Aurora MySQL que usam consultas paralelas podem sofrer uma reinicialização do banco de dados ao executar um grande número de consultas paralelas simultâneas.
- Corrigido um problema que pode fazer com que uma instância de banco de dados seja reiniciada durante a execução de workloads do leitor intensivas de E/S.
- Corrigido um problema que pode fazer com que a instância do banco de dados seja reiniciada ao tentar ler uma página do banco de dados que pertence a uma tabela descartada.
- Corrigido um problema que pode fazer com que a instância do leitor seja reiniciada quando a instância do gravador aumenta o volume do banco de dados para um múltiplo de 160 GB.
- Corrigido um problema que pode causar indisponibilidade do cluster de banco de dados caso a instância do gravador seja reiniciada enquanto o banco de dados está criando ou descartando gatilhos nas tabelas internas do sistema.
- Corrigido um problema que pode fazer com que uma instância do leitor seja reiniciada ao executar consultas de linguagem de manipulação de dados (DML) em uma tabela contendo um índice de texto completo.
- Corrigido um problema que pode fazer com que uma instância do leitor seja reiniciada enquanto executa uma consulta que utiliza um plano de execução de consulta paralela do Aurora.
- Corrigido um problema que pode fazer com que a instância do gravador seja reiniciada ao executar a consulta `OPTIMIZE TABLE` em uma tabela com um índice de pesquisa de texto completo (FTS).
- A inserção rápida não está habilitada nessa versão do Aurora MySQL, devido a um problema que pode causar inconsistências ao executar consultas como `INSERT INTO`, `SELECT` e `FROM`.

Consulte mais informações sobre a otimização de inserção rápida em [Aprimoramentos de desempenho do Amazon Aurora MySQL](#).

Melhorias gerais:

- Corrigido um problema em que pequenas instâncias de réplica de leitura podem apresentar maior atraso de replicação após a atualização de versões anteriores à 2.11.*.
- Corrigido um problema que pode causar mensagens de log excessivas ao consultar a [tabela de concessões procs_priv](#) para verificação de solicitações que envolvem rotinas armazenadas.
- Corrigido um problema de gerenciamento de memória que pode fazer com que a instância de banco de dados use memória excessiva ao executar consultas usando a otimização de junção de hash.
- Corrigido um problema que causava a reinicialização do banco de dados ao executar declarações SELECT com tabelas particionadas (criadas em uma versão do MySQL compatível com o antigo manipulador de partições `ha_partition`) e a consulta paralela era escolhida pelo planejador de consultas.
- Corrigido um problema que pode impedir o estabelecimento de novas conexões de clientes com o banco de dados quando o encaminhamento de gravação está ativado.
- Redução do atraso na replicação do log binário (binlog) quando uma réplica do log binário do Aurora MySQL executa eventos QUERY gravados no arquivo de log binário da fonte sem um banco de dados padrão definido pelo comando USE.
- Corrigido um problema envolvendo verificações de índice em que um resultado impreciso poderia ser retornado ao executar uma consulta SELECT com a cláusula GROUP BY e o parâmetro `aurora_parallel_query` definido como ON.
- Adicionado suporte para habilitar e desabilitar o log binário no nível da sessão. Consulte [Stored Procedures - Replicating](#) no Guia do usuário do Amazon Aurora.
- Corrigido um problema que pode fazer com que uma réplica do log binário seja reiniciada caso a variável de sistema `server_uuid` da origem esteja ausente ou tenha um valor inválido.
- Adicionado suporte para definir o formato de log binário no nível da sessão. Consulte [Stored Procedures - Replicating](#) no Guia do usuário do Amazon Aurora.
- Corrigido um problema que pode fazer com que a métrica CommitLatency do CloudWatch seja relatada incorretamente quando o parâmetro `innodb_flush_log_at_trx_commit` não está definido como 1.

- Corrigido um problema para evitar que as estatísticas do InnoDB fiquem obsoletas, o que às vezes pode gerar um plano de execução de consulta abaixo do ideal, que pode levar a um aumento no tempo de execução da consulta.
- Corrigido um problema que pode fazer com que o banco de dados seja reiniciado quando consumidores de log binário (binlog) conectados estão usando IDs de servidor de replicação de log binário (binlog) duplicados.

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bugs da comunidade até a 5.7.12, incluindo as indicadas abaixo. Para obter mais informações, consulte [MySQL bugs fixed by Aurora MySQL 2.x database engine updates](#).

- Replicação: alguns eventos de logs binários nem sempre foram tratados corretamente. (Bug nº 34617506)
- Corrigido um problema que pode causar maior utilização da CPU devido à rotação do certificado TLS em segundo plano (Correção de bug nº 34284186 da comunidade).
- Em declarações preparadas, alguns tipos de subconsultas poderiam causar uma saída do servidor. (Bug nº 33100586)

Recursos não compatíveis com o Aurora MySQL versão 2

No momento, os recursos a seguir não são compatíveis com o Aurora MySQL versão 2 (compatível com MySQL 5.7).

- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL `CREATE TABLESPACE`

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2023-06-09 (versão 2.11.3, compatível com MySQL 5.7.12): versão do Suporte estendido do RDS

Versão: 2.11.3

O Aurora MySQL 2.11.3 está disponível para o público. As versões do Aurora MySQL 2.11 são compatíveis com o MySQL 5.7.12. Para obter mais informações sobre alterações da comunidade, consulte [Changes in MySQL 5.7.12 \(2016-04-11, General Availability\)](#).

As versões atualmente compatíveis do Aurora MySQL são 2.07.*, 2.11.*, 3.01.*, 3.02.* e 3.03.*.

Você pode atualizar um cluster de banco de dados existente do Aurora MySQL 2.* para o Aurora MySQL 2.11.3. Também é possível restaurar um snapshot de qualquer versão atualmente compatível do Aurora MySQL para o Aurora MySQL 2.11.3.

Se você atualizar um banco de dados global do Aurora MySQL para uma versão 2.11.*, deverá atualizar os clusters de banco de dados primário e secundário para exatamente a mesma versão, inclusive o nível de patch. Para obter mais informações sobre como fazer upgrade da versão secundária de um banco de dados global do Aurora, consulte [Minor version upgrades](#).

Imediatamente após o upgrade da versão do mecanismo no local para o Aurora MySQL 2.11.*, um upgrade do sistema operacional é aplicado automaticamente a todas as instâncias afetadas nas classes de instância de banco de dados db.r4, db.r5, db.t2 e db.t3, caso as instâncias estejam executando uma versão antiga do sistema operacional. As instâncias do leitor de um cluster de banco de dados multi-AZ aplicam a atualização do sistema operacional primeiro. Quando a atualização do sistema operacional na primeira instância do leitor é concluída, ocorre um failover e o upgrade da instância anterior do gravador.

Note

O upgrade do sistema operacional não é aplicado automaticamente aos bancos de dados globais do Aurora durante os upgrades de versões principais.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correção de problemas de segurança e CVEs listados abaixo:

- As cifras SSL padrão usadas pelo Aurora MySQL foram atualizadas para excluir os valores menos seguros de DES-CBC3-SHA do parâmetro do banco de dados [SSL_CIPHER](#). Se você encontrar problemas de conexão SSL devido à remoção da cifra DES-CBC3-SHA, use uma cifra segura aplicável desta lista, [ConfiguringCipherSuites](#). Mais informações sobre a [configuração de cifra de conexão](#) do cliente MySQL podem ser encontradas na documentação do MySQL.
- [CVE-2023-21963](#)
- [CVE-2023-21912](#)
- [CVE-2023-0215](#)

- [CVE-2022-43551](#)
- [CVE-2022-37434](#)

Melhorias de disponibilidade:

- Corrigido um problema na criptografia de eventos do fluxo de atividades do banco de dados (DAS) que pode causar a reinicialização do banco de dados.
- Corrigidos dois problemas que podem fazer com que a reinicialização do banco de dados falhe caso ocorra durante a execução de uma consulta de linguagem de definição de dados (DDL).
- A inserção rápida não está habilitada nessa versão do Aurora MySQL, devido a um problema que pode causar inconsistências ao executar consultas como `INSERT INTO`, `SELECT` e `FROM`. Consulte mais informações sobre a otimização de inserção rápida em [Aprimoramentos de desempenho do Amazon Aurora MySQL](#).

Melhorias gerais:

- Introdução de otimizações de desempenho de gerenciamento de arquivos em réplicas de log binário para ajudar a reduzir a contenção ao gravar em arquivos de log de retransmissão.
- Corrigido um problema em que definir o parâmetro `aurora_disable_hash_join` como 1 ou ON poderia não impedir que o otimizador usasse uma junção de hash.
- Corrigido um problema que pode fazer com que o contador `buffer_pool_read_requests` seja relatado incorretamente nas métricas `information_schema`.
- Corrigido um problema que pode fazer com que o armazenamento local fique cheio durante a execução das operações `LOAD FROM S3` ou `SELECT INTO S3`. O problema também pode levar a uma maior utilização da CPU, à reinicialização do banco de dados devido à pouca memória e ao aumento da latência dessas consultas.

Recursos não compatíveis com o Aurora MySQL versão 2

No momento, os recursos a seguir não são compatíveis com o Aurora MySQL versão 2 (compatível com MySQL 5.7).

- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL CREATE TABLESPACE

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2023-03-24 (versão 2.11.2, compatível com MySQL 5.7.12): versão do Suporte estendido do RDS

Versão: 2.11.2

O Aurora MySQL 2.11.2 está disponível para o público. As versões do Aurora MySQL 2.11 são compatíveis com o MySQL 5.7.12. Para obter mais informações sobre alterações da comunidade, consulte [Changes in MySQL 5.7.12 \(2016-04-11, General Availability\)](#).

As versões atualmente compatíveis do Aurora MySQL são 2.07.*, 2.11.*, 3.01.*, 3.02.* e 3.03.*.

Você pode atualizar um cluster de banco de dados existente do Aurora MySQL 2.* para o Aurora MySQL 2.11.2. Também é possível restaurar um snapshot de qualquer versão atualmente compatível do Aurora MySQL para o Aurora MySQL 2.11.2.

Se você atualizar um banco de dados global do Aurora MySQL para uma versão 2.11.*, deverá atualizar os clusters de banco de dados primário e secundário para exatamente a mesma versão, inclusive o nível de patch. Para obter mais informações sobre como fazer upgrade da versão secundária de um banco de dados global do Aurora, consulte [Minor version upgrades](#).

Imediatamente após o upgrade da versão do mecanismo no local para o Aurora MySQL 2.11.*, um upgrade do sistema operacional é aplicado automaticamente a todas as instâncias afetadas nas classes de instância de banco de dados db.r4, db.r5, db.t2 e db.t3, caso as instâncias estejam executando uma versão antiga do sistema operacional. As instâncias do leitor de um cluster de banco de dados multi-AZ aplicam a atualização do sistema operacional primeiro. Quando a atualização do sistema operacional na primeira instância do leitor é concluída, ocorre um failover e o upgrade da instância anterior do gravador.

 Note

O upgrade do sistema operacional não é aplicado automaticamente aos bancos de dados globais do Aurora durante os upgrades de versões principais.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

 Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Melhorias gerais:

- Corrigido um problema em que instâncias de banco de dados que usam replicação de log binário podem apresentar um aumento na utilização da CPU e falhas de conexão quando vários consumidores de replicação de log binário estão anexados.

- Corrigido um problema que pode fazer com que uma instância do leitor em uma região secundária do banco de dados global fique fora de sincronia após a atualização para o Aurora MySQL versão 2.11 caso o gravador do banco de dados principal esteja no Aurora MySQL versão 2.10.

Melhorias de disponibilidade:

- A inserção rápida não está habilitada nessa versão do Aurora MySQL, devido a um problema que pode causar inconsistências ao executar consultas como `INSERT INTO`, `SELECT` e `FROM`. Consulte mais informações sobre a otimização de inserção rápida em [Aprimoramentos de desempenho do Amazon Aurora MySQL](#).

Recursos não compatíveis com o Aurora MySQL versão 2

No momento, os recursos a seguir não são compatíveis com o Aurora MySQL versão 2 (compatível com MySQL 5.7).

- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha

- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL CREATE TABLESPACE

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2023-02-14 (versão 2.11.1, compatível com MySQL 5.7.12): versão do Suporte estendido do RDS

Versão: 2.11.1

O Aurora MySQL 2.11.1 está disponível para o público. As versões do Aurora MySQL 2.11 são compatíveis com o MySQL 5.7.12. Para obter mais informações sobre alterações da comunidade, consulte [Changes in MySQL 5.7.12 \(2016-04-11, General Availability\)](#).

As versões atualmente compatíveis do Aurora MySQL são 1.19.5, 1.19.6, 1.22.*, 1.23.*, 2.07.*, 2.09.*, 2.10.*, 2.11.*, 3.01.* e 3.02.*.

Você pode atualizar um cluster de banco de dados existente do Aurora MySQL 2.* para o Aurora MySQL 2.11.1. Para clusters que executam o Aurora MySQL versão 1, você pode atualizar um cluster existente do Aurora MySQL 1.23 ou posterior diretamente para 2.11.1. Também é possível restaurar um snapshot de qualquer versão atualmente compatível do Aurora MySQL para o Aurora MySQL 2.11.1.

Se você atualizar um banco de dados global do Aurora MySQL para uma versão 2.11.* e tiver o encaminhamento de gravação ativado, deverá atualizar os clusters de banco de dados primário e secundários para exatamente a mesma versão, incluindo o nível de patch, para continuar usando o encaminhamento de gravação. Para obter mais informações sobre como fazer upgrade da versão secundária de um banco de dados global do Aurora, consulte [Minor version upgrades](#).

Imediatamente após o upgrade da versão do mecanismo no local para o Aurora MySQL 2.11.*, um upgrade do sistema operacional é aplicado automaticamente a todas as instâncias afetadas nas classes de instância de banco de dados db.r4, db.r5, db.t2 e db.t3, caso as instâncias estejam executando uma versão antiga do sistema operacional. As instâncias do leitor de um cluster de banco de dados multi-AZ aplicam a atualização do sistema operacional primeiro. Quando a atualização do sistema operacional na primeira instância do leitor é concluída, ocorre um failover e o upgrade da instância anterior do gravador.

Note

O upgrade do sistema operacional não é aplicado automaticamente aos bancos de dados globais do Aurora durante os upgrades de versões principais.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correção de problemas de segurança e CVEs listados abaixo:

Correções e outras melhorias para ajustar o tratamento em um ambiente gerenciado. Correções adicionais do CVE abaixo:

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

Melhorias de disponibilidade:

- Corrigido um problema em que surtos de conexão podem causar maior latência de consulta ou reiniciar a instância de banco de dados.
- Corrigido um problema que, em casos raros, pode fazer com que uma réplica do Aurora seja reiniciada durante a execução simultânea de grandes operações de atualização ou workloads de linguagem de definição de dados (DDL) na instância do gravador e operações de leitura no mesmo conjunto de tabelas na réplica do Aurora.

- Corrigido um problema em que surtos de conexão poderiam fazer com que o processo de estabelecimento da conexão demorasse mais para ser concluído ou falhasse com erros de tempo limite.
- Corrigido um problema em que a rotação do log de auditoria avançada pode reduzir a memória liberável, o que poderia levar à reinicialização da instância de banco de dados.
- A inserção rápida não está habilitada nessa versão do Aurora MySQL, devido a um problema que pode causar inconsistências ao executar consultas como INSERT INTO, SELECT e FROM. Consulte mais informações sobre a otimização de inserção rápida em [Aprimoramentos de desempenho do Amazon Aurora MySQL](#).

Melhorias gerais:

- Corrigido um problema em que as [variáveis de status do servidor SSL](#) não eram preenchidas.
- Corrigido um problema em que as instruções da linguagem de manipulação de dados (DML) que executam gravações duplicadas podiam levar ao registro em log excessivo de erros e ao aumento das latências de consulta.

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7).

- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para funções AWS Lambda chamadas de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL CREATE TABLESPACE

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2022-10-25 (versão 2.11.0, compatível com MySQL 5.7.12): versão do Suporte estendido do RDS

Versão: 2.11.0

O Aurora MySQL 2.11.0 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com o MySQL 5.7.12. Para obter mais informações sobre alterações da comunidade, consulte [Changes in MySQL 5.7.12 \(2016-04-11, General Availability\)](#).

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Você pode atualizar um cluster de banco de dados existente do Aurora MySQL 2.* para o Aurora MySQL 2.11.0. Para clusters que executam o Aurora MySQL versão 1, você pode atualizar um

cluster existente do Aurora MySQL 1.23 ou posterior diretamente para 2.11.0. Também é possível restaurar um snapshot de qualquer versão atualmente compatível do Aurora MySQL para o Aurora MySQL 2.11.0.

Se você atualizar um banco de dados global do Aurora MySQL para uma versão 2.11.* e tiver o encaminhamento de gravação ativado, deverá atualizar os clusters de banco de dados primário e secundários para exatamente a mesma versão, incluindo o nível de patch, para continuar usando o encaminhamento de gravação. Para obter mais informações sobre como fazer upgrade da versão secundária de um banco de dados global do Aurora, consulte [Minor version upgrades](#).

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correção de problemas de segurança e CVEs:

- O provedor SSL/TLS foi alterado de OpenSSL para [AWS-LC](#).

Isso traz uma série de mudanças, incluindo, mas não se limitando à remoção do suporte para cifras DHE-RSA-* menos seguras.

Para obter mais informações, consulte [Usar TLS com clusters de banco de dados do Aurora MySQL](#).

As seguintes correções de CVE estão incluídas nesta versão:

- [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](#)

Novos atributos:

- Com o lançamento do Aurora MySQL versão 2.11, uma nova atualização do sistema operacional está disponível. Recomendamos que você aplique essa atualização pendente do sistema operacional a todas as instâncias de banco de dados do Aurora MySQL após o upgrade para a versão 2.11. Para obter mais informações, consulte [Trabalhar com atualizações do sistema operacional](#).
- Uma nova opção de configuração dinâmica, `innodb_deadlock_detect`, pode ser usada para desabilitar a detecção de impasse. Em sistemas de alta simultaneidade, a detecção de deadlock pode causar uma desaceleração quando vários segmentos aguardam o mesmo bloqueio. Às

vezes, pode ser mais eficiente desabilitar a detecção de impasse e confiar na configuração `innodb_lock_wait_timeout` para reversão de transações quando ocorre um impasse. (Bug nº 23477773) Mais informações sobre a detecção de impasse do InnoDB podem ser encontradas na [documentação do MySQL](#).

- As funções `UUID_TO_BIN`, `BIN_TO_UUID` e `IS_UUID` do MySQL 8.0 foram adicionadas. Mais informações sobre o uso dessas funções podem ser encontradas em [MySQL Miscellaneous function](#).
- Adicionado suporte para dicas do otimizador, permitindo que o usuário habilite ou desabilite a consulta paralela do Aurora MySQL por tabela ou por consulta.
 - [Como trabalhar com a consulta paralela do Amazon Aurora MySQL](#)
 - [Dicas do Aurora MySQL](#)
- Removido o suporte ao tipo de instância R3.
- Adicionado suporte para instâncias R6i.

Melhorias de disponibilidade:

- Corrigido um problema que pode impedir a replicação lógica entre regiões em um cluster de banco de dados devido ao arquivo de log binário incorreto e à posição gravada nos logs de erros. Esse problema pode ocorrer quando o mecanismo é reiniciado após a execução de uma declaração de DDL.
- Corrigido um problema que, em raras condições, pode fazer com que as instâncias do leitor do Aurora sejam reiniciadas ao executar instruções da lista de controle de acesso (ACL), como `GRANT` e `FLUSH` na instância do gravador. É mais provável que esse problema afete instâncias do leitor com um grande número de usuários e operações de ACL (por exemplo, alterações de permissão).
- Corrigido um problema que, em raras condições, pode fazer com que a instância do gravador seja reiniciada ou executada um failover quando uma transação acessa uma linha que está sendo excluída por outra transação.
- Melhora no desempenho da pesquisa de frases em texto completo para reduzir significativamente o tempo necessário para pesquisar frases em uma tabela com índices de texto completo.
- Corrigido um problema em que, após a reinicialização de uma instância do gravador, ela travava em uma recuperação lenta e, posteriormente, reiniciava novamente. Esse problema ocorre quando há um grande número de linhas não confirmadas no banco de dados no momento da reinicialização inicial.

- Corrigido um problema que, em raras ocasiões, faz com que o servidor de banco de dados seja reiniciado por conta de uma longa espera de semáforo quando o [thread do detector de impasse](#) trava.
- Corrigido um problema que, em raras ocasiões, pode fazer com que o banco de dados seja reiniciado por conta de uma longa espera de semáforo quando o thread de E/S atinge um impasse.
- A inserção rápida não está habilitada nessa versão do Aurora MySQL, devido a um problema que pode causar inconsistências ao executar consultas como INSERT INTO, SELECT e FROM. Consulte mais informações sobre a otimização de inserção rápida em [Aprimoramentos de desempenho do Amazon Aurora MySQL](#).

Melhorias gerais:

- Corrigido um problema que pode fazer com que o servidor de banco de dados seja reiniciado quando todas as condições a seguir são verdadeiras:
 - ALLOW_INVALID_DATES está desabilitado no MODO SQL.
 - O servidor do banco de dados está processando uma declaração INSERT, UPDATE, DELETE ou SELECT com um valor inválido do tipo DATETIME, de forma que o mês não esteja entre 1 e 12.
- Corrigido um problema em que o período de retenção de logs binários não era respeitado quando o log binário era definido como OFF, resultando em uma utilização de armazenamento maior do que a esperada. Após essa correção, os logs binários serão eliminados com base no período de retenção. Mais informações sobre como configurar o período de retenção de logs binários podem ser encontradas no [Guia do usuário do Aurora MySQL](#).
- Corrigido um problema que pode fazer com que a memória disponível na instância do banco de dados diminua quando determinadas declarações SQL da linguagem de controle de dados (DCL), como GRANT, FLUSH PRIVILEGES etc. são executadas nessa instância. O uso frequente dessas declarações pode fazer com que a memória que pode ser liberada continue diminuindo e que a instância de banco de dados seja reiniciada devido a problemas de falta de memória. O uso dessas declarações na instância do gravador também pode reduzir a memória que pode ser liberada nas instâncias do leitor.
- Apresentado um tamanho maior de buffer de leitura para leituras realizadas a partir dos logs de retransmissão para minimizar o número de operações de E/S de leitura, o que reduz a contenção entre os threads de E/S e SQL.
- Corrigido um problema que fazia com que o procedimento armazenado mysql.rds_rotate_slow_log falhasse com a mensagem de erro “A tabela 'mysql.slow_log_backup' não existe”.

- Corrigido um problema em que a invalidação excessiva do cache de consulta causa um uso de CPU e latências maiores do que o esperado na réplica de leitura devido à necessidade da réplica de leitura ler os dados do disco em vez do cache de consulta.
- Corrigido um problema que permitia que os usuários executassem os comandos `INSTALL PLUGIN` e `UNINSTALL PLUGIN` em uma instância do leitor, o que pode causar um impasse em `LOCK_plugin`, `LOCK_system_variables_hash`, `LOCK_global_system_variables`. Agora, essas declarações só podem ser executadas na instância do gravador em um cluster de banco de dados.
- Corrigido um problema em que clusters podem ter uma latência de confirmação maior do que a esperada quando log binário é habilitado. Isso afeta todas as transações que geram grandes eventos de log binário (com mais de 500 MB).
- Corrigido um problema que pode fazer com que a métrica `trx_active_transactions` na tabela `INFORMATION_SCHEMA.INNODB_METRICS` tenha um valor incorreto.
- Corrigido um problema que pode interromper a replicação lógica devido ao arquivo de log binário se tornar inconsistente ao executar uma reversão para o ponto de salvamento de uma transação grande.
- Por padrão, hashes de credenciais mascarados em log geral, log de consulta lenta e log de auditoria usando um segredo de máscara consistente. Isso pode ser configurado por meio do parâmetro `aurora_mask_password_hashes_type`.
- Corrigido um problema em que a duração do reinício com zero tempo de inatividade (ZDR) era informada incorretamente nos eventos observados pelo cliente.
- Corrigido um problema que pode fazer com que as chamadas para [mysql_rds_import_binlog_ssl_material](#) falhem com o [ERRO 1457 do servidor MySQL](#).
- Corrigido um problema em que a inicialização do encadeamento de despejo poderia ficar bloqueada com o thread para limpar os logs binários. Isso pode impedir que o arquivo de log binário ativo alterne e, em vez disso, continue crescendo ou causar problemas com novas conexões de réplica de log binário.
- Corrigido um problema em que o cache de consultas pode retornar resultados obsoletos em uma réplica de leitura do Aurora.

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bugs da comunidade até 5.7, incluindo as indicadas abaixo. Para obter mais informações, consulte [MySQL bugs fixed by Aurora MySQL 2.x database engine updates](#).

- Corrigido um problema em que o código para ler informações do conjunto de caracteres das tabelas de eventos de declarações do Performance Schema (por exemplo, `events_statements_current`) não impedia a gravação simultânea nas informações desse conjunto de caracteres. Como resultado, o conjunto de caracteres de texto da consulta SQL pode ser inválido, o que pode resultar na saída do servidor. Com essa correção, um conjunto de caracteres inválido causa o truncamento da coluna `SQL_TEXT` e impede a saída do servidor. (Bug nº 23540008)
- InnoDB: reversão de uma correção para o Bug nº 25189192, Bug nº 84038 da comunidade. Corrigido um problema em que, após uma operação `RENAME TABLE` que movia uma tabela para um esquema diferente, o InnoDB falhava ao atualizar a tabela do dicionário de dados do `INNODB_SYS_DATAFILES`. Isso resultou em um erro na reinicialização, indicando que não foi possível localizar o arquivo de dados do espaço de tabela.
- InnoDB: corrigido um problema em que o servidor descartava um índice de chave externa definido internamente ao adicionar um novo índice e tentava utilizar um índice secundário definido em uma coluna virtual gerada como o índice de chave externa, causando a saída do servidor. Agora, o InnoDB permite que uma restrição de chave externa faça referência a um índice secundário definido em uma coluna virtual gerada. (Bug #23533396)
- Corrigido um problema em que duas sessões executavam simultaneamente uma operação `INSERT... ON DUPLICATE KEY UPDATE` gerava um impasse. Durante a reversão parcial de uma tupla, outra sessão poderia atualizá-la. A correção desse bug reverte as correções do Bug nº 11758237, Bug nº 17604730 e Bug nº 20040791. (Bug nº 25966845)
- Reversão de uma correção para o Bug nº 27407480 da comunidade: corrigido um problema em que os privilégios `EXECUTE` e `ALTER ROUTINE` não eram concedidos corretamente aos criadores de rotinas, mesmo com `automatic_sp_privileges` habilitado.
- Reversão da correção do Bug nº 24671968 da comunidade: corrigido um problema em que uma consulta poderia produzir resultados incorretos se a cláusula `WHERE` contivesse uma subconsulta dependente, a tabela tivesse um índice secundário nas colunas na lista de seleção seguido pelas colunas na subconsulta e `GROUP BY` ou `DISTINCT` permitisse que a consulta usasse uma varredura de índice solta.
- Corrigido um problema em que a replicação era interrompida se uma instrução de exclusão de várias tabelas fosse emitida em várias tabelas com chaves estrangeiras. (Bug nº 80821)
- Corrigido um problema em que, em casos especiais, certos erros secundários não eram ignorados, mesmo com [slave_skip_errors](#) habilitado. Nos casos em que a abertura e o bloqueio de uma tabela falham ou quando as conversões de campo falham em um servidor que executa a replicação baseada em linhas, o erro é considerado crítico e o estado de [slave_skip_errors](#) é ignorado.

A correção garante que, com [slave_skip_errors](#) habilitado, todos os erros relatados durante a aplicação de uma transação sejam tratados corretamente. (Bug nº 70640, Bug nº 17653275)

- Corrigido um problema em que uma declaração [SET PASSWORD](#) era replicada de uma entidade principal do MySQL 5.6 para uma entidade secundária do MySQL 5.7 ou de uma entidade principal do MySQL 5.7 com a variável de sistema [log_built_in_as_identified_by_password](#) definida como ON para uma entidade secundária do MySQL 5.7. O próprio hash da senha também era codificado antes de ser armazenado na entidade secundária. O problema agora foi corrigido e o hash da senha replicada é armazenado conforme passado originalmente para a entidade secundária. (Bug nº 24687073)
- Corrigido um problema em que a serialização de um valor JSON que consiste em um grande subdocumento encapsulado em vários níveis de matrizes ou objetos JSON, ou ambos, às vezes exigia muito tempo para ser concluída. (Bug nº 23031146)
- Declarações que não podem ser analisadas (devido, por exemplo, a erros de sintaxe) não são mais gravadas no log de consultas lentas. (Bug nº 33732907)

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7).

- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para funções AWS Lambda chamadas de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL `CREATE TABLESPACE`

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2022-11-01 (versão 2.10.3) (obsoleta)

Versão: 2.10.3

O Aurora MySQL 2.10.3 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Você pode atualizar um cluster de banco de dados existente do Aurora MySQL 2.* para o Aurora MySQL 2.10.3. Para clusters que executam o Aurora MySQL versão 1, você pode atualizar um

cluster existente do Aurora MySQL 1.23 ou posterior diretamente para 2.10.3. Também é possível restaurar um snapshot de qualquer versão atualmente compatível do Aurora MySQL para o Aurora MySQL 2.10.3.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

 Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correção de problemas de segurança e CVEs listados abaixo:

Correções e outras melhorias para ajustar o tratamento em um ambiente gerenciado. Correções adicionais do CVE abaixo:

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

Melhorias gerais:

- Corrigido um problema que, em raras ocasiões, faz com que o servidor de banco de dados seja reiniciado por conta de uma longa espera de semáforo quando o [thread do detector de impasse](#) trava.
- Corrigido um problema que pode fazer com que a memória disponível na instância do banco de dados diminua quando determinadas declarações SQL da linguagem de controle de dados (DCL), como GRANT, FLUSH PRIVILEGES etc. são executadas nessa instância. O uso frequente

dessas declarações pode fazer com que a memória que pode ser liberada continue diminuindo e que a instância de banco de dados seja reiniciada devido a problemas de falta de memória. O uso dessas declarações na instância do gravador também pode reduzir a memória que pode ser liberada nas instâncias do leitor.

- Corrigido um problema em que as consultas na tabela `performance_schema.events_waits_summary_global_by_event_name` podem ficar lentas quando uma instância de banco de dados está sob carga pesada com o evento de espera de `performance_schema` `wait/io/aurora_respond_to_client` habilitado.
- Corrigido um problema que, em raras condições, pode fazer com que o servidor de banco de dados seja interrompido e reiniciado quando as transações são parcialmente revertidas devido a uma violação de restrição nos índices secundários.
- Corrigido um problema que, em raras condições, pode fazer com que a instância do gravador seja reiniciada ou executada um failover quando uma transação acessa uma linha que está sendo excluída por outra transação.
- Corrigido um problema que, em raras ocasiões, pode fazer com que o banco de dados seja reiniciado por conta de uma longa espera de semáforo quando o thread de E/S atinge um impasse.
- Corrigido um problema que pode fazer com que a réplica de leitura seja reiniciada durante o failover em raras condições quando o arquivo de bloqueio de soquete Unix está em uso.
- Corrigido um problema em que a invalidação excessiva do cache de consulta causa um uso de CPU e latências maiores do que o esperado na réplica de leitura devido à necessidade da réplica de leitura ler os dados do disco em vez do cache de consulta.

Integração de correções de bug do MySQL Community Edition

Essa versão inclui todas as correções de bugs da comunidade até 5.7, incluindo as indicadas abaixo. Para obter mais informações, consulte [MySQL bugs fixed by Aurora MySQL 2.x database engine updates](#).

- Corrigido um problema em que o código para ler informações do conjunto de caracteres das tabelas de eventos de declarações do Performance Schema (por exemplo, `events_statements_current`) não impedia a gravação simultânea nas informações desse conjunto de caracteres. Como resultado, o conjunto de caracteres de texto da consulta SQL pode ser inválido, o que pode resultar na saída do servidor. Com essa correção, um conjunto de caracteres inválido causa o truncamento da coluna `SQL_TEXT` e impede a saída do servidor. (Bug nº 23540008)

- Corrigido um problema quando UPDATE exigia uma tabela temporária com uma chave primária maior que 1024 bytes e essa tabela era criada usando o InnoDB, o servidor poderia sair. (Bug nº 25153670)
- Corrigido um problema em que duas sessões executavam simultaneamente uma operação INSERT... ON DUPLICATE KEY UPDATE gerava um impasse. Durante a reversão parcial de uma tupla, outra sessão poderia atualizá-la. A correção desse bug reverte as correções do Bug nº 11758237, Bug nº 17604730 e Bug nº 20040791. (Bug nº 25966845)

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7).

- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para funções AWS Lambda chamadas de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plugin de replicação de grupo

- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL `CREATE TABLESPACE`

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2022-01-26 (versão 2.10.2) (obsoleta)

Versão: 2.10.2

O Aurora MySQL 2.10.2 está geralmente disponível. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Você pode atualizar um cluster de banco de dados existente do Aurora MySQL 2.* para o Aurora MySQL 2.10.0. Para clusters que executam o Aurora MySQL versão 1, você pode atualizar um cluster existente do Aurora MySQL 1.23 ou superior diretamente para 2.10.0. Também é possível restaurar um snapshot de qualquer versão atualmente compatível do Aurora MySQL para o Aurora MySQL 2.10.0.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

 Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correção de problemas de segurança e CVEs listados abaixo:

Correções e outras melhorias para ajustar o tratamento em um ambiente gerenciado. Correções adicionais do CVE abaixo:

- [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](#)

Melhorias gerais:

- Inclusão de otimização de performance para ajudar a reduzir a latência de E/S do banco de dados em classes de instância 24XL.
- Inclusão de suporte para cifras SSL ECDHE. Para obter mais informações sobre como configurar seus clientes para usar essas cifras SSL, consulte a seguinte documentação do MySQL: [cifras de protocolos de conexão criptografados](#)
- Correção de problemas de segurança relacionados à integração do Aurora MySQL com outros serviços da AWS, como o Simple Storage Service (Amazon S3), o Amazon ML e o AWS Lambda.

- Correção de um problema que podia fazer com que a reinicialização da instância do banco de dados falhasse se o banco de dados tivesse aproximadamente 1 GB de combinações de usuários e privilégios.
- Correção de um problema com o Parallel Query que pode fazer com que o banco de dados retorne agrupamentos incorretos ou uma ordem de classificação incorreta ao executar consultas com a cláusula GROUP BY e uma cláusula WHERE contendo um predicado de intervalo.
- Correção de um problema que pode fazer com que tabelas general_log e slow_log fiquem inacessíveis após a atualização da versão principal no local do Aurora MySQL 1.x (compatível com MySQL 5.6) para o Aurora MySQL 2.x (compatível com MySQL 5.7).
- Correção de um problema que, em casos raros, pode fazer com que a instância de banco de dados seja reiniciada quando tabelas innodb_trx, innodb_locks ou innodb_lockwaits são consultadas durante cenários de workload pesado no banco de dados. Ferramentas e recursos de monitoramento, como o Performance Insights, podem consultar essas tabelas.
- Corrigido um problema em que o valor de uma coluna TIMESTAMP de uma linha existente era atualizado para o timestamp mais recente quando todas as seguintes condições eram atendidas:
 1. Existe um acionador para a tabela.
 2. Um INSERT é executado na tabela que tem uma cláusula ON DUPLICATE KEY UPDATE.
 3. A linha inserida causa uma violação de valor duplicado em um índice UNIQUE ou PRIMARY KEY.
 4. Uma ou mais colunas são do tipo de dados TIMESTAMP e têm um valor padrão de CURRENT_TIMESTAMP.
- Correção de um problema que, em casos raros, poderia impedir que uma réplica de log binário se conectasse a uma instância com o log binário ativado.
- Correção de um problema em que, em raras condições, as transações não podiam ser confirmadas ao serem executadas em uma instância com o log binário ativado.
- Correção de um problema em que novas conexões não podiam ser estabelecidas para uma instância com o log binário ativado.
- Correção de um problema que pode causar registro em log interno excessivo ao tentar aplicar patches de tempo de inatividade zero e reiniciar, fazendo com que o armazenamento local fosse preenchido.
- Correção de um problema que fazia com que uma réplica de log binário parasse com um erro HA_ERR_FOUND_DUPP_KEY ao replicar determinadas instruções DDL e DCL. Esse problema ocorre quando a instância de origem está configurada com o formato de registro em log binário MIXED e o nível de isolamento READ COMMITTED ou READ UNCOMMITTED.

- Correção de um problema em que o thread de E/S de replicação de log binário não conseguia acompanhar a instância principal quando a replicação de vários segmentos estava habilitada
- Correção de um problema em que, em raras condições, um grande número de conexões ativas com a instância do banco de dados pode fazer com que a métrica CommitLatency do CloudWatch fosse relatada incorretamente.
- Correção de um problema que fazia com que o armazenamento local em instâncias do Graviton seja preenchido ao executar LOAD FROM S3 ou SELECT INTO S3.
- Correção de um problema que pode causar resultados de consulta errados ao consultar uma tabela com uma chave externa e ambas as condições a seguir são atendidas:
 1. Cache de consultas habilitado
 2. Uma transação com uma exclusão ou atualização em cascata nessa tabela é revertida
- Correção de um problema que, em raras condições, poderia fazer com que as instâncias do leitor do Aurora fossem reiniciadas. A chance de esse problema ocorrer aumenta à medida que o número de reversões de transações aumenta.
- Corrigido um problema em que o número de ocorrências mutex 'LOCK_epoch_id_master' no esquema de performance aumenta quando uma sessão é aberta e fechada.
- Correção de um problema que pode causar um número crescente de deadlocks para workloads que têm muitas transações atualizando o mesmo conjunto de linhas simultaneamente.
- Corrigido um problema que, em raras condições, poderia fazer com que as instâncias fossem reiniciadas quando o volume do banco de dados aumentasse para um múltiplo de 160 GB.
- Correção de um problema com o Parallel Query que poderia fazer com que o banco de dados fosse reiniciado ao executar instruções SQL com uma cláusula LIMIT.
- Correção de um problema que, em raras condições, poderia fazer com que a instância do banco de dados fosse reiniciada ao utilizar transações XA no nível de isolamento READ COMMITTED.
- Correção de um problema em que, após uma instância de leitura do Aurora ser reiniciada, ela pode ser reiniciada novamente se houver um workload DDL pesado durante a reinicialização.
- Correção de um problema com a geração de relatórios incorretos de atraso de replicação do leitor do Aurora.
- Correção de um problema que, em raras condições, pode fazer com que uma instância de gravador seja reiniciada quando uma verificação de integridade de dados na memória apresenta falhas.

- Correção de um problema que, em raras condições, mostrava o gráfico “Database Load” em sessões do Performance Insights (PI) utilizando a CPU ativamente, mesmo que as sessões tivessem encerrado o processamento e estivessem ociosas.
- Correção de um problema que, em raras condições, pode fazer com que o servidor de banco de dados seja reiniciado quando uma consulta é processada usando o Parallel Query.
- Correção de um problema que, em condições raras, pode fazer com que a instância de gravador no cluster primário do banco de dados global seja reiniciada devido a uma condição de corrida durante a replicação global do banco de dados.
- Correção de um problema que pode ocorrer durante a reinicialização de uma instância de banco de dados, que pode causar mais de uma reinicialização.

Integração de correções de bug do MySQL Community Edition

- Corrigido um problema no InnoDB em que um erro no código relacionado às estatísticas de tabela gerava uma afirmação no arquivo fonte dict0stats.cc. (Bug #24585978)
- Correção de um problema em que um índice secundário sobre uma coluna virtual ficava corrompido quando o índice era construído online. Para instruções [UPDATE](#), corrigimos isso da seguinte maneira: se o valor da coluna virtual do registro de índice estiver definido como NULL, geramos esse valor a partir do registro de índice do cluster. (Bug #30556595)
- Correção de um problema no InnoDB em que a exclusão de linhas marcadas era capaz de adquirir um bloqueio de leitura externo antes de uma reversão parcial ser concluída. O bloqueio de leitura externo impediu a conversão de um bloqueio implícito em um bloqueio explícito durante a reversão parcial, causando uma falha de afirmação. (Bug #29195848)
- Correção de um problema em que os nomes de host vazios nas contas podiam fazer com que o servidor se comportasse mal. (Bug #28653104)
- Correção de um problema no InnoDB em que uma interrupção de consulta durante uma espera de bloqueio causava um erro. (Bug #28068293)
- Correção de um problema na replicação em que as transações intercaladas às vezes podiam travar o aplicador secundário quando o nível de isolamento da transação era definido como [REPEATABLE READ](#). (Erro nº 25040331)
- Correção de um problema que pode fazer com que as réplicas de log binário parassem devido ao tempo limite de espera de bloqueio. (Bug #27189701)

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7).

- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para funções AWS Lambda chamadas de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta

- Filtragem de replicação
- A declaração SQL CREATE TABLESPACE

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-10-21 (versão 2.10.1) (obsoleta)

Versão: 2.10.1

O Aurora MySQL 2.10.1 está disponível para o público em geral. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Você pode atualizar um cluster de banco de dados existente do Aurora MySQL 2.* para o Aurora MySQL 2.10.0. Para clusters que executam o Aurora MySQL versão 1, você pode atualizar um cluster existente do Aurora MySQL 1.23 ou superior diretamente para 2.10.0. Também é possível restaurar um snapshot de qualquer versão atualmente compatível do Aurora MySQL para o Aurora MySQL 2.10.0.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correção de problemas de segurança e CVEs listados abaixo:

Correções e outras melhorias para ajustar o tratamento em um ambiente gerenciado. Correções adicionais do CVE abaixo:

- [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](#)

Melhorias de disponibilidade:

- Adicionada a capacidade de desligar o cluster de forma limpa para futuros upgrades de versões principais.

Melhorias gerais:

- Corrigido um problema que poderia causar alto consumo de CPU nas instâncias de leitor devido ao log excessivo de mensagens informativas em arquivos de log de diagnóstico internos.
- Corrigido um problema em que o valor de uma coluna `TIMESTAMP` de uma linha existente era atualizado para o timestamp mais recente quando todas as seguintes condições eram atendidas:
 1. Existe um acionador para a tabela.
 2. Um `INSERT` é executado na tabela que tem uma cláusula `ON DUPLICATE KEY UPDATE`.
 3. A linha inserida causa uma violação de valor duplicado em um índice `UNIQUE` ou `PRIMARY KEY`.
 4. Uma ou mais colunas são do tipo de dados `TIMESTAMP` e têm um valor padrão de `CURRENT_TIMESTAMP`.
- Corrigido um problema introduzido na versão 2.10.0 que fazia com que o uso da função `json_merge` gerasse um código de erro em determinados casos. Especificamente, quando a função `json_merge` é utilizada em uma DDL contendo colunas geradas, ela pode retornar o código de erro 1305.

- Corrigido um problema em que, em raras condições, réplicas de leitura reiniciam quando o histórico de atualizações de um objeto grande está sendo validado para a visualização de leitura de uma transação na réplica de leitura.
- Corrigido um problema que, em raras condições, faz com que uma instância de gravador seja reiniciada quando uma verificação de integridade de dados na memória falha.

Integração de correções de bug da edição MySQL community

- CURRENT_TIMESTAMP PRODUCES ZEROS IN TRIGGER. (Bug #25209512)

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7).

- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para funções AWS Lambda chamadas de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL CREATE TABLESPACE

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-05-25 (versão 2.10.0) (obsoleta)

Versão: 2.10.0

O Aurora MySQL 2.10.0 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Você pode atualizar um cluster de banco de dados existente do Aurora MySQL 2.* para o Aurora MySQL 2.10.0. Para clusters que executam o Aurora MySQL versão 1, você pode atualizar um cluster existente do Aurora MySQL 1.23 ou superior diretamente para 2.10.0. Também é possível restaurar um snapshot de qualquer versão atualmente compatível do Aurora MySQL para o Aurora MySQL 2.10.0.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

 Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correção de problemas de segurança e CVEs listados abaixo:

Correções e outras melhorias para ajustar o tratamento em um ambiente gerenciado. Correções adicionais do CVE abaixo:

- [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](#)

Novos recursos:

- A classe de instância `db.t3.large` agora é compatível com o Aurora MySQL.
- Replicação de log binário:

- Introduzido o cache de E/S de binlog para melhorar a performance do binlog, reduzindo a contenção entre processos do gravador e processos de despejo. Para obter mais informações, consulte [Otimização da replicação de log binário](#) no Guia do usuário do Amazon Aurora.
- No [Aurora MySQL versão 2.08](#), introduzimos melhorias no processamento de log binários (binlog) para reduzir o tempo de recuperação de falhas e a latência de tempo de confirmação quando transações muito grandes estão envolvidas. Agora, essas melhorias são compatíveis com clusters que têm o GTID ativado.
- Maior disponibilidade da instância do leitor:
 - Anteriormente, quando uma instância do gravador era reiniciada, todas as instâncias do leitor em um cluster do Aurora MySQL também eram reiniciadas. Com o lançamento de hoje, as instâncias de leitor na região continuam a atender solicitações de leitura durante a reinicialização da instância de gravador, melhorando a disponibilidade de leitura no cluster. Para obter mais informações, consulte [Rebooting an Aurora MySQL cluster \(version 2.10 and higher\)](#) no Guia do usuário do Amazon Aurora.

 Important

Depois de atualizar para o Aurora MySQL 2.10, a reinicialização da instância de gravador não executará uma reinicialização de todo o cluster. Se você quiser reinicializar todo o cluster, agora você reinicializa todas as instâncias do leitor no cluster depois de reinicializar a instância de gravador.

- Melhorada a performance das leituras de página de leitura antecipada solicitadas pela técnica lógica de leitura antecipada (LRA) . Isso foi feito agrupando as várias leituras de página em lotes em uma única solicitação enviada ao armazenamento do Aurora. Como resultado, as consultas que usam a otimização LRA são executadas até três vezes mais rápido.
- Reinicializações e aplicação de patches com zero tempo de inatividade:
 - Melhorias na reinicialização com tempo de inatividade zero (ZDR) e aplicação de patch com tempo de inatividade zero (ZDP) para habilitar o ZDR e o ZDP em vários cenários, incluindo suporte adicional para casos em que o registro em log binário está habilitado. Além disso, foi melhorada a visibilidade dos eventos de ZDR e ZDP. Consulte a documentação para obter detalhes: [Zero-downtime restart \(ZDR – Reinício com tempo de inatividade zero\) para Amazon Aurora MySQL](#) e [Como usar os patches com tempo de inatividade zero](#) no Guia do usuário do Amazon Aurora.

Melhorias de disponibilidade:

- Melhorias para uma inicialização mais rápida quando o banco de dados tem um grande número de índices temporários e tabelas criados durante uma atividade DDL interrompida anteriormente.
- Correção de vários problemas relacionados a reinicializações repetidas durante a recuperação após falha de operações de DDL interrompidas, como `DROP TRIGGER`, `ALTER TABLE` e, especificamente, `ALTER TABLE`, que modifica o tipo de particionamento ou o número de partições em uma tabela.
- Corrigido um problema que poderia causar a reinicialização do servidor durante o processamento de log do Database Activity Streams (DAS).
- Corrigido um problema ao imprimir uma mensagem de erro durante o processamento de uma consulta `ALTER` em tabelas do sistema.

Melhorias gerais:

- Corrigido um problema em que o cache de consultas poderia retornar resultados obsoletos em uma instância de leitor.
- Corrigido um problema em que algumas métricas de confirmação do Aurora não estavam sendo atualizadas quando a variável do sistema `innodb_flush_log_at_trx_commit` estava definida como 0 ou 2.
- Corrigido um problema em que um resultado de consulta armazenado no cache de consulta não era atualizado por transações com várias instruções.
- Corrigido um problema que poderia fazer com que o carimbo de data/hora da última modificação dos arquivos de log binário não fosse atualizado corretamente. Isso podia levar à limpeza prematura os arquivos de log binários, antes de atingir o período de retenção configurado pelo cliente.
- Corrigida a posição e o nome de arquivo do binlog relatados como incorretos do InnoDB após a recuperação de falhas.
- Correção de um problema que poderia fazer com que grandes transações gerassem eventos de logs binários incorretos se o parâmetro `binlog_checksum` fosse definido como `NONE`.
- Correção de um problema que fazia com que uma réplica de log binário parasse com um erro, se a transação replicada contivesse uma instrução DDL e um grande número de alterações de linha.
- Corrigido um problema que levava a uma reinicialização em uma instância de leitor ao descartar uma tabela.

- Corrigido um problema que fazia com que os conectores de código aberto falhassem ao tentar consumir um arquivo de binlog com uma grande transação.
- Corrigido um problema que poderia levar a resultados de consulta incorretos na coluna de geometria grande depois de criar um índice espacial na tabela com os grandes valores de geometria.
- O banco de dados agora recria o tablespace temporário durante a reinicialização, o que permite que o espaço de armazenamento associado seja liberado e recuperado.
- Correção de um problema que impedia que as tabelas `performance_schema` fossem truncadas em instâncias de leitor do Aurora.
- Corrigido um problema que fazia com que uma réplica de log binário parasse com um erro `HA_ERR_KEY_NOT_FOUND`.
- Corrigido um problema que fazia com que o banco de dados fosse reiniciado ao executar a instrução `FLUSH TABLES WITH READ LOCK`.
- Corrigido um problema que impedia o uso de funções de bloqueio no nível do usuário em instâncias de leitor do Aurora.

Integração de correções de bug da edição MySQL community

- As transações intercaladas poderiam, por vezes, impedir o aplicador de réplica quando o nível de isolamento da transação foi definido como [LEITURA REPETIDA](#). (Erro nº 25040331)
- Quando um procedimento armazenado continha uma instrução referente a uma visão que, por sua vez, se referia a outra visão, não era possível invocar o procedimento com sucesso mais de uma vez. (Erro n.º 87858, erro n.º 26864199)
- Para consultas com muitas condições OR, o otimizador agora é mais eficiente em memória e menos provável exceder o limite de memória imposto pela variável de sistema [range_optimizer_max_mem_size](#). Além disso, o valor padrão dessa variável foi aumentado de 1.536.000 para 8.388.608. (Erro n.º 79450, erro n.º 22283790)
- Replicação: na função `next_event()`, que é chamada por um processo SQL de uma réplica para ler o próximo evento do log de retransmissão, o processo SQL não liberava o `relaylog.log_lock` adquirido quando ocorria um erro (por exemplo, devido a um log de retransmissão fechado), fazendo com que todos os outros processos ficassem esperando para adquirir um bloqueio no log de retransmissão para continuar. Com essa correção, o bloqueio é liberado antes que o processo SQL deixe a função sob a situação. (Bug nº 21697821)

- Corrigindo uma corrupção de memória em ALTER TABLE com coluna virtual. (Erro n.º 24961167, erro n.º 24960450)
- Replicação: réplicas de vários processos não poderiam ser configuradas com pequenos tamanhos de fila usando [slave_pending_jobs_size_max](#) se precisassem processar transações maiores que esse tamanho. Qualquer pacote maior que [slave_pending_jobs_size_max](#) era rejeitado com o erro ER_MTS_EVENT_BIGGER_PENDING_JOBS_SIZE_MAX, mesmo que o pacote fosse menor que o limite definido por [slave_max_allowed_packet](#). Com essa correção, [slave_pending_jobs_size_max](#) se torna um limite flexível em vez de um limite rígido. Se o tamanho de um pacote exceder [slave_pending_jobs_size_max](#), mas for menor que [slave_max_allowed_packet](#), a transação será retida até que todos os operadores de réplica tenham filas vazias e, em seguida, é processada. Todas as transações subsequentes são retidas até que a grande transação seja concluída. O tamanho da fila para operadores de réplica pode, portanto, ser limitado e ainda permitir transações maiores ocasionais. (Erro n.º 21280753, erro n.º 77406)
- Replicação: ao usar uma réplica de vários processos, os erros do aplicador exibiram dados de ID do trabalhador que eram inconsistentes com os dados externalizados nas tabelas de replicação do Performance Schema. (Erro n.º 25231367)
- Replicação: em uma réplica de replicação baseada em GTID em execução com [-gtid-mode=ON](#), [-log-bin=OFF](#) e utilizando [-slave-skip-errors](#), quando era encontrado um erro que deveria ser ignorado, Exec_Master_Log_Pos não era atualizado corretamente, fazendo com que Exec_Master_Log_Pos perdesse a sincronia com Read_master_log_pos. Se um GTID_NEXT não estava especificado, a réplica nunca atualizaria seu estado GTID ao reverter de uma única transação de instrução. O Exec_Master_Log_Pos não seria atualizado porque, mesmo que a transação tivesse sido concluída, seu estado GTID mostraria o contrário. A correção remove a restrição da atualização do estado GTID quando uma transação é revertida somente se GTID_NEXT estiver especificado. (Erro n.º 22268777)
- Replicação: uma instrução com falha parcial não estava consumindo corretamente um GTID gerado ou especificado automaticamente quando o registro em log binário era desabilitado. A correção garante que um [DROP TABLE](#), [DROP USER](#) ou [DROP VIEW](#) com falha parcial consuma respectivamente o GTID relevante e o salve em @@GLOBAL.GTID_EXECUTED e na tabela mysql.gtid_executed quando o registro em log binário está desabilitado. (Erro n.º 21686749)
- Replicação: as réplicas que executam o MySQL 5.7 não conseguiam se conectar a uma origem do MySQL 5.5 devido a um erro ao recuperar o [server_uuid](#), que não faz parte do MySQL 5.5.0 Isso foi causado por mudanças no método de recuperação do server_uuid. (Erro n.º 22748612)

- Replicação: o mecanismo de salto de transação GTID que ignora silenciosamente uma transação GTID executada anteriormente não funcionava corretamente para transações XA. (Erro nº 25041920)
- Declarações [">XA ROLLBACK](#), que falharam porque foi fornecido o ID incorreto de uma transação, poderiam ser registradas no log binário com o ID correto da transação e, portanto, poderiam ser acionadas por réplicas de replicação. Agora é feita uma verificação da situação de erro antes que o registro em log binário ocorra, e ROLLBACK as instruções XA com falha não são registradas em log. (Erro nº 26618925)
- Replicação: se uma réplica fosse configurada utilizando uma instrução [CHANGE MASTER TO](#) que não especificava o nome do arquivo e a posição do log de origem e depois fosse desligada antes da emissão de [START SLAVE](#) e em seguida reiniciada com a opção [-relay-log-recovery](#) definida, a replicação não era iniciada. Isso aconteceu porque o processo do receptor não havia sido iniciado antes da tentativa de recuperação do log de retransmissão, portanto, nenhum evento de rotação de log estava disponível no log de retransmissão para fornecer o nome de arquivo e a posição do log de origem. Nessa situação, a réplica agora ignora a recuperação do log de retransmissão e registra um aviso. Em seguida, prossegue para iniciar a replicação. (Erro n.º 28996606, erro n.º 93397)
- Replicação: na replicação baseada em linha, uma mensagem que exibia incorretamente comprimentos de campo era retornada ao replicar de uma tabela com uma coluna `utf8mb3` para uma tabela da mesma definição em que a coluna foi definida com um conjunto de caracteres `utf8mb4`. (Erro n.º 25135304, erro n.º 83918)
- Replicação: quando uma instrução [RESET SLAVE](#) era emitida em uma réplica de replicação com GTIDs em uso, os arquivos de log de retransmissão existentes eram removidos, mas o novo arquivo de log de retransmissão de substituição era gerado antes que o conjunto de GTIDs recebidos para o canal fosse limpo. Portanto, o antigo conjunto GTID era gravado no novo arquivo de log de retransmissão como o evento `PREVIOUS_GTIDS`, causando um erro fatal na replicação informando que a réplica tinha mais GTIDs do que a origem, mesmo que o conjunto `gtid_executed` definido para ambos os servidores estivesse vazio. Agora, quando `RESET SLAVE` é emitido, o conjunto de GTIDs recebidos é apagado antes que o novo arquivo de log de retransmissão seja gerado, para que essa situação não ocorra. (Erro nº 27411175)
- Replicação: com GTIDs em uso para replicação, transações incluindo instruções que causaram um erro de análise ([ER_PARSE_ERROR](#)) não puderam ser ignoradas manualmente pelo método recomendado de injetar uma transação vazia ou de substituição com o mesmo GTID. Essa ação deve resultar na réplica que identifica o GTID como já usado e, portanto, ignorando a transação indesejada que compartilhou seu GTID. No entanto, no caso de um erro de análise, como a

instrução era analisada antes que o GTID fosse verificado para ver se ele precisava ser ignorado, o processo do aplicador de replicação parava devido ao erro de análise, mesmo que a intenção fosse ignorar a transação de qualquer maneira. Com essa correção, o processo do aplicador de replicação agora ignora erros de análise se a transação em questão precisar ser ignorada porque o GTID já foi usado. Observe que essa alteração de comportamento não se aplica no caso de workloads que consistem em saída de log binário produzida por `mysqlbinlog`. Nessa situação, haveria o risco de que uma transação com um erro de análise que viesse imediatamente após uma transação ignorada também fosse ignorada silenciosamente, quando deveria gerar um erro. (Erro nº 27638268)

- Replicação: habilite o processo SQL para que o GTID ignore uma transação parcial. (Erro nº 25800025)
- Replicação: quando um parâmetro de tempo limite negativo ou fracionário era fornecido a `WAIT_UNTIL_SQL_THREAD_AFTER_GTIDS()`, o servidor se comportava de maneiras inesperadas. Com esta correção:
 - Um valor de tempo limite fracionário é lido como está, sem arredondamento.
 - Um valor de tempo limite negativo é rejeitado com um erro se o servidor estiver em um modo SQL estrito. Se o servidor não estiver em um modo SQL estrito, o valor fará com que a função retorne NULL imediatamente sem qualquer espera e, em seguida, emita um aviso. (Erro nº 24976304, erro nº 83537)
- Replicação: se a função `WAIT_FOR_EXECUTED_GTID_SET()` fosse usada com um valor de tempo limite incluindo uma parte fracionada (por exemplo, 1,5), um erro na lógica de transmissão significava que o tempo limite foi arredondado para o segundo inteiro mais próximo e para zero para valores menores que 1 segundo (por exemplo, 0,1). Agora a lógica de transmissão foi corrigida para que o valor do tempo limite seja aplicado conforme especificado originalmente, sem arredondamento. Agradecemos a Dirkjan Bussink pela contribuição. (Erro nº 29324564, erro nº 94247)
- Com os GTIDs ativados, o [XA COMMIT](#) em uma transação XA desconectada dentro de uma transação de várias instruções gerava uma afirmação. (Erro nº 22173903)
- Replicação: uma afirmação era gerada em compilações de depuração se uma instrução [XA ROLLBACK](#) fosse emitida para um identificador de transação desconhecido quando o valor de `gtid_next` tivesse sido definido manualmente. Agora, o servidor não tenta atualizar o estado de GTID se uma instrução XA ROLLBACK falhar com um erro. (Erro nº 27928837, erro nº 90640)
- Corrigido o problema de ordem de classificação incorreta quando várias funções CASE são usadas na cláusula `ORDER BY` (Erro nº 22810883).

- Algumas consultas que usavam ordenação podiam acessar uma coluna não inicializada durante a otimização e causar a saída do servidor. (Bug nº 27389294)

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7).

- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para funções AWS Lambda chamadas de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens

- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL `CREATE TABLESPACE`

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-11-12 (versão 2.09.3) (obsoleta)

Versão: 2.09.3

O Aurora MySQL 2.09.3 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Você pode atualizar um cluster de banco de dados existente do Aurora MySQL 2.* para o Aurora MySQL 2.10.0. Para clusters que executam o Aurora MySQL versão 1, você pode atualizar um cluster existente do Aurora MySQL 1.23 ou superior diretamente para 2.10.0. Também é possível restaurar um snapshot de qualquer versão atualmente compatível do Aurora MySQL para o Aurora MySQL 2.10.0.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo pelo Console de gerenciamento da AWS, pela AWS CLI ou pela API do Amazon RDS.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correções de segurança:

Correções e outras melhorias para ajustar o tratamento em um ambiente gerenciado. Correções adicionais do CVE abaixo:

- [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](#)

Melhorias de disponibilidade:

- Introdução de uma otimização capaz de reduzir a contenção de consultas executadas em tabelas em `information_schema`.
- Inclusão de suporte para cifras SSL ECDHE.

Melhorias gerais:

- Correção de um problema que, em raras condições, pode fazer com que uma instância de gravador seja reiniciada quando uma verificação de integridade de dados na memória apresenta falhas.
- Correção de um problema que, em raras condições, pode fazer com que a instância de banco de dados seja reiniciada quando o volume do cluster está se expandindo enquanto o registro em log binário está habilitado.
- Correção de uma condição rara de corrida durante a reinicialização de uma instância de banco de dados que pode causar mais de uma reinicialização.
- Correção de um problema que podia fazer com que a reinicialização da instância do banco de dados falhasse se o banco de dados tivesse muitas combinações de usuários e privilégios.
- Correção de um problema com consulta paralela que poderia fazer com que o banco de dados fosse reiniciado ao executar instruções SQL com a cláusula LIMIT.
- Correção de um problema com a geração de relatórios incorretos de atraso de replicação do Aurora.
- Correção de um problema que pode fazer com que tabelas `general_log` e `slow_log` fiquem inacessíveis após o upgrade da versão principal no local do Aurora-MySQL 1.x (com base no MySQL 5.6) para o Aurora-MySQL 2.x (baseado no MySQL 5.7).
- Correção de um problema que, em casos raros, pode fazer com que a instância de banco de dados seja reiniciada quando tabelas `innodb_trx`, `innodb_locks` ou `innodb_lockwaits` são consultadas durante cenários de workload pesada no banco de dados. Ferramentas e recursos de monitoramento, como o Performance Insights, podem consultar essas tabelas.
- Correção de um problema que pode fazer com que uma instância de banco de dados seja reiniciada quando a instrução SQL “FLUSH TABLES WITH READ LOCK” é executada.
- Correção de um problema em que o processo de limpeza do InnoDB é pausado durante a exclusão de uma instância de leitor, resultando em um aumento temporário no comprimento da lista de histórico.
- Correção de um problema com consultas paralelas que pode fazer com que o banco de dados seja reiniciado ao executar uma instrução SQL em uma tabela contendo uma coluna virtual.
- Correção de um problema com consultas paralelas que pode fazer com que o banco de dados retorne agrupamentos incorretos ou uma ordem de classificação incorreta ao executar consultas com a cláusula GROUP BY e uma cláusula WHERE contendo um predicado de intervalo.
- Correção de um problema em consultas paralelas que, em raras condições, pode fazer com que o banco de dados seja reiniciado ao executar instruções SQL com funções JSON.

- Correção de um problema que, em condições raras, pode fazer com que a instância de gravador no cluster primário do Banco de dados global seja reiniciada devido a uma condição de corrida durante a Replicação global do banco de dados.
- Correção de um problema que fazia com que uma réplica de binlog parasse com um erro `HA_ERR_FOUND_DUPP_KEY` ao replicar determinadas instruções DDL e DCL. Esse problema ocorre quando a instância de origem está configurada com o formato de registro em log binário `MIXED` e o nível de isolamento `READ COMMITTED` ou `READ UNCOMMITTED`.
- Corrigido um problema que, em raras condições, poderia fazer com que a instância do banco de dados fosse reiniciada ao utilizar transações XA no nível de isolamento `READ COMMITTED`.
- Correção de um problema em que o valor de uma coluna `TIMESTAMP` de uma linha existente era atualizado para o timestamp mais recente quando todas as seguintes condições eram atendidas: 1. existe um acionador para a tabela; 2. é feito um `INSERT` na tabela que tem uma cláusula `ON DUPLICATE KEY UPDATE`; 3. a linha inserida pode causar uma violação de valor duplicado em um índice `UNIQUE` ou `PRIMARY KEY`; e 4. uma ou mais colunas são do tipo de dados `TIMESTAMP` e têm um valor padrão de `CURRENT_TIMESTAMP`.
- Correção de um problema que, em raras condições, poderia fazer com que uma instância de leitor fosse reiniciada devido a um processamento de verificação incorreto.
- Correção de um problema que pode fazer com que a instância de leitor seja reiniciada quando a instância de gravador aumenta o volume do banco de dados a ponto de ultrapassar limites de tamanho de volume específicos.
- Correção de um problema que pode causar tempos de reinicialização mais longos para instâncias de banco de dados utilizando volumes de cluster clonados.
- Correção de um problema em que a reinicialização de uma instância de banco de dados pode falhar uma ou mais vezes depois que uma operação `TRUNCATE TABLE` é executada na instância de gravador.
- Correção de um problema que, em raras condições, poderia fazer com que a instância do banco de dados fosse reiniciada.
- Correção de um problema que, em raras condições, poderia fazer com que a instância do gravador seja reiniciada quando o volume do banco de dados crescesse para um múltiplo de 160 GB.

Integração de correções de bug da edição MySQL community

- Bug #23533396 - Ao adicionar um novo índice, o servidor descartava um índice de chave externa internamente definido e tentava utilizar um índice secundário definido em uma coluna gerada

virtual como o índice de chave externa, causando um encerramento do servidor. Agora, o InnoDB permite que uma restrição de chave externa faça referência a um índice secundário definido em uma coluna virtual gerada.

- Bug #29550513 - Replicação: um problema de bloqueio na função `WAIT_FOR_EXECUTED_GTID_SET()` pode fazer com que o servidor trave em certas circunstâncias. O problema já foi corrigido.

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7).

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para funções AWS Lambda chamadas de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL `CREATE TABLESPACE`

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-02-26 (versão 2.09.2) (obsoleta)

Versão: 2.09.2

O Aurora MySQL 2.09.2 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Você pode atualizar um cluster de banco de dados existente do Aurora MySQL 2.* para o Aurora MySQL 2.09.2. Para clusters que executam a versão 1 do Aurora MySQL, você pode atualizar um cluster existente da versão 1.23 ou superior do Aurora MySQL diretamente para a 2.09.2. Também é possível restaurar um snapshot de qualquer versão atualmente compatível do Aurora MySQL para a versão 2.09.2 do Aurora MySQL.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo pelo Console de gerenciamento da AWS, pela AWS CLI ou pela API do Amazon RDS.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

 Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Novos atributos:

- Agora os clusters do Aurora MySQL oferecem suporte às seguintes instâncias do EC2 R6g com a tecnologia dos processadores AWS Graviton2 com base em Arm: r6g.large, r6g.xlarge, r6g.2xlarge, r6g.4xlarge, r6g.8xlarge, r6g.12xlarge, r6g.16xlarge. Para obter mais informações, consulte [Classes de instância de banco de dados Aurora](#) no Guia do usuário do Amazon Aurora.

Correções de segurança:

Correções e outras melhorias para ajustar o tratamento em um ambiente gerenciado. Correções adicionais do CVE abaixo:

- [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](#)

Melhorias de disponibilidade:

- Correção de um problema introduzido na versão 2.09.0 que pode causar latência de gravação elevada durante o dimensionamento do volume de armazenamento de cluster.

- Correção de um problema no recurso de redimensionamento dinâmico que poderia fazer com que as réplicas de leitura do Aurora reiniciassem.
- Correção de um problema que poderia causar mais tempo de inatividade durante a atualização de 1.23.* para 2.09.*.
- Correção de um problema em que um DDL ou DML poderia causar reinicialização do mecanismo durante uma solicitação de pré-busca de página.
- Correção de um problema que fazia com que uma réplica de log binário parasse com um erro, se a transação replicada contivesse uma instrução DDL e um grande número de alterações de linha.
- Correção de um problema em que um banco de dados agindo como uma réplica de log binário podia ser reiniciado durante a replicação de um evento DDL na tabela de mysql time_zone.
- Correção de um problema que poderia fazer com que grandes transações gerassem eventos de logs binários incorretos se o parâmetro binlog_checksum fosse definido como NONE.
- Correção de um problema que fazia com que uma réplica de log binário parasse com um erro HA_ERR_KEY_NOT_FOUND.
- Melhoria da estabilidade geral.

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas esses recursos não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7)

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para funções AWS Lambda chamadas de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).

- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL `CREATE TABLESPACE`

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-12-11 (versão 2.09.1) (obsoleta)

Versão: 2.09.1

O Aurora MySQL 2.09.1 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Você pode atualizar um cluster de banco de dados existente do Aurora MySQL 2.* para a versão 2.09.1 do Aurora MySQL. Para clusters que executam a versão 1 do Aurora MySQL, você pode atualizar um cluster existente da versão 1.23 ou superior do Aurora MySQL diretamente para a 2.09.1. Também é possível restaurar um snapshot de qualquer versão atualmente compatível do Aurora MySQL para a versão 2.09.1 do Aurora MySQL.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo pelo Console de gerenciamento da AWS, pela AWS CLI ou pela API do RDS.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correções de segurança:

Correções e outras melhorias para ajustar o tratamento em um ambiente gerenciado. Correções adicionais do CVE abaixo:

- [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](#)

Alterações incompatíveis:

Esta versão introduz uma alteração de permissão que afeta o comportamento do `mysqldump` comando. Os usuários devem ter o `PROCESS` privilégio para acessar a `INFORMATION_SCHEMA.FILES` tabela. Para executar o `mysqldump` comando sem alterações, conceda o `PROCESS` privilégio ao usuário do banco de dados ao qual o `mysqldump` comando se conecta. Você também pode executar o comando `mysqldump` com a opção `--no-tablespaces`. Com essa opção, a saída `mysqldump` não inclui nenhuma `CREATE LOGFILE GROUP` ou instruções do `CREATE TABLESPACE`. Nesse caso, o comando `mysqldump` não acessa a tabela `INFORMATION_SCHEMA.FILES` e você não precisa conceder a permissão `PROCESS`.

Melhorias de disponibilidade:

- Corrigido um problema que poderia fazer com que uma sessão de cliente travasse quando o mecanismo de banco de dados encontrava um erro ao ler ou gravar na rede.
- Corrigido um vazamento de memória no recurso de redimensionamento dinâmico, introduzido em 2.09.0.

Bancos de dados globais:

- Corrigido vários problemas em que as réplicas de uma região secundária de banco de dados global podiam ser reiniciadas quando atualizadas para a versão 2.09.0 enquanto o gravador de região principal estava em uma versão mais antiga.

Integração de correções de bug da edição MySQL community

- Replicação: As transações intercaladas poderiam, por vezes, impedir o aplicador escravo quando o nível de isolamento da transação foi definido como [LEITURA REPETIDA](#). (Erro nº 25040331)
- Para uma tabela com uma coluna [TIMESTAMP](#) ou [DATETIME](#) cujo valor padrão é [CURRENT_TIMESTAMP](#), a coluna podia ser inicializada em `0000-00-00 00:00:00` caso a tabela tivesse um acionador `BEFORE INSERT`. (Erro nº 25209512, erro nº 84077)

- Para uma instrução [INSERIR](#) para a qual a VALUES lista produziu valores para a segunda linha ou linha posterior usando uma subconsulta contendo uma junção, o servidor pode sair depois de não resolver os privilégios necessários. (Erro nº 23762382)

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas esses recursos não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7)

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para funções AWS Lambda chamadas de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plugin de replicação de grupo
- Maior tamanho de página

- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL `CREATE TABLESPACE`

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-09-17 (versão 2.09.0) (obsoleta)

Versão: 2.09.0

Aurora MySQL 2.09.0 está disponível para o público em geral. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Você pode restaurar um snapshot de Aurora MySQL 1.23.* para Aurora MySQL 2.09.0. Também existe a opção de atualizar os clusters de banco de dados existentes do Aurora MySQL 2.* para o Aurora MySQL 2.09.0. Não é possível atualizar um cluster existente do Aurora MySQL 1.23.* diretamente para 2.09.0. No entanto, é possível restaurar seu snapshot para o Aurora MySQL 2.09.0.

Important

As melhorias no Aurora armazenamento nesta versão limitam os caminhos de atualização disponíveis de Aurora MySQL 1.* para Aurora MySQL 2.09. Quando você atualiza um cluster Aurora MySQL 1.* para 2.09, você deve atualizar a partir de Aurora MySQL 1.23.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo pelo Console de gerenciamento da AWS, pela AWS CLI ou pela API do RDS.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Novos atributos:

- Com essa versão, é possível criar instâncias de banco de dados MySQL do Amazon Aurora com até 128 tebibytes (TiB) de armazenamento. O novo limite de armazenamento é um aumento em relação aos 64 TiB anteriores. O tamanho de armazenamento de 128 TiB oferece suporte a bancos de dados maiores. Esse recurso não é compatível com tamanhos de instâncias pequenas (db.t2 ou db.t3). Um único tablespace não pode expandir além de 64 TiB devido às [limitações do InnoDB com tamanho de página de 16 KB](#).

O Aurora alerta quando o tamanho do volume do cluster está próximo de 128 TiB, para que seja possível executar uma ação antes de atingir o limite de tamanho. Os alertas aparecem no log mysql e RDS Events no Console de gerenciamento da AWS.

- Agora você pode ativar ou desativar a consulta paralela para um cluster existente alterando o valor do parâmetro do cluster de banco de dados `aurora_parallel_query`. Não é necessário usar a configuração `parallelquery` para o parâmetro `--engine-mode` ao criar o cluster.

A consulta paralela foi expandida para estar disponível em todas as regiões onde o Aurora MySQL está disponível.

Há uma série de outros aprimoramentos de funcionalidade e alterações nos procedimentos para atualizar e habilitar a consulta paralela em um cluster do Aurora. Para obter mais informações, consulte [Como trabalhar com a consulta paralela do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

- O Aurora redimensiona dinamicamente o espaço de armazenamento em cluster. Com o redimensionamento dinâmico, o espaço de armazenamento do cluster de bancos de dados

Aurora diminui automaticamente ao remover dados do cluster de banco de dados. Para obter mais informações, consulte [Escalabilidade de armazenamento](#) no Guia do usuário do Amazon Aurora.

 Note

O recurso de redimensionamento dinâmico está sendo implantado em fases nas regiões da AWS em que o Aurora está disponível. Dependendo da região em que o cluster está, esse recurso pode não estar disponível ainda. Para obter mais informações, consulte [o anúncio de Novidades](#).

Correções de alta prioridade:

- Backport de comunidade do erro nº 27659490: SELECIONAR USANDO INTERVALO DINÂMICO E MESCLAGEM DE ÍNDICE USA MUITA MEMÓRIA (OOM)
- Erro nº 26881508: MYSQL#1: DISABLE_ABORT_ON_ERROR IN AUTH_COMMON.H
- Backport de comunidade do erro nº 24437124: POSSÍVEL EXCESSO DE BUFFER NA CRIAÇÃO DA TABELA
- Backport do erro nº 27158030: INNODB ONLINE ALTERA TRAVAMENTOS COM DML SIMULTÂNEO
- Erro nº 29770705: SERVIDOR TRAVOU AO EXECUTAR SELEÇÃO COM CLÁUSULA WHERE ESPECÍFICA
- Backport do erro nº 26502135: MYSQLD SEGFAULTS EM MDL_CONTEXT::TRY_ACQUIRE_LOCK_IMPL
- Backport do erro nº 26935001: A ALTERAÇÃO DA TABELA AUTO_INCREMENT TENTA LER O ÍNDICE DO ESPAÇO DE TABELA DESCARTADO
- Erro nº 28491099: [FATAL] O BLOCO DE MEMÓRIA É INVÁLIDO | INNODB: FALHA DE ASSERÇÃO: UT0UT.CC:670
- Erro nº 30499288: GCC 9.2.1 REPORTA UM NOVO AVISO PARA OS_FILE_GET_PARENT_DIR
- Erro nº 29952565, em que MYSQLD TEM SINAL 11 AO EXECUTAR UMA CONSULTA (UNIÃO + ORDEM POR + SUBCONSULTA)
- Erro nº 30628268: TRAVAMENTO POR FALTA DE MEMÓRIA
- Erro nº 30441969: BUG #29723340: TRAVAMENTO DO SERVIDOR MYSQL APÓS CONSULTA SQL COM DADOS ?AST

- Erro nº 30569003: INTERRUPÇÃO DA REPLICAÇÃO DE 5.7 COM ERRO DE SINTAXE COM GERENCIAMENTO DE CONCESSÕES
- Erro nº 29915479: EXECUÇÃO COM_REGISTER_SLAVE SEM COM_BINLOG_DUMP PODE RESULTAR EM SAÍDA DO SERVIDOR
- Erro nº 30569003: INTERRUPÇÃO DA REPLICAÇÃO DE 5.7 COM ERRO DE SINTAXE COM GERENCIAMENTO DE CONCESSÕES
- Erro nº 29915479: EXECUÇÃO COM_REGISTER_SLAVE SEM COM_BINLOG_DUMP PODE RESULTAR EM SAÍDA DO SERVIDOR
- Erro nº 20712046: A EXIBIÇÃO DAS TABELAS PROCESSLIST E PERFORMANCE_SCHEMA NÃO OCULTA A SENHA NA CONSULTA
- Backport do erro nº 18898433: PERFORMANCE EXTREMAMENTE LENTA COM JUNÇÕES EXTERNAS E BUFFER DE JUNÇÃO (corrigido na versão 5.7.21). As consultas com muitas junções à esquerda ficavam lentas se o buffer de junção fosse usado (por exemplo, usando o algoritmo de loop aninhado em bloco). (Erro nº 18898433, erro nº 72854)
- Backport do erro nº 26402045: MYSQLD TRAVA DURANTE A CONSULTA (corrigido no MySQL 5.7.23). Certos casos de materialização da subconsulta poderiam causar uma saída do servidor. Essas consultas agora produzem um erro sugerindo que a materialização seja desabilitada. (Erro nº 26402045)
- [Backport do MySQL] usuários que não sejam rdsadmin não têm permissão para atualizar a tabela pfs na réplica do leitor.
- Correção do problema em que o cliente não pode atualizar o perfschema na réplica do leitor
- Erro nº 26666274: LOOP INFINITO NO CONTÊINER DE BUFFER DO ESQUEMA DE PERFORMANCE
- [Erro nº 26997096](#): o valor relay_log_space não é atualizado de forma sincronizada, de modo que seu valor é às vezes muito maior do que o espaço em disco real usado pelos logs de retransmissão.
- ERRO Nº 25082593: A VALIDAÇÃO DE CHAVE EXTERNA NÃO PRECISA ADQUIRIR BLOQUEIO DE LACUNA EM LEITURA CONFIRMADA
- [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](#)

Melhorias de disponibilidade:

- Habilite a correção de ABA do gerenciador de bloqueio por padrão.
- Correção de um problema no gerenciador de bloqueio em que uma condição de corrida podia fazer com que um bloqueio fosse compartilhado por duas transações, fazendo com que o banco de dados fosse reiniciado.
- Corrigido um problema ao criar uma tabela temporária com formato de linha compactado que poderia resultar em uma reinicialização.
- Corrija o valor padrão de `table_open_cache` em instâncias 16XL e 24XL que podem causar failovers repetidos e alta utilização da CPU em classes de instâncias grandes (R4/R5-16XL, R5-12XL, R5-24XL). Isso impactou as versões 2.07.x.
- Corrigido um problema em que a restauração de um cluster do Amazon S3 para o Aurora MySQL versão 2.08.0 levava mais tempo do que o esperado quando o backup do S3 não incluía a tabela `mysql.host`.
- Corrigido um problema que poderia causar failovers repetidos devido a atualizações de colunas virtuais com índices secundários.
- Correção de um problema relacionado ao gerenciamento de memória de bloqueio de transação com transações de gravação de longa duração resultando em uma reinicialização do banco de dados.

- Vários problemas em que o mecanismo podia travar durante a aplicação de patches com tempo de inatividade zero durante a conferência de pontos de segurança para aplicação de patches foram corrigidos.
- Corrigido um problema para ignorar o registro em log redo para tabelas temporárias, que anteriormente causava uma falha.
- Corrigida uma condição de corrida no gerenciador de bloqueio entre a eliminação de conexão/consulta e a sessão eliminada.
- Correção de um problema em que o banco de dados poderia travar se fosse uma réplica de binlog e recebesse um evento DDL sobre a tabela `time_zone` do MySQL.

Bancos de dados globais:

- A visualização de `INFORMATION_SCHEMA.REPLICA_HOST_STATUS` do MySQL em uma região secundária agora mostra as entradas para as réplicas pertencentes a essa região.
- Corrigidas falhas de consulta inesperadas que poderiam ocorrer em uma região secundária do banco de dados global após problemas temporários de conectividade de rede entre as regiões primária e secundária.
-

Consulta paralela:

- Corrigido o plano EXPLAIN para uma consulta paralela, que está incorreto para uma consulta simples de tabela única.
- Corrigido o deadlatch automático que pode ocorrer quando a consulta paralela está habilitada.

Melhorias gerais:

- A exportação para o S3 agora oferece suporte à palavra-chave `ENCRYPTION`.
- O parâmetro `aurora_binlog_replication_max_yield_seconds` agora tem um valor máximo de 36.000. O valor máximo anterior aceito era de 45. Esse parâmetro funciona somente quando o parâmetro `aurora_binlog_use_large_read_buffer` é definido como 1.
- Alterado o comportamento para mapear `MIXED binlog_format` para `ROW` em vez de `STATEMENT` ao executar `LOAD DATA FROM INFILE | S3`.

- Corrigido um problema em que uma réplica de binlog conectada a uma primária de binlog do Aurora MySQL poderia mostrar dados incompletos quando a primária executava `LOAD DATA FROM S3` e `binlog_format` estava definido como `STATEMENT`.
- Maior comprimento máximo permitido para variáveis de sistema de auditoria `server_audit_incl_users` e `server_audit_excl_users` de 1.024 bytes para 2.000 bytes.
- Corrigido um problema em que os usuários podiam perder acesso ao banco de dados ao baixar o parâmetro `max_connections` no grupo de parâmetros quando as conexões atuais fossem maiores do que o valor que estava sendo definido.
- Corrigido um problema em streams de atividade de dados em que aspas simples e barra invertida não eram escapadas corretamente.

Integração de correções de bug da edição MySQL community

- Erro nº27659490: SELECIONAR USANDO INTERVALO DINÂMICO E MESCLAGEM DE ÍNDICE USA MUITA MEMÓRIA (OOM)
- Erro nº 26881508: MYSQL#1: DISABLE_ABORT_ON_ERROR IN AUTH_COMMON.H
- Erro nº 24437124: POSSÍVEL EXCESSO DE BUFFER DURANTE A CRIAÇÃO DA TABELA
- Erro nº 27158030: INNODB ONLINE ALTERA TRAVAMENTOS COM DML SIMULTÂNEO
- Erro nº29770705: SERVIDOR TRAVOU AO EXECUTAR SELEÇÃO COM CLÁUSULA WHERE ESPECÍFICA
- Erro nº 26502135: MYSQLD SEGFAULTS EM MDL_CONTEXT::TRY_ACQUIRE_LOCK_IMPL
- Erro nº26935001: ALTERAR TABELA AUTO_INCREMENT TENTA LER ÍNDICE DO TABLESPACE DESCARTADO
- Erro nº 28491099: [FATAL] O BLOCO DE MEMÓRIA É INVÁLIDO | INNODB: FALHA DE ASSERTÇÃO: UT0UT.CC:670
- Erro nº 30499288: GCC 9.2.1 REPORTA UM NOVO AVISO PARA OS_FILE_GET_PARENT_DIR
- Erro nº 29952565: em que MYSQLD TEM SINAL 11 AO EXECUTAR UMA CONSULTA (UNIÃO + ORDEM POR + SUBCONSULTA)
- Erro nº 30628268: TRAVAMENTO POR FALTA DE MEMÓRIA
- Erro nº 30441969: BUG #29723340: TRAVAMENTO DO SERVIDOR MYSQL APÓS CONSULTA SQL COM DADOS ?AST
- Erro nº 30569003: INTERRUPÇÃO DA REPLICAÇÃO DE 5.7 COM ERRO DE SINTAXE COM GERENCIAMENTO DE CONCESSÕES

- Erro nº 29915479: EXECUÇÃO COM_REGISTER_SLAVE SEM COM_BINLOG_DUMP PODE RESULTAR EM SAÍDA DO SERVIDOR
- Erro nº 30569003: INTERRUPÇÃO DA REPLICAÇÃO DE 5.7 COM ERRO DE SINTAXE COM GERENCIAMENTO DE CONCESSÕES
- Erro nº 29915479: EXECUÇÃO COM_REGISTER_SLAVE SEM COM_BINLOG_DUMP PODE RESULTAR EM SAÍDA DO SERVIDOR
- Erro nº 20712046: A EXIBIÇÃO DAS TABELAS PROCESSLIST E PERFORMANCE_SCHEMA NÃO OCULTA A SENHA NA CONSULTA
- Erro nº 18898433: PERFORMANCE EXTREMAMENTE LENTA COM JUNÇÕES EXTERNAS E BUFFER DE JUNÇÃO (corrigido na versão 5.7.21)
- Erro nº 26402045: MYSQLD TRAVA DURANTE A CONSULTA (corrigido no MySQL 5.7.23)
- Erro nº 23103937: PS_TRUNCATE_ALL_TABLES() NÃO FUNCIONA NO MODO SUPER_READ_ONLY
- Erro nº 26666274: LOOP INFINITO NO CONTÊINER DE BUFFER DO ESQUEMA DE PERFORMANCE
- Erro nº 26997096: o valor relay_log_space não é atualizado de forma sincronizada, de modo que seu valor é às vezes muito maior do que o espaço em disco real usado pelos logs de retransmissão. (<https://github.com/mysql/mysql-server/commit/78f25d2809ad457e81f90342239c9bc32a36cdfa>)
- Erro nº 25082593: A VALIDAÇÃO DE CHAVE EXTERNA NÃO PRECISA ADQUIRIR BLOQUEIO DE LACUNA EM LEITURA CONFIRMADA
- Erro nº 24764800: REPLICAÇÃO FALHA NO SUBORDINADO COM XAER_RMFAIL ERROR.
- Erro nº 81441: AVISO SOBRE O LOCALHOST AO USAR SKIP-NAME-RESOLVE.

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas esses recursos não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7)

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.

- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para funções AWS Lambda chamadas de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL CREATE TABLESPACE

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2022-01-06 (versão 2.08.4) (obsoleta)

Versão: 2.08.4

O Aurora MySQL 2.08.4 está geralmente disponível. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Você pode atualizar um cluster de banco de dados existente do Aurora MySQL 2.* para o Aurora MySQL 2.10.0. Para clusters que executam o Aurora MySQL versão 1, você pode atualizar um cluster existente do Aurora MySQL 1.23 ou superior diretamente para 2.10.0. Também é possível restaurar um snapshot de qualquer versão atualmente compatível do Aurora MySQL para o Aurora MySQL 2.10.0.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo pelo Console de gerenciamento da AWS, pela AWS CLI ou pela API do Amazon RDS.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correções de segurança e melhorias gerais:

- Correção de problemas de segurança relacionados à integração do Aurora MySQL com outros serviços da AWS, como o Simple Storage Service (Amazon S3), o Amazon ML e o AWS Lambda.

- Correção de um problema em que o valor de uma coluna `TIMESTAMP` de uma linha existente era atualizado para o timestamp mais recente quando todas as seguintes condições eram atendidas: 1. existe um acionador para a tabela; 2. é feito um `INSERT` na tabela que tem uma cláusula `ON DUPLICATE KEY UPDATE`; 3. a linha inserida pode causar uma violação de valor duplicado em um índice `UNIQUE` ou `PRIMARY KEY`; e 4. uma ou mais colunas são do tipo de dados `TIMESTAMP` e têm um valor padrão de `CURRENT_TIMESTAMP`.
- Corrigido um problema que, em raras condições, faz com que uma instância de gravador seja reiniciada quando uma verificação de integridade de dados na memória falha.
- Correção de um problema com o `Parallel Query` que poderia fazer com que o banco de dados fosse reiniciado ao executar instruções SQL com uma cláusula `LIMIT`.

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas esses recursos não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7)

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para funções AWS Lambda chamadas de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de

indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL `CREATE TABLESPACE`

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-11-12 (versão 2.08.3) (obsoleta)

Versão: 2.08.3

O Aurora MySQL 2.08.3 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Você pode atualizar clusters de banco de dados Aurora MySQL 2.* existentes diretamente para Aurora MySQL 2.08.3. Você pode atualizar um cluster Aurora MySQL 1.* existente diretamente para 2.07.3 ou superior e, em seguida, atualizar diretamente para 2.08.3.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo pelo Console de gerenciamento da AWS, pela AWS CLI ou pela API do RDS.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

 Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correções de segurança:

Correções e outras melhorias para ajustar o tratamento em um ambiente gerenciado. Correções adicionais do CVE abaixo:

- [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](#)

Alterações incompatíveis:

Esta versão introduz uma alteração de permissão que afeta o comportamento do `mysqldump` comando. Os usuários devem ter o `PROCESS` privilégio para acessar a `INFORMATION_SCHEMA.FILES` tabela. Para executar o `mysqldump` comando sem alterações,

conceda o `PROCESS` privilégio ao usuário do banco de dados ao qual o `mysqldump` comando se conecta. Você também pode executar o comando `mysqldump` com a opção `--no-tablespaces`. Com essa opção, a saída `mysqldump` não inclui nenhuma `CREATE LOGFILE GROUP` ou instruções do `CREATE TABLESPACE`. Nesse caso, o comando `mysqldump` não acessa a tabela `INFORMATION_SCHEMA.FILES` e você não precisa conceder a permissão `PROCESS`.

Integração de correções de bug da edição MySQL community

- Erro #23762382 - INSERIR CONSULTA DE VALORES COM JUNÇÃO EM UMA SELEÇÃO QUE CAUSA COMPORTAMENTO INCORRETO.
- Erro #25209512 - `CURRENT_TIMESTAMP` PRODUZ ZEROS NO GATILHO.

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas esses recursos não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7)

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para funções AWS Lambda chamadas de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de

indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL `CREATE TABLESPACE`

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-08-28 (versão 2.08.2) (obsoleta)

Versão: 2.08.2

O Aurora MySQL 2.08.2 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

É possível restaurar um snapshot de uma versão atualmente compatível do Aurora MySQL para o Aurora MySQL 2.08.2. Você tem a opção de atualizar os clusters de banco de dados Aurora MySQL 2.* existentes para o Aurora MySQL 2.08.2. Não é possível atualizar um cluster do Aurora MySQL 1.* existente diretamente para o 2.08.2. No entanto, é possível restaurar seu snapshot no Aurora MySQL 2.08.2. Para obter mais informações sobre a restauração de snapshots, consulte [Restauração de um snapshot de um cluster de banco de dados](#) no Guia do usuário do Amazon Aurora.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo pelo Console de gerenciamento da AWS, pela AWS CLI ou pela API do RDS.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correções críticas:

- Correção de um problema que poderia causar uma paralisação não planejada e afetar a disponibilidade do banco de dados.

Correções de disponibilidade:

- Correção de um problema em que o banco de dados Aurora MySQL poderia ser reiniciado se fosse uma réplica binlog e replicasse um evento DDL sobre a tabela `time_zone mysql`.

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas esses recursos não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7)

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.

- Funções nativas para funções AWS Lambda chamadas de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL CREATE TABLESPACE

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-06-18 (versão 2.08.1) (obsoleta)

Versão: 2.08.1

O Aurora MySQL 2.08.1 está disponível para o público em geral. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

É possível restaurar um snapshot de uma versão atualmente compatível do Aurora MySQL para o Aurora MySQL 2.08.1. Você tem a opção de atualizar os clusters de banco de dados Aurora MySQL 2.* existentes para o Aurora MySQL 2.08.1. Não é possível atualizar um cluster do Aurora MySQL 1.* existente diretamente para o 2.08.1. No entanto, é possível restaurar seu snapshot no Aurora MySQL 2.08.1.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo pelo Console de gerenciamento da AWS, pela AWS CLI ou pela API do RDS.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Novos atributos:

- Encaminhamento de gravação do banco de dados global. Em um banco de dados global Aurora, agora é possível executar determinadas operações de gravação, como instruções DML, enquanto está conectado a um cluster secundário. As operações de gravação são encaminhadas para o cluster primário e qualquer alteração é replicada de volta para os clusters secundários. Para obter mais informações, consulte [Como usar o encaminhamento de gravação em um banco de dados global Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Correções de estabilidade geral:

- Corrigido um problema em que a restauração de um cluster do Amazon S3 para o Aurora MySQL versão 2.08.0 demorava mais do que o esperado se o backup do S3 não incluísse a tabela `mysql.host`.

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas esses recursos não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7)

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para funções AWS Lambda chamadas de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plugin de replicação de grupo
- Maior tamanho de página

- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL `CREATE TABLESPACE`

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-06-02 (versão 2.08.0) (obsoleta)

Versão: 2.08.0

O Aurora MySQL 2.08.0 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

É possível restaurar um snapshot de uma versão atualmente compatível do Aurora MySQL para o Aurora MySQL 2.08.0. Você tem a opção de atualizar os clusters de banco de dados Aurora MySQL 2.* existentes para o Aurora MySQL 2.08.0. Não é possível atualizar um cluster do Aurora MySQL 1.* existente diretamente para o 2.08.0. No entanto, é possível restaurar seu snapshot no Aurora MySQL 2.08.0.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo pelo Console de gerenciamento da AWS, pela AWS CLI ou pela API do RDS.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

 Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Novos atributos:

- Melhoria no processamento de log binário (binlog) para reduzir o tempo de recuperação de falhas e a latência de tempo de confirmação quando transações muito grandes estão envolvidas.
- Execução do recurso Stream de atividades do banco de dados (DAS) para o Aurora MySQL. Esse recurso fornece um fluxo de dados quase em tempo real da atividade do banco de dados do seu banco de dados relacional para ajudar você a monitorar a atividade. Para obter mais informações, consulte [Monitorar o Amazon Aurora com o recurso Database Activity Streams](#) no Guia do usuário do Amazon Aurora.
- Atualização de arquivos de fuso horário para oferecer suporte à alteração de fuso horário mais recente do Brasil.
- Apresentação de novas palavras-chave no SQL para exercer a funcionalidade de junção de hash para uma tabela específica e/ou tabela interna: HASH_JOIN, HASH_JOIN_PROBING e HASH_JOIN_BUILDING. Para obter mais detalhes, consulte [Dicas do Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Apresentação do suporte de dicas de ordem de junção no Aurora MySQL 5.7, que faz portabilidade retroativa de [um recurso MySQL 8.0](#). As novas dicas são JOIN_FIXED_ORDER, JOIN_ORDER, JOIN_PREFIX e JOIN_SUFFIX. Para obter documentação detalhada do suporte de dicas de ordem de junção, consulte [WL#9158: Join Order Hints](#).
- O Aurora Machine Learning agora oferece suporte a funções definidas pelo usuário com MEDIUMINT como o tipo de retorno.
- O procedimento `lambda_async()` armazenado agora oferece suporte a todos os caracteres utf8 do MySQL.

Correções de alta prioridade:

- Correção de um problema que poderia fazer com que uma instância de banco de dados do leitor retornasse resultados incompletos para uma consulta FTS depois que a tabela `INFORMATION_SCHEMA.INNODB_SYS_TABLES` é consultada na instância de banco de dados do gravador.
- [CVE-2019-5443](#)
- [CVE-2019-3822](#)

Melhorias de disponibilidade:

- Correção de um problema que resultava na reinicialização do banco de dados após uma instrução de várias consultas que acessa várias tabelas ou bancos de dados ser executada com o cache de consulta habilitado.
- Correção de uma condição de corrida no gerenciador de bloqueio que resultava em uma reinicialização ou um failover do banco de dados durante a reversão da transação.
- Correção de um problema que acionava a reinicialização ou o failover do banco de dados quando várias conexões estavam tentando atualizar a mesma tabela com um índice de Pesquisa de texto completo.
- Correção de um problema que poderia acionar uma reinicialização ou um failover do banco de dados durante um comando `kill session`. Se encontrar esse problema, entre em contato com o suporte da AWS para habilitar essa correção em sua instância.
- Correção de um problema que fazia com que a instância de banco de dados do leitor reiniciasse durante uma transação com várias instruções `SELECT` e uma workload de gravação intensiva na instância de banco de dados do gravador com `AUTOCOMMIT` habilitado.
- Correção de um problema que fazia com que a instância de banco de dados do leitor reiniciasse após a execução de consultas de longa duração enquanto a instância de banco de dados de gravação estava sob uma workload OLTP de gravação intensiva.

Melhorias gerais:

- Melhoria no tempo de recuperação de banco de dados e na latência de confirmação para transações de longa execução quando o log binário está habilitado.
- Melhoria do algoritmo para gerar melhores estatísticas para estimar contagens de valores distintas em colunas indexadas, incluindo colunas com distribuições de dados distorcidos.
- Redução do tempo de resposta e da utilização da CPU de consultas de junção que acessam tabelas temporárias MyISAM e os resultados são despejados no armazenamento local.

- Correção de um problema que impedia que os snapshots do Aurora MySQL 5.6 com nomes de banco de dados ou tabelas contendo espaços fossem restaurados para um novo cluster do Aurora MySQL 5.7.
- Inclusão de informações de transação da vítima quando o deadlock é resolvido no `show engine innodb status`.
- Correção de um problema que fazia com que as conexões ficassem presas quando clientes de várias versões diferentes estavam conectados ao mesmo banco de dados e acessavam o cache de consulta.
- Correção de um vazamento de memória resultante de várias invocações do fluxo de trabalho de patches com tempo de inatividade zero (ZDP) e de reinicialização com tempo de inatividade zero (ZDR) durante toda a vida útil de uma instância de banco de dados.
- Correção de uma mensagem de erro nas operações de patches com tempo de inatividade zero (ZDP) e de reinicialização com tempo de inatividade zero (ZDR) que informa de forma errada que a última transação foi abortada se o sinalizador de confirmação automática estiver desativado.
- Correção de um problema nas operações de patches com tempo de inatividade zero (ZDP) que poderia levar a uma mensagem de erro de falha do servidor ao restaurar variáveis de sessão do usuário no novo processo de banco de dados.
- Corrigido um problema nas operações ZDP (Zero Inatividade Patch) que podia causar falhas intermitentes no banco de dados quando há consultas de execução longa durante a aplicação de patches.
- Correção de um problema em que consultas, incluindo uma função do Aurora Machine Learning, retornavam mensagens de erro vazias devido a uma resposta de erro manipulada incorretamente dos serviços de machine learning, como o Amazon SageMaker AI e o Amazon Comprehend.
- Correção de um problema na funcionalidade de monitoramento de falta de memória que não respeitava um valor personalizado do parâmetro `table_definition_cache`.
- A mensagem de erro “A execução da consulta foi interrompida” é retornada se uma consulta do Aurora Machine Learning for interrompida. Anteriormente, a mensagem genérica “Erro interno no processamento de solicitação de ML” era retornada.
- Correção de um problema que poderia fazer com que um operador de log binário tivesse um tempo limite de conexão quando o parâmetro `slave_net_timeout` fosse menor que o parâmetro `aurora_binlog_replication_max_yield_seconds` e houvesse baixa workload no cluster mestre de log binário.
- Melhoria no monitoramento do progresso de recuperação de log binário com envio de mensagens informativas no log de erros com uma frequência de uma mensagem por minuto.

- Correção de um problema que poderia fazer com que transações ativas não fossem relatadas pela consulta `SHOW ENGINE INNODB STATUS`.

Integração de correções de bug da edição MySQL community

- [Bug #25289359](#): um bloqueio de cache de texto completo obtido quando os dados foram sincronizados não foi liberado se o tamanho de cache de texto completo excedeu o limite respectivo.
- [Bug #29138644](#): alterar manualmente a hora do sistema enquanto o servidor MySQL estava sendo executado causou atrasos no thread do agente de limpeza de página.
- [Bug #25222337](#): um nome de campo de coluna virtual NULL em um índice virtual causou uma saída do servidor durante uma comparação de nome de campo que ocorre ao preencher colunas virtuais afetadas por uma restrição de chave externa.
- [Bug #25053286](#): executar um procedimento armazenado que contém uma consulta que acessou uma visualização pode alocar memória que não foi liberada até o término da sessão.
- [Bug #25586773](#): executar um procedimento armazenado que contém uma instrução que criou uma tabela com base no conteúdo de determinadas instruções `SELECT` pode resultar em um vazamento de memória.
- [Bug #28834208](#): durante o aplicativo de log, após uma operação `OPTIMIZE TABLE`, o InnoDB não preencheu as colunas virtuais antes de verificar se há atualizações de índice de coluna virtual.
- [Bug n.º 26666274](#): loop infinito no contêiner de buffer de esquema de performance devido a excesso de inteiros não assinados de 32 bits.

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas esses recursos não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7)

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.

- Funções nativas para funções AWS Lambda chamadas de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL CREATE TABLESPACE

Atualizações do mecanismo de banco de dados Aurora MySQL 2023-08-15 (versão 2.07.10) (Obsoleto)

Versão: 2.07.10

O Aurora MySQL 2.07.10 está disponível para o público. As versões do Aurora MySQL 2.07 são compatíveis com o MySQL 5.7.12. Para obter mais informações sobre alterações da comunidade, consulte [Changes in MySQL 5.7.12 \(2016-04-11, General Availability\)](#).

As versões atualmente compatíveis do Aurora MySQL são 2.07.*, 2.11.*, 3.01.*, 3.02.*, 3.03.* e 3.04.*.

É possível restaurar um snapshot de uma versão atualmente compatível do Aurora MySQL para o Aurora MySQL 2.07.10. Você também tem a opção de atualizar os clusters de bancos de dados do Aurora MySQL 2.* existentes para o Aurora MySQL 2.07.10. A atualização no local está disponível para clusters do Aurora MySQL 1.* para o Aurora MySQL 2.* (consulte [Realizar a atualização da versão principal de um cluster de bancos de dados do Amazon Aurora MySQL](#)). Ela também está disponível para clusters do Aurora MySQL 2.* para o Aurora MySQL 3.* (consulte [Fazer upgrade do Aurora MySQL 2.x para 3.x](#)).

Imediatamente após o upgrade da versão do mecanismo no local para o Aurora MySQL 2.07.10, um upgrade do sistema operacional é aplicado automaticamente a todas as instâncias afetadas nas classes de instância de banco de dados db.r4, db.r5, db.t2 e db.t3, caso as instâncias estejam executando uma versão antiga do sistema operacional. As instâncias do leitor de um cluster de banco de dados multi-AZ aplicam a atualização do sistema operacional primeiro. Quando a atualização do sistema operacional na primeira instância do leitor é concluída, ocorre um failover e o upgrade da instância anterior do gravador.

Note

O upgrade do sistema operacional não é aplicado automaticamente aos bancos de dados globais do Aurora durante os upgrades de versões principais.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Problemas de segurança corrigidos e CVEs listados abaixo:

Correções e outras melhorias para ajustar o tratamento em um ambiente gerenciado. Correções adicionais do CVE abaixo:

- [CVE-2023-21963](#)
- [CVE-2023-21912](#)
- [CVE-2023-0215](#)
- [CVE-2022-43551](#)
- [CVE-2022-37434](#)
- Corrigido um problema em que os eventos relatados durante o processamento das rotações de log de auditoria podiam não ser gravados no log de auditoria.
- [As cifras SSL padrão usadas pelo Aurora MySQL foram atualizadas para excluir os valores DES CBC3 - -SHA menos seguros do parâmetro do banco de dados SSL_CIPHER.](#) Se você encontrar problemas de conexão SSL devido à remoção da cifra DES- CBC3 -SHA, use uma cifra segura aplicável a partir das seguintes informações, [Configurando conjuntos de criptografia para conexões com clusters de banco de dados Aurora MySQL.](#)
- O OpenSSL foi atualizado para a versão 1.0.2zh.

Melhorias gerais:

- Adicionado suporte para cifras SSL ECDHE-RSA que usam tamanhos de chave menores para criptografia.
- Corrigido um problema de gerenciamento de memória ao executar consultas com junções de hash.

Recursos não compatíveis com o Aurora MySQL versão 2

No momento, os recursos a seguir não são compatíveis com o Aurora MySQL versão 2 (compatível com MySQL 5.7).

- Pré-busca de chave assíncrona (AKP).
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\).](#)

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- A declaração SQL CREATE TABLESPACE
- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- Protocolo X

Atualizações do mecanismo de banco de dados Aurora MySQL 2023-05-04 (versão 2.07.9) (Obsoleto)

Versão: 2.07.9

O Aurora MySQL 2.07.9 está disponível para o público. As versões do Aurora MySQL 2.07 são compatíveis com o MySQL 5.7.12. Para obter mais informações sobre alterações da comunidade, consulte [Changes in MySQL 5.7.12 \(2016-04-11, General Availability\)](#).

As versões atualmente compatíveis do Aurora MySQL são 2.07.*, 2.11.*, 3.01.*, 3.02.* e 3.03.*.

É possível restaurar um snapshot de uma versão atualmente compatível do Aurora MySQL para o Aurora MySQL 2.07.9. Você também tem a opção de atualizar os clusters de bancos de dados do

Aurora MySQL 2.* existentes para o Aurora MySQL 2.07.9. A atualização no local está disponível para clusters do Aurora MySQL 1.* para o Aurora MySQL 2.* (consulte [Realizar a atualização da versão principal de um cluster de bancos de dados do Amazon Aurora MySQL](#)). Ela também está disponível para clusters do Aurora MySQL 2.* para o Aurora MySQL 3.* (consulte [Fazer upgrade do Aurora MySQL 2.x para 3.x](#)).

Imediatamente após o upgrade da versão do mecanismo no local para o Aurora MySQL 2.07.9, um upgrade do sistema operacional é aplicado automaticamente a todas as instâncias afetadas nas classes de instância de banco de dados db.r4, db.r5, db.t2 e db.t3, caso as instâncias estejam executando uma versão antiga do sistema operacional. As instâncias do leitor de um cluster de banco de dados multi-AZ aplicam a atualização do sistema operacional primeiro. Quando a atualização do sistema operacional na primeira instância do leitor é concluída, ocorre um failover e o upgrade da instância anterior do gravador.

Note

O upgrade do sistema operacional não é aplicado automaticamente aos bancos de dados globais do Aurora durante os upgrades de versões principais.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Problemas de segurança corrigidos e CVEs listados abaixo:

Correções e outras melhorias para ajustar o tratamento em um ambiente gerenciado. Correções adicionais do CVE abaixo:

- [CVE-2022-32221](#)

Melhorias de disponibilidade:

- Corrigido um problema em que a rotação do log de auditoria avançada pode reduzir a memória liberável, o que poderia levar à reinicialização da instância de banco de dados.
- Corrigido um problema que pode ocorrer durante a reinicialização do banco de dados e que faz com que o banco de dados não seja inicializado com êxito por um longo período.

Melhorias gerais:

- Corrigido um problema que, em raras condições, poderia fazer com que as instâncias fossem reiniciadas quando o volume do banco de dados aumentasse para um múltiplo de 160 GB.

Recursos não compatíveis com o Aurora MySQL versão 2

No momento, os recursos a seguir não são compatíveis com o Aurora MySQL versão 2 (compatível com MySQL 5.7).

- Pré-busca de chave assíncrona (AKP).
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha

- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL `CREATE TABLESPACE`

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2022-06-16 (versão 2.07.8) (obsoleta)

Versão: 2.07.8

O Aurora MySQL 2.07.8 está disponível para o público. As versões 2.* do Aurora MySQL são compatíveis com o MySQL 5.7 e as versões 1.* do Aurora MySQL são compatíveis com o MySQL 5.6.

Note

Esta versão é designada como uma versão de suporte de longo prazo (LTS). Para obter mais informações, consulte [Versões de suporte de longo prazo \(LTS\) do Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

É possível restaurar um snapshot de uma versão atualmente compatível do Aurora MySQL para o Aurora MySQL 2.07.8. Você também tem a opção de atualizar os clusters de bancos de dados do Aurora MySQL 2.* existentes para o Aurora MySQL 2.07.8. A atualização no local está disponível para clusters do Aurora MySQL 1.* para o Aurora MySQL 2.* (consulte [Realizar a atualização da versão principal de um cluster de bancos de dados do Amazon Aurora MySQL](#)). Ela também está disponível para clusters do Aurora MySQL 2.* para o Aurora MySQL 3.* (consulte [Fazer upgrade do Aurora MySQL 2.x para 3.x](#)).

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo pelo Console de gerenciamento da AWS, pela AWS CLI ou pela API do RDS.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

 Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correções de segurança:

Correções e outras melhorias para ajustar o tratamento em um ambiente gerenciado. Correções adicionais do CVE abaixo:

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

Melhorias gerais:

- Corrigido um problema que, em raras ocasiões, faz com que o servidor de banco de dados seja reiniciado quando o thread do detector de impasse trava devido a uma condição de corrida.

Integração de correções de bug da edição MySQL community

- Quando UPDATE exigia uma tabela temporária com uma chave primária maior que 1024 bytes e essa tabela era criada usando o InnoDB, o servidor poderia sair. (Bug nº 25153670)

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas esses recursos não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7)

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.

- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para funções AWS Lambda chamadas de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL CREATE TABLESPACE

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-11-24 (versão 2.07.7) (obsoleta)

Versão: 2.07.7

O Aurora MySQL 2.07.7 está disponível para o público. As versões 2.* do Aurora MySQL são compatíveis com o MySQL 5.7 e as versões 1.* do Aurora MySQL são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

É possível restaurar um snapshot de uma versão atualmente compatível do Aurora MySQL para o Aurora MySQL 2.07.7. Você também tem a opção de atualizar os clusters de bancos de dados do Aurora MySQL 2.* existentes para o Aurora MySQL 2.07.7. Não é possível atualizar um cluster existente do Aurora MySQL 1.* diretamente para a versão 2.07.7. No entanto, é possível restaurar o seu snapshot para o Aurora MySQL 2.07.7.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo pelo Console de gerenciamento da AWS, pela AWS CLI ou pela API do RDS.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correções de segurança:

Correções e outras melhorias para ajustar o tratamento em um ambiente gerenciado. Correções adicionais do CVE abaixo:

- [CVE-2019-17543](#)

- [CVE-2019-2960](#)

Melhorias gerais:

- Correção de problemas de segurança relacionados à integração do Aurora MySQL com outros serviços da AWS, como o Amazon S3, o Amazon ML e o Lambda.
- Correção de um problema com a geração de relatórios incorretos de um atraso de replicação do Aurora.
- Correção de um problema que podia fazer com que a reinicialização da instância do banco de dados falhasse se o banco de dados tivesse muitas combinações de usuários e privilégios.
- Correção de um problema que pode fazer com que tabelas `general_log` e `slow_log` fiquem inacessíveis após o upgrade da versão principal no local do Aurora MySQL 1.x (com base no MySQL 5.6) para o Aurora MySQL 2.x (baseado no MySQL 5.7).
- Correção de um problema que, em raras condições, poderia fazer com que uma instância de leitor fosse reiniciada devido a um processamento de verificação incorreto.
- Correção de um problema que em condições raras mostrava o gráfico “Database Load” (Carga do banco de dados) em sessões do Performance Insights (PI) utilizando a CPU ativamente, mesmo que as sessões tivessem encerrado o processamento e estivessem ociosas.
- Correção de um problema com consulta paralela que poderia fazer com que o banco de dados fosse reiniciado ao executar instruções SQL com uma cláusula `LIMIT`.
- Corrigido um problema em que o valor de uma coluna `TIMESTAMP` de uma linha existente era atualizado para o timestamp mais recente quando todas as seguintes condições eram atendidas: 1. Existe um acionador para a tabela; 2. é feito um `INSERT` na tabela que tem uma cláusula `ON DUPLICATE KEY UPDATE`; 3. a linha inserida pode causar uma violação de valor duplicado em um índice `UNIQUE` ou `PRIMARY KEY`; e 4. uma ou mais colunas são do tipo de dados `TIMESTAMP` e têm um valor padrão de `CURRENT_TIMESTAMP`.
- Corrigido um problema que, em raras condições, poderia fazer com que a instância do banco de dados fosse reiniciada ao utilizar transações XA no nível de isolamento `READ COMMITTED`.

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas esses recursos não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7)

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para funções AWS Lambda chamadas de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL CREATE TABLESPACE

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-09-02 (versão 2.07.6) (obsoleta)

Versão: 2.07.6

A versão 2.07.6 do Aurora MySQL está disponível para o público. As versões 2.* do Aurora MySQL são compatíveis com o MySQL 5.7 e as versões 1.* do Aurora MySQL são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

É possível restaurar um snapshot de uma versão atualmente compatível do Aurora MySQL para a versão 2.07.6 do Aurora MySQL. Você também pode atualizar os clusters de banco de dados existentes da versão 2.* do Aurora MySQL para a 2.07.6. Não é possível atualizar um cluster existente Aurora MySQL 1.* diretamente para a versão 2.07.6. No entanto, é possível restaurar o seu snapshot para a versão 2.07.6 do Aurora MySQL.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo pelo Console de gerenciamento da AWS, pela AWS CLI ou pela API do RDS.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Integração de correções de bug da edição MySQL community

- INSERIR REGISTROS COM 64K É MUITO DEMORADO. ([Bug n.º#23031146](#))

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas esses recursos não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7)

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.

- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para funções AWS Lambda chamadas de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL CREATE TABLESPACE

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-07-06 (versão 2.07.5) (obsoleta)

Versão: 2.07.5

O Aurora MySQL 2.07.5 está disponível para o público. As versões 2.* do Aurora MySQL são compatíveis com o MySQL 5.7 e as versões 1.* do Aurora MySQL são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

É possível restaurar um snapshot de uma versão atualmente compatível do Aurora MySQL para o Aurora MySQL 2.07.5. Você tem a opção de atualizar os clusters de banco de dados Aurora MySQL 2.* existentes para o Aurora MySQL 2.07.5. Não é possível atualizar um cluster Aurora MySQL 1.* existente diretamente para o 2.07.5. No entanto, é possível restaurar seu snapshot no Aurora MySQL 2.07.5.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo pelo Console de gerenciamento da AWS, pela AWS CLI ou pela API do RDS.

Note

Esta versão é designada como uma versão de suporte de longo prazo (LTS). Para obter mais informações, consulte [Versões de suporte de longo prazo \(LTS\) do Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Melhorias de disponibilidade:

- Correção de um problema em que não eram permitidos bloqueios no nível de usuário em uma réplica do Aurora

- Correção de um problema que poderia causar reinicialização de um banco de dados ao usar transações XA no nível de isolamento READ COMMITTED.
- Comprimento máximo permitido estendido para 2000 para os parâmetros globais `server_audit_incl_users` e `server_audit_excl_users`.

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas esses recursos não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7)

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para funções AWS Lambda chamadas de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plugin de replicação de grupo

- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL `CREATE TABLESPACE`

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-03-04 (versão 2.07.4) (obsoleta)

Versão: 2.07.4

O Aurora MySQL 2.07.4 está disponível para o público. As versões 2.* do Aurora MySQL são compatíveis com o MySQL 5.7 e as versões 1.* do Aurora MySQL são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

É possível restaurar um snapshot de uma versão atualmente compatível do Aurora MySQL na versão 2.07.4 do Aurora MySQL. Também existe a opção de atualizar os clusters de banco de dados existentes Aurora MySQL 2.* para o Aurora MySQL 2.07.4. Não é possível atualizar um cluster existente Aurora MySQL 1.* diretamente para o 2.07.4. No entanto, é possível restaurar seu snapshot no Aurora MySQL 2.07.4.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo pelo Console de gerenciamento da AWS, pela AWS CLI ou pela API do RDS.

Note

Esta versão é designada como uma versão de suporte de longo prazo (LTS). Para obter mais informações, consulte [Versões de suporte de longo prazo \(LTS\) do Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correções de segurança:

- [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](#)

Melhorias de disponibilidade:

- Correção de um problema que poderia fazer um cliente travar em caso de erro de rede durante a leitura ou gravação de um pacote de rede.
- Melhora dos tempos de reinicialização do mecanismo em alguns casos após DDL ser interrompido.
- Correção de um problema em que um DDL ou DML poderia causar reinicialização do mecanismo durante uma solicitação de pré-busca de página.
- Correção de um problema em que uma réplica podia ser reiniciada durante a realização de uma verificação reversa de uma tabela/índice em uma réplica de leitura do Aurora.
- Correção de um problema na operação de cluster de clonagem que poderia fazer com que o clone demorasse mais tempo.
- Correção de um problema que poderia causar uma reinicialização de um banco de dados ao usar otimização de consulta paralela para colunas geoespaciais.
- Correção de um problema que fazia com que uma réplica de log binário parasse com um erro HA_ERR_KEY_NOT_FOUND.

Integração de correções de bug da edição MySQL community

- Correção de um problema no analisador de ngram de texto completo ao lidar com tokens contendo ' ' (espaço), '%', ou ','. Os clientes devem reconstruir seus índices FTS se estiverem usando o analisador ngram. (Bug n.º 25873310)
- Correção de um problema que poderia causar a reinicialização do mecanismo durante a execução da consulta com visualizações SQL aninhadas. (Bug n.º 27214153, bug n.º 26864199)

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas esses recursos não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7)

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para funções AWS Lambda chamadas de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL `CREATE TABLESPACE`

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-11-10 (versão 2.07.3) (obsoleta)

Versão: 2.07.3

O Aurora MySQL 2.07.3 está disponível para o público. As versões 2.* do Aurora MySQL são compatíveis com o MySQL 5.7 e as versões 1.* do Aurora MySQL são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

É possível restaurar um snapshot de uma versão atualmente compatível do Aurora MySQL para a versão Aurora MySQL 2.07.3. Também existe a opção de atualizar os clusters de banco de dados existentes do Aurora MySQL 2.* para o Aurora MySQL 2.07.3. Não é possível atualizar um cluster existente do Aurora MySQL 1.* diretamente para 2.07.3. No entanto, é possível restaurar seu snapshot para o Aurora MySQL 2.07.3.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo pelo Console de gerenciamento da AWS, pela AWS CLI ou pela API do RDS.

 Note

Esta versão é designada como uma versão de suporte de longo prazo (LTS). Para obter mais informações, consulte [Versões de suporte de longo prazo \(LTS\) do Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correções de segurança:

Correções e outras melhorias para ajustar o tratamento em um ambiente gerenciado.

- [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](#)

Alterações incompatíveis:

Esta versão introduz uma alteração de permissão que afeta o comportamento do `mysql_dump` comando. Os usuários devem ter o `PROCESS` privilégio para acessar a

INFORMATION_SCHEMA.FILES tabela. Para executar o mysqldump comando sem alterações, conceda o PROCESS privilégio ao usuário do banco de dados ao qual o mysqldump comando se conecta. Você também pode executar o comando mysqldump com a opção --no-tablespaces. Com essa opção, a saída mysqldump não inclui nenhuma CREATE LOGFILE GROUP ou instruções do CREATE TABLESPACE. Nesse caso, o comando mysqldump não acessa a tabela INFORMATION_SCHEMA.FILES e você não precisa conceder a permissão PROCESS.

Melhorias de disponibilidade:

- Corrigida uma condição de corrida no gerenciador de bloqueio entre a queda de uma conexão/consulta e o término da sessão, o que gera em uma reinicialização do banco de dados.
- Corrigido um problema que resultava na reinicialização do banco de dados após uma instrução de várias consultas que acessa várias tabelas ou bancos de dados ser executada com o cache de consulta habilitado.
- Corrigido um problema que poderia causar reinicializações repetidas devido a atualizações de colunas virtuais com índices secundários.

Integração de correções de bug da edição MySQL community

- InnoDB: Transações XA simultâneas que foram executadas com sucesso para o estágio de preparação XA no mestre em conflito quando repetidas no escravo, o que gera um tempo limite de espera de bloqueio no applier thread. O conflito foi devido ao intervalo de bloqueio GAP que diferiu quando as transações foram repetidas em série no escravo. Para evitar esse tipo de conflito, os bloqueios do GAP realizados pelas transações XA no nível de isolamento [LEITURA CONFIRMADA](#) agora são liberados (e não mais herdados) quando as transações XA atingem o estágio de preparação. (Erro nº 27189701, erro nº 25866046)
- InnoDB: Um bloqueio de gap foi realizado desnecessariamente durante a validação de chave estrangeira ao usar o nível de isolamento [LEITURA CONFIRMADA](#). (Erro nº 25082593)
- Replicação: Ao usar transações XA, se um tempo limite de espera de bloqueio ou um deadlock ocorreu para o thread do aplicador (SQL) em um escravo de replicação, a repetição automática não funcionou. A causa era que, mesmo que o thread SQL fizesse uma reversão, ele não reverteria a transação XA de volta. Isso significava que, quando a transação foi repetida, o primeiro evento era XA START, que era inválido, pois a transação XA já estava em andamento, levando a um erro XAER_RMFAIL. (Erro nº 24764800)
- Replicação: As transações intercaladas poderiam, por vezes, impedir o aplicador escravo quando o nível de isolamento da transação foi definido como [LEITURA REPETIDA](#). (Erro nº 25040331)

- Replicação: O valor retornado por uma instrução [MOSTRAR STATUS ESCRAVO](#) para o tamanho total combinado de todos os arquivos de log de retransmissão existentes (Relay_Log_Space) pode se tornar muito maior do que o espaço em disco real usado pelos arquivos de log de retransmissão. O thread de E/S não bloqueou a variável enquanto atualizava o valor, de modo que o thread SQL poderia excluir automaticamente um arquivo de log de retransmissão e gravar um valor reduzido antes que o thread de E/S terminasse a atualização do valor. O thread de E/S gravou seu cálculo de tamanho original, ignorando a atualização do thread SQL e, portanto, adicionando de volta o espaço para o arquivo excluído. O valor Relay_Log_Space agora está bloqueado durante as atualizações para evitar atualizações simultâneas e garantir um cálculo preciso. (Erro nº 26997096, erro nº 87832)
- Para uma instrução [INSERIR](#) para a qual a lista VALORES produziu valores para a segunda linha ou linha posterior com uma subconsulta contendo uma junção, o servidor pode sair depois de não resolver os privilégios necessários. (Erro nº 23762382)
- Para uma tabela com uma coluna [TIMESTAMP](#) ou [DATETIME](#) cujo valor padrão é [CURRENT_TIMESTAMP](#), a coluna podia ser inicializada em 0000-00-00 00:00:00 caso a tabela tivesse um acionador BEFORE INSERT. (Erro nº 25209512, erro nº 84077)
- Uma saída do servidor pode gerar tentativas simultâneas de vários threads para registrar e cancelar o registro de objetos do Performance Schema de metadados. (Erro nº 26502135)
- Executar um procedimento armazenado que contém uma instrução que criou uma tabela com base no conteúdo de determinadas instruções [SELECIONAR](#) que pode resultar em um vazamento de memória. (Erro nº 25586773)
- Executar um procedimento armazenado que contém uma consulta que acessou uma visualização poderia alocar memória que não foi liberada até o término da sessão. (Erro nº 25053286)
- Certos casos de materialização da subconsulta poderiam causar uma saída do servidor. Essas consultas agora produzem um erro sugerindo que a materialização seja desabilitada. (Erro nº 26402045)
- As consultas com muitas junções à esquerda ficavam lentas se o buffer de junção fosse usado (por exemplo, usando o algoritmo de loop aninhado em bloco). (Erro nº 18898433, erro nº 72854)
- O otimizador ignorou a segunda coluna em um índice composto ao executar uma junção interna com uma LIKE cláusula contra a segunda coluna. (Erro nº 28086754)

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas esses recursos não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7)

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para funções AWS Lambda chamadas de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens

- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL CREATE TABLESPACE

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-04-17 (versão 2.07.2) (obsoleta)

Versão: 2.07.2

O Aurora MySQL 2.07.2 está disponível para o público. As versões 2.* do Aurora MySQL são compatíveis com o MySQL 5.7 e as versões 1.* do Aurora MySQL são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

É possível restaurar um snapshot de uma versão atualmente compatível do Aurora MySQL para o Aurora MySQL 2.07.2. Você tem a opção de atualizar os clusters de bancos de dados Aurora MySQL 2.* existentes para o Aurora MySQL 2.07.2. Não é possível atualizar um cluster do Aurora MySQL 1.* existente diretamente para o 2.07.2. No entanto, é possível restaurar seu snapshot no Aurora MySQL 2.07.2.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo por meio da Console de gerenciamento da AWS, da ou da API do AWS CLI RDS.

Note

Esta versão é designada como uma versão de suporte de longo prazo (LTS). Para obter mais informações, consulte [Versões de suporte de longo prazo \(LTS\) do Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correções de segurança:

- [CVE-2016-8287](#)
- [CVE-2016-5634](#)

Correções de alta prioridade:

- Corrigido um problema que fazia com que a clonagem demorasse mais tempo em alguns clusters de banco de dados com altas cargas de gravação.
- Corrigido um problema que poderia causar consultas em uma instância de banco de dados de leitor com planos de execução usando índices secundários para retornar dados não confirmados. O problema é limitado a dados afetados por operações de linguagem de manipulação de dados (DML) que modificam as principais colunas de índice primário ou secundário.

Melhorias gerais:

- Corrigido um problema que resultava em uma restauração lenta de um cluster de banco de dados Aurora 1.x contendo índices FTS (Full Text Search) para um cluster de banco de dados Aurora 2.x.
- Corrigido um problema que causava restaurações mais lentas de um snapshot de banco de dados Aurora 1.x contendo tabelas particionadas com caracteres especiais em nomes de tabela para um cluster de banco de dados Aurora 2.x.
- Corrigido um problema que causava erros ao consultar logs de consulta lenta e logs gerais em instâncias de banco de dados de leitor.

Integração de correções de bug da edição MySQL community

- Erro nº 23104498: corrigido um problema no esquema de performance ao relatar a memória total usada. (<https://github.com/mysql/mysql-server/commit/20b6840df5452f47313c6f9a6ca075bfbc00a96b>)
- Erro nº 22551677: corrigido um problema no esquema de performance que poderia fazer com que o mecanismo de banco de dados falhasse ao tentar colocá-lo offline. (<https://github.com/mysql/mysql-server/commit/05e2386eccd32b6b444b900c9f8a87a1d8d531e9>)

- Erro nº 23550835, erro nº 23298025, erro nº 81464: correção de um problema no esquema de performance que causava uma falha no mecanismo de banco de dados devido porque excedia a capacidade de um buffer interno. (<https://github.com/mysql/mysql-server/commit/b4287f93857bf2f99b18fd06f555bbe5b12debfc>)

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas esses recursos não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7)

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para invocar AWS Lambda funções de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plugin de replicação de grupo

- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL `CREATE TABLESPACE`

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-12-23 (versão 2.07.1) (obsoleta)

Versão: 2.07.1

O Aurora MySQL 2.07.1 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

É possível restaurar um snapshot de uma versão atualmente compatível do Aurora MySQL para o Aurora MySQL 2.07.1. Você tem a opção de atualizar os clusters de bancos de dados Aurora MySQL 2.* existentes para o Aurora MySQL 2.07.1. Não é possível atualizar um cluster do Aurora MySQL 1.* existente diretamente para o 2.07.1. No entanto, é possível restaurar seu snapshot no Aurora MySQL 2.07.1.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo por meio da Console de gerenciamento da AWS, da ou da API do AWS CLI RDS.

Note

No momento, essa versão não está disponível nas seguintes AWS regiões: AWS GovCloud (Leste dos EUA) [us-gov-east-1], AWS GovCloud (Oeste dos EUA) [-1], China (Ningxia) [us-

gov-westcn-northwest-1], Ásia-Pacífico (Hong Kong) [ap-east-1] e Oriente Médio (Bahrein) [me-south-1]. Haverá um anúncio separado assim que ela for disponibilizada.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correções de alta prioridade:

- Corrigido um vazamento de memória lento no subsistema de rastreamento e registro em log de banco de dados do Aurora específico que reduz a memória disponível.

Correções de estabilidade geral:

- Corrigido um travamento durante a execução de uma consulta complexa envolvendo junções de várias tabelas e agregação que usam tabelas intermediárias internamente.

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas esses recursos não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7)

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).

- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL `CREATE TABLESPACE`

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-11-25 (versão 2.07.0) (obsoleta)

Versão: 2.07.0

O Aurora MySQL 2.07.0 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

É possível restaurar um snapshot de uma versão atualmente compatível do Aurora MySQL para a versão Aurora MySQL 2.07.0. Você tem a opção de atualizar os clusters de bancos de dados Aurora MySQL 2.* existentes para o Aurora MySQL 2.07.0. Não é possível atualizar um cluster do Aurora MySQL 1.* existente diretamente para o 2.07.0. No entanto, é possível restaurar seu snapshot no Aurora MySQL 2.07.0.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo por meio da Console de gerenciamento da AWS, da ou da API do AWS CLI RDS.

Note

No momento, essa versão não está disponível nas seguintes AWS regiões: AWS GovCloud (Leste dos EUA) [us-gov-east-1], (Oeste dos EUA) [-1], China AWS GovCloud (Ningxia) [us-gov-westcn-northwest-1], Ásia-Pacífico (Hong Kong) [ap-east-1], Oriente Médio (Bahrein) [me-south-1] e América do Sul (São Paulo) [sa-east-1] -east-1]. Haverá um anúncio separado assim que ela for disponibilizada.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Novos recursos:

- Os bancos de dados globais agora permitem adicionar regiões secundárias de réplica somente para leitura para clusters de banco de dados implantados nessas AWS regiões: regiões: Leste dos EUA (Norte da Virgínia) [us-east-1], Leste dos EUA (Ohio) [us-east-2], Oeste dos EUA (Norte da

Califórnia) [us-west-1], Oeste dos EUA (Oregon) [us-west-2], Europa (Irlanda) [eu-west-1] Europa (Londres) [eu-west-2], Europa (Paris) [eu-west-3], Ásia-Pacífico (Tóquio) [ap-northeast-1], Ásia-Pacífico (Seul) [ap-northeast-2], Ásia-Pacífico (Seul) [ap-northeast-2], Ásia-Pacífico (Cingapura) [ap-southeast-1] Ásia-Pacífico (Sydney) [ap-southeast-2], Canadá (Central) [ca-central-1], Europa (Frankfurt) [eu-central-1] e Ásia-Pacífico (Mumbai) [ap-south-1].

- O Amazon Aurora Machine Learning é uma integração altamente otimizada entre o banco de dados Aurora MySQL e os serviços de aprendizado de AWS máquina (ML). O machine learning do Aurora permite que os desenvolvedores adicionem uma variedade de previsões baseadas em ML aos seus aplicativos de banco de dados invocando modelos de ML com a linguagem de programação SQL familiar que já usam para o desenvolvimento de bancos de dados, sem a necessidade de criar integrações personalizadas nem aprender ferramentas separadas. Para obter mais informações, consulte [Usar recursos de machine learning \(ML\) com o Amazon Aurora](#).
- Adicionamos suporte para o nível de isolamento READ COMMITTED ANSI nas réplicas de leitura. Esse nível de isolamento permite que as consultas de longa execução na réplica de leitura sejam executadas sem afetar a alta taxa de transferência das gravações no nó do gravador. Para obter mais informações, consulte [Níveis de isolamento do Aurora MySQL](#).

Correções críticas:

- [CVE-2019-2922](#)
- [CVE-2019-2923](#)
- [CVE-2019-2924](#)
- [CVE-2019-2910](#)

Correções de alta prioridade:

- Corrigimos um problema na recuperação DDL que resultava em um tempo de inatividade prolongado do banco de dados. Os clusters que se tornarem indisponíveis após a execução de uma instrução de descarte de várias tabelas, por exemplo, `DROP TABLE t1, t2, t3`, deverão ser atualizados para esta versão.
- Corrigimos um problema na recuperação DDL que resultava em um tempo de inatividade prolongado do banco de dados. Os clusters que se tornarem indisponíveis após a execução de instruções DDL `INPLACE ALTER TABLE` deverão ser atualizados para esta versão.

Correções de estabilidade geral:

- Corrigimos um problema que gerava dados inconsistentes na tabela `information_schema.replica_host_status`.

Integração de correções de bug da edição MySQL community

- Bug nº 26251621: INCORRECT BEHAVIOR WITH TRIGGER AND GCOL
- Bug nº 22574695: ASSERTION `!TABLE || (!TABLE->READ_SET || BITMAP_IS_SET(TABLE->READ_SET, FIEL
- Bug nº 25966845: INSERT ON DUPLICATE KEY GENERATE A DEADLOCK
- Bug nº 23070734: CONCURRENT TRUNCATE TABLES CAUSE STALL
- Bug nº 26191879: FOREIGN KEY CASCADES USE EXCESSIVE MEMORY
- Bug nº 20989615: INNODB AUTO_INCREMENT PRODUCES SAME VALUE TWICE

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas esses recursos não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7)

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

O Aurora MySQL 2.07.0 é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, o Aurora MySQL 2.07.0 não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plug-in de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL `CREATE TABLESPACE`

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-11-22 (versão 2.06.0) (obsoleta)

Versão: 2.06.0

O Aurora MySQL 2.06.0 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

É possível restaurar um snapshot de uma versão atualmente compatível do Aurora MySQL para a versão Aurora MySQL 2.06.0. Você tem a opção de atualizar os clusters de bancos de dados Aurora MySQL 2.* existentes para o Aurora MySQL 2.06.0. Não é possível atualizar um cluster do Aurora MySQL 1.* existente diretamente para o 2.06.0. No entanto, é possível restaurar seu snapshot no Aurora MySQL 2.06.0.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo por meio da Console de gerenciamento da AWS, da ou da API do AWS CLI RDS.

Note

No momento, essa versão não está disponível nas seguintes AWS regiões: AWS GovCloud (Leste dos EUA) [us-gov-east-1], AWS GovCloud (Oeste dos EUA) [-1], China (Ningxia) [us-gov-westcn-northwest-1], Ásia-Pacífico (Hong Kong) [ap-east-1] e Oriente Médio (Bahrein) [me-south-1]. Haverá um anúncio separado assim que ela for disponibilizada.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Novos recursos:

- Os clusters do Aurora MySQL já oferecem suporte para os tipos de instância db.r5.8xlarge, db.r5.16xlarge e db.r5.24xlarge. Para obter mais informações sobre tipos de instância para clusters do Aurora MySQL, consulte [Classes de instância de banco de dados Aurora](#) no Guia do usuário do Amazon Aurora.
- O recurso de junção de hash já está disponível para o público e não exige o modo de laboratório do Aurora definido como ativado. Esse recurso pode melhorar a performance da consulta quando você precisa unir uma grande quantidade de dados usando um equijoin. Para obter mais informações sobre como usar esse recurso, consulte [Usar a API de dados do Aurora Serverless](#) no Guia do usuário do Amazon Aurora.
- O recurso disputa de linhas dinâmica já está disponível para o público e não exige o modo de laboratório do Aurora definido como ativado. Esse recurso melhora substancialmente as taxas de transferência para workloads com muitas transações disputando linhas na mesma página.
- O Aurora MySQL 2.06 e posterior oferece suporte ao retrocesso de um cluster de banco de dados para um período específico, sem restaurar os dados de um backup. Esse recurso, conhecido como

retrocesso, oferece uma maneira rápida de se recuperar de erros do usuário, como descartar a tabela errada ou excluir a linha errada. O retrocesso é concluído em segundos, mesmo para bancos de dados grandes. Leia [o AWS blog](#) para obter uma visão geral e consulte [Backtracking an Aurora DB](#) Cluster no Guia do usuário do Amazon Aurora para obter mais detalhes.

- O Aurora 2.06 e versões posteriores oferecem suporte a AWS Lambda invocações síncronas por meio da função nativa. `lambda_sync()` A função nativa `lambda_async()` também está disponível e pode ser usada como uma alternativa ao procedimento armazenado existente para a invocação assíncrona do Lambda. Para obter informações sobre como chamar funções do Lambda, consulte [Como invocar uma função do Lambda com uma função nativa do Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Correções críticas:

Nenhum.

Correções de alta prioridade:

Correções de segurança

- [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](#)

Tratamento de conexões

- A disponibilidade do banco de dados foi aprimorada para melhor atender a um aumento nas conexões de clientes durante a execução de uma ou mais DDLs. Ela é tratada com a criação temporária de threads adicionais quando necessário. O sistema solicitará a atualização se o banco de dados não responder após a elevação nas conexões durante o processamento de DDL.

Reinicialização do mecanismo

- Corrigimos um problema de indisponibilidade prolongada na reinicialização do mecanismo. Isso resolve o problema na inicialização do grupo de buffers. Esse problema ocorre raramente, mas pode afetar qualquer versão compatível.
- Corrigido um problema que faz com que um banco de dados configurado como principal do log binário (binlog) seja reiniciado durante a execução de uma workload de gravação intensa.

Correções de estabilidade geral:

- Fizemos melhorias nos casos em que as consultas que acessam dados não armazenados em cache podem ficar mais lentas do que o normal. Incentivamos os clientes que tiverem latência de leitura elevada sem explicação ao acessarem dados não armazenados em cache a fazerem a atualização, pois talvez tenham esse problema.
- Corrigimos um problema de falha ao restaurar tabelas particionadas de um snapshot de banco de dados. Os clientes que encontrarem erros ao acessarem tabelas particionadas em um banco de dados restaurado de um snapshot de um banco de dados do Aurora MySQL 1.* devem usar esta versão.
- Melhoramos a estabilidade das réplicas do Aurora corrigindo a contenção de bloqueio entre os threads que atendem a consultas e os que aplicam alterações do esquema enquanto uma consulta DDL está em andamento na instância de banco de dados do gravador.
- Corrigimos um problema de estabilidade relacionado à atualização da tabela `mysql.innodb_table_stats`, disparada por operações DDL.
- Corrigimos um problema que relatava incorretamente um `ERROR 1836` quando uma consulta aninhada era executada em uma tabela temporária na réplica do Aurora.

Melhorias de desempenho:

- Melhoramos a performance da replicação do log binário evitando chamadas desnecessárias de API ao cache se o cache de consultas tiver sido desabilitado no operador do log binário.

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas esses recursos não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7)

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

O Aurora MySQL 2.06.0 é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, o Aurora MySQL 2.06.0 não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plug-in de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL CREATE TABLESPACE

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-11-11 (versão 2.05.0) (obsoleta)

Versão: 2.05.0

O Aurora MySQL 2.05.0 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 2.01.*, 2.02.*, 2.03.* e 2.04.*.

É possível restaurar um snapshot de uma versão atualmente compatível do Aurora MySQL para a versão Aurora MySQL 2.05.0. Você tem a opção de atualizar os clusters de bancos de dados Aurora MySQL 2.* existentes, até o 2.04.6, para o Aurora MySQL 2.05.0. Não é possível atualizar um cluster do Aurora MySQL 1.* existente diretamente para o 2.05.0. No entanto, é possível restaurar seu snapshot no Aurora MySQL 2.05.0.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo por meio da Console de gerenciamento da AWS, da ou da API do AWS CLI RDS.

Note

No momento, essa versão não está disponível nas seguintes AWS regiões: AWS GovCloud (Leste dos EUA) [-1], (Oeste dos EUA) [us-gov-east-1], China AWS GovCloud (Ningxia) [us-gov-westcn-northwest-1], Ásia-Pacífico (Hong Kong) [ap-east-1], Europa (Estocolmo) [eu-north-1] e Oriente Médio (Bahrein) [me-south-1] -south-1]. Haverá um anúncio separado assim que ela for disponibilizada.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

 Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correções críticas:

- [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](#)
- Correção de um problema em que os eventos no arquivo de log binário atual do mestre não eram replicados no operador se o valor do parâmetro `sync_binlog` não estivesse definido como 1.

Correções de alta prioridade:

- É altamente recomendável que os clientes com tamanho de banco de dados próximo a 64 terabytes (TiB) atualizem para esta versão a fim de evitar tempo de inatividade causado por erros de estabilidade que afetam volumes próximos ao limite de armazenamento do Aurora.
- O valor padrão do parâmetro `aurora_binlog_replication_max_yield_seconds` foi alterado para zero a fim de evitar um aumento no atraso da replicação em favor do desempenho da consulta em primeiro plano no mestre do log binário.

Integração de correções de bugs do MySQL

- Bug #23054591: PURGE BINARY LOGS TO está lendo todo o arquivo de log binário e causando MySQL a paralisação

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas esses recursos não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7)

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para invocar AWS Lambda funções de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

O Aurora MySQL 2.05.0 é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, o Aurora MySQL 2.05.0 não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plug-in de replicação de grupo
- Maior tamanho de página

- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL `CREATE TABLESPACE`

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-08-14 (versão 2.04.9) (obsoleta)

Versão: 2.04.9

Aurora MySQL 2.04.9 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo por meio da Console de gerenciamento da AWS, da ou da API do AWS CLI RDS.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

No momento, essa versão não está disponível nas seguintes AWS regiões: AWS GovCloud (Leste dos EUA) [us-gov-east-1], AWS GovCloud (Oeste dos EUA) [us-gov-west-1], Ásia-Pacífico (Hong Kong) [ap-east-1] e Oriente Médio (Bahrein) [me-south-1]. Haverá um anúncio separado assim que ela for disponibilizada.

 Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correções de alta prioridade:

Correções críticas:

- [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](#)

Melhorias de disponibilidade:

- Correção de um problema que poderia causar uma reinicialização ou um failover do bando de dados devido à execução de um comando `kill session`. Se você encontrar esse problema, entre em contato com o AWS suporte para ativar essa correção na sua instância.
- Corrigido um problema que fazia com que um banco de dados reiniciasse durante a execução de uma consulta complexa envolvendo junções de várias tabelas e agregação que usam tabelas intermediárias internamente.
- Corrigido um problema que fazia com que o banco de dados reiniciasse devido a uma interrupção de `DROP TABLE` em várias tabelas.
- Corrigido um problema que fazia com que um banco de dados fizesse failover durante a recuperação do banco de dados.
- Corrigido uma reinicialização do banco de dados causada por relatórios incorretos de `threads_running` quando auditoria e logs de consulta lentos são habilitados.
- Corrigido um problema em que um comando `kill query` podia ficar travado durante a execução.
- Correção de uma condição de corrida no gerenciador de bloqueio que resultava em uma reinicialização ou um failover do banco de dados durante a reversão da transação.
- Correção de um problema que acionava a reinicialização ou o failover do banco de dados quando várias conexões estavam tentando atualizar a mesma tabela com um índice de Pesquisa de texto completo.
- Corrigido um problema que podia causar um deadlatch ao limpar um índice, resultando em um failover ou em uma reinicialização.

Melhorias gerais:

- Corrigidos problemas que podiam fazer com que consultas em réplicas de leitura usassem dados de uma transação não confirmada. Esse problema é limitado às transações que são iniciadas imediatamente após uma reinicialização do banco de dados.
- Corrigido um problema encontrado durante `INPLACE ALTER TABLE` para uma tabela com triggers definidos e quando a DDL não continha uma cláusula `RENAME`.
- Corrigido um problema que fazia com que a clonagem demorasse mais tempo em alguns clusters de banco de dados com alta carga de gravação.
- Corrigido um problema encontrado durante uma atualização quando uma tabela particionada tinha espaços incorporados no nome.
- Corrigido um problema em que a réplica de leitura podia ver resultados parciais transitórios de uma transação confirmada recentemente no gravador.
- Corrigido um problema em que consultas em uma réplica de leitura em uma tabela FTS podiam produzir resultados obsoletos. Isso só ocorrerá quando a consulta FTS na réplica de leitura segue de perto uma consulta em `INFORMATION_SCHEMA.INNODB_SYS_TABLES` para a mesma tabela FTS no gravador.
- Corrigido um problema que resultava em uma restauração lenta do cluster de banco de dados Aurora 1.x contendo índices FTS (Full-Text Search – pesquisa de texto completo) em um cluster de banco de dados Aurora 2.x.
- Comprimento máximo permitido estendido para 2000 para os parâmetros globais `server_audit_incl_users` e `server_audit_excl_users`.
- Corrigido um problema em que a restauração do Aurora 1.x para o Aurora 2.x poderia levar um tempo prolongado para ser concluída.
- Corrigido um problema em que uma invocação `lambda_async` por meio do procedimento armazenado não funcionava com Unicode.
- Corrigido um problema encontrado quando um índice espacial não manipula corretamente uma coluna de geometria fora do registro.
- Corrigido um problema que poderia fazer com que uma consulta falhasse em uma instância de banco de dados do leitor com o erro `InternalFailureException` e a mensagem "Operation terminated (internal error)" [Operação encerrada (erro interno)].

Integração de correções de bugs do MySQL

- Bug #23070734, Bug #80060: O `TRUNCATE` simultâneo causa paralisações `TABLES`

- Erro nº 23103937: PS_TRUNCATE_ALL_TABLES() NÃO FUNCIONA NO MODO SUPER_READ_ONLY
- Erro nº 22551677: ao colocar o servidor offline, uma condição de corrida dentro do esquema de performance poderia levar a uma saída do servidor.
- Erro nº 27082268: sincronização de FTS inválida.
- ERRO nº 12589870: corrigido um problema que causa uma reinicialização com instrução de várias consultas quando o cache de consulta está habilitado.
- Erro nº 26402045: certos casos de materialização da subconsulta poderiam causar uma saída do servidor. Essas consultas agora produzem um erro sugerindo que a materialização seja desabilitada.
- Erro nº 18898433: consultas com muitas junções à esquerda ficavam lentas se o buffer de junção fosse usado (por exemplo, usando o algoritmo de loop aninhado em bloco).
- Erro nº 25222337: um nome de campo de coluna virtual NULL em um índice virtual causava uma saída do servidor durante uma comparação de nome de campo que ocorre ao preencher colunas virtuais afetadas por uma restrição de chave externa. (<https://github.com/mysql/mysql-server/commit/273d5c9d7072c63b6c47dbef6963d7dc491d5131>)
- Erro nº 25053286: executar um procedimento armazenado que contém uma consulta que acessou uma visualização poderia alocar memória que não foi liberada até o término da sessão. (<https://github.com/mysql/mysql-server/commit/d7b37d4d141a95f577916448650c429f0d6e193d>)
- Bug #25586773: Executar um procedimento armazenado contendo uma instrução que criou uma tabela a partir do conteúdo de determinadas instruções SELECT (<https://dev.mysql.com/doc/refman/5.7/en/select.html>) pode resultar em um vazamento de memória. (<https://github.com/mysql/mysql-server/commit/88301e5adab65f6750f66af284be410c4369d0c1>)
- Erro nº 26666274: LOOP INFINITO NO CONTÊINER DE BUFFER DO ESQUEMA DE PERFORMANCE.
- Erro nº 23550835, erro nº 23298025, erro nº 81464: tabelas do esquema de performance SELECT quando um buffer interno estava cheio poderia causar uma saída do servidor.

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas esses recursos não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7)

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para invocar AWS Lambda funções de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

O Aurora MySQL 2.04.9 é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, o Aurora MySQL 2.04.9 não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plug-in de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL CREATE TABLESPACE

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-11-20 (versão 2.04.8) (obsoleta)

Versão: 2.04.8

O Aurora MySQL 2.04.8 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

É possível restaurar um snapshot de qualquer versão 2.* do Aurora MySQL para a versão Aurora MySQL 2.04.8. Você tem a opção de atualizar os clusters de bancos de dados Aurora MySQL 2.* existentes para o Aurora MySQL 2.04.8.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo por meio da Console de gerenciamento da AWS, da ou da API do AWS CLI RDS.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

No momento, essa versão não está disponível nas seguintes AWS regiões: AWS GovCloud (Leste dos EUA) [us-gov-east-1], AWS GovCloud (Oeste dos EUA) [us-gov-west-1], Ásia-Pacífico (Hong Kong) [ap-east-1] e Oriente Médio (Bahrein) [me-south-1]. Haverá um anúncio separado assim que ela for disponibilizada.

Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Novos recursos:

- Melhorias da réplica de leitura:
 - Reduzimos o tráfego de rede da instância de gravador transmitindo com eficiência os dados para as instâncias de leitor no cluster de bancos de dados Aurora. Essa melhoria é habilitada por padrão porque ajuda a impedir atrasos e reinicialização das réplicas. O parâmetro para esse recurso é `aurora_enable_repl_bin_log_filtering`.
 - Reduzimos o tráfego de rede das instâncias de gravador para instâncias de leitor no cluster de bancos de dados Aurora usando compactação. Essa melhoria é habilitada por padrão somente para as classes de instância 8xlarge e 16xlarge porque elas podem tolerar sobrecarga adicional de CPU para compactação. O parâmetro para esse recurso é `aurora_enable_replica_log_compression`.

Correções de alta prioridade:

- Melhoramos o gerenciamento de memória na instância de gravador do Aurora que impede a reinicialização do gravador causada por falta de memória durante uma workload pesada na presença de instâncias de leitor no cluster de bancos de dados Aurora.
- Correção para uma condição não determinística no programador que resulta na reinicialização do mecanismo ao acessar o objeto de esquema de performance simultaneamente.

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas esses recursos não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7)

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.

- Funções nativas para invocar AWS Lambda funções de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

O Aurora MySQL 2.04.8 é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, o Aurora MySQL 2.04.8 não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plug-in de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL `CREATE TABLESPACE`

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-11-14 (versão 2.04.7) (obsoleta)

Versão: 2.04.7

O Aurora MySQL 2.04.7 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

É possível restaurar um snapshot de uma versão atualmente compatível do Aurora MySQL para a versão Aurora MySQL 2.04.7. Você tem a opção de atualizar os clusters de bancos de dados Aurora MySQL 2.* existentes para o Aurora MySQL 2.04.7. Não é possível atualizar um cluster do Aurora MySQL 1.* existente diretamente para o 2.04.7. No entanto, é possível restaurar seu snapshot no Aurora MySQL 2.04.7.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo por meio da Console de gerenciamento da AWS, da ou da API do AWS CLI RDS.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

No momento, essa versão não está disponível nas seguintes AWS regiões: AWS GovCloud (Leste dos EUA) [us-gov-east-1], AWS GovCloud (Oeste dos EUA) [us-gov-west-1], Ásia-Pacífico (Hong Kong) [ap-east-1] e Oriente Médio (Bahrein) [me-south-1]. Haverá um anúncio separado assim que ela for disponibilizada.

Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correções de alta prioridade:

Tratamento de conexões

- A disponibilidade do banco de dados foi aprimorada para melhor atender a um aumento nas conexões de clientes durante a execução de uma ou mais DDLs. Ela é tratada com a criação temporária de threads adicionais quando necessário. O sistema solicitará a atualização se o banco de dados não responder após a elevação nas conexões durante o processamento de DDL.
- Corrigido um problema que resultava em um valor incorreto para a variável de status `Threads_running` global.

Reinicialização do mecanismo

- Corrigimos um problema de indisponibilidade prolongada na reinicialização do mecanismo. Isso resolve o problema na inicialização do grupo de buffers. Esse problema ocorre raramente, mas pode afetar qualquer versão compatível.

Correções de estabilidade geral:

- Fizemos melhorias nos casos em que as consultas que acessam dados não armazenados em cache podem ficar mais lentas do que o normal. Incentivamos os clientes que tiverem latências de leitura elevadas sem explicação ao acessarem dados não armazenados em cache a fazerem a atualização, pois talvez tenham esse problema.

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas esses recursos não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7)

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para invocar AWS Lambda funções de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).

- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

O Aurora MySQL 2.04.7 é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, o Aurora MySQL 2.04.7 não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plug-in de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL `CREATE TABLESPACE`

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-09-19 (versão 2.04.6) (obsoleta)

Versão: 2.04.6

O Aurora MySQL 2.04.6 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Você tem a opção de atualizar os clusters de bancos de dados Aurora MySQL 2.* existentes para o Aurora MySQL 2.04.6. Não permitimos uma atualização vigente dos clusters do Aurora MySQL 1.*. Essa restrição será suspensa em uma versão posterior do Aurora MySQL 2.*. É possível restaurar snapshots do Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 2.01.*, 2.02.*, 2.03.* e 2.04.* para o Aurora MySQL 2.04.6.

Para usar uma versão mais antiga do Aurora MySQL, você pode criar novos clusters de banco de dados especificando a versão do mecanismo por meio da API Console de gerenciamento da AWS, da AWS CLI ou da Amazon RDS.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

No momento, essa versão não está disponível nas seguintes AWS regiões: Europa (Londres) [eu-west-2] AWS GovCloud , (Leste dos EUA) [-1], (Oeste dos EUA) us-gov-east [-1] AWS GovCloud , China (Ningxia) us-gov-west [cn-northwest-1] e Ásia-Pacífico (Hong Kong) [ap-east-1]. Haverá um anúncio separado assim que ela for disponibilizada.

Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

- Correção de um problema em que os eventos no arquivo de log binário atual do mestre não eram replicados no operador se o valor do parâmetro `sync_binlog` não estivesse definido como 1.
- O valor padrão do parâmetro `aurora_binlog_replication_max_yield_seconds` foi alterado para zero a fim de evitar um aumento no atraso da replicação em favor do desempenho da consulta em primeiro plano no mestre do log binário.

Integração de correções de bugs do MySQL

- Bug #23054591: PURGE BINARY LOGS TO está lendo todo o arquivo de log binário e causando MySQL a paralisação

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas esses recursos não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7)

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para invocar AWS Lambda funções de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

O Aurora MySQL 2.04.6 é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

No momento, o Aurora MySQL 2.04.6 não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plug-in de replicação de grupo
- Maior tamanho de página

- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL CREATE TABLESPACE

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-07-08 (versão 2.04.5) (obsoleta)

Versão: 2.04.5

O Aurora MySQL 2.04.5 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Você tem a opção de atualizar os clusters de bancos de dados Aurora MySQL 2.* existentes para o Aurora MySQL 2.04.5. Não permitimos uma atualização vigente dos clusters do Aurora MySQL 1.*. Essa restrição será suspensa em uma versão posterior do Aurora MySQL 2.*. Você pode restaurar snapshots do Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 2.01.*, 2.02.*, 2.03.* e 2.04.* para o Aurora MySQL 2.04.5.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correções de segurança:

- [CVE-2016-3518](#)

Correções gerais:

- Uma condição de corrida foi corrigida durante o crescimento do volume de armazenamento que fazia com que o banco de dados fosse reiniciado.
- Uma falha de comunicação interna foi corrigida durante o volume aberto que fazia com que o banco de dados fosse reiniciado.
- Suporte de recuperação DDL adicionado ao `ALTER TABLE ALGORITHM=INPLACE` em tabelas particionadas.
- Foi corrigido um problema com a recuperação DDL do `ALTER TABLE ALGORITHM=COPY` que fazia com que o banco de dados fosse reiniciado.
- Estabilidade de réplica do Aurora melhorada mediante workload de exclusão pesada no gravador.
- Foi corrigida uma reinicialização do banco de dados causada por um deadlatch entre o thread que executava a sincronização do índice de pesquisa de texto completo e o thread que executava a remoção da tabela de pesquisa de texto completo do cache de dicionário.
- Correção de um problema de estabilidade no operador do log binário durante a replicação DDL enquanto a conexão com o mestre do log binário é instável.
- Corrigido um out-of-memory problema no código de pesquisa de texto completo que fazia com que o banco de dados fosse reiniciado.
- Foi corrigido um problema no Aurora Writer que fazia com que ele reiniciasse quando todo o volume de 64 tebibytes (TiB) fosse usado.
- Foi corrigida uma condição de corrida no recurso de esquema de performance que fazia com que o banco de dados fosse reiniciado.
- Correção de um problema que causava conexões anuladas ao solucionar um erro no gerenciamento do protocolo de rede.

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas esses recursos não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7)

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para invocar AWS Lambda funções de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

O Aurora MySQL 2.04.5 é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, o Aurora MySQL 2.04.5 não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plug-in de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers

- Plugin de validação de senha
- Plugins de gravação de consulta
- Filtragem de replicação
- A declaração SQL `CREATE TABLESPACE`

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-05-29 (versão 2.04.4) (obsoleta)

versão: 2.04.4

O Aurora MySQL 2.04.4 está disponível para o público geral. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Ao criar um novo cluster de bancos de dados Aurora MySQL, incluindo a restauração de um snapshot, você tem a opção de escolher a compatibilidade com o MySQL 5.7 ou o MySQL 5.6. Não permitimos a atualização local de clusters do Aurora MySQL 1.* ou a restauração de clusters do Aurora MySQL 1.* de um backup do Amazon S3 para o Aurora MySQL 2.04.4. Pretendemos remover essas restrições em uma versão posterior do Aurora MySQL 2.*.

Você pode restaurar snapshots do Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 2.01.*, 2.02.*, 2.03.* e 2.04.* para o Aurora MySQL 2.04.4.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

Atualmente, essa versão não está disponível nas regiões AWS GovCloud (Oeste dos EUA) [us-gov-west-1], Europa (Estocolmo) [eu-north-1], China (Ningxia) [cn-northwest-1] e Ásia-Pacífico (Hong Kong) [ap-east-1]. AWS Haverá um anúncio separado assim que ela for disponibilizada.

 Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

- Correção de um problema que poderia causar falhas ao carregar dados no Aurora do S3.
- Correção de um problema que poderia causar falhas ao carregar dados do Aurora para o S3.
- Correção de um problema que causava conexões anuladas ao solucionar um erro no gerenciamento do protocolo de rede.
- Correção de um problema que poderia causar uma falha ao processar tabelas particionadas.
- Correção de um problema com a indisponibilidade do recurso Performance Insights em algumas regiões.

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas esses recursos não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7)

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para invocar AWS Lambda funções de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).

- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

O Aurora MySQL 2.04.4 é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, o Aurora MySQL 2.04.4 não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plug-in de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL `CREATE TABLESPACE`

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-05-09 (versão 2.04.3) (obsoleta)

Versão: 2.04.3

O Aurora MySQL 2.04.3 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Ao criar um novo cluster de bancos de dados Aurora MySQL, incluindo a restauração de um snapshot, você tem a opção de escolher a compatibilidade com o MySQL 5.7 ou o MySQL 5.6. Não permitimos a atualização local de clusters do Aurora MySQL 1.* ou a restauração de clusters do Aurora MySQL 1.* de um backup do Amazon S3 para o Aurora MySQL 2.04.3. Pretendemos remover essas restrições em uma versão posterior do Aurora MySQL 2.*.

Você pode restaurar snapshots do Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 2.01.*, 2.02.*, 2.03.* e 2.04.* para o Aurora MySQL 2.04.3.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

Atualmente, esta versão não está disponível nas regiões AWS GovCloud (Oeste dos EUA) [us-gov-west-1] e China (Ningxia) [cn-northwest-1]. AWS Haverá um anúncio separado assim que ela for disponibilizada.

Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

- Corrigido um erro na replicação do log binário que pode provocar um problema em instâncias do Aurora configuradas como operador do log binário.
- Foi corrigida uma out-of-memory condição ao lidar com grandes rotinas armazenadas.
- Corrigido um erro ao tratar determinados tipos de comando ALTER TABLE.
- Corrigido um problema com conexões anuladas devido a um erro no gerenciamento do protocolo da rede.

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas esses recursos não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7)

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para invocar AWS Lambda funções de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

O Aurora MySQL 2.04.3 é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, o Aurora MySQL 2.04.3 não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plug-in de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers

- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL CREATE TABLESPACE

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-05-02 (versão 2.04.2) (obsoleta)

Versão: 2.04.2

O Aurora MySQL 2.04.2 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Ao criar um novo cluster de bancos de dados Aurora MySQL, incluindo a restauração de um snapshot, você tem a opção de escolher a compatibilidade com o MySQL 5.7 ou o MySQL 5.6. Não permitimos a atualização local de clusters do Aurora MySQL 1.* ou a restauração de clusters do Aurora MySQL 1.* de um backup do Amazon S3 para o Aurora MySQL 2.04.2. Pretendemos remover essas restrições em uma versão posterior do Aurora MySQL 2.*.

Você pode restaurar snapshots do 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 para o Aurora MySQL 2.04.2.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

Atualmente, esta versão não está disponível nas regiões AWS GovCloud (Oeste dos EUA) [us-gov-west-1] e China (Ningxia) [cn-northwest-1]. AWS Haverá um anúncio separado assim que ela for disponibilizada.

 Note

Para obter informações sobre como atualizar seu cluster de banco de dados do Aurora MySQL, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

- Adicionado suporte para a replicação de log binário do SSL que usa certificados personalizados. Para obter informações sobre como usar a replicação de log binário do SSL no Aurora MySQL, consulte [mysql_rds_import_binlog_ssl_material](#).
- Corrigido um deadlatch na instância principal do Aurora que ocorre quando uma tabela com um índice de Pesquisa de texto completo está sendo otimizada.
- Corrigido um problema nas réplicas do Aurora em que a performance de determinadas consultas que usam `SELECT( *)` pode ser afetada em tabelas que têm índices secundários.
- Corrigida uma condição que resultava na postagem do Erro 1032.
- Melhorada a estabilidade de réplicas do Aurora por meio da correção de vários deadlatches.

Integração de correções de bugs do MySQL

- Bug nº 24829050 - A OTIMIZAÇÃO DE INDEX_MERGE_INTERSECTION PROVOCA RESULTADOS INCORRETOS NA CONSULTA

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas esses recursos não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7)

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.

- Funções nativas para invocar AWS Lambda funções de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

O Aurora MySQL 2.04.2 é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, o Aurora MySQL 2.04.2 não é compatível com os seguintes recursos do MySQL 5.7:

- Plug-in de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL `CREATE TABLESPACE`

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-03-25 (versão 2.04.1) (obsoleta)

Versão: 2.04.1

O Aurora MySQL 2.04.1 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Ao criar um novo cluster de bancos de dados Aurora MySQL, incluindo a restauração de um snapshot, você tem a opção de escolher a compatibilidade com o MySQL 5.7 ou o MySQL 5.6. Não permitimos a atualização local de clusters do Aurora MySQL 1.* ou a restauração de clusters do Aurora MySQL 1.* de um backup do Amazon S3 para o Aurora MySQL 2.04.1. Pretendemos remover essas restrições em uma versão posterior do Aurora MySQL 2.*.

Você pode restaurar snapshots do Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 2.01.*, 2.02.*, 2.03.*, 2.04.0 para o Aurora MySQL 2.04.1.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

 Note

No momento, esta versão não está disponível na região AWS GovCloud (Oeste dos EUA) [us-gov-west-1]. Haverá um anúncio separado assim que ela for disponibilizada.

 Note

O procedimento para atualizar o cluster de banco de dados mudou. Para obter mais informações, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

- Corrigido um problema em que um snapshot do Aurora MySQL 5.6 para versões inferiores à 1.16 não podia ser restaurado para o cluster mais recente do Aurora MySQL 5.7.

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas esses recursos não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7)

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para invocar AWS Lambda funções de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens

- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL CREATE TABLESPACE

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-03-25 (versão 2.04.0) (obsoleta)

Versão: 2.04

O Aurora MySQL 2.04 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Ao criar um novo cluster de bancos de dados Aurora MySQL, incluindo a restauração de um snapshot, você tem a opção de escolher a compatibilidade com o MySQL 5.7 ou o MySQL 5.6. Não permitimos a atualização local de clusters do Aurora MySQL 1.* ou a restauração de clusters do Aurora MySQL 1.* de um backup do Amazon S3 para o Aurora MySQL 2.04.0. Pretendemos remover essas restrições em uma versão posterior do Aurora MySQL 2.*.

Você pode restaurar snapshots do Aurora MySQL 1.19.*, 2.01.*, 2.02.* e 2.03.* para o Aurora MySQL 2.04.0. Não é possível restaurar snapshots do Aurora MySQL 1.14.* ou inferior, 1.15.*, 1.16.*, 1.17.*, 1.18.* no Aurora MySQL 2.04.0. Essa restrição é removida no Aurora MySQL 2.04.1.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

No momento, esta versão não está disponível na região AWS GovCloud (Oeste dos EUA) [us-gov-west-1]. Haverá um anúncio separado assim que ela for disponibilizada.

 Note

O procedimento para atualizar o cluster de banco de dados mudou. Para obter mais informações, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

- Oferece suporte à replicação baseada em GTID. Para obter informações sobre como usar a replicação baseada em GTID com o Aurora MySQL, consulte [Usar a replicação baseada em GTID para o Amazon RDS for MySQL](#) no Guia do usuário do Amazon Aurora.
- Corrigido um problema em que uma réplica do Aurora lança incorretamente um erro `Running in read-only mode` quando uma instrução que exclui ou atualiza linhas em uma tabela temporária contém uma subconsulta do InnoDB.

Integração de correções de bugs do MySQL

- Bug nº 26225783: FALHA NO MYSQL AO CRIAR TABELA (REPRODUZÍVEL) -> INNODB: JUNTO COM ESPERA DE SEMÁFORO.

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas esses recursos não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7)

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para invocar AWS Lambda funções de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

Esta versão do Aurora MySQL é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, essa versão do Aurora MySQL não oferece suporte aos seguintes recursos do MySQL 5.7:

- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL `CREATE TABLESPACE`

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-02-07 (versão 2.03.4) (obsoleta)

Versão: 2.03.4

O Aurora MySQL 2.03.4 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

Ao criar um novo cluster de bancos de dados Aurora MySQL (inclusive restaurar um snapshot), você pode escolher a compatibilidade com o MySQL 5.7 ou o MySQL 5.6.

Não permitimos a atualização no local de clusters do Aurora MySQL 1.* para o Aurora MySQL 2.03.4 ou a restauração para o Aurora MySQL 2.03.4 de um backup do Amazon S3. Pretendemos remover essas restrições em uma versão posterior do Aurora MySQL 2.*.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

 Note

No momento, essa versão não está disponível nas regiões AWS GovCloud (Oeste dos EUA) [us-gov-west-1] e China (Pequim) [cn-north-1]. Haverá um anúncio separado assim que ela for disponibilizada.

 Note

O procedimento para atualizar o cluster de banco de dados mudou. Para obter mais informações, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

- Support para agrupamento UTF8 MB4 Unicode 9.0 com e sem distinção entre maiúsculas e minúsculas, `utf8mb4_0900_as_ci`

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7).

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para invocar AWS Lambda funções de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

O Aurora MySQL 2.03.4 é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, o Aurora MySQL 2.03.4 não oferece suporte aos seguintes recursos do MySQL 5.7:

- Identificadores globais de transações (GTIDs). O Aurora MySQL é compatível com a versão 2.04 e GTIDs superior.
- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação

- A declaração SQL CREATE TABLESPACE
- Protocolo X

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-01-18 (versão 2.03.3) (obsoleta)

Versão: 2.03.3

O Aurora MySQL 2.03.3 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

Ao criar um novo cluster de bancos de dados Aurora MySQL (inclusive restaurar um snapshot), você pode escolher a compatibilidade com o MySQL 5.7 ou o MySQL 5.6.

Não permitimos a atualização no local de clusters do Aurora MySQL 1.* para o Aurora MySQL 2.03.3 ou a restauração para o Aurora MySQL 2.03.3 de um backup do Amazon S3. Pretendemos remover essas restrições em uma versão posterior do Aurora MySQL 2.*.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

No momento, essa versão não está disponível nas regiões AWS GovCloud (Oeste dos EUA) [us-gov-west-1] e China (Pequim) [cn-north-1]. Haverá um anúncio separado assim que ela for disponibilizada.

Note

O procedimento para atualizar o cluster de banco de dados mudou. Para obter mais informações, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correções CVE

- [CVE-2016-5436](#)

Correções críticas:

- Corrigido um problema em que a réplica do Aurora pode se tornar dead-latched ao executar uma verificação retroativa em um índice.
- Corrigido um problema em que uma réplica do Aurora pode ser reiniciada quando a instância principal do Aurora executa operações de DDL no local em tabelas particionadas.
- Corrigido um problema em que uma réplica do Aurora pode ser reiniciada durante a invalidação do cache de consulta depois de uma operação de DDL na instância principal do Aurora.
- Corrigido um problema em que uma réplica do Aurora pode ser reiniciada durante uma consulta SELECT em uma tabela enquanto a instância principal do Aurora executa um truncamento nessa tabela.
- Corrigido um problema de resultado incorreto com tabelas temporárias do MyISAM em que apenas colunas indexadas são acessadas.
- Corrigido um problema em logs lentos que geravam valores grandes incorretos para `query_time` e `lock_time` periodicamente depois de aproximadamente 40.000 consultas.
- Corrigido um problema em que um esquema denominado “tmp” poderia fazer com que a migração do RDS for MySQL para o Aurora MySQL MySQL ficasse travada.
- Corrigido um problema em que o log de auditoria pode ter eventos ausentes durante uma rotação de log.
- Corrigido um problema em que a instância primária do Aurora restaurada de um snapshot do Aurora 5.6 pode ser reiniciada quando o recurso Fast DDL no modo de laboratório está habilitado.
- Corrigido um problema em que o uso de 100% de CPU é provocado pelo thread de estatísticas do dicionário.
- Corrigido um problema em que uma réplica do Aurora pode ser reiniciada ao executar uma instrução CHECK TABLE.

Integração de correções de bugs do MySQL

- Bug nº 25361251: COMPORTAMENTO INCORRETO COM CHAVE DE INSERÇÃO OU DE DUPLICAÇÃO NO SP
- Bug nº 26734162: COMPORTAMENTO INCORRETO COM ATUALIZAÇÃO DE CHAVE DE INSERÇÃO DE BLOB + DUPLICAÇÃO
- Bug nº 27460607: COMPORTAMENTO INCORRETO DE IODKU QUANDO A TABELA DE ORIGEM DE INSERÇÃO DA SELEÇÃO ESTÁ VAZIA
- Uma consulta usando uma cláusula DISTINCT ou GROUP BY pode retornar resultados incorretos. (MySQL 5.7 Bug #79591, Bug #22343910)
- A DELETE de tabelas unificadas usando uma tabela derivada na cláusula de WHERE falha com o erro 1093 (Erro nº 23074801).
- GCOLS: COMPORTAMENTO INCORRETO COM ALTERAÇÕES NO CONJUNTO DE CARACTERES (Bug nº 25287633).

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7).

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para invocar AWS Lambda funções de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

O Aurora MySQL 2.03.3 é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, o Aurora MySQL 2.03.3 não oferece suporte aos seguintes recursos do MySQL 5.7:

- Identificadores globais de transações (GTIDs). O Aurora MySQL é compatível com a versão 2.04 e GTIDs superior.
- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL `CREATE TABLESPACE`
- Protocolo X

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-01-09 (versão 2.03.2) (obsoleta)

Versão: 2.03.2

O Aurora MySQL 2.03.2 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

Ao criar um novo cluster de bancos de dados Aurora MySQL (inclusive restaurar um snapshot), você pode escolher a compatibilidade com o MySQL 5.7 ou o MySQL 5.6.

Não permitimos a atualização no local de clusters do Aurora MySQL 1.* para o Aurora MySQL 2.03.2 ou a restauração para o Aurora MySQL 2.03.2 de um backup do Amazon S3. Pretendemos remover essas restrições em uma versão posterior do Aurora MySQL 2.*.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

 Note

No momento, essa versão não está disponível nas regiões AWS GovCloud (Oeste dos EUA) [us-gov-west-1] e China (Pequim) [cn-north-1]. Haverá um anúncio separado assim que ela for disponibilizada.

 Note

O procedimento para atualizar o cluster de banco de dados mudou. Para obter mais informações, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

- Aurora Version Selector: desde o Aurora MySQL 2.03.2, é possível escolher uma das várias versões do Aurora compatíveis com o MySQL 5.7 no Console de gerenciamento da AWS. Para obter mais informações, consulte [Verificação ou especificação de versões de mecanismo do Aurora MySQL por meio da AWS](#) no Guia do usuário do Amazon Aurora.

Correções críticas:

- [CVE-2016-3495](#)

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7).

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para invocar AWS Lambda funções de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

O Aurora MySQL 2.03.2 é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, o Aurora MySQL 2.03.2 não é compatível com os seguintes recursos do MySQL 5.7:

- Identificadores globais de transações (GTIDs). O Aurora MySQL é compatível com a versão 2.04 e GTIDs superior.
- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens

- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL CREATE TABLESPACE
- Protocolo X

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-10-24 (versão 2.03.1) (obsoleta)

Versão: 2.03.1

O Aurora MySQL 2.03.1 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

Ao criar um novo cluster de bancos de dados Aurora MySQL, escolha a compatibilidade com MySQL 5.7 ou MySQL 5.6. Ao restaurar um snapshot compatível com MySQL 5.6, escolha a compatibilidade com MySQL 5.7 ou MySQL 5.6.

Você pode restaurar snapshots do Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 2.01.*, 2.02.* e 2.03 para o Aurora MySQL 2.03.1.

Não permitimos a atualização no local de clusters do Aurora MySQL 1.* para o Aurora MySQL 2.03.1 ou a restauração para o Aurora MySQL 2.03.1 de um backup do Amazon S3. Pretendemos remover essas restrições em uma versão posterior do Aurora MySQL 2.*.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

No momento, essa versão não está disponível nas regiões AWS GovCloud (Oeste dos EUA) [us-gov-west-1] e China (Pequim) [cn-north-1]. Haverá um anúncio separado assim que ela for disponibilizada.

Melhorias

- Corrija um problema em que o Aurora Writer possa reiniciar durante a detecção de deadlock da transação.

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7).

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para invocar AWS Lambda funções de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

O Aurora MySQL 2.03.1 é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, o Aurora MySQL 2.03.1 não oferece suporte aos seguintes recursos do MySQL 5.7:

- Identificadores globais de transações (GTIDs). O Aurora MySQL é compatível com a versão 2.04 e GTIDs superior.

- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL `CREATE TABLESPACE`
- Protocolo X

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-10-11 (versão 2.03) (obsoleta)

Versão: 2.03

O Aurora MySQL 2.03 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

Ao criar um novo cluster de bancos de dados Aurora MySQL, escolha a compatibilidade com MySQL 5.7 ou MySQL 5.6. Ao restaurar um snapshot compatível com MySQL 5.6, escolha a compatibilidade com MySQL 5.7 ou MySQL 5.6.

Você pode restaurar snapshots do Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 2.01.*, 2.02.* para o Aurora MySQL 2.03.

Não permitimos a atualização no local de clusters do Aurora MySQL 1.* para o Aurora MySQL 2.03 ou a restauração para o Aurora MySQL 2.03 de um backup do Amazon S3. Pretendemos remover essas restrições em uma versão posterior do Aurora MySQL 2.*.

 Note

No momento, essa versão não está disponível nas regiões AWS GovCloud (Oeste dos EUA) [us-gov-west-1] e China (Pequim) [cn-north-1]. Haverá um anúncio separado assim que ela for disponibilizada.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

- O esquema de performance está disponível.
- Corrigido um problema em que sessões zumbi com estado eliminado podem consumir mais CPU.
- Corrigido um problema de trava inoperante quando uma transação somente leitura está adquirindo um bloqueio em um registro no Aurora Writer.
- Corrigido um problema em que a réplica do Aurora sem workload do cliente pode apresentar alta utilização da CPU.
- Várias correções em edições que podem causar a reinicialização da réplica do Aurora ou do gravador do Aurora.
- Adicionado recurso para ignorar o registro em log do diagnóstico quando o limite da taxa de transferência do disco é atingido.
- Corrigido um problema de vazamento da memória quando o log binário está habilitado no gravador do Aurora.

Integração de correções de bug da edição MySQL community

- VARREDURA REVERSA EM UMA TABELA PARTICIONADA REALIZA ICP - PEDIDO DE DESC (Bug n° 24929748).
- JSON_OBJECT CRIA CÓDIGO JSON INVÁLIDO (Bug n° 26867509).
- INSERIR DADOS JSON GRANDES DEMORA MUITO (Bug n° 22843444).
- TABELAS PARTICIONADAS USAM MAIS MEMÓRIA EM 5.7 DO QUE EM 5.6 (Bug n° 25080442).

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7).

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para invocar AWS Lambda funções de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

O Aurora MySQL 2.03 é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, o Aurora MySQL 2.03 não oferece suporte aos seguintes recursos do MySQL 5.7:

- Identificadores globais de transações (GTIDs). O Aurora MySQL é compatível com a versão 2.04 e GTIDs superior.
- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens

- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL CREATE TABLESPACE
- Protocolo X

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-10-08 (versão 2.02.5) (obsoleta)

Versão: 2.02.5

O Aurora MySQL 2.02.5 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

Ao criar um novo cluster de bancos de dados Aurora MySQL, escolha a compatibilidade com MySQL 5.7 ou MySQL 5.6. Ao restaurar um snapshot compatível com MySQL 5.6, escolha a compatibilidade com MySQL 5.7 ou MySQL 5.6.

Você pode restaurar snapshots do Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 2.01.*, 2.02.* para o Aurora MySQL 2.02.5. Você também pode realizar uma atualização no local do Aurora MySQL 2.01.* ou 2.02.* para o Aurora MySQL 2.02.5.

Não permitimos a atualização no local de clusters do Aurora MySQL 1.* para o Aurora MySQL 2.02.5 ou a restauração para o Aurora MySQL 2.02.5 de um backup do Amazon S3. Pretendemos remover essas restrições em uma versão posterior do Aurora MySQL 2.*.

O esquema de performance está desabilitado para esta versão do Aurora MySQL 5.7. Atualização para o Aurora 2.03 para suporte ao esquema de performance.

Note

No momento, essa versão não está disponível nas regiões AWS GovCloud (Oeste dos EUA) [us-gov-west-1] e China (Pequim) [cn-north-1]. Haverá um anúncio separado assim que ela for disponibilizada.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

- Corrija um problema em que uma réplica do Aurora pode reiniciar ao fazer uma varredura reversa em uma tabela.

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7).

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para invocar AWS Lambda funções de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

O Aurora MySQL 2.02.5 é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, o Aurora MySQL 2.02.5 não oferece suporte aos seguintes recursos do MySQL 5.7:

- Identificadores globais de transações (GTIDs). O Aurora MySQL é compatível com a versão 2.04 e GTIDs superior.
- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL CREATE TABLESPACE
- Protocolo X

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-09-21 (versão 2.02.4) (obsoleta)

Versão: 2.02.4

O Aurora MySQL 2.02.4 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

Ao criar um novo cluster de bancos de dados Aurora MySQL, escolha a compatibilidade com MySQL 5.7 ou MySQL 5.6. Ao restaurar um snapshot compatível com MySQL 5.6, escolha a compatibilidade com MySQL 5.7 ou MySQL 5.6.

Você pode restaurar snapshots do Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 2.01.*, 2.02.* para o Aurora MySQL 2.02.4. Você também pode realizar uma atualização no local do Aurora MySQL 2.01.* ou 2.02.* para o Aurora MySQL 2.02.4.

Não permitimos a atualização no local de clusters do Aurora MySQL 1.* para o Aurora MySQL 2.02.4 ou a restauração para o Aurora MySQL 2.02.4 de um backup do Amazon S3. Pretendemos remover essas restrições em uma versão posterior do Aurora MySQL 2.*.

O esquema de performance está desabilitado para esta versão do Aurora MySQL 5.7. Atualização para o Aurora 2.03 para suporte ao esquema de performance.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

- Corrigido um problema de estabilidade para índices de pesquisa de texto completo em tabelas restauradas com base em um snapshot do Aurora MySQL 5.6.

Integração de correções de bug da edição MySQL community

- 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

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7).

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para invocar AWS Lambda funções de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Compatibilidade com o MySQL 5.7

O Aurora MySQL 2.02.4 é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, o Aurora MySQL 2.02.4 não oferece suporte aos seguintes recursos do MySQL 5.7:

- Identificadores globais de transações (GTIDs). O Aurora MySQL é compatível com a versão 2.04 e GTIDs superior.
- Plugin de replicação de grupo

- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL `CREATE TABLESPACE`
- Protocolo X

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-08-23 (versão 2.02.3) (obsoleta)

Versão: 2.02.3

O Aurora MySQL 2.02.3 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

Ao criar um novo cluster de bancos de dados Aurora MySQL, escolha a compatibilidade com MySQL 5.7 ou MySQL 5.6. Ao restaurar um snapshot compatível com MySQL 5.6, escolha a compatibilidade com MySQL 5.7 ou MySQL 5.6.

Você pode restaurar snapshots do Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 2.01.*, 2.02.* para o Aurora MySQL 2.02.3. Você também pode realizar uma atualização no local do Aurora MySQL 2.01.* ou 2.02.* para o Aurora MySQL 2.02.3.

Não permitimos a atualização no local de clusters do Aurora MySQL 1.* para o Aurora MySQL 2.02.3 ou a restauração para o Aurora MySQL 2.02.3 de um backup do Amazon S3. Pretendemos remover essas restrições em uma versão posterior do Aurora MySQL 2.*.

O esquema de performance está desabilitado para esta versão do Aurora MySQL 5.7. Atualização para o Aurora 2.03 para suporte ao esquema de performance.

 Note

No momento, essa versão não está disponível nas regiões AWS GovCloud (Oeste dos EUA) [us-gov-west-1] e China (Pequim) [cn-north-1]. Haverá um anúncio separado assim que ela for disponibilizada.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Comparação com o Aurora MySQL versão 1

Os seguintes recursos do Amazon Aurora MySQL são compatíveis no Aurora MySQL versão 1 (compatível com o MySQL 5.6), mas não são compatíveis atualmente no Aurora MySQL versão 2 (compatível com o MySQL 5.7).

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para invocar AWS Lambda funções de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Atualmente, o Aurora MySQL 2.01 não oferece suporte aos recursos adicionados ao Aurora MySQL versão 1.16 e posterior. Para obter informações sobre o Aurora MySQL versão 1.16, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).

Compatibilidade com o MySQL 5.7

O Aurora MySQL 2.02.3 é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, o Aurora MySQL 2.02.3 não oferece suporte aos seguintes recursos do MySQL 5.7:

- Identificadores globais de transações (GTIDs). O Aurora MySQL é compatível com a versão 2.04 e GTIDs superior.
- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL `CREATE TABLESPACE`
- Protocolo X

Diferenças de CLI entre o Aurora MySQL 2.x e o Aurora MySQL 1.x

- O nome do mecanismo para o Aurora MySQL 2.x é `aurora-mysql`; o nome do mecanismo para o Aurora MySQL 1.x continua sendo `aurora`.
- O parameter group padrão para o Aurora MySQL 2.x é `default.aurora-mysql5.7`; o parameter group padrão para o Aurora MySQL 1.x continua sendo `default.aurora5.6`.
- O nome de família de parameter groups de cluster de banco de dados para o Aurora MySQL 2.x é `aurora-mysql5.7`; nome de família de parameter groups de cluster de banco de dados para o Aurora MySQL 1.x continua sendo `aurora5.6`.

Consulte a documentação do Aurora para ver o conjunto completo de comandos da CLI e as diferenças entre o Aurora MySQL 2.x e o Aurora MySQL 1.x.

Melhorias

- Foi corrigido um problema em que uma réplica do Aurora pode ser reiniciada ao usar restaurações de cursor otimistas enquanto lê registros.
- Atualizado o valor padrão do parâmetro `innodb_stats_persistent_sample_pages` para 128 a fim de aprimorar estatísticas de índice.
- Corrigido um problema em que uma réplica do Aurora pode reiniciar ao acessar uma tabela pequena modificada simultaneamente na instância principal do Aurora.
- Corrigido `ANALYZE TABLE` para interromper o abastecimento do cache de definição da tabela.
- Corrigido um problema em que a instância principal do Aurora ou uma réplica do Aurora pode reiniciar ao converter uma consulta de ponto geoespacial em um intervalo de pesquisa.

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-06-04 (versão 2.02.2) (obsoleta)

Versão: 2.02.2

O Aurora MySQL 2.02.2 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

Ao criar um novo cluster de bancos de dados Aurora MySQL, escolha a compatibilidade com MySQL 5.7 ou MySQL 5.6. Ao restaurar um snapshot compatível com MySQL 5.6, escolha a compatibilidade com MySQL 5.7 ou MySQL 5.6.

Você pode restaurar snapshots do Aurora MySQL 1.14*, 1.15*, 1.16*, 1.17*, 2.01* e 2.02 para o Aurora MySQL 2.02.2. Você também pode realizar uma atualização no local do Aurora MySQL 2.01* ou 2.02 para o Aurora MySQL 2.02.2.

Não permitimos a atualização no local de clusters do Aurora MySQL 1.* para o Aurora MySQL 2.02.2 ou a restauração para o Aurora MySQL 2.02.2 de um backup do Amazon S3. Pretendemos remover essas restrições em uma versão posterior do Aurora MySQL 2.*.

O esquema de performance está desabilitado para esta versão do Aurora MySQL 5.7. Atualização para o Aurora 2.03 para suporte ao esquema de performance.

 Note

No momento, essa versão não está disponível nas regiões AWS GovCloud (Oeste dos EUA) [us-gov-west-1] e China (Pequim) [cn-north-1]. Haverá um anúncio separado assim que ela for disponibilizada.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correções críticas:

- [CVE-2016-3486](#)

Comparação com o Aurora MySQL 5.6

Os recursos a seguir do Amazon Aurora MySQL são compatíveis com o Aurora MySQL 5.6, mas atualmente não são compatíveis com o Aurora MySQL 5.7.

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para invocar AWS Lambda funções de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Atualmente, o Aurora MySQL 2.01 não oferece suporte aos recursos adicionados ao Aurora MySQL versão 1.16 e posterior. Para obter informações sobre o Aurora MySQL versão 1.16, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).

Compatibilidade com o MySQL 5.7

O Aurora MySQL 2.02.2 é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, o Aurora MySQL 2.02.2 não oferece suporte aos seguintes recursos do MySQL 5.7:

- Identificadores globais de transações (GTIDs). O Aurora MySQL é compatível com a versão 2.04 e GTIDs superior.
- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL CREATE TABLESPACE
- Protocolo X

Diferenças de CLI entre o Aurora MySQL 2.x e o Aurora MySQL 1.x

- O nome do mecanismo para o Aurora MySQL 2.x é `aurora-mysql`; o nome do mecanismo para o Aurora MySQL 1.x continua sendo `aurora`.
- O parameter group padrão para o Aurora MySQL 2.x é `default.aurora-mysql5.7`; o parameter group padrão para o Aurora MySQL 1.x continua sendo `default.aurora5.6`.

- O nome de família de parameter groups de cluster de banco de dados para o Aurora MySQL 2.x é `aurora-mysql5.7`; nome de família de parameter groups de cluster de banco de dados para o Aurora MySQL 1.x continua sendo `aurora5.6`.

Consulte a documentação do Aurora para ver o conjunto completo de comandos da CLI e as diferenças entre o Aurora MySQL 2.x e o Aurora MySQL 1.x.

Melhorias

- Foi corrigido um problema no gravador do Aurora que, ocasionalmente, pode ser reiniciado ao rastrear o andamento de réplica do Aurora.
- Foi corrigido um problema em que uma réplica do Aurora é reiniciada ou gera um erro quando uma tabela particionada é acessada depois da execução de instruções de criação ou de descarte de índices na tabela no gravador do Aurora.
- Corrigido um problema em que uma tabela em uma réplica do Aurora ficava inacessível enquanto aplicava as alterações causadas pela execução das instruções da ADD/DROP coluna da tabela ALTER no Aurora Writer.

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-05-03 (versão 2.02) (obsoleta)

Versão: 2.02

O Aurora MySQL 2.02 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

Ao criar um novo cluster de bancos de dados Aurora MySQL, escolha a compatibilidade com MySQL 5.7 ou MySQL 5.6. Ao restaurar um snapshot compatível com MySQL 5.6, escolha a compatibilidade com MySQL 5.7 ou MySQL 5.6.

Você pode restaurar snapshots do Aurora MySQL 1.14*, 1.15*, 1.16*, 1.17* e 2.01* para o Aurora MySQL 2.02. Você também pode realizar uma atualização no local do Aurora MySQL 2.01* ou para o Aurora MySQL 2.02.

Não permitimos a atualização no local de clusters do Aurora MySQL 1.x para o Aurora MySQL 2.02 ou a restauração para o Aurora MySQL 2.02 de um backup do Amazon S3. Pretendemos eliminar essas restrições em uma versão posterior do Aurora MySQL 2.x.

O esquema de performance está desabilitado para esta versão do Aurora MySQL 5.7. Atualização para o Aurora 2.03 para suporte ao esquema de performance.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Comparação com o Aurora MySQL 5.6

Os recursos a seguir do Amazon Aurora MySQL são compatíveis com o Aurora MySQL 5.6, mas atualmente não são compatíveis com o Aurora MySQL 5.7.

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para invocar AWS Lambda funções de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Atualmente, o Aurora MySQL 2.01 não oferece suporte aos recursos adicionados ao Aurora MySQL versão 1.16 e posterior. Para obter informações sobre o Aurora MySQL versão 1.16, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).

Compatibilidade com o MySQL 5.7

O Aurora MySQL 2.02 é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, o Aurora MySQL 2.02 não oferece suporte aos seguintes recursos do MySQL 5.7:

- Identificadores globais de transações (GTIDs). O Aurora MySQL é compatível com a versão 2.04 e GTIDs superior.
- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL `CREATE TABLESPACE`
- Protocolo X

Diferenças de CLI entre o Aurora MySQL 2.x e o Aurora MySQL 1.x

- O nome do mecanismo para o Aurora MySQL 2.x é `aurora-mysql`; o nome do mecanismo para o Aurora MySQL 1.x continua sendo `aurora`.
- O parameter group padrão para o Aurora MySQL 2.x é `default.aurora-mysql5.7`; o parameter group padrão para o Aurora MySQL 1.x continua sendo `default.aurora5.6`.
- O nome de família de parameter groups de cluster de banco de dados para o Aurora MySQL 2.x é `aurora-mysql5.7`; nome de família de parameter groups de cluster de banco de dados para o Aurora MySQL 1.x continua sendo `aurora5.6`.

Consulte a documentação do Aurora para ver o conjunto completo de comandos da CLI e as diferenças entre o Aurora MySQL 2.x e o Aurora MySQL 1.x.

Melhorias

- Correção de um problema onde um Aurora Writer reinicia ao executar instruções INSERT e explorar a otimização de Inserção rápida.
- Correção de um problema onde um Aurora Replica reinicia ao executar instruções de ALTER DATABASE no Aurora Replica.
- Correção de um problema onde Aurora Replica reinicia ao executar consultas em tabelas que tinham acabado de ser colocadas no Aurora Writer.
- Correção de um problema onde Aurora Replica reinicia ao definir `innodb_adaptive_hash_index` como OFF no Aurora Replica.
- Correção de um problema onde um Aurora Replica reinicia ao executar consultas de TRUNCATE TABLE no Aurora Writer.
- Correção de um problema onde Aurora Writer congela em determinadas circunstâncias ao executar INSERT. Em um cluster de nó múltiplo, isso pode resultar em um failover.
- Correção de um vazamento de memória associado com definição das variáveis de sessão.
- Correção de um problema onde Aurora Writer congela em determinadas circunstâncias associadas com reversão da limpeza de tabelas com colunas geradas.
- Correção de um problema onde Aurora Writer pode, às vezes, reiniciar quando log binário é habilitado.

Integração de correções de bugs do MySQL

- A junção à esquerda retorna resultados incorretos na parte externa (Bug nº 22833364).

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-03-13 (versão 2.01.1) (obsoleta)

Versão: 2.01.1

O Aurora MySQL 2.01.1 está disponível para o público. As versões do Aurora MySQL 2.x são compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x são compatíveis com o MySQL 5.6.

Ao criar um novo cluster de bancos de dados Aurora MySQL, escolha a compatibilidade com MySQL 5.7 ou MySQL 5.6. Ao restaurar um snapshot compatível com MySQL 5.6, escolha a compatibilidade com MySQL 5.7 ou MySQL 5.6.

Você pode restaurar snapshots do Aurora MySQL 1.14*, 1.15*, 1.16* e 2.01* para o Aurora MySQL 2.01.1.

Não permitimos a atualização no local de clusters do Aurora MySQL 1.x para o Aurora MySQL 2.01.1 ou a restauração para o Aurora MySQL 2.01.1 de um backup do Amazon S3. Pretendemos eliminar essas restrições em uma versão posterior do Aurora MySQL 2.x.

O esquema de performance está desabilitado para esta versão do Aurora MySQL 5.7. Atualização para o Aurora 2.03 para suporte ao esquema de performance.

Comparação com o Aurora MySQL 5.6

Os recursos a seguir do Amazon Aurora MySQL são compatíveis com o Aurora MySQL 5.6, mas atualmente não são compatíveis com o Aurora MySQL 5.7.

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para invocar AWS Lambda funções de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Atualmente, o Aurora MySQL 2.01.1 não oferece suporte aos recursos adicionados ao Aurora MySQL versão 1.16 e posterior. Para obter informações sobre o Aurora MySQL versão 1.16, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).

Compatibilidade com o MySQL 5.7

O Aurora MySQL 2.01.1 é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, o Aurora MySQL 2.01.1 não oferece suporte aos seguintes recursos do MySQL 5.7:

- Identificadores globais de transações (GTIDs). O Aurora MySQL é compatível com a versão 2.04 e GTIDs superior.
- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta
- Filtragem de replicação
- A declaração SQL `CREATE TABLESPACE`
- Protocolo X

Diferenças de CLI entre o Aurora MySQL 2.x e o Aurora MySQL 1.x

- O nome do mecanismo para o Aurora MySQL 2.x é `aurora-mysql`; o nome do mecanismo para o Aurora MySQL 1.x continua sendo `aurora`.
- O parameter group padrão para o Aurora MySQL 2.x é `default.aurora-mysql5.7`; o parameter group padrão para o Aurora MySQL 1.x continua sendo `default.aurora5.6`.
- O nome de família de parameter groups de cluster de banco de dados para o Aurora MySQL 2.x é `aurora-mysql5.7`; nome de família de parameter groups de cluster de banco de dados para o Aurora MySQL 1.x continua sendo `aurora5.6`.

Consulte a documentação do Aurora para ver o conjunto completo de comandos da CLI e as diferenças entre o Aurora MySQL 2.x e o Aurora MySQL 1.x.

Melhorias

- Foi corrigido um problema com a restauração de snapshots em que privilégios de banco de dados específicos ao Aurora eram criados incorretamente quando um snapshot compatível com o MySQL 5.6 era restaurado com compatibilidade com o MySQL 5.7.
- Suporte adicionado para as restaurações de snapshots 1.17.

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-02-06 (versão 2.01) (obsoleta)

Versão: 2.01

O Aurora MySQL 2.01 está disponível para o público. Daqui em diante, as versões do Aurora MySQL 2.x serão compatíveis com as versões do MySQL 5.7, e as versões do Aurora MySQL 1.x serão compatíveis com o MySQL 5.6.

Ao criar um novo cluster de bancos de dados Aurora MySQL, inclusive os restaurados de snapshots, escolha a compatibilidade com MySQL 5.7 ou MySQL 5.6.

Você pode restaurar snapshots do Aurora MySQL 1.14*, 1.15* e 1.16* para o Aurora MySQL 2.01.

Não permitimos a atualização no local de clusters do Aurora MySQL 1.x para o Aurora MySQL 2.01 ou a restauração para o Aurora MySQL 2.01 de um backup do Amazon S3. Pretendemos eliminar essas restrições em uma versão posterior do Aurora MySQL 2.x.

O esquema de performance está desabilitado para esta versão do Aurora MySQL 5.7. Atualização para o Aurora 2.03 para suporte ao esquema de performance.

Comparação com o Aurora MySQL 5.6

Os recursos a seguir do Amazon Aurora MySQL são compatíveis com o Aurora MySQL 5.6, mas atualmente não são compatíveis com o Aurora MySQL 5.7.

- Pré-busca de chave assíncrona (AKP). Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.

- Hash joins. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Funções nativas para invocar AWS Lambda funções de forma síncrona. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Lote de varredura. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).
- Migrar dados do MySQL usando um bucket do Amazon S3. Para obter mais informações, consulte [Migrar dados de um banco de dados MySQL externo para um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Atualmente, o Aurora MySQL 2.01 não oferece suporte aos recursos adicionados ao Aurora MySQL versão 1.16 e posterior. Para obter informações sobre o Aurora MySQL versão 1.16, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#).

Compatibilidade com o MySQL 5.7

O Aurora MySQL 2.01 é compatível com o MySQL 5.7 e inclui recursos como suporte a JSON, índices espaciais e colunas geradas. O Aurora MySQL usa uma implementação nativa de indexação espacial com curvas de ordem z para oferecer performance de gravação 20 vezes melhor e performance de leitura 10 vezes melhor do que os conjuntos de dados espaciais do MySQL 5.7.

Atualmente, o Aurora MySQL 2.01 não oferece suporte aos seguintes recursos do MySQL 5.7:

- Identificadores globais de transações (GTIDs). O Aurora MySQL é compatível com a versão 2.04 e GTIDs superior.
- Plugin de replicação de grupo
- Maior tamanho de página
- Carregamento de grupo de buffers InnoDB na inicialização
- Plugin de analisador de texto completo do InnoDB
- Replicação em várias origens
- Redimensionamento online do grupo de buffers
- Plugin de validação de senha
- Plugins de regravação de consulta

- Filtragem de replicação
- A declaração SQL `CREATE TABLESPACE`
- Protocolo X

Diferenças de CLI entre o Aurora MySQL 2.x e o Aurora MySQL 1.x

- O nome do mecanismo para o Aurora MySQL 2.x é `aurora-mysql`; o nome do mecanismo para o Aurora MySQL 1.x continua sendo `aurora`.
- O parameter group padrão para o Aurora MySQL 2.x é `default.aurora-mysql5.7`; o parameter group padrão para o Aurora MySQL 1.x continua sendo `default.aurora5.6`.
- O nome de família de parameter groups de cluster de banco de dados para o Aurora MySQL 2.x é `aurora-mysql5.7`; nome de família de parameter groups de cluster de banco de dados para o Aurora MySQL 1.x continua sendo `aurora5.6`.

Consulte a documentação do Aurora para ver o conjunto completo de comandos da CLI e as diferenças entre o Aurora MySQL 2.x e o Aurora MySQL 1.x.

Atualizações feitas no mecanismo de banco de dados do Amazon Aurora MySQL versão 1 (obsoleta)

As atualizações a seguir referem-se ao mecanismo de banco de dados da versão 1 do Amazon Aurora:

- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-09-30 \(versão 1.23.4\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-06-28 \(versão 1.23.3\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-03-18 \(versão 1.23.2\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-11-24 \(versão 1.23.1\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-09-02 \(versão 1.23.0\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-06-03 \(versão 1.22.5\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-03-04 \(versão 1.22.4\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-11-09 \(versão 1.22.3\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-03-05 \(versão 1.22.2\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-12-23 \(versão 1.22.1\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-11-25 \(versão 1.22.0\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-11-25 \(versão 1.21.0\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-03-05 \(versão 1.20.1\) \(obsoleta\)](#)

- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-11-11 \(versão 1.20.0\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-03-05 \(versão 1.19.6\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-09-19 \(versão 1.19.5\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-06-05 \(versão 1.19.2\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-05-09 \(versão 1.19.1\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-02-07 \(versão 1.19.0\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-09-20 \(versão 1.18.0\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-03-05 \(versão 1.17.9\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-01-17 \(versão 1.17.8\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-10-08 \(versão 1.17.7\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-09-06 \(versão 1.17.6\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-08-14 \(versão 1.17.5\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-08-07 \(versão 1.17.4\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-06-05 \(versão 1.17.3\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-04-27 \(versão 1.17.2\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-03-23 \(versão 1.17.1\) \(obsoleta\)](#)

- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-03-13 \(versão 1.17\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 \(versão 1.16\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-11-20 \(versão 1.15.1\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-10-24 \(versão 1.15\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-03-13 \(versão 1.14.4\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL: 2017-09-22 \(versão 1.14.1\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL: 2017-08-07 \(versão 1.14\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL: 2017-05-15 \(versão 1.13\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL: 2017-04-05 \(versão 1.12\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL: 2017-02-23 \(versão 1.11\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL: 2017-01-12 \(versão 1.10.1\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL: 2016-12-14 \(versão 1.10\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL: 2016-11-10 \(versão 1.9.0, 1.9.1\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL: 2016-10-26 \(versão 1.8.1\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL: 2016-10-18 \(versão 1.8\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL: 2016-09-20 \(versão 1.7.1\) \(obsoleta\)](#)

- [Atualizações do mecanismo de banco de dados do Aurora MySQL: 2016-08-30 \(versão 1.7.0\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL: 2016-06-01 \(versão 1.6.5\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL: 2016-04-06 \(versão 1.6\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL: 2016-01-11 \(versão 1.5\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL: 2015-12-03 \(versão 1.4\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL: 2015-10-16 \(versão 1.2, 1.3\) \(obsoleta\)](#)
- [Atualizações do mecanismo de banco de dados do Aurora MySQL: 2015-08-24 \(versão 1.1\) \(obsoleta\)](#)

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-09-30 (versão 1.23.4) (obsoleta)

Versão: 1.23.4

A versão 1.23.4 do Aurora MySQL está disponível para o público. As versões 2.* do Aurora MySQL são compatíveis com o MySQL 5.7 e as versões 1.* do Aurora MySQL são compatíveis com o MySQL 5.6.

Essa versão de mecanismo está programada para ser descontinuada em 28 de fevereiro de 2023. Para obter mais informações, consulte [Preparar para o fim da vida útil do Amazon Aurora, edição compatível com MySQL versão 1](#).

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo pelo console do RDS, pela AWS CLI ou pela API do Amazon RDS.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Melhorias gerais:

- Corrigido um problema que poderia causar alto consumo de CPU nas instâncias de leitor devido ao log excessivo de mensagens informativas em arquivos de log de diagnóstico internos.

Correções de alta prioridade:

- [CVE-2021-2307](#)
- [CVE-2021-2226](#)
- [CVE-2021-2160](#)
- [CVE-2021-2154](#)
- [CVE-2021-2060](#)
- [CVE-2021-2032](#)
- [CVE-2021-2001](#)

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-06-28 (versão 1.23.3) (obsoleta)

Versão: 1.23.3

O Aurora MySQL 1.23.3 está disponível para o público. As versões 1.* do Aurora MySQL são compatíveis com o MySQL 5.6 e as versões 2.* do Aurora MySQL são compatíveis com o MySQL 5.7.

Essa versão de mecanismo está programada para ser descontinuada em 28 de fevereiro de 2023. Para obter mais informações, consulte [Preparar para o fim da vida útil do Amazon Aurora, edição compatível com MySQL versão 1](#).

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo pelo console do RDS, pela AWS CLI ou pela API do Amazon RDS.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Melhorias gerais de estabilidade e disponibilidade.

Correções de segurança:

- [CVE-2021-23841](#)
- [CVE-2021-3449](#)
- [CVE-2020-28196](#)

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-03-18 (versão 1.23.2) (obsoleta)

Versão: 1.23.2

O Aurora MySQL 1.23.2 está disponível para o público. As versões 1.* do Aurora MySQL são compatíveis com o MySQL 5.6 e as versões 2.* do Aurora MySQL são compatíveis com o MySQL 5.7.

Essa versão de mecanismo está programada para ser descontinuada em 28 de fevereiro de 2023. Para obter mais informações, consulte [Preparar para o fim da vida útil do Amazon Aurora, edição compatível com MySQL versão 1](#).

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo pelo console do RDS, pela AWS CLI ou pela API do Amazon RDS.

Note

Essa versão não está disponível nas seguintes regiões: AWS GovCloud (EUA-Leste) [us-gov-east-1], AWS GovCloud (EUA-Oeste) [us-gov-west-1]. Haverá um anúncio separado assim que ela for disponibilizada.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correções de alta prioridade:

- [CVE-2020-14867](#)
- [CVE-2020-14812](#)
- [CVE-2020-14769](#)
- [CVE-2020-14765](#)
- [CVE-2020-14793](#)
- [CVE-2020-14672](#)
- [CVE-2020-1971](#)
- [CVE-2018-3143](#)

Melhorias de disponibilidade:

- Correção de um problema no recurso de redimensionamento do armazenamento de clusters que poderia fazer com que as instâncias de banco de dados do leitor reiniciassem.
- Correção de um problema de failover devido a uma condição de corrida na declaração RESET QUERY CACHE.
- Correção de uma falha em uma chamada de procedimento armazenada aninhada com cache de consulta.
- Correção de um problema para evitar o reinício repetido do mysqld ao se recuperar de um truncamento incompleto de tabelas particionadas ou subparticionadas.
- Correção de um problema que poderia impedir o sucesso da migração do ambiente on-premise ou do RDS for MySQL para o Aurora MySQL.
- Correção de uma condição de corrida rara em que o banco de dados pode reiniciar durante o dimensionamento do volume de armazenamento.
- Correção de um problema no gerenciador de bloqueio em que uma condição de corrida podia fazer com que um bloqueio fosse compartilhado por duas transações, fazendo com que o banco de dados fosse reiniciado.

- Correção de um problema relacionado ao gerenciamento de memória de bloqueio de transação com transações de gravação de longa duração resultando em uma reinicialização do banco de dados.
- Correção de uma condição de corrida no gerenciador de bloqueio que resultava em uma reinicialização ou um failover do banco de dados durante a reversão da transação.
- Correção de um problema durante a atualização de 5.6 para 5.7 quando o DDL Online Rápido estava habilitado na tabela no modo de laboratório do 5.6.
- Vários problemas em que o mecanismo podia ser reiniciado durante a aplicação de patches com tempo de inatividade zero durante a conferência de um ponto desativado na atividade do banco de dados para aplicação de patches foram corrigidos.
- Correção de vários problemas relacionados a reinicializações repetidas por operações de DDL interrompidas, como `DROP TRIGGER`, `ALTER TABLE` e, especificamente, `ALTER TABLE`, que modificam o tipo de particionamento ou o número de partições em uma tabela.
- Atualizado o valor padrão de `table_open_cache` em instâncias 16XL e 24XL que podem causar failovers repetidos e alta utilização da CPU em classes de instâncias grandes (R4/R5-16XL, R5-12XL, R5-24XL). Foram afetadas as versões 1.21.x e 1.22.x.
- Correção de um problema que fazia com que uma réplica de log binário parasse com um erro `HA_ERR_KEY_NOT_FOUND`.

Integração de correções de bug da edição MySQL community

- Replicação: Enquanto uma instrução `SHOW BINLOG EVENTS` estava sendo executada, todas as transações paralelas foram bloqueadas. A correção garante que o processo `SHOW BINLOG EVENTS` agora só adquira um bloqueio durante o cálculo da posição final do arquivo, portanto, as transações paralelas não são bloqueadas por longas durações. (Erro n.º 76618, erro n.º 20928790)

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-11-24 (versão 1.23.1) (obsoleta)

Versão: 1.23.1

O Aurora MySQL 1.23.1 está disponível para o público. As versões 1.* do Aurora MySQL são compatíveis com o MySQL 5.6 e as versões 2.* do Aurora MySQL são compatíveis com o MySQL 5.7.

Essa versão de mecanismo está programada para ser descontinuada em 28 de fevereiro de 2023. Para obter mais informações, consulte [Preparar para o fim da vida útil do Amazon Aurora, edição compatível com MySQL versão 1](#).

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo pelo console do RDS, pela AWS CLI ou pela API do Amazon RDS.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correções de segurança:

Correções e outras melhorias para ajustar o tratamento em um ambiente gerenciado. Correções adicionais do CVE abaixo:

- [CVE-2020-14559](#)
- [CVE-2020-14539](#)

Alterações incompatíveis:

Esta versão introduz uma alteração de permissão que afeta o comportamento do `mysqldump` comando. Os usuários devem ter o `PROCESS` privilégio para acessar a `INFORMATION_SCHEMA.FILES` tabela. Para executar o `mysqldump` comando sem alterações, conceda o `PROCESS` privilégio ao usuário do banco de dados ao qual o `mysqldump` comando se conecta. Você também pode executar o comando `mysqldump` com a opção `--no-tablespaces`. Com essa opção, a saída `mysqldump` não inclui nenhuma `CREATE LOGFILE GROUP` ou instruções do `CREATE TABLESPACE`. Nesse caso, o comando `mysqldump` não acessa a tabela `INFORMATION_SCHEMA.FILES` e você não precisa conceder a permissão `PROCESS`.

Melhorias de disponibilidade:

- Corrigido um problema que fazia com que uma instância de Aurora leitor em um cluster secundário de banco de dados global com execução 1.23.0 seja reiniciada repetidamente.
- Corrigido um problema em que as réplicas de uma região secundária de banco de dados global podiam ser reiniciadas quando atualizadas para a versão 1.23.0 enquanto o gravador de Região principal estava em uma versão mais antiga.
- Corrigido um vazamento de memória no recurso de redimensionamento dinâmico, introduzido em Aurora MySQL 1.23.0.
- Corrigido um problema que poderia causar a reinicialização do servidor durante a execução de uma consulta usando o recurso de consulta paralela.
- Corrigido um problema que poderia fazer com que uma sessão de cliente travasse quando o mecanismo de banco de dados encontrava um erro ao ler ou gravar na rede.

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-09-02 (versão 1.23.0) (obsoleta)

Versão: 1.23.0

O Aurora MySQL 1.23.0 está disponível para o público. As versões 1.* do Aurora MySQL são compatíveis com o MySQL 5.6 e as versões 2.* do Aurora MySQL são compatíveis com o MySQL 5.7.

Essa versão de mecanismo está programada para ser descontinuada em 28 de fevereiro de 2023. Para obter mais informações, consulte [Preparar para o fim da vida útil do Amazon Aurora, edição compatível com MySQL versão 1](#).

As versões atualmente compatíveis do Aurora MySQL são 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.*.

É possível restaurar o snapshot de um banco de dados Aurora MySQL 1.* para o Aurora MySQL 1.23.0.

Important

As melhorias no Aurora armazenamento nesta versão limitam os caminhos de atualização disponíveis de Aurora MySQL 1.23 para Aurora MySQL 2.*. Quando você atualiza um cluster Aurora MySQL 1.23 para 2.*, você deve atualizar para Aurora MySQL 2.09.0 ou posterior.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo por meio do console do RDS, da CLI AWS ou da API do Amazon RDS.

Note

No momento, essa versão não está disponível nas seguintes regiões: AWS GovCloud (Leste dos EUA) [us-gov-east-1], AWS GovCloud (Oeste dos EUA) [us-gov-west-1]. Haverá um anúncio separado assim que ela for disponibilizada.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Novos recursos:

- Agora você pode ativar ou desativar a consulta paralela para um cluster existente alterando o valor do parâmetro do cluster de banco de dados `aurora_parallel_query`. Não é necessário usar a configuração `parallelquery` para o parâmetro `--engine-mode` ao criar o cluster.

A consulta paralela foi expandida para estar disponível em todas as regiões onde o Aurora MySQL está disponível.

Há uma série de outros aprimoramentos de funcionalidade e alterações nos procedimentos para atualizar e habilitar a consulta paralela em um cluster do Aurora. Para obter mais informações, consulte [Como trabalhar com a consulta paralela do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

- Com essa versão, é possível criar instâncias de banco de dados MySQL do Amazon Aurora com até 128 terabytes (TiB) de armazenamento. O novo limite de armazenamento é um aumento em relação aos 64 TiB anteriores. O tamanho de armazenamento de 128 TiB oferece suporte a bancos de dados maiores. Esse recurso não é compatível com tamanhos de instâncias pequenas (db.t2 ou db.t3). Um único tablespace não pode expandir além de 64 TiB devido às [limitações do InnoDB com tamanho de página de 16 KB](#).

O Aurora alerta quando o tamanho do volume do cluster está próximo de 128 TiB, para que seja possível executar uma ação antes de atingir o limite de tamanho. Os alertas aparecem no log mysql e RDS Events no Console de gerenciamento da AWS.

- Melhoria no processamento de log binário (binlog) para reduzir o tempo de recuperação de falhas e a latência de tempo de confirmação quando transações muito grandes estão envolvidas.
- O Aurora redimensiona dinamicamente o espaço de armazenamento em cluster. Com o redimensionamento dinâmico, o espaço de armazenamento do cluster de bancos de dados Aurora diminui automaticamente ao remover dados do cluster de banco de dados. Para obter mais informações, consulte [Escalabilidade de armazenamento](#) no Guia do usuário do Amazon Aurora.

 Note

O recurso de redimensionamento dinâmico está sendo implantado em fases nas AWS regiões em que o Aurora está disponível. Dependendo da região em que o cluster está, esse atributo pode não estar disponível ainda. Para obter mais informações, consulte [o anúncio de Novidades](#).

Correções de alta prioridade:

- [CVE-2019-2911](#)
- [CVE-2019-2537](#)
- [CVE-2018-2787](#)
- [CVE-2018-2784](#)
- [CVE-2018-2645](#)
- [CVE-2018-2640](#)

Melhorias de disponibilidade:

- Correção de um problema no gerenciador de bloqueio em que uma condição de corrida podia fazer com que um bloqueio fosse compartilhado por duas transações, fazendo com que o banco de dados fosse reiniciado.
- Correção de um problema relacionado ao gerenciamento de memória de bloqueio de transação com transações de gravação de longa duração resultando em uma reinicialização do banco de dados.
- Correção de uma condição de corrida no gerenciador de bloqueio que resultava em uma reinicialização ou um failover do banco de dados durante a reversão da transação.

- Correção de um problema durante a atualização de 5.6 para 5.7 quando o `innodb_file_format` era alterado em uma tabela que tinha o DDL Fast ativado.
- Vários problemas em que o mecanismo podia ser reiniciado durante a aplicação de patches com tempo de inatividade zero durante a conferência de um ponto desativado na atividade do banco de dados para aplicação de patches foram corrigidos.
- Correção de um problema relacionado à recuperação DDL que afeta a reinicialização da instância de banco de dados ao recuperar uma operação `DROP TRIGGER` interrompida.
- Correção de um bug que poderia causar indisponibilidade do banco de dados se ocorresse uma falha durante a execução de determinadas operações de particionamento. Especificamente, uma operação `ALTER TABLE` interrompida que modifica o tipo de particionamento ou o número de partições em uma tabela.
- Corrija o valor padrão de `table_open_cache` em instâncias 16XL e 24XL que podem causar failovers repetidos e alta utilização da CPU em classes de instâncias grandes (R4/R5-16XL, R5-12XL, R5-24XL). Isso afetou as versões 1.21.x e 1.22.x.

Bancos de dados globais:

- Preencha os dados ausentes na visualização do `INFORMATION_SCHEMA.REPLICA_HOST_STATUS` MySQL nas regiões primária e AWS secundária em um banco de dados global Aurora.
- Correção de falhas de consulta inesperadas que poderiam ocorrer em uma região secundária de banco de dados global devido à coleta de lixo de registros UNDO na região principal, após problemas temporários de conectividade de rede entre as regiões principal e secundária.

Consulta paralela:

- Correção de um problema em que uma consulta paralela poderia fazer com que uma consulta de longa duração retornasse um resultado vazio.
- Correção de um problema em que uma consulta em uma pequena tabela na réplica de leitura do Aurora poderia levar mais de um segundo.
- Correção de um problema que poderia causar uma reinicialização quando uma consulta paralela e uma instrução DML estavam sendo executadas simultaneamente sob uma workload pesada.

Melhorias gerais:

- Correção de um problema em que consultas usando índice espacial podiam retornar resultados parciais se o índice espacial fosse criado em tabelas com valores espaciais grandes já existentes.
- Maior comprimento máximo permitido para variáveis de sistema de auditoria `server_audit_incl_users` e `server_audit_excl_users` de 1.024 bytes para 2.000 bytes.
- Correção de um problema em que uma réplica binlog conectada a uma principal de log binário do Aurora MySQL poderia mostrar dados incompletos quando o log binário principal do Aurora MySQL carregava dados do S3 em `statement binlog_format`.
- Para o carregamento de dados, mapear `binlog_format mixed` para `row` em vez de `statement`, a fim de estar em conformidade com o comportamento da comunidade.
- Correção de um problema que fazia com que a replicação de binlog parasse de funcionar quando o usuário fechava a conexão e a sessão estava usando tabelas temporárias.
- Tempo de resposta melhorado de uma consulta envolvendo tabelas temporárias MyISAM.
- Corrija o problema de permissão quando o operador de log binário executa uma função do Lambda nativa.
- Correção de um problema nas réplicas de leitura do Aurora ao tentar consultar ou alternar o log lento ou o log geral.
- Corrigido um problema que interrompia a replicação lógica quando o parâmetro `binlog_checksum` era definido com valores diferentes no mestre e na réplica.
- Corrigido um problema em que a réplica de leitura podia ver resultados parciais transitórios de uma transação confirmada recentemente no gravador.
- Inclua informações da transação revertida em `show engine innodb status` quando um deadlock for resolvido.

Integração de correções de bug da edição MySQL community

- Eventos de binlog com `ALTER TABLE ADD COLUMN ALGORITHM=QUICK` serão regravados `ALGORITHM=DEFAULT` para serem compatíveis com a edição da comunidade.
- ERRO nº 22350047: SE O CLIENTE FOI ENCERRADO APÓS A REVERSÃO PARA OS STMTS ANTERIORES DO PONTO DE SALVAMENTO CONFIRMADO
- Erro nº 29915479: EXECUÇÃO COM_REGISTER_SLAVE SEM COM_BINLOG_DUMP PODE RESULTAR EM SAÍDA DO SERVIDOR
- Erro nº 30441969: BUG #29723340: TRAVAMENTO DO SERVIDOR MYSQL APÓS CONSULTA SQL COM DADOS ?AST

- Erro nº 30628268: TRAVAMENTO POR FALTA DE MEMÓRIA
- Erro nº 27081349: COMPORTAMENTO INESPERADO AO EXCLUIR COM FUNÇÃO ESPACIAL
- Erro nº 27230859: COMPORTAMENTO INESPERADO AO LIDAR COM POLÍGONO INVÁLIDO"
- Erro nº 27081349: COMPORTAMENTO INESPERADO AO EXCLUIR COM ESPACIAL"
- Erro nº 26935001: ALTERAR TABELA AUTO_INCREMENT TENTA LER ÍNDICE DO TABLESPACE DESCARTADO
- Erro nº 29770705: SERVIDOR TRAVOU AO EXECUTAR SELEÇÃO COM CLÁUSULA WHERE ESPECÍFICA
- Erro nº 27659490: SELECIONAR USANDO INTERVALO DINÂMICO E MESCLAGEM DE ÍNDICE USA MUITA MEMÓRIA (OOM)
- Erro nº 24786290: A REPLICAÇÃO É INTERROMPIDA APÓS O Erro nº 74145 OCORRER NO MASTER
- Erro nº 27703912: USO EXCESSIVO DE MEMÓRIA COM MUITA PREPARAÇÃO
- Bug #20527363: TRUNCATE TEMPORARY TABLE CRASH:!! DICT_ TF2 _FLAG_IS_SET (TABELA, DICT_ _TEMPORARY) TF2
- Erro nº 23103937: PS_TRUNCATE_ALL_TABLES() NÃO FUNCIONA NO MODO SUPER_READ_ONLY
- Erro nº 25053286: USAR VISUALIZAÇÃO COM CONDIÇÃO NO PROCEDIMENTO CAUSA COMPORTAMENTO INCORRETO (corrigido em 5.6.36)
- Erro nº 25586773: COMPORTAMENTO INCORRETO PARA CRIAR TABELA SELECT EM UM LOOP EM SP (corrigido em 5.6.39)
- Erro nº 27407480: AUTOMATIC_SP_PRIVILEGES REQUER OS PRIVILÉGIOS DE INSERÇÃO PARA A TABELA MYSQL.USER
- Erro nº 26997096: o valor `relay_log_space` não é atualizado de forma sincronizada, de modo que seu valor é às vezes muito maior do que o espaço em disco real usado pelos logs de retransmissão.
- Erro nº 15831300 SLAVE_TYPE_CONVERSIONS=ALL_NON_LOSSY NÃO ESTÁ FUNCIONANDO CONFORME O ESPERADO
- Erro de backport SSL nº 17087862, erro nº 20551271
- Erro nº 16894092: REGRESSÃO DE PERFORMANCE EM 5.6.6+ PARA INSERIR EM... SELECT ... DE (corrigido em 5.6.15).
- Transfira uma correção de erro relacionada a SLAVE_TYPE_CONVERSIONS.

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-06-03 (versão 1.22.5) (obsoleta)

Versão: 1.22.5

O Aurora MySQL 1.22.5 está disponível para o público. As versões 1.* do Aurora MySQL são compatíveis com o MySQL 5.6 e as versões 2.* do Aurora MySQL são compatíveis com o MySQL 5.7.

Essa versão de mecanismo está programada para ser descontinuada em 28 de fevereiro de 2023. Para obter mais informações, consulte [Preparar para o fim da vida útil do Amazon Aurora, edição compatível com MySQL versão 1](#).

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo pelo console do RDS, pela AWS CLI ou pela API do Amazon RDS.

Note

Esta versão é designada como uma versão de suporte de longo prazo (LTS). Para obter mais informações, consulte [Versões de suporte de longo prazo \(LTS\) do Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Melhorias de disponibilidade:

- Resolvido um problema que poderia fazer com que o banco de dados parasse e, posteriormente, reiniciasse ou executasse um failover devido a um conflito de simultaneidade entre os processos internos de limpeza.
- Resolvido um problema que poderia fazer com que o cluster ficasse indisponível se o banco de dados fosse reiniciado enquanto mantinha transações XA no estado preparado e, em seguida,

reiniciado novamente antes que essas transações fossem confirmadas ou revertidas. Antes dessa correção, você pode resolver o problema restaurando o cluster para um ponto no tempo antes da primeira reinicialização.

- Resolvido um problema que poderia fazer com que a limpeza do InnoDB ficasse bloqueada se o banco de dados fosse reiniciado durante o processamento de uma instrução DDL. Como resultado, o comprimento da lista de histórico do InnoDB aumentaria e o volume de armazenamento do cluster continuaria crescendo até que ele fosse preenchido, tornando o banco de dados indisponível. Antes dessa correção, você pode mitigar o problema reiniciando o banco de dados novamente para desbloquear a expurgação.

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-03-04 (versão 1.22.4) (obsoleta)

Versão: 1.22.4

O Aurora MySQL 1.22.4 está disponível para o público. As versões 1.* do Aurora MySQL são compatíveis com o MySQL 5.6 e as versões 2.* do Aurora MySQL são compatíveis com o MySQL 5.7.

Essa versão de mecanismo está programada para ser descontinuada em 28 de fevereiro de 2023. Para obter mais informações, consulte [Preparar para o fim da vida útil do Amazon Aurora, edição compatível com MySQL versão 1](#).

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo pelo console do RDS, pela AWS CLI ou pela API do Amazon RDS.

Note

Esta versão é designada como uma versão de suporte de longo prazo (LTS). Para obter mais informações, consulte [Versões de suporte de longo prazo \(LTS\) do Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correções de segurança:

Correções e outras melhorias para ajustar o tratamento em um ambiente gerenciado. Correções adicionais do CVE abaixo:

- [CVE-2020-14867](#)
- [CVE-2020-14812](#)
- [CVE-2020-14793](#)
- [CVE-2020-14769](#)
- [CVE-2020-14765](#)
- [CVE-2020-14672](#)
- [CVE-2020-1971](#)

Melhorias de disponibilidade:

- Correção de um problema que poderia acionar uma reinicialização ou um failover do banco de dados durante um comando `kill session`. Se encontrar esse problema, entre em contato com o suporte da AWS para habilitar essa correção em sua instância.
- Melhoria no processamento de log binário para reduzir o tempo de recuperação de falhas e a latência de tempo de commit quando transações muito grandes estão envolvidas.
- Correção de um problema que fazia com que uma réplica de log binário parasse com um erro `HA_ERR_KEY_NOT_FOUND`.

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-11-09 (versão 1.22.3) (obsoleta)

Versão: 1.22.3

O Aurora MySQL 1.22.3 está disponível para o público. As versões 1.* do Aurora MySQL são compatíveis com o MySQL 5.6 e as versões 2.* do Aurora MySQL são compatíveis com o MySQL 5.7.

Essa versão de mecanismo está programada para ser descontinuada em 28 de fevereiro de 2023. Para obter mais informações, consulte [Preparar para o fim da vida útil do Amazon Aurora, edição compatível com MySQL versão 1](#).

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo pelo console do RDS, pela AWS CLI ou pela API do Amazon RDS.

Note

Esta versão é designada como uma versão de suporte de longo prazo (LTS). Para obter mais informações, consulte [Versões de suporte de longo prazo \(LTS\) do Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Em caso de dúvidas ou preocupações, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correções de segurança:

Correções e outras melhorias para ajustar o tratamento em um ambiente gerenciado. Correções adicionais do CVE abaixo:

- [CVE-2020-14559](#)
- [CVE-2020-14539](#)
- [CVE-2020-2579](#)
- [CVE-2020-2812](#)
- [CVE-2020-2780](#)

- [CVE-2020-2763](#)

Alterações incompatíveis:

Esta versão introduz uma alteração de permissão que afeta o comportamento do `mysqldump` comando. Os usuários devem ter o `PROCESS` privilégio para acessar a `INFORMATION_SCHEMA.FILES` tabela. Para executar o `mysqldump` comando sem alterações, conceda o `PROCESS` privilégio ao usuário do banco de dados ao qual o `mysqldump` comando se conecta. Você também pode executar o comando `mysqldump` com a opção `--no-tablespaces`. Com essa opção, a saída `mysqldump` não inclui nenhuma `CREATE LOGFILE GROUP` ou instruções do `CREATE TABLESPACE`. Nesse caso, o comando `mysqldump` não acessa a tabela `INFORMATION_SCHEMA.FILES` e você não precisa conceder a permissão `PROCESS`.

Melhorias de disponibilidade:

- Corrigidos problemas que poderiam causar reinicializações do servidor durante a recuperação de uma instrução DDL que não foi confirmada.
- Corrigidas as condições de corrida no gerenciador de bloqueio que podem gerar uma reinicialização do servidor.
- Correção de um problema que poderia fazer com que o agente de monitoramento reiniciasse o servidor durante a recuperação de uma transação grande

Melhorias gerais:

- Alterado o comportamento para mapear `MIXED binlog_format` para `ROW` em vez de `STATEMENT` ao executar `LOAD DATA FROM INFILE | S3`.
- Corrigido um problema em que uma réplica de binlog conectada a uma primária de binlog do Aurora MySQL poderia mostrar dados incompletos quando a primária executava `LOAD DATA FROM S3` e `binlog_format` estava definido como `STATEMENT`.

Integração de correções de bug da edição MySQL community

- Bug #26654685: Um ID de índice corrompido encontrado durante uma verificação de chave estrangeira gerou uma declaração
- Erro nº 15831300: Por padrão, ao promover inteiros de um tipo menor no mestre para um tipo maior no escravo (por exemplo, de uma coluna [SMALLINT](#) no mestre para uma coluna [BIGINT](#) no

escravo), os valores promovidos são tratados como se fossem assinados. Agora, nesses casos, é possível modificar ou substituir esse comportamento usando um ou ambos de `ALL_SIGNED`, `ALL_UNSIGNED` no conjunto de valores especificados para a variável de sistema de servidor [slave_type_conversions](#) . Para obter mais informações, consulte [Replicação baseada em linha: promoção e rebaixamento de atributos](#), bem como a descrição da variável.

- Ero nº17449901: Com `foreign_key_checks=0`, o InnoDB permitiu que um índice exigido por uma restrição de chave estrangeira fosse descartado, colocando a tabela em um inconsistente e fazendo com que a falha na verificação de chave estrangeira ocorresse na carga da tabela. O InnoDB agora evita descartar um índice exigido por uma restrição de chave estrangeira, mesmo com `foreign_key_checks=0`. A restrição de chave estrangeira deve ser removida antes de soltar o índice de chave estrangeira.
- ERRO N.º 20768847: Um [ALTER TABLE... A operação DROP INDEX](#) em uma tabela com dependências de chave estrangeira levantou uma declaração.

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-03-05 (versão 1.22.2) (obsoleta)

Versão: 1.22.2

O Aurora MySQL 1.22.2 está disponível para o público. As versões 1.* do Aurora MySQL são compatíveis com o MySQL 5.6 e as versões 2.* do Aurora MySQL são compatíveis com o MySQL 5.7.

Essa versão de mecanismo está programada para ser descontinuada em 28 de fevereiro de 2023. Para obter mais informações, consulte [Preparar para o fim da vida útil do Amazon Aurora, edição compatível com MySQL versão 1](#).

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo por meio do console do RDS, da CLI AWS ou da API do Amazon RDS.

Note

No momento, essa versão não está disponível nas seguintes regiões: AWS GovCloud (Leste dos EUA) [us-gov-east-1], AWS GovCloud (Oeste dos EUA) [us-gov-west-1]. Haverá um anúncio separado assim que ela for disponibilizada.

Esta versão é designada como uma versão de suporte de longo prazo (LTS). Para obter mais informações, consulte [Versões de suporte de longo prazo \(LTS\) do Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correções de alta prioridade:

- Corrigido um problema de falhas de conexão intermitentes após a rotação do certificado.
- Corrigido um problema que fazia com que a clonagem demorasse mais tempo em alguns clusters de banco de dados com altas cargas de gravação.
- Corrigido um problema que interrompia a replicação lógica quando o parâmetro `binlog_checksum` era definido com valores diferentes no mestre e na réplica.
- Corrigido um problema em que o log lento e o log geral não giravam corretamente nas réplicas de leitura.
- Corrigido um problema com o comportamento do nível de isolamento de leitura confirmada do ANSI.

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-12-23 (versão 1.22.1) (obsoleta)

Versão: 1.22.1

O Aurora MySQL 1.22.1 está disponível para o público. As versões 1.* do Aurora MySQL são compatíveis com o MySQL 5.6 e as versões 2.* do Aurora MySQL são compatíveis com o MySQL 5.7.

Essa versão de mecanismo está programada para ser descontinuada em 28 de fevereiro de 2023. Para obter mais informações, consulte [Preparar para o fim da vida útil do Amazon Aurora, edição compatível com MySQL versão 1](#).

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo por meio da API RDS AWS CLI ou da API do Console de gerenciamento da AWS RDS. Você tem a opção de atualizar os clusters de bancos de dados Aurora MySQL 1.* existentes para o Aurora MySQL 1.22.1.

Note

No momento, essa versão não está disponível nas seguintes AWS regiões: AWS GovCloud (Leste dos EUA) [us-gov-east-1], (Oeste dos EUA) [-1], China AWS GovCloud (Ningxia) [us-gov-westcn-northwest-1], Ásia-Pacífico (Hong Kong) [ap-east-1] e Oriente Médio (Bahrein) [me-south-1]. Haverá um anúncio separado assim que ela for disponibilizada.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

O procedimento para atualizar o cluster de banco de dados mudou. Para obter mais informações, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correções críticas:

- Corrigidos problemas que impediam a recuperação do mecanismo envolvendo bloqueios de tabela e tabelas temporárias.
- Melhorada a estabilidade do log binário quando tabelas temporárias são usadas.

Correções de alta prioridade:

- Corrigido um vazamento de memória lento no subsistema de rastreamento e registro em log de banco de dados do Aurora específico que reduz a memória disponível.

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-11-25 (versão 1.22.0) (obsoleta)

Versão: 1.22.0

O Aurora MySQL 1.22.0 está disponível para o público. As versões 1.* do Aurora MySQL são compatíveis com o MySQL 5.6 e as versões 2.* do Aurora MySQL são compatíveis com o MySQL 5.7.

Essa versão de mecanismo está programada para ser descontinuada em 28 de fevereiro de 2023. Para obter mais informações, consulte [Preparar para o fim da vida útil do Amazon Aurora, edição compatível com MySQL versão 1](#).

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo por meio da API RDS AWS CLI ou da API do Console de gerenciamento da AWS RDS. Você tem a opção de atualizar os clusters de bancos de dados Aurora MySQL 1.* existentes para o Aurora MySQL 1.22.0.

Note

No momento, essa versão não está disponível nas seguintes AWS regiões: AWS GovCloud (Leste dos EUA) [us-gov-east-1], (Oeste dos EUA) [-1], China AWS GovCloud (Ningxia) [us-gov-westcn-northwest-1], Ásia-Pacífico (Hong Kong) [ap-east-1], Oriente Médio (Bahrein) [me-south-1] e América do Sul (São Paulo) [sa-east-1] -east-1]. Haverá um anúncio separado assim que ela for disponibilizada.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

 Note

O procedimento para atualizar o cluster de banco de dados mudou. Para obter mais informações, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Novos recursos:

- Os clusters do Aurora MySQL já oferecem suporte para os tipos de instância r5.8xlarge, r5.16xlarge e r5.24xlarge.
- O log binário tem novas melhorias para atenuar a latência do tempo de confirmação quando há transações muito grandes envolvidas.
- O Aurora MySQL agora tem um mecanismo para reduzir a janela de tempo durante a qual os eventos de uma transação grande são gravados no log binário na confirmação. Ele impede com eficácia a recuperação offline demorada que ocorre quando o banco de dados trava durante essa janela de tempo. Esse recurso também corrige o problema que ocorre quando uma transação grande bloqueia transações pequenas na confirmação do log binário. Esse recurso é desativado por padrão e poderá ser habilitado pela equipe de manutenção se for necessário para sua workload. Quando habilitado, ele será acionado quando o tamanho de uma transação for > 500 MB.
- Adicionamos suporte para o nível de isolamento READ COMMITTED ANSI nas réplicas de leitura. Esse nível de isolamento permite que as consultas de longa execução na réplica de leitura sejam executadas sem afetar a alta taxa de transferência das gravações no nó do gravador. Para obter mais informações, consulte [Níveis de isolamento do Aurora MySQL](#).
- Os bancos de dados globais agora permitem adicionar regiões secundárias de réplica somente para leitura para clusters de banco de dados implantados nessas AWS regiões: regiões: Leste dos EUA (Norte da Virgínia) [us-east-1], Leste dos EUA (Ohio) [us-east-2], Oeste dos EUA (Norte da Califórnia) [us-west-1], Oeste dos EUA (Oregon) [us-west-2], Europa (Irlanda) [eu-west-1] Europa (Londres) [eu-west-2], Europa (Paris) [eu-west-3], Ásia-Pacífico (Tóquio) [ap-northeast-1], Ásia-Pacífico (Seul) [ap-northeast-2], Ásia-Pacífico (Seul) [ap-northeast-2], Ásia-Pacífico (Cingapura) [ap-southeast-1] Ásia-Pacífico (Sydney) [ap-southeast-2], Canadá (Central) [ca-central-1], Europa (Frankfurt) [eu-central-1] e Ásia-Pacífico (Mumbai) [ap-south-1].

- O recurso disputa de linhas dinâmica já está disponível para o público e não exige o modo de laboratório do Aurora definido como ativado. Esse recurso melhora substancialmente as taxas de transferência para workloads com muitas transações disputando linhas na mesma página.
- Esta versão tem arquivos de fuso horário atualizados para oferecer suporte à atualização de fuso horário do Brasil mais recente para novos clusters.

Correções críticas:

- [CVE-2019-2922](#)
- [CVE-2019-2923](#)
- [CVE-2019-2924](#)
- [CVE-2019-2910](#)

Correções de alta prioridade:

- [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](#)
- Corrigimos um problema no componente de recuperação DDL que resultava em um tempo de inatividade prolongado do banco de dados. Os clusters que se tornarem indisponíveis após a execução da consulta `TRUNCATE TABLE` em uma tabela com a coluna `AUTO_INCREMENT` deverão ser atualizados.
- Corrigimos um problema no componente de recuperação DDL que resultava em um tempo de inatividade prolongado do banco de dados. Os clusters que se tornarem indisponíveis após a execução da consulta `DROP TABLE` em várias tabelas em paralelo deverão ser atualizados.

Correções de estabilidade geral:

- Corrigimos um problema que fazia com que as réplicas de leitura fossem inicializadas durante uma transação de execução longa. Os clientes que observarem a reinicialização de réplicas com uma queda acelerada na memória disponível deverão considerar a atualização para esta versão.
- Corrigimos um problema que relatava incorretamente ERROR 1836 quando uma consulta aninhada era executada em uma tabela temporária na réplica de leitura.
- Corrigimos um erro de anulação de consulta paralela em uma instância de leitor do Aurora enquanto uma workload de gravação intensa estava em execução na instância de gravador do Aurora.
- Corrigimos um problema que faz com que um banco de dados configurado como mestre do log binário seja reiniciado durante a execução de uma workload de gravação intensa.
- Corrigimos um problema de indisponibilidade prolongada na reinicialização do mecanismo. Isso resolve o problema na inicialização do grupo de buffers. Esse problema ocorre raramente, mas pode afetar qualquer versão compatível.
- Corrigimos um problema que gerava dados inconsistentes na tabela `information_schema.replica_host_status`.
- Corrigimos um comportamento de disputa entre a consulta paralela e os caminhos de execução padrão que fazia com que os nós do leitor fossem reiniciados de forma intermitente.
- Melhoramos a estabilidade do banco de dados quando o número de conexões do cliente excede o valor do parâmetro `max_connections`.
- Melhoramos a estabilidade das instâncias de leitor bloqueando o DDL incompatível e as consultas `LOAD FROM S3`.

Integração de correções de bug da edição MySQL community

- Bug nº 16346241 - SERVER CRASH IN ITEM_PARAM::QUERY_VAL_STR
- Bug nº 17733850 - NAME_CONST() CRASH IN ITEM_NAME_CONST::ITEM_NAME_CONST()
- Bug nº 20989615 - INNODB AUTO_INCREMENT PRODUCES SAME VALUE TWICE
- Erro nº 20181776 – O CONTROLE DE ACESSO NÃO CORRESPONDE AO HOST MAIS ESPECÍFICO QUANDO CONTÉM CARACTERES CURINGA
- Bug #27326796 - FALHA NO MYSQL COM FALHA DE ASSERÇÃO DO INNODB NO ARQUIVO PARS.CC PARS0
- Bug nº 20590013 - IF YOU HAVE A FULLTEXT INDEX AND DROP IT YOU CAN NO LONGER PERFORM ONLINE DDL

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-11-25 (versão 1.21.0) (obsoleta)

Versão: 1.21.0

O Aurora MySQL 1.21.0 está disponível para o público. As versões 1.* do Aurora MySQL são compatíveis com o MySQL 5.6 e as versões 2.* do Aurora MySQL são compatíveis com o MySQL 5.7.

As versões atualmente compatíveis do Aurora MySQL são 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.*. Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo por meio da API RDS AWS CLI ou da API do Console de gerenciamento da AWS RDS. Você tem a opção de atualizar os clusters de bancos de dados Aurora MySQL 1.* existentes para o Aurora MySQL 1.21.0.

Note

No momento, essa versão não está disponível nas seguintes AWS regiões: AWS GovCloud (Leste dos EUA) [-1], (Oeste dos EUA) [us-gov-east-1], China AWS GovCloud (Ningxia) [us-gov-westcn-northwest-1], Ásia-Pacífico (Hong Kong) [ap-east-1], Europa (Estocolmo) [eu-north-1] e Oriente Médio (Bahrein) [me-south-1] -south-1]. Haverá um anúncio separado assim que ela for disponibilizada.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

O procedimento para atualizar o cluster de banco de dados mudou. Para obter mais informações, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correções críticas:

- [CVE-2018-0734](#)
- [CVE-2019-2534](#)
- [CVE-2018-2612](#)
- [CVE-2017-3599](#)
- [CVE-2018-2562](#)
- [CVE-2017-3329](#)
- [CVE-2018-2696](#)
- [CVE-2015-4737](#)

Correções de alta prioridade:

- É altamente recomendável que os clientes com tamanho de banco de dados próximo a 64 terabytes (TiB) atualizem para esta versão a fim de evitar tempo de inatividade causado por erros de estabilidade que afetam volumes próximos ao limite de armazenamento do Aurora.

Correções de estabilidade geral:

- Correção de um erro de anulação de consulta em instâncias de leitor do Aurora enquanto uma workload de leitura pesada estava em execução na instância de gravador do Aurora.
- Correção de um problema em instâncias de leitor do Aurora que reduziam a memória livre durante transações de longa execução enquanto havia um tráfego de confirmação de transação pesado na instância de gravador.
- Agora o valor do parâmetro `aurora_disable_hash_join` persiste após a reinicialização do banco de dados ou a substituição do host.
- Correção de um problema relacionado ao cache da pesquisa de texto completo que fazia com que a instância do Aurora ficasse sem memória. Os clientes que usam pesquisa de texto completo devem realizar a atualização.
- Melhoria na estabilidade do banco de dados quando o recurso de junção de hash está habilitado e a instância está com pouca memória. Os clientes que usam junção de hash devem realizar a atualização.
- Correção de um problema no cache de consulta em que o erro "Muitas conexões" poderia causar uma reinicialização.
- Corrigimos o cálculo de memória livre em instâncias T2 para que seja incluído o espaço de memória de troca a fim de evitar reinicializações desnecessárias.

Integração de correções de bug da edição MySQL community

- Bug #19929406: `HANDLE_FATAL_SIGNAL (SIG=11)` EM `__MEMMOVE__BACK FROM STRING: :COPY SSSE3`
- Bug #17059925: para instruções [UNION](#), o valor das linhas examinadas era calculado de maneira incorreta. Isso era manifestado como valores grandes demais para a coluna `ROWS_EXAMINED` das tabelas de instrução do Performance Schema (como [events_statements_current](#)).
- Bug #11827369: algumas consultas com subconsultas aninhadas `SELECT ... FROM DUAL` geravam uma afirmação.
- Bug nº 16311231: resultados incorretos eram retornados se uma consulta contivesse uma subconsulta em uma cláusula `IN`, que contivesse uma operação [XOR](#) na cláusula `WHERE`.

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-03-05 (versão 1.20.1) (obsoleta)

Versão: 1.20.1

O Aurora MySQL 1.20.1 está disponível para o público. As versões 1.* do Aurora MySQL são compatíveis com o MySQL 5.6 e as versões 2.* do Aurora MySQL são compatíveis com o MySQL 5.7.

As versões atualmente compatíveis do Aurora MySQL são 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.*. Você pode restaurar o snapshot de um banco de dados do Aurora MySQL 1.* para o Aurora MySQL 1.20.1.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo por meio do console do RDS, da CLI AWS ou da API do Amazon RDS.

Note

No momento, essa versão não está disponível nas seguintes regiões: AWS GovCloud (Leste dos EUA) [us-gov-east-1], AWS GovCloud (Oeste dos EUA) [us-gov-west-1]. Haverá um anúncio separado assim que ela for disponibilizada.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correções de alta prioridade:

- Corrigido um problema de falhas de conexão intermitentes após a rotação do certificado.
- Corrigido um problema relacionado à simultaneidade de fechamento de conexão que resultava em um failover sob workload pesada.

Correções de estabilidade geral:

- Corrigida uma falha durante a execução de uma consulta complexa envolvendo junções de várias tabelas e agregações que usam tabelas intermediárias internamente.

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-11-11 (versão 1.20.0) (obsoleta)

Versão: 1.20.0

O Aurora MySQL 1.20.0 está disponível para o público. As versões 1.* do Aurora MySQL são compatíveis com o MySQL 5.6 e as versões 2.* do Aurora MySQL são compatíveis com o MySQL 5.7.

As versões atualmente compatíveis do Aurora MySQL são 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.*, 1.20.*, 2.01.*, 2.02.*, 2.03.* e 2.04.*. Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo por meio da API RDS AWS CLI ou da API do Console de gerenciamento da AWS RDS. Você tem a opção de atualizar os clusters de bancos de dados Aurora MySQL 1.* existentes, até o 1.19.5, para o Aurora MySQL 1.20.0.

Note

No momento, essa versão não está disponível nas seguintes AWS regiões: AWS GovCloud (Leste dos EUA) [-1], (Oeste dos EUA) [us-gov-east-1], China AWS GovCloud (Ningxia) [us-gov-westcn-northwest-1], Ásia-Pacífico (Hong Kong) [ap-east-1], Europa (Estocolmo) [eu-

north-1] e Oriente Médio (Bahrein) [me-south-1] -south-1]. Haverá um anúncio separado assim que ela for disponibilizada.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

 Note

O procedimento para atualizar o cluster de banco de dados mudou. Para obter mais informações, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correções críticas:

- [CVE-2018-0734](#)
- [CVE-2019-2534](#)
- [CVE-2018-2612](#)
- [CVE-2017-3599](#)
- [CVE-2018-2562](#)
- [CVE-2017-3329](#)
- [CVE-2018-2696](#)
- [CVE-2015-4737](#)

Correções de alta prioridade:

- É altamente recomendável que os clientes com tamanho de banco de dados próximo a 64 terabytes (TiB) atualizem para esta versão a fim de evitar tempo de inatividade causado por erros de estabilidade que afetam volumes próximos ao limite de armazenamento do Aurora.

Correções de estabilidade geral:

- Correção de um erro de anulação de consulta em instâncias de leitor do Aurora enquanto uma workload de leitura pesada estava em execução na instância de gravador do Aurora.
- Correção de um problema em instâncias de leitor do Aurora que reduziam a memória livre durante transações de longa execução enquanto havia um tráfego de confirmação de transação pesado na instância de gravador.
- Agora o valor do parâmetro `aurora_disable_hash_join` persiste após a reinicialização do banco de dados ou a substituição do host.
- Correção de um problema relacionado ao cache da pesquisa de texto completo que fazia com que a instância do Aurora ficasse sem memória. Os clientes que usam pesquisa de texto completo devem realizar a atualização.
- Melhoria na estabilidade do banco de dados quando o recurso de junção de hash está habilitado e a instância está com pouca memória. Os clientes que usam junção de hash devem realizar a atualização.
- Correção de um problema no cache de consulta em que o erro "Muitas conexões" poderia causar uma reinicialização.
- Corrigimos o cálculo de memória livre em instâncias T2 para que seja incluído o espaço de memória de troca a fim de evitar reinicializações desnecessárias.

Integração de correções de bug da edição MySQL community

- Bug #19929406: `HANDLE_FATAL_SIGNAL (SIG=11) EM __MEMMOVE__BACK FROM STRING: :COPY SSSE3`
- Bug #17059925: para instruções [UNION](#), o valor das linhas examinadas era calculado de maneira incorreta. Isso era manifestado como valores grandes demais para a coluna `ROWS_EXAMINED` das tabelas de instrução do Performance Schema (como [events_statements_current](#)).
- Bug #11827369: algumas consultas com subconsultas aninhadas `SELECT ... FROM DUAL` geravam uma afirmação.
- Bug nº 16311231: resultados incorretos eram retornados se uma consulta contivesse uma subconsulta em uma cláusula `IN`, que contivesse uma operação [XOR](#) na cláusula `WHERE`.

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-03-05 (versão 1.19.6) (obsoleta)

Versão: 1.19.6

O Aurora MySQL 1.19.6 está disponível para o público. As versões 1.* do Aurora MySQL são compatíveis com o MySQL 5.6 e as versões 2.* do Aurora MySQL são compatíveis com o MySQL 5.7.

Essa versão de mecanismo está programada para ser descontinuada em 28 de fevereiro de 2023. Para obter mais informações, consulte [Preparar para o fim da vida útil do Amazon Aurora, edição compatível com MySQL versão 1](#).

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Você pode restaurar o snapshot de um banco de dados do Aurora MySQL 1.* para o Aurora MySQL 1.19.6.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo por meio do console do RDS, da CLI AWS ou da API do Amazon RDS.

Note

No momento, essa versão não está disponível nas seguintes regiões: AWS GovCloud (Leste dos EUA) [us-gov-east-1], AWS GovCloud (Oeste dos EUA) [us-gov-west-1]. Haverá um anúncio separado assim que ela for disponibilizada.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correções de alta prioridade:

- Corrigido um problema de falhas de conexão intermitentes após a rotação do certificado.

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-09-19 (versão 1.19.5) (obsoleta)

Versão: 1.19.5

O Aurora MySQL 1.19.5 está disponível para o público. As versões 1.* do Aurora MySQL são compatíveis com o MySQL 5.6 e as versões 2.* do Aurora MySQL são compatíveis com o MySQL 5.7.

Essa versão de mecanismo está programada para ser descontinuada em 28 de fevereiro de 2023. Para obter mais informações, consulte [Preparar para o fim da vida útil do Amazon Aurora, edição compatível com MySQL versão 1](#).

As versões atualmente compatíveis do Aurora MySQL são 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.*.

Você tem a opção de atualizar os clusters de banco de dados existentes para o Aurora MySQL 1.19.5. É possível restaurar snapshots do Aurora MySQL 1.14.*, 1.15.*, 1.16.*, 1.17.*, 1.18.*, 1.19.1 e 1.19.2 para o Aurora MySQL 1.19.5.

Para usar uma versão mais antiga do Aurora MySQL, você pode criar novos clusters de banco de dados especificando a versão do mecanismo por meio da API Console de gerenciamento da AWS, da ou da API do RDS AWS CLI.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

No momento, essa versão não está disponível nas seguintes AWS regiões: Europa (Londres) [eu-west-2] AWS GovCloud , (Leste dos EUA) [-1], (Oeste dos EUA) us-gov-east [-1] AWS GovCloud , China (Ningxia) us-gov-west [cn-northwest-1] e Ásia-Pacífico (Hong Kong) [ap-east-1]. Haverá um anúncio separado assim que ela for disponibilizada.

 Note

O procedimento para atualizar o cluster de banco de dados mudou. Para obter mais informações, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

- Correção de um problema em instâncias de leitor do Aurora que reduziam a memória livre durante transações de longa execução enquanto havia um tráfego de confirmação de transação pesado na instância de gravador.
- Correção de um erro de anulação de consulta em instâncias de leitor do Aurora enquanto uma workload de leitura pesada estava em execução na instância de gravador do Aurora.
- Agora o valor do parâmetro `aurora_disable_hash_join` persiste após a reinicialização do banco de dados ou a substituição do host.
- Correção de um problema relacionado ao cache da pesquisa de texto completo que fazia com que a instância do Aurora ficasse sem memória.
- Melhoria na estabilidade do banco de dados quando o tamanho do volume se aproxima do limite de volume de 64 terabytes (TiB) com a reserva de 160 GB de espaço para que o fluxo de trabalho de recuperação seja concluído sem um failover.
- Melhoria na estabilidade do banco de dados quando o recurso de junção de hash está habilitado e a instância está com pouca memória.
- Correção do cálculo de memória livre para que seja incluído o espaço de memória de troca em instâncias T2 que fazia com que elas fossem reinicializadas de forma prematura.
- Correção de um problema no cache de consulta em que o erro "Muitas conexões" poderia causar uma reinicialização.

Integração de correções de bug da edição MySQL community

- [CVE-2018-2696](#)
- [CVE-2015-4737](#)
- Bug #19929406: `HANDLE_FATAL_SIGNAL (SIG=11) EM __MEMMOVE__ _BACK FROM STRING: :COPY SSSE3`

- Bug #17059925: para instruções [UNION](#), o valor das linhas examinadas era calculado de maneira incorreta. Isso era manifestado como valores grandes demais para a coluna ROWS_EXAMINED das tabelas de instrução do Performance Schema (como [events_statements_current](#)).
- Bug #11827369: algumas consultas com subconsultas aninhadas `SELECT ... FROM DUAL` geravam uma afirmação.
- Bug nº 16311231: resultados incorretos eram retornados se uma consulta contivesse uma subconsulta em uma cláusula `IN`, que contivesse uma operação [XOR](#) na cláusula `WHERE`.

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-06-05 (versão 1.19.2) (obsoleta)

Versão: 1.19.2

O Aurora MySQL 1.19.2 está disponível para o público. Todos os novos clusters de bancos de dados Aurora MySQL com compatibilidade com o MySQL 5.6, inclusive os restaurados de snapshots, serão criados com 1.17.8, 1.19.0, 1.19.1 ou 1.19.2. Você tem a opção, mas não a obrigatoriedade, de atualizar os clusters de banco de dados existentes para o Aurora MySQL 1.19.2. Para usar uma versão mais antiga, você pode criar novos clusters de banco de dados no Aurora MySQL 1.14.4, Aurora MySQL 1.15.1, Aurora MySQL 1.16, Aurora MySQL 1.17.8 ou Aurora MySQL 1.18. Você pode fazer isso usando a API do Amazon RDS AWS CLI ou especificando a versão do mecanismo.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

Atualmente, essa versão não está disponível nas regiões AWS GovCloud (Oeste dos EUA) [us-gov-west-1], Europa (Estocolmo) [eu-north-1], China (Ningxia) [cn-northwest-1] e Ásia-Pacífico (Hong Kong) [ap-east-1]. AWS Haverá um anúncio separado assim que ela for disponibilizada.

 Note

O procedimento para atualizar o cluster de banco de dados mudou. Para obter mais informações, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

- Correção de um problema que poderia causar falhas ao carregar dados no Aurora do Amazon S3.
- Correção de um problema que poderia causar falhas ao carregar dados do Aurora para o Amazon S3.
- Correção de um problema que criou sessões zumbi em um estado eliminado.
- Correção de um problema que causava conexões anuladas ao solucionar um erro no gerenciamento do protocolo de rede.
- Correção de um problema que poderia causar uma falha ao processar tabelas particionadas.
- Correção de um problema relacionado à replicação do log binário de criação de gatilho.

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-05-09 (versão 1.19.1) (obsoleta)

Versão: 1.19.1

O Aurora MySQL 1.19.1 está disponível para o público. Todos os novos clusters de bancos de dados Aurora MySQL com compatibilidade com o MySQL 5.6, inclusive os restaurados de snapshots, podem ser criados com o 1.17.8, 1.19.0 ou 1.19.1. Você tem a opção, mas não a obrigatoriedade, de atualizar os clusters de banco de dados existentes para o Aurora MySQL 1.19.1. Para usar uma versão mais antiga, você pode criar novos clusters de banco de dados no Aurora MySQL 1.14.4, Aurora MySQL 1.15.1, Aurora MySQL 1.16, Aurora MySQL 1.17.8 ou Aurora MySQL 1.18. Você pode fazer isso usando a API do Amazon RDS AWS CLI ou especificando a versão do mecanismo.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

 Note

No momento, essa versão não está disponível nas regiões AWS GovCloud (Oeste dos EUA) [us-gov-west-1] e China (Pequim) [cn-north-1]. Haverá um anúncio separado assim que ela for disponibilizada.

 Note

O procedimento para atualizar o cluster de banco de dados mudou. Para obter mais informações, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

- Corrigido um erro na replicação do log binário que pode provocar um problema em instâncias do Aurora configuradas como operador do log binário.
- Corrigido um erro ao tratar determinados tipos de comando ALTER TABLE.
- Corrigido um problema com conexões anuladas devido a um erro no gerenciamento do protocolo da rede.

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-02-07 (versão 1.19.0) (obsoleta)

Versão: 1.19.0

O Aurora MySQL 1.19.0 está disponível para o público. Todos os novos clusters de bancos de dados Aurora MySQL com compatibilidade com o MySQL 5.6, inclusive os restaurados de snapshots, serão criados com 1.17.8 ou 1.19.0. Você tem a opção, mas não a obrigatoriedade, de atualizar os clusters de banco de dados existentes para o Aurora MySQL 1.19.0. Para usar uma versão mais antiga, você pode criar novos clusters de banco de dados no Aurora MySQL 1.14.4, Aurora MySQL 1.15.1, Aurora MySQL 1.16, Aurora MySQL 1.17.8 ou Aurora MySQL 1.18.0. Você pode fazer isso usando a API do Amazon RDS AWS CLI ou especificando a versão do mecanismo.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

 Note

No momento, essa versão não está disponível nas regiões AWS GovCloud (Oeste dos EUA) [us-gov-west-1] e China (Pequim) [cn-north-1]. Haverá um anúncio separado assim que ela for disponibilizada.

 Note

O procedimento para atualizar o cluster de banco de dados mudou. Para obter mais informações, consulte [Atualizando a versão secundária ou o nível de patch de um cluster de banco de dados de Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Recursos

- Seletor de versão do Aurora - a partir do Aurora MySQL 1.19.0, você pode escolher entre várias versões do MySQL 5.6 compatíveis com o Aurora no console do Amazon RDS. Para obter mais informações, consulte [Verificação ou especificação de versões de mecanismo do Aurora MySQL por meio da AWS](#) no Guia do usuário do Amazon Aurora.

Melhorias

- Corrigido um problema de estabilidade relacionado à consulta CHECK TABLE em uma réplica do Aurora.
- Introduzida uma nova variável de usuário global `aurora_disable_hash_join` para desabilitar junção hash.
- Corrigido um problema de estabilidade ao gerar a linha de saída durante junção hash de várias tabelas.
- Corrigido um problema que retornava um resultado incorreto devido a uma alteração de plano durante a verificação da aplicabilidade de junção hash.

- A Aplicação de patch com tempo de inatividade zero é compatível com transações de execução longa. Essa melhoria entrará em vigor ao atualizar da versão 1.19 para uma superior.
- A aplicação de patch com tempo de inatividade zero agora é compatível quando o log binário está habilitado. Essa melhoria entrará em vigor ao atualizar da versão 1.19 para uma superior.
- Corrigido um problema que provocava um pico de utilização de CPU na réplica do Aurora não relacionado à workload.
- Corrigida uma condição de corrida no gerenciador de bloqueio que resultava em uma reinicialização do banco de dados.
- Corrigida uma condição de corrida no componente do gerenciador de bloqueio para melhorar a estabilidade de instâncias do Aurora.
- Estabilidade do detector de deadlock melhorada dentro do componente do gerenciador de bloqueio.
- INSERTA operação em uma tabela será proibida, se o InnoDB detectar que o índice está corrompido.
- Corrigido um problema de estabilidade no Fast DDL.
- A estabilidade do Aurora foi melhorada reduzindo o consumo de memória em lotes de verificação para subconsulta de linha única.
- Corrigido um problema de estabilidade que ocorria depois que uma chave estrangeira era descartada enquanto a variável do sistema `foreign_key_checks` estava definida como 0.
- Corrigido um problema no recurso de Impedimento de memória insuficiente que substituíria incorretamente alterações no valor de `table_definition_cache` feitas pelo usuário.
- Corrigidos problemas de estabilidade no recurso de Impedimento de memória insuficiente.
- Corrigido um problema que definia `query_time` e `lock_time` em `slow_query_log` como valores de lixo.
- Corrigido um problema de estabilidade de consulta paralela disparado pelo tratamento inadequado de agrupamento de strings internamente.
- Corrigido um problema de estabilidade de consulta paralela disparado por uma pesquisa de índice secundário.
- Corrigido um problema de estabilidade de consulta paralela disparado por uma atualização de várias tabelas.

Integração de correções de bug da edição MySQL community

- BUG N° 32917: DETECTAR ARQUIVOS DE GRUPOS TEMPORÁRIOS ÓRFÃOS E TRATAR COM TRANQUILIDADE
- BUG N° 63144: BLOQUEIO DE METADADOS DO COMANDO CREATE TABLE IF NOT EXISTS É MUITO RESTRITIVO

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-09-20 (versão 1.18.0) (obsoleta)

Versão: 1.18.0

O Aurora MySQL 1.18.0 está disponível para o público. Todos os novos clusters de consulta paralela do Aurora MySQL compatíveis com o MySQL 5.6, inclusive os restaurados de snapshots, serão criados no Aurora MySQL 1.18.0. Você tem a opção, mas não a obrigatoriedade, de atualizar os clusters de consulta paralela existentes para o Aurora MySQL 1.18.0. É possível criar clusters de banco de dados no Aurora MySQL 1.14.4, Aurora MySQL 1.15.1, Aurora MySQL 1.16 ou Aurora MySQL 1.17.6. Você pode fazer isso usando a API do Amazon RDS AWS CLI ou especificando a versão do mecanismo.

Com a versão 1.18.0 do Aurora MySQL, estamos usando um modelo de aplicação de patch de cluster em que todos os nós em um cluster de bancos de dados Aurora recebem patch ao mesmo tempo.

Important

O Aurora MySQL 1.18.0 só se aplica a clusters de consulta paralela do Aurora. Se você atualizar um cluster 5.6.10a provisionado, a versão resultante será 1.17.8. Se você atualizar um cluster de consulta paralela 5.6.10a, a versão resultante será 1.18.0.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Recursos

- A consulta paralela está disponível com esta versão para novos clusters e snapshots restaurados. A consulta paralela do Aurora MySQL é uma otimização que paraleliza parte da computação envolvida no processamento de consultas com uso intenso de I/O dados. O trabalho que é paralelizado inclui linhas de recuperação de armazenamento, extração de valores de colunas e determinação de quais linhas correspondem às condições na cláusula WHERE e em cláusulas de junção. O trabalho com uso intensivo de dados é delegado (em termos de otimização de banco de dados, empurrados para baixo) para vários nós na camada de armazenamento distribuído do Aurora. Sem as consultas paralelas, cada consulta traz todos os dados de varredura para um nó único dentro do cluster do Aurora MySQL (o nó de cabeçalho) e realiza todo o processamento de consultas lá.
- Quando o recurso de consulta paralela está habilitado, o mecanismo do Aurora MySQL determina automaticamente quando as consultas podem se beneficiar, sem exigir alterações de SQL como avisos ou atributos de tabela.

Para obter mais informações, consulte [Como trabalhar com a consulta paralela do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

- OOM Avoidance: este recurso monitora a memória do sistema e acompanha a memória consumida por diversos componentes do banco de dados. Quando fica sem memória, o sistema realiza uma lista de ações para liberá-la em diversos componentes acompanhados em uma tentativa de evitar que o banco de dados fique sem Out of Memory (OOM – Sem memória) e, assim, evitar uma reinicialização do banco de dados. Esse recurso permanece habilitado por padrão para instâncias t2 e pode ser habilitado em outras classes de instância por meio de um novo parâmetro de instância chamado `aurora_oom_response`. O parâmetro de instância recebe uma string de ações separadas por vírgula que uma instância de banco de dados deve realizar quando a memória está baixa. Entre as ações válidas estão "print", "tune", "decline", "kill_query" ou qualquer combinação delas. Qualquer string vazia significa que não há ações a serem executadas e desabilita o recurso. As ações padrão para o recurso é "print, tune". Exemplos de uso:
 - "print" – Só imprime as consultas que utilizam muita memória.
 - "tune" – Ajusta os caches de tabela internos para liberar parte de memória novamente ao sistema.
 - "decline" – recusa novas consultas quando a instância tem pouca memória.
 - "kill_query" – Encerra as consultas em ordem decrescente de consumo da memória até a memória da instância superar o limite mínimo. As instruções Data Definition Language (DDL) não são encerradas.

- "print, tune" – Realiza ações descritas para "print" e "tune".
- "tune, decline, kill_query" – realiza as ações descritas para "tune", "decline" e "kill_query".

Para obter informações sobre out-of-memory condições de tratamento e outros conselhos sobre solução de problemas, consulte [Problemas de falta de memória do Amazon Aurora MySQL no Guia do usuário do](#) Amazon Aurora.

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-03-05 (versão 1.17.9) (obsoleta)

Versão: 1.17.9

O Aurora MySQL 1.17.9 está disponível para o público. As versões 1.* do Aurora MySQL são compatíveis com o MySQL 5.6 e as versões 2.* do Aurora MySQL são compatíveis com o MySQL 5.7.

As versões atualmente compatíveis do Aurora MySQL são 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.*. Você pode restaurar o snapshot de um banco de dados do Aurora MySQL 1.* para o Aurora MySQL 1.17.9.

Para criar um cluster com uma versão mais antiga do Aurora MySQL, especifique a versão do mecanismo por meio do console do RDS, da CLI AWS ou da API do Amazon RDS.

Note

No momento, essa versão não está disponível nas seguintes regiões: AWS GovCloud (Leste dos EUA) [us-gov-east-1], AWS GovCloud (Oeste dos EUA) [us-gov-west-1]. Haverá um anúncio separado assim que ela for disponibilizada.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

Correções de alta prioridade:

- Corrigido um problema de falhas de conexão intermitentes após a rotação do certificado.

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-01-17 (versão 1.17.8) (obsoleta)

Versão: 1.17.8

O Aurora MySQL 1.17.8 está disponível para o público. Todos os novos clusters de bancos de dados Aurora MySQL com compatibilidade com o MySQL 5.6, inclusive os restaurados de snapshots, serão criados no Aurora MySQL 1.17.8. Você tem a opção, mas não a obrigatoriedade, de atualizar os clusters de banco de dados existentes para o Aurora MySQL 1.17.8. Para usar uma versão anterior, você pode criar novos clusters de banco de dados no Aurora MySQL 1.14.4, 1.15.1, 1.16 ou 1.17.7. Você pode fazer isso usando a API do Amazon RDS AWS CLI ou especificando a versão do mecanismo.

Com a versão 1.17.8 do Aurora MySQL, estamos usando um modelo de aplicação de patch de cluster em que todos os nós em um cluster de bancos de dados Aurora recebem patch ao mesmo tempo.

Note

No momento, essa versão não está disponível nas regiões AWS GovCloud (Oeste dos EUA) [us-gov-west-1] e China (Pequim) [cn-north-1]. Haverá um anúncio separado assim que ela for disponibilizada.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

- Corrigido um problema de performance que aumentou a utilização da CPU em uma réplica do Aurora após uma reinicialização.
- Corrigido um problema de estabilidade para consultas SELECT que usavam junção de hash.

Integração de correções de bug da edição MySQL community

- BUG N° 13418638: CREATE TABLE IF NOT EXISTS METADATA LOCK IS TOO RESTRICTIVE

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-10-08 (versão 1.17.7) (obsoleta)

Versão: 1.17.7

O Aurora MySQL 1.17.7 está disponível para o público. Todos os novos clusters de bancos de dados Aurora MySQL com compatibilidade com o MySQL 5.6, inclusive os restaurados de snapshots, serão criados no Aurora MySQL 1.17.7. Você tem a opção, mas não a obrigatoriedade, de atualizar os clusters de banco de dados existentes para o Aurora MySQL 1.17.7. Para usar uma versão anterior, você pode criar novos clusters de banco de dados no Aurora MySQL 1.14.4, 1.15.1, 1.16 ou 1.17.6. Você pode fazer isso usando a API do Amazon RDS AWS CLI ou especificando a versão do mecanismo.

Com a versão 1.17.7 do Aurora MySQL, estamos usando um modelo de aplicação de patch de cluster em que todos os nós em um cluster de bancos de dados Aurora recebem patch ao mesmo tempo.

Note

No momento, essa versão não está disponível nas regiões AWS GovCloud (Oeste dos EUA) [us-gov-west-1] e China (Pequim) [cn-north-1]. Haverá um anúncio separado assim que ela for disponibilizada.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

- A variável de status InnoDB `innodb_buffer_pool_size` foi tornada publicamente visível para os clientes modificarem.
- Corrigido um problema de estabilidade no cluster do Aurora ocorrido durante failovers.

- Melhorada disponibilidade do cluster com a correção de um problema de recuperação DDL ocorrido após uma operação TRUNCATE malsucedida.
- Corrigido um problema de estabilidade relacionado à atualização da tabela `mysql.innodb_table_stats`, disparada por operações DDL.
- Corrigidos problemas de estabilidade do Aurora Replica disparados durante a invalidação do cache de consulta após uma operação DDL.
- Corrigido um problema de estabilidade disparado por acesso de memória inválido durante a remoção do cache de dicionário periódica no segundo plano.

Integração de correções de bug da edição MySQL community

- Bug nº 16208542: índice perdido em uma coluna de chave estranha causa perda de tabela.
- Bug nº 76349: vazamento da memória em `add_derived_key()`.
- Bug nº 16862316: em tabelas particionadas, as consultas podem retornar resultados diferentes, dependendo do uso de mesclagem de índice.
- Bug nº 17588348: consultas que usam a otimização de mesclagem de índice (consulte [Otimização de mesclagem de índice](#)) podem retornar resultados inválidos quando executadas em tabelas que foram particionadas por HASH.

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-09-06 (versão 1.17.6) (obsoleta)

Versão: 1.17.6

O Aurora MySQL 1.17.6 está disponível para o público. Todos os novos clusters de bancos de dados Aurora MySQL com compatibilidade com o MySQL 5.6, inclusive os restaurados de snapshots, serão criados no Aurora MySQL 1.17.6. Você tem a opção, mas não a obrigatoriedade, de atualizar os clusters de banco de dados existentes para o Aurora MySQL 1.17.6. Para usar uma versão anterior, você pode criar novos clusters de banco de dados no Aurora MySQL 1.14.4, 1.15.1, 1.16 ou 1.17.5. Você pode fazer isso usando a API do Amazon RDS AWS CLI ou especificando a versão do mecanismo.

Com a versão 1.17.6 do Aurora MySQL, estamos usando um modelo de aplicação de patch de cluster em que todos os nós em um cluster de bancos de dados Aurora recebem patch ao mesmo tempo.

 Note

No momento, essa versão não está disponível nas regiões AWS GovCloud (Oeste dos EUA) [us-gov-west-1] e China (Pequim) [cn-north-1]. Haverá um anúncio separado assim que ela for disponibilizada.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

- Corrigido um problema de estabilidade no Aurora Reader para consultas SELECT enquanto o Aurora Writer está realizando operações DDL na mesma tabela.
- Corrigido um problema de estabilidade causado pela criação e exclusão de registros DDL para tabelas temporárias que usam Heap/Memory o mecanismo.
- Correção de um problema de estabilidade no operador do log binário quando instruções DDL estão sendo replicadas enquanto a conexão com o mestre do log binário está instável.
- Corrigido um problema de estabilidade encontrado durante a gravação no log de consulta lento.
- Corrigido um problema com a tabela de status da réplica que expôs informações incorretas de atraso do Aurora Reader.

Integração de correções de bug da edição MySQL community

- Para uma instrução [ALTER TABLE](#) que renomeou ou alterou o valor padrão de uma coluna [BINARY](#), a alteração foi feita usando uma cópia da tabela e não está no local. (Bug nº 67141, Bug nº 14735373, Bug nº 69580, Bug nº 17024290)
- Uma união externa entre uma tabela regular e uma tabela derivada que agrupa implicitamente pode causar uma saída do servidor. (Bug nº 16177639)

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-08-14 (versão 1.17.5) (obsoleta)

Versão: 1.17.5

O Aurora MySQL 1.17.5 está disponível para o público. Todos os novos clusters de bancos de dados Aurora MySQL com compatibilidade com o MySQL 5.6, inclusive os restaurados de snapshots, serão criados no Aurora MySQL 1.17.5. Você tem a opção, mas não a obrigatoriedade, de atualizar os clusters de banco de dados existentes para o Aurora MySQL 1.17.5. Para usar uma versão anterior, você pode criar novos clusters de banco de dados no Aurora MySQL 1.14.4, 1.15.1, 1.16 ou 1.17.4. Você pode fazer isso usando a API do Amazon RDS AWS CLI ou especificando a versão do mecanismo.

Com a versão 1.17.5 do Aurora MySQL, estamos usando um modelo de aplicação de patch de cluster em que todos os nós em um cluster de bancos de dados Aurora recebem patch ao mesmo tempo.

 Note

No momento, essa versão não está disponível nas regiões AWS GovCloud (Oeste dos EUA) [us-gov-west-1] e China (Pequim) [cn-north-1]. Haverá um anúncio separado assim que ela for disponibilizada.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

- Corrigido um problema em que um Aurora Writer pode passar por uma reinicialização após a correção de um cluster do Aurora com o recurso Zero-Downtime Patching.

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-08-07 (versão 1.17.4) (obsoleta)

Versão: 1.17.4

O Aurora MySQL 1.17.4 está disponível para o público. Todos os novos clusters de bancos de dados Aurora MySQL com compatibilidade com o MySQL 5.6, inclusive os restaurados de snapshots, serão criados no Aurora MySQL 1.17.4. Você tem a opção, mas não a obrigatoriedade, de atualizar os clusters de banco de dados existentes para o Aurora MySQL 1.17.4. Para usar uma versão

anterior, você pode criar novos clusters de banco de dados no Aurora MySQL 1.14.4, 1.15.1, 1.16 ou 1.17.3. Você pode fazer isso usando a API do Amazon RDS AWS CLI ou especificando a versão do mecanismo.

Com a versão 1.17.4 do Aurora MySQL, estamos usando um modelo de aplicação de patch de cluster em que todos os nós em um cluster de bancos de dados Aurora recebem patch ao mesmo tempo.

Note

No momento, essa versão não está disponível nas regiões AWS GovCloud (Oeste dos EUA) [us-gov-west-1] e China (Pequim) [cn-north-1]. Haverá um anúncio separado assim que ela for disponibilizada.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

- Melhorias de replicação
 - Tráfego de rede reduzido por meio da não transmissão de registros do log binário para réplicas de cluster. Por padrão, essa melhoria está habilitada.
 - Tráfego de rede reduzido por meio da compactação de mensagens de replicação. Essa melhoria é habilitada por padrão para as classes de instância 8xlarge e 16xlarge. Essas grandes instâncias podem sustentar um volume pesado de tráfego de gravação que resulta em tráfego de rede substancial para mensagens de replicação.
 - Correções no cache de consulta de réplica.
- Foi corrigido um problema em que `ORDER BY LOWER(col_name)` podia produzir ordenação incorreta ao usar o agrupamento `utf8_bin`.
- Corrigido um problema em que as instruções (especialmente `TRUNCATE TABLE`) podiam causar problemas em réplicas do Aurora, inclusive instabilidade ou tabelas não encontradas.
- Foi corrigido um problema em que os soquetes eram deixados em um estado entreaberto quando os nós de armazenamento eram reiniciados.
- Os seguintes parâmetros de cluster de banco de dados novos estão disponíveis:

- `aurora_enable_zdr` – Permita conexões abertas em um Aurora Replica permaneçam ativas na reinicialização da réplica.
- `aurora_enable_replica_log_compression` – Permita a compactação de cargas úteis de replicação para melhor a utilização da largura de banda da rede entre o mestre e as réplicas do Aurora.
- `aurora_enable_repl_bin_log_filtering` – Permita a filtragem de registros de replicação inutilizáveis por réplicas do Aurora no mestre.

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-06-05 (versão 1.17.3) (obsoleta)

Versão: 1.17.3

O Aurora MySQL 1.17.3 está disponível para o público. Todos os novos clusters de bancos de dados Aurora MySQL com compatibilidade com o MySQL 5.6, inclusive os restaurados de snapshots, serão criados no Aurora MySQL 1.17.3. Você tem a opção, mas não a obrigatoriedade, de atualizar os clusters de banco de dados existentes para o Aurora MySQL 1.17.3. É possível criar clusters de banco de dados no Aurora MySQL 1.14.4, Aurora MySQL 1.15.1 ou Aurora MySQL 1.16. Você pode fazer isso usando a API do Amazon RDS AWS CLI ou especificando a versão do mecanismo.

Com a versão 1.17.3 do Aurora MySQL, estamos usando um modelo de aplicação de patch de cluster em que todos os nós em um cluster de bancos de dados Aurora recebem patch ao mesmo tempo.

Note

No momento, essa versão não está disponível nas regiões AWS GovCloud (Oeste dos EUA) [us-gov-west-1] e China (Pequim) [cn-north-1]. Haverá um anúncio separado assim que ela for disponibilizada.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

- Foi corrigido um problema em que uma réplica do Aurora pode ser reiniciada ao usar restaurações de cursor otimistas enquanto lê registros.
- Corrigido um problema em que um Aurora Writer era reiniciado ao tentar encerrar uma sessão do MySQL (eliminar "<session id>") com o esquema de desempenho ativado.
- Foi corrigido um problema em que um gravador do Aurora é reiniciado ao computar um limite para coleta de lixo.
- Foi corrigido um problema em um gravador do Aurora que pode ocasionalmente ser reiniciado ao rastrear o andamento de uma réplica do Aurora no aplicativo de registro em log.
- Foi corrigido um problema com o cache de consulta quando a confirmação automática está desativada e que pode causar leituras obsoletas.

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-04-27 (versão 1.17.2) (obsoleta)

Versão: 1.17.2

O Aurora MySQL 1.17.2 está disponível para o público. Todos os novos clusters de bancos de dados Aurora MySQL com compatibilidade com o MySQL 5.6, inclusive os restaurados de snapshots, serão criados no Aurora MySQL 1.17.2. Você tem a opção, mas não a obrigatoriedade, de atualizar os clusters de banco de dados existentes para o Aurora MySQL 1.17.2. É possível criar clusters de banco de dados no Aurora MySQL 1.14.4, Aurora MySQL 1.15.1 ou Aurora MySQL 1.16. Você pode fazer isso usando a API do Amazon RDS AWS CLI ou especificando a versão do mecanismo.

Com a versão 1.17.2 do Aurora MySQL, estamos usando um modelo de aplicação de patch de cluster em que todos os nós em um cluster de bancos de dados Aurora recebem patch ao mesmo tempo.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

- Correção de um problema que estava causando reinícios durante determinadas operações de partição de DDL.

- Corrigido um problema que fazia com que o suporte para invocação de AWS Lambda funções por meio de funções nativas do Aurora MySQL fosse desativado.
- Correção de um problema com invalidação de cachê que estava causando reinícios em Aurora Replicas.
- Correção de um problema no gerenciador de bloqueio que estava causando reinícios.

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-03-23 (versão 1.17.1) (obsoleta)

Versão: 1.17.1

O Aurora MySQL 1.17.1 está disponível para o público. Todos os novos clusters de banco de dados, inclusive os restaurados de snapshots, serão criados no Aurora MySQL 1.17.1. Você tem a opção, mas não a obrigatoriedade, de atualizar os clusters de banco de dados existentes para o Aurora MySQL 1.17.1. É possível criar clusters de banco de dados no Aurora MySQL 1.15.1, Aurora MySQL 1.16 ou Aurora MySQL 1.17. Você pode fazer isso usando a API do Amazon RDS AWS CLI ou especificando a versão do mecanismo.

Com a versão 1.17.1 do Aurora MySQL, estamos usando um modelo de aplicação de patch de cluster em que todos os nós em um cluster de bancos de dados Aurora recebem patch ao mesmo tempo. Esta versão corrige alguns problemas conhecidos e regressões do mecanismo.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Note

Há um problema na versão mais recente do mecanismo do Aurora MySQL. Após a atualização para 1.17.1, a versão do mecanismo relatada é incorretamente como 1.17. Se você atualizou para 1.17.1, pode confirmar a atualização verificando a coluna Maintenance (Manutenção) do cluster de banco de dados no Console de gerenciamento da AWS. Se ela exibir none, o mecanismo está atualizado na versão 1.17.1.

Melhorias

- Foi corrigido um problema na recuperação de logs binários que resultava em tempos de recuperação mais longos em situações com grandes arquivos de índice de log binário que podem acontecer se os logs binários forem alternados com muita frequência.
- Foi corrigido um problema no otimizador de consultas que gerava um plano de consultas ineficiente para tabelas particionadas.
- Foi corrigido um problema no otimizador de consultas devido ao qual uma consulta de intervalo resultava em uma reinicialização do mecanismo de banco de dados.

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-03-13 (versão 1.17) (obsoleta)

Versão: 1.17

O Aurora MySQL 1.17 está disponível para o público. As versões 1.x do Aurora MySQL só são compatíveis com o MySQL 5.6, e não com o MySQL 5.7. Todos os novos clusters de banco de dados compatíveis com 5.6, inclusive os restaurados de snapshots, serão criados no Aurora 1.17. Você tem a opção, mas não a obrigatoriedade, de atualizar os clusters de banco de dados existentes para o Aurora 1.17. É possível criar novos clusters de banco de dados no Aurora 1.14.1, Aurora 1.15.1 ou Aurora 1.16. Você pode fazer isso usando a AWS CLI ou a API do Amazon RDS e especificando a versão do mecanismo.

Com a versão 1.17 do Aurora, estamos usando um modelo de aplicação de patch de cluster em que todos os nós em um cluster de bancos de dados Aurora recebem patch ao mesmo tempo. Oferecemos suporte a aplicação de patches com tempo de inatividade zero, o que funciona com base em melhor esforço para preservar as conexões de clientes durante o processo de aplicação de patches. Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#).

Aplicação de patches com tempo de inatividade zero

O recurso de aplicação de patches com tempo de inatividade zero (ZDP) tenta, com o melhor esforço, preservar as conexões do cliente por meio de um patch de mecanismo. Para obter mais

informações sobre ZDP, consulte [Como usar os patches com tempo de inatividade zero](#) no Guia do usuário do Amazon Aurora.

Novos recursos

- O Aurora MySQL agora oferece suporte à compactação de bloqueio, que otimiza o uso de memória do gerenciador de bloqueio. Desde a versão 1.17, você pode usar esse recurso sem habilitar o modo laboratório.

Melhorias

- Corrigido um problema visto predominantemente em instâncias com menos núcleos em que um único núcleo pode ter 100% de utilização da CPU mesmo quando o banco de dados está inativo.
- A performance da busca de logs binários de clusters do Aurora foi melhorada.
- Foi corrigido um problema onde as réplicas do Aurora tentavam gravar estatísticas da tabela no armazenamento persistente e falhavam.
- Foi corrigido um problema em que o cache de consultas não funcionava como esperado em réplicas do Aurora.
- Foi corrigida uma condição de corrida no gerenciador de bloqueio que resultava em uma reinicialização do mecanismo.
- Foi corrigido um problema em que os bloqueios obtidos por transações de somente de leitura e de confirmação automática resultavam em uma reinicialização do mecanismo.
- Foi corrigido um problema em que algumas consultas não eram gravadas nos logs de auditoria.
- Foi corrigido um problema com a recuperação de determinadas operações de manutenção de partição em caso de failover.

Integração de correções de bugs do MySQL

- LAST_INSERT_ID é replicado incorretamente se os filtros de replicação forem usados (Bug 69861)
- A consulta retorna resultados diferentes dependendo da configuração INDEX_MERGE (Bug 16862316)
- Nova execução de procedimento de consulta de rotina armazenada, plano de consulta ineficiente (Bug 16346367)
- INNODB FTS : declaração em FTS_CACHE_APPEND_DELETED_DOC_IDS (BUG 18079671)

- Declaração RBT_EMPTY(INDEX_CACHE->WORDS) na COLUNA DE ALTERAÇÃO DE ALTER TABLE (BUG 17536995)
- Pesquisa INNODB fulltext não encontra registros quando pontos de gravação estão envolvidos (BUG 70333, BUG 17458835)

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-12-11 (versão 1.16) (obsoleta)

Versão: 1.16

O Aurora MySQL 1.16 está disponível para o público. Todos os novos clusters de banco de dados, inclusive os restaurados de snapshots, serão criados no Aurora 1.16. Você tem a opção, mas não a obrigatoriedade, de atualizar os clusters de banco de dados existentes para o Aurora 1.16. É possível criar novos clusters de banco de dados no Aurora 1.14.1 ou Aurora 1.15.1. Você pode fazer isso usando a API do Amazon RDS AWS CLI ou especificando a versão do mecanismo.

Com a versão 1.16 do Aurora, estamos usando um modelo de aplicação de patch de cluster em que todos os nós em um cluster de bancos de dados Aurora recebem patch ao mesmo tempo. Estamos habilitando a aplicação de patches com tempo de inatividade zero, que funciona no melhor dos esforços para preservar as conexões de clientes ao longo do processo de aplicação de patches. Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#).

Aplicação de patches com tempo de inatividade zero

O recurso de aplicação de patches com tempo de inatividade zero (ZDP) tenta, com o melhor esforço, preservar as conexões do cliente por meio de um patch de mecanismo. Para obter mais informações sobre ZDP, consulte [Como usar os patches com tempo de inatividade zero](#) no Guia do usuário do Amazon Aurora.

Novos recursos

- O Aurora MySQL agora oferece suporte a AWS Lambda invocações síncronas por meio da função nativa. `lambda_sync()` A função nativa `lambda_async()` também está disponível e pode ser

usada como uma alternativa ao procedimento armazenado existente para a invocação assíncrona do Lambda. Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

- O Aurora MySQL agora é compatível com hash joins para acelerar consultas equijoin. O otimizador baseado em custo do Aurora pode decidir automaticamente quando usar junções hash. Também é possível forçar o uso em um plano de consultas. Para obter mais informações, consulte [Otimizando grandes consultas de junção do Aurora MySQL com junções hash](#) no Guia do usuário do Amazon Aurora.
- Agora, o Aurora MySQL oferece suporte ao lote de varredura para acelerar significativamente as consultas em memória, orientadas pela varredura. O recurso aumenta a performance de varreduras completas de tabela, de índice e de intervalo de índice por processamento em lote.

Melhorias

- Correção de um problema em que as réplicas de leitura travavam ao executar consultas em tabelas que tinham acabado de ser colocadas na tabela mestre.
- Correção de um problema em que a reinicialização do gravador em um cluster de banco de dados com um número muito grande de índices FULLTEXT resultava em uma recuperação mais longa do que a esperada.
- Correção de um problema em que a liberação de logs binários ocasionava incidentes LOST_EVENTS nos eventos de log binário.
- Correção de problemas de estabilidade com o programador quando o esquema de performance está ativado.
- Correção de um problema em que uma subconsulta usando tabelas temporárias retornava resultados parciais.

Integração de correções de bugs do MySQL

Nenhum

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-11-20 (versão 1.15.1) (obsoleta)

Versão: 1.15.1

O Aurora MySQL 1.15.1 está disponível para o público. Todos os novos clusters de banco de dados, inclusive os restaurados de snapshots, serão criados no Aurora 1.15.1. Você tem a opção, mas não a obrigatoriedade, de atualizar os clusters de banco de dados existentes para o Aurora 1.15.1. É possível criar novos clusters de banco de dados no Aurora 1.14.1. Você pode fazer isso usando a API do Amazon RDS AWS CLI ou especificando a versão do mecanismo.

Com a versão 1.15.1 do Aurora, estamos usando um modelo de aplicação de patch de cluster em que todos os nós em um cluster de bancos de dados Aurora recebem patch ao mesmo tempo. Estamos habilitando a aplicação de patches com tempo de inatividade zero, que funciona no melhor dos esforços para preservar as conexões de clientes ao longo do processo de aplicação de patches. Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Aplicação de patches com tempo de inatividade zero

O recurso de aplicação de patches com tempo de inatividade zero (ZDP) tenta, com o melhor esforço, preservar as conexões do cliente por meio de um patch de mecanismo. Para obter mais informações sobre ZDP, consulte [Como usar os patches com tempo de inatividade zero](#) no Guia do usuário do Amazon Aurora.

Melhorias

- Correção de um problema no seletor de segmento adaptativo para uma solicitação de leitura que faria com que ele escolhesse o mesmo segmento duas vezes, causando um aumento na latência de leitura sob certas condições.
- Correção de um problema decorrente de uma otimização no Aurora MySQL para o programador de threads. Este problema se manifesta em erros espúrios durante a gravação no log lento, enquanto as consultas associadas em si estão normais.
- Correção de um problema de estabilidade das réplicas de leitura em volumes grandes (> 5 TB).
- Correção de um problema em que a contagem de threads do operador aumenta continuamente devido a uma contagem falsa de conexão excepcional.
- Corrigido um problema com bloqueios de tabela que provocavam esperas longas de semáforo durante workloads de inserção.

- Revertidas as seguintes correções de bugs do MySQL incluídas no Aurora MySQL 1.15:
 - Instância do MySQL paralisada ao "fazer um índice SYNC" (Bug nº 73816)
 - Declaração RBT_EMPTY(INDEX_CACHE->WORDS) na COLUNA DE ALTERAÇÃO DE ALTER TABLE (Bug 17536995)
 - Pesquisa de texto completo de InnoDB não encontra registros quando pontos de salvamento estão envolvidos (erro #70333)

Integração de correções de bugs do MySQL

Nenhum

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-10-24 (versão 1.15) (obsoleta)

Versão: 1.15

O Aurora MySQL 1.15 está disponível para o público. Todos os novos clusters de banco de dados, inclusive os restaurados de snapshots, serão criados no Aurora 1.15. Você tem a opção, mas não a obrigatoriedade, de atualizar os clusters de banco de dados existentes para o Aurora 1.15. É possível criar novos clusters de banco de dados no Aurora 1.14.1. Você pode fazer isso usando a API do Amazon RDS AWS CLI ou especificando a versão do mecanismo.

Com a versão 1.15 do Aurora, estamos usando um modelo de aplicação de patch de cluster em que todos os nós em um cluster de bancos de dados Aurora recebem patch ao mesmo tempo. Como as atualizações exigem a reinicialização do banco de dados, você terá de 20 a 30 segundos de tempo de inatividade, após o qual poderá continuar usando seus clusters de banco de dados. Se os clusters de banco de dados estiverem executando o Aurora 1.14 ou o Aurora 1.14.1, o recurso de aplicação de patch com tempo de inatividade zero no Aurora MySQL poderá permitir conexões de clientes à instância principal do Aurora MySQL para persistir na atualização, dependendo da workload.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Aplicação de patches com tempo de inatividade zero

O recurso de aplicação de patches com tempo de inatividade zero (ZDP) tenta, com o melhor esforço, preservar as conexões do cliente por meio de um patch de mecanismo. Para obter mais informações sobre ZDP, consulte [Como usar os patches com tempo de inatividade zero](#) no Guia do usuário do Amazon Aurora.

Novos recursos

- **Asynchronous Key Prefetch:** o Asynchronous Key Prefetch (AKP – Pré-busca de chave assíncrona) é um recurso destinado a melhorar a performance de junções de índice não armazenadas em cache pré-buscando chaves na memória quando necessário. O caso de uso principal segmentado por AKP é uma junção de índice entre uma tabela externa pequena e interna grande, em que o índice é altamente seletivo na tabela maior. Além disso, quando a interface Multi-Range Read (MRR – Leitura de vários intervalos) estiver habilitada, o AKP será aproveitado para uma pesquisa de índice de secundária para primária. Instâncias menores que tenham restrições de memória podem, em alguns casos, ser capazes de utilizar o AKP, dada a cardinalidade da chave certa. Para obter mais informações, consulte [Otimizar consultas de junção indexadas do Aurora MySQL com pré-busca de chave assíncrona](#) no Guia do usuário do Amazon Aurora.
- **Fast DDL** – estendemos o recurso que foi lançado no [Aurora 1.13](#) para operações que incluem valores padrão. Com essa extensão, o DDL rápido se aplica a operações que adicionam uma coluna anulável ao final de uma tabela, com ou sem valores padrão. O recurso permanece em modo de laboratório do Aurora. Para obter mais informações, consulte [Alterar tabelas no Amazon Aurora usando a DDL rápida](#) no Guia do usuário do Amazon Aurora.

Melhorias

- Corrigido um erro de cálculo durante a otimização de consultas WITHIN/CONTAINS espaciais que anteriormente resultava em um conjunto de resultados vazio.
- Corrigiu o comando SHOW VARIABLE para mostrar o valor de parâmetro `innodb_buffer_pool_size` atualizado sempre que é alterado no parameter group.
- A maior estabilidade da instância primária durante a inserção em lote em uma tabela alterada usando o DDL rápido quando a indexação de hash adaptável está desabilitada e o registro a ser inserido é o primeiro de uma página.

- Maior estabilidade do Aurora quando o usuário tenta definir o valor do parâmetro do cluster de banco de dados `server_audit_events` como **default**.
- Corrigido um problema no qual uma alteração feita no conjunto de caracteres do banco de dados de uma instrução `ALTER TABLE` executada na instância primária do Aurora não replicada nas réplicas do Aurora até serem reiniciadas.
- Maior estabilidade corrigindo uma condição de corrida na instância principal que o permitiu registrar uma réplica do Aurora, mesmo que a instância primária tenha fechado o próprio volume.
- Aprimoramento da performance da instância primária durante a criação do índice em uma tabela grande alterando o protocolo de bloqueio para habilitar instruções DML durante a criação do índice.
- Corrigiu a inconsistência de metadados de InnoDB durante a consulta `ALTER TABLE RENAME` que aumentou a estabilidade. Exemplo: quando colunas da tabela `t1(c1, c2)` são renomeadas de maneira cíclica `t1(c2, c3)` dentro da mesma instrução `ALTER`.
- Maior estabilidade de réplicas do Aurora para o cenário em que uma réplica do Aurora não tem workload ativa e a instância primária não responde.
- Maior disponibilidade de réplicas do Aurora para um cenário no qual a réplica do Aurora mantém um bloqueio explícito em uma tabela e impede o thread de replicação de aplicar qualquer alteração feita no DDL recebida da instância primária.
- Maior estabilidade da instância primária quando uma chave externa e uma coluna são adicionadas a uma tabela de duas sessões separadas ao mesmo tempo e o DDL rápido foi habilitado.
- Maior estabilidade do thread de purga na instância principal durante uma workload de gravação pesada bloqueando o truncamento de registros para desfazer até serem purgados.
- Maior estabilidade corrigindo a ordem de versão do bloqueio durante o processo de confirmação de transações que descartam tabelas.
- Corrigiu um defeito em réplicas do Aurora em que a instância de banco de dados não pode ser inicializada e reclamou que a porta 3306 já estava em uso.
- Corrigiu uma condição de corrida em que uma execução de consulta `SELECT` em determinadas tabelas `information_schema` (`innodb_trx`, `innodb_lock`, `innodb_lock_waits`) aumentou a instabilidade do cluster.

Integração de correções de bugs do MySQL

- `CREATE USER` aceita plugin e hash de senha, mas ignora o hash de senha (erro #78033)

- O mecanismo de particionamento adiciona campos ao conjunto de bits de leitura capazes de retornar entradas classificadas de um índice particionado. Isso leva o buffer de junção a tentar ler campos desnecessários. Corrigido não adicionando todos os campos de particionamento ao `read_set`, mas em vez de somente classificar nos campos de prefixo definidos no `read_set`. Adicionado um `DEBUG_ASSERT` que, se `key_cmp`, pelo menos o primeiro campo deverá ser lido (Bug 16367691).
- Instância do MySQL paralisada ao "fazer um índice SYNC" (Bug nº 73816)
- Declaração `RBT_EMPTY(INDEX_CACHE->WORDS)` na COLUNA DE ALTERAÇÃO DE `ALTER TABLE` (Bug 17536995)
- Pesquisa de texto completo de InnoDB não encontra registros quando pontos de salvamento estão envolvidos (erro #70333)

Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-03-13 (versão 1.14.4) (obsoleta)

Versão: 1.14.4

O Aurora MySQL 1.14.4 está disponível para o público. Você pode criar novos clusters de banco de dados no Aurora 1.14.4, usando a CLI AWS ou a API do Amazon RDS e especificando a versão do mecanismo. Você tem a opção, mas não a obrigatoriedade, de atualizar os clusters de banco de dados 1.14.x existentes para o Aurora 1.14.4.

Com a versão 1.14.4 do Aurora, estamos usando um modelo de aplicação de patch de cluster em que todos os nós em um cluster de bancos de dados Aurora recebem patch ao mesmo tempo. Oferecemos suporte a aplicação de patches com tempo de inatividade zero, o que funciona com base em melhor esforço para preservar as conexões de clientes durante o processo de aplicação de patches. Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#). Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Aplicação de patches com tempo de inatividade zero

O recurso de aplicação de patches com tempo de inatividade zero (ZDP) tenta, com o melhor esforço, preservar as conexões do cliente por meio de um patch de mecanismo. Para obter mais

informações sobre ZDP, consulte [Como usar os patches com tempo de inatividade zero](#) no Guia do usuário do Amazon Aurora.

Novos recursos

- Agora o Aurora MySQL oferece suporte às classes de instâncias db.r4.

Melhorias

- Foi corrigido um problema em que LOST_EVENTS eram gerados ao gravar grandes eventos de binlog.

Integração de correções de bugs do MySQL

- Eventos ignoráveis não funcionam e não são testados (bug N° 74683)
- NOVO->FALHA EM ASSERT ANTIGA "GTID_MODE > 0" (erro #20436436)

Atualizações do mecanismo de banco de dados do Aurora MySQL: 2017-09-22 (versão 1.14.1) (obsoleta)

Versão: 1.14.1

O Aurora MySQL 1.14.1 está disponível para o público. Todos os novos clusters de banco de dados, inclusive os restaurados de snapshots, serão criados no Aurora MySQL 1.14.1. O Aurora MySQL 1.14.1 também é uma atualização obrigatória para clusters de banco de dados existentes do Aurora MySQL. Para obter mais informações, consulte [Anúncio: Extensão do cronograma de atualização obrigatório para o Amazon Aurora](#) no site do AWS Developer Forums.

Com a versão 1.14.1 do Aurora MySQL, estamos usando um modelo de aplicação de patch de cluster em que todos os nós em um cluster de bancos de dados Aurora MySQL recebem patch ao mesmo tempo. Como as atualizações exigem a reinicialização do banco de dados, você terá de 20 a 30 segundos de tempo de inatividade, após o qual poderá continuar usando seus clusters de banco de dados. Se os clusters de banco de dados estiverem executando a versão 1.13 ou posterior, o recurso de patches com tempo de inatividade zero do Aurora MySQL poderá permitir que conexões de clientes à instância principal do Aurora MySQL persistam após a atualização, dependendo da workload.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#).

Melhorias

- Corrigiu condições de corrida associadas a inserções e purga para aumentar a estabilidade do recurso DDL rápido, que permanece em modo de laboratório do Aurora MySQL.

Atualizações do mecanismo de banco de dados do Aurora MySQL: 2017-08-07 (versão 1.14) (obsoleta)

Versão: 1.14

O Aurora MySQL 1.14 está disponível para o público. Todos os novos clusters de banco de dados, inclusive os restaurados de snapshots, serão criados no Aurora MySQL 1.14. O Aurora MySQL 1.14 também é uma atualização obrigatória para clusters de banco de dados existentes do Aurora MySQL. Enviaremos um anúncio separado com a linha do tempo para reprovar versões anteriores do Aurora MySQL.

Com a versão 1.14 do Aurora MySQL, estamos usando um modelo de aplicação de patch de cluster em que todos os nós em um cluster de bancos de dados Aurora recebem patch ao mesmo tempo. Como as atualizações exigem a reinicialização do banco de dados, você terá de 20 a 30 segundos de tempo de inatividade, após o qual poderá continuar usando seus clusters de banco de dados. Se os clusters de banco de dados estiverem executando a versão 1.13, o recurso de patches com tempo de inatividade zero do Aurora poderá permitir conexões cliente à instância primária do Aurora para manter a atualização, dependendo da workload.

Se você tiver alguma dúvida ou preocupação, o AWS Support está disponível nos fóruns da comunidade e por meio do [AWS Support](#).

Aplicação de patches com tempo de inatividade zero

O recurso de aplicação de patches com tempo de inatividade zero (ZDP) tenta, com o melhor esforço, preservar as conexões do cliente por meio de um patch de mecanismo. Para obter mais informações sobre ZDP, consulte [Como usar os patches com tempo de inatividade zero](#) no Guia do usuário do Amazon Aurora.

Melhorias

- Corrigiu um erro "record not found" incorreto quando um registro foi encontrado no índice secundário, mas não no índice primário.
- Corrigiu um problema de estabilidade que pode ocorrer por causa de uma declaração defensiva (adicionada no 1.12) que estava muito forte quando um indivíduo escreve algo abrangendo mais de 32 páginas. Essa situação pode ocorrer, por exemplo, com valores BLOB grandes.
- Corrigiu um problema de estabilidade por causa de inconsistências entre os caches do espaço de tabela e do dicionário.
- Corrigiu um problema no qual uma réplica do Aurora se tornará sem resposta depois que exceder o número máximo de tentativas de conexão com a instância primária. Uma réplica do Aurora será reiniciada se o período de inatividade for além do período de pulsação usado para verificação de integridade pela instância primária.
- Corrigiu um livelock que pode ocorrer sob simultaneidade muito alta quando uma conexão tenta adquirir um Meta Data Lock (MDL – Bloqueio de metadado) exclusivo enquanto emite um comando, como `ALTER TABLE`.
- Corrigido um problema de estabilidade em uma réplica de leitura do Aurora na presença de logical/parallel leitura antecipada.
- Melhorou `LOAD FROM S3` de duas maneiras:
 1. Processamento melhor de erros de tempo limite do Amazon S3 usando a nova tentativa do SDK, além da nova tentativa existente.
 2. Otimização de performance durante o carregamento de arquivos muito grandes ou muitos arquivos armazenando em cache e reutilizando o estado do cliente.
- Corrigiu os seguintes problemas de estabilidade com DDL rápido para operações `ALTER TABLE`:
 1. Quando a instrução `ALTER TABLE` tem vários comandos `ADD COLUMN` e os nomes de coluna não estão em ordem crescente.
 2. Quando a string de nome da coluna a ser atualizada e a string de nome correspondente, buscadas na tabela do sistema interno, difere por um caractere de encerramento nulo (`/0`).
 3. Em determinadas operações de divisão da árvore B.
 4. Quando a tabela tem uma chave principal de comportamento variável.
- Corrigiu um problema de estabilidade com réplicas do Aurora quando demora muito para criar o cache de índice Full Text Search (FTS – Pesquisa de texto completa) consistente com essa instância primária. Isso poderá acontecer se uma parte grande das entradas de índice FTS recém-criadas na instância primária ainda não tiver sido enviada para disco.

- Corrigiu um problema de estabilidade que pode acontecer durante a criação do índice.
- Nova infraestrutura que rastreia o consumo de memória por conexão e a telemetria associada que será usada para criar estratégias de prevenção Out-Of-Memory (OOM).
- Corrigiu um problema em que `ANALYZE TABLE` estava permitido incorretamente em réplicas do Aurora. Isso já foi bloqueado.
- Corrigiu um problema de estabilidade causado por um deadlock raro por causa de uma condição de corrida entre a leitura lógica e a limpeza.

Integração de correções de bugs do MySQL

- Uma pesquisa de texto completo combinada com tabelas derivadas (subconsultas na cláusula `FROM`) causou uma saída do servidor. Agora, se uma operação de texto completa depender de uma tabela derivada, o servidor produzirá um erro indicando que uma pesquisa de texto completa não pode ser realizada em uma tabela materializada. (Bug nº 68751, Bug nº 16539903)

Atualizações do mecanismo de banco de dados do Aurora MySQL: 2017-05-15 (versão 1.13) (obsoleta)

Versão: 1.13

Note

Habilitamos um novo recurso – `SELECT INTO OUTFILE S3` – no Aurora MySQL versão 1.13 depois da versão inicial e atualizamos as notas de release para refletir essa alteração.

O Aurora MySQL 1.13 está disponível para o público. Todos os novos clusters de banco de dados, inclusive os restaurados de snapshots, serão criados no Aurora MySQL 1.13. Você tem a opção, mas não a obrigatoriedade, de atualizar os clusters de banco de dados existentes para o Aurora MySQL 1.13. Com a versão 1.13 do Aurora, estamos usando um modelo de aplicação de patch de cluster em que todos os nós em um cluster de bancos de dados Aurora recebem patch ao mesmo tempo. Estamos habilitando a aplicação de patches com tempo de inatividade zero, que funciona no melhor dos esforços para preservar as conexões de clientes ao longo do processo de aplicação de patches. Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Aplicação de patches com tempo de inatividade zero

O recurso de aplicação de patches com tempo de inatividade zero (ZDP) tenta, com o melhor esforço, preservar as conexões do cliente por meio de um patch de mecanismo. Para obter mais informações sobre ZDP, consulte [Como usar os patches com tempo de inatividade zero](#) no Guia do usuário do Amazon Aurora.

Novos atributos:

- **SELECT INTO OUTFILE S3** – O Aurora MySQL já permite fazer upload dos resultados de uma consulta para um ou mais arquivos em um bucket do Amazon S3. Para obter mais informações, consulte [Salvar dados a partir de um cluster de banco de dados do Amazon Aurora MySQL em arquivos de texto de um bucket do Amazon S3](#) no Guia do usuário do Amazon Aurora.

Melhorias:

- Implementado o truncamento de arquivos de log no formato CSV na inicialização do mecanismo para evitar um longo tempo de recuperação. Agora tabelas `general_log_backup`, `general_log`, `slow_log_backup` e `slow_log` não sobrevivem a uma reinicialização do banco de dados.
- Correção de um problema em que a migração de um banco de dados chamado **test** falhava.
- Melhoria na estabilidade do coletor de lixo do gerenciador de bloqueios, reutilizando os segmentos de bloqueio corretos.
- Melhoria na estabilidade do gerenciador de bloqueios, removendo declarações inválidas durante o algoritmo de detecção de deadlock.
- Reabilitação da replicação assíncrona e correção de um problema associado que fazia com que o atraso de réplica incorreto fosse informado em uma workload sem carga ou somente leitura. As melhorias no pipeline de replicação que foram apresentadas na versão 1.10. Essas melhorias foram apresentadas para aplicar atualizações no stream de log ao cache do buffer de uma Réplica do Aurora, o que ajuda a melhorar a performance e a estabilidade de leitura em réplicas do Aurora.
- Correção de um problema em que `autocommit=OFF` faz com que eventos agendados sejam bloqueados e transações longas permaneçam abertas até o servidor reinicializar.
- Correção de um problema em que logs de consultas lentas, gerais e de auditoria não podiam registrar consultas processadas por confirmação assíncrona.

- Melhoria na performance do recurso LRA (logical read ahead) em até 2,5 vezes. Isso foi feito permitindo que as pré-buscas continuassem em páginas intermediárias em uma árvore B.
- Adição da validação de parâmetros para variáveis de auditoria a fim de remover espaços desnecessários.
- Correção de uma regressão, apresentada no Aurora MySQL versão 1.11, em que as consultas podem retornar resultados incorretos ao usarem a opção `SQL_CALC_FOUND_ROWS` e invocando a função `FOUND_ROWS()`.
- Correção de um problema de estabilidade quando a lista Metadata Lock era formada incorretamente.
- Estabilidade aprimorada quando `sql_mode` está definido como `PAD_CHAR_TO_FULL_LENGTH` e o comando `SHOW FUNCTION STATUS WHERE Db= 'string'` é executado.
- Correção de um caso raro quando instâncias não surgiam após a atualização da versão do Aurora devido a uma verificação de consistência de volume falsa.
- Correção do problema de performance, apresentado no Aurora MySQL versão 1.12, em que a performance do gravador do Aurora era reduzida quando os usuários tinham um grande número de tabelas.
- Melhoria em um problema de estabilidade quando o gravador do Aurora está configurado como um operador de log binário e o número de conexões se aproxima de 16.000.
- Correção de um problema raro em que uma Réplica do Aurora podia ser reiniciada quando uma conexão fosse bloqueada à espera de Metadata Lock ao executar a DDL no Aurora mestre.

Integração de correções de bugs do MySQL

- Com uma tabela do InnoDB vazia, não é possível diminuir o valor de `auto_increment` usando uma instrução `ALTER TABLE`, mesmo quando a tabela está vazia. (Bug nº 69882)
- `MATCH() ... AGAINST` que usam uma string longa como argumento para `AGAINST()` podem resultar em um erro quando executadas em uma tabela do InnoDB com um índice de pesquisa de texto completo. (Bug nº 17640261)
- A manipulação de `SQL_CALC_FOUND_ROWS` em combinação com `ORDER BY` e `LIMIT` podia provocar resultados incorretos para `FOUND_ROWS()`. (Bug nº 68458, Bug nº 16383173)
- `ALTER TABLE` não permite alterar a nulidade da coluna quando existe uma chave externa. (Bug nº 77591)

Atualizações do mecanismo de banco de dados do Aurora MySQL: 2017-04-05 (versão 1.12) (obsoleta)

Versão: 1.12

Agora, o Aurora MySQL 1.12 é a versão preferencial para a criação de novos clusters de banco de dados, incluindo restaurações de snapshots.

Esta não é uma atualização obrigatória para clusters existentes. Você terá a opção de atualizar os clusters existentes para a versão 1.12 depois de concluir o patch no âmbito de frotas para a versão 1.11 (consulte as [notas de release](#) do Aurora 1.11 e o comunicado do [fórum correspondente](#)). Com a versão 1.12 do Aurora, estamos usando um modelo de aplicação de patch de cluster em que todos os nós em um cluster de bancos de dados Aurora recebem patch ao mesmo tempo. Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Novos recursos

- **Fast DDL** – Agora o Aurora MySQL permite executar uma operação `ALTER TABLE tbl_name ADD COLUMN col_name column_definition` quase instantaneamente. A operação é concluída sem exigir que a tabela seja copiada e sem causar impacto material em outras instruções DML. Como ela não consome armazenamento temporário para a cópia da tabela, as instruções de DDL são práticas mesmo para tabelas grandes em classes de instâncias pequenas. O DDL rápido atualmente tem suporte apenas para adicionar uma coluna anulável, sem um valor padrão, no final de uma tabela. Esse recurso está atualmente disponível no modo de laboratório do Aurora. Para obter mais informações, consulte [Alterar tabelas no Amazon Aurora usando a DDL rápida](#) no Guia do usuário do Amazon Aurora.
- **Mostrar o status do volume** – Adicionamos um novo comando de monitoramento, `SHOW VOLUME STATUS`, para exibir o número de nós e discos em um volume. Para obter mais informações, consulte [Exibir o status do volume para um cluster de banco de dados do Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

- **Implementação de alterações na compactação de bloqueio** para reduzir ainda mais a memória alocada por objeto de bloqueio. Essa melhoria está disponível no modo de laboratório.

- Correção de um problema em que a métrica `trx_active_transactions` diminui rapidamente mesmo quando o banco de dados está ocioso.
- Correção de uma mensagem de erro inválida sobre a sintaxe de consulta de injeção de falhas ao simular falhas em discos e nós.
- Correção de vários problemas relacionados a condições de corrida e travamentos inativos no gerenciador de bloqueio.
- Correção de um problema que causava um estouro de buffer no otimizador de consulta.
- Correção de um problema de estabilidade em réplicas de leitura do Aurora quando os nós de armazenamento subjacentes apresentam baixa memória disponível.
- Correção de um problema em que conexões ociosas persistiam após a configuração do parâmetro `wait_timeout`.
- Correção de um problema em que `query_cache_size` retorna um valor inesperado após a reinicialização da instância.
- Correção de um problema de performance que é o resultado de um thread de diagnóstico que sonda a rede muitas vezes no caso em que as gravações não estão progredindo para o armazenamento.

Integração de correções de bugs do MySQL

- Recarregar uma tabela que foi removida enquanto vazia fez com que um valor `AUTO_INCREMENT` fosse redefinido. (Bug nº 21454472, Bug nº 77743)
- Um registro de índice não foi encontrado na reversão devido a inconsistências na estrutura de `purge_node_t`. A inconsistência resultou em avisos e mensagens de erro, como "erro na atualização da entrada do índice sec", "impossível limpar um registro" e "tentativa de limpar a entrada do índice sec não marcada para exclusão". (Bug nº 19138298, Bug nº 70214, Bug nº 21126772, Bug nº 21065746)
- O cálculo incorreto do tamanho da pilha para a operação `qsort` resulta em excedente de pilha. (Bug nº 73979)
- Registro não encontrado em um índice após reversão. (Bug nº 70214, Bug nº 72419)
- `ALTER TABLE` adicionar coluna `TIMESTAMP` na atualização `CURRENT_TIMESTAMP` insere ZERO-datas (Bug nº 17392)

Atualizações do mecanismo de banco de dados do Aurora MySQL: 2017-02-23 (versão 1.11) (obsoleta)

Versão: 1.11

Aplicaremos patches em todos os clusters de bancos de dados Aurora MySQL com a versão mais recente durante um curto período após o lançamento. Os clusters de banco de dados são corrigidos usando o procedimento legado com um tempo de inatividade de cerca de 5 a 30 segundos.

A aplicação de patches ocorre durante a janela de manutenção do sistema que você especificou para cada uma das suas instâncias de banco de dados. Você pode visualizar ou alterar essa janela usando o Console de gerenciamento da AWS. Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Como alternativa, você pode aplicar o patch imediatamente no Console de gerenciamento da AWS escolhendo um cluster de banco de dados, escolhendo Cluster Actions e, em seguida, escolhendo Upgrade Now.

Com a versão 1.11 do Aurora MySQL, estamos usando um modelo de aplicação de patch de cluster em que todos os nós em um cluster de bancos de dados Aurora recebem patch ao mesmo tempo.

Novos recursos

- Opção MANIFEST para LOAD DATA FROM S3 – LOAD DATA FROM S3 foi lançado na versão 1.8. As opções para esse comando foram expandidas, e agora você pode especificar uma lista de arquivos a serem carregados em um cluster de bancos de dados Aurora do Amazon S3 usando um arquivo manifesto. Isso facilita o carregamento de dados de arquivos específicos em um ou mais locais, ao contrário do carregamento de dados de um único arquivo usando a opção FILE ou do carregamento de dados de vários arquivos que têm a mesma localização e prefixo usando a opção PREFIX. O formato do arquivo manifesto é o mesmo usado pelo Amazon Redshift. Para obter mais informações sobre como usar LOAD DATA FROM S3 com a opção MANIFEST, consulte [Usar um manifesto para especificar arquivos de dados para carregamento](#) no Guia do usuário do Amazon Aurora.
- Indexação espacial habilitada por padrão – esse recurso foi lançado no modo de laboratório na versão 1.10 e agora está ativado por padrão. A indexação espacial melhora a performance das consultas em conjuntos de dados grandes para consultas que utilizam dados espaciais. Para obter mais informações sobre o uso da indexação espacial, consulte [Amazon Aurora MySQL e dados espaciais](#) no Guia do usuário do Amazon Aurora.

- Alteração no cronograma de auditoria avançada: esse recurso foi lançado na versão 1.10.1 para fornecer uma facilidade de alta performance para a auditoria de atividades do banco de dados. Nessa versão, a precisão dos carimbos de data/hora do log de auditoria foi alterada de um segundo para um microssegundo. Os carimbos de data/hora mais precisos permitem compreender melhor quando um evento de auditoria ocorreu. Para obter mais informações sobre auditoria, consulte [Como utilizar a auditoria avançada em um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

- Modificação do parâmetro `thread_handling` para evitar que você o defina como algo diferente de **multiple-connections-per-thread**, que é o único modelo com suporte no grupo de threads do Aurora.
- Correção de um problema causado quando você define o parâmetro `buffer_pool_size` e `query_cache_size` para ser maior que a memória total do cluster de banco de dados. Nessa circunstância, o Aurora define o parâmetro modificado como o valor padrão e, portanto, o cluster de banco de dados pode ser inicializado e não falhar.
- Correção de um problema no cache de consulta em que uma transação obtém resultados de leitura obsoletos se a tabela é invalidada em outra transação.
- Correção de um problema em que arquivos binlog marcados para exclusão são removidos depois de um pequeno atraso em vez de serem removidos imediatamente.
- Correção de um problema em que um banco de dados criado com o nome **tmp** é tratado como um banco de dados do sistema armazenado em armazenamento temporário e não persistente no armazenamento distribuído do Aurora.
- Modificação do comportamento de `SHOW TABLES` para excluir determinadas tabelas internas do sistema. Essa alteração ajuda a evitar um failover desnecessário causado depois que `mysqldump` bloqueia todos os arquivos listados em `SHOW TABLES`, o que, por sua vez, evita gravações na tabela interna do sistema, causando o failover.
- Correção de um problema em que uma Réplica do Aurora é reiniciada incorretamente ao criar uma tabela temporária de uma consulta que invoca uma função cujo argumento é uma coluna de uma tabela do InnoDB.
- Correção de um problema relacionado a um conflito de bloqueio de metadados em um nó de Réplica do Aurora que faz com que a Réplica do Aurora fique atrás do cluster de banco de dados principal e acabe sendo reiniciada.

- Correção de uma trava inoperante no pipeline de replicação em nós leitores, o que faz com que uma Réplica do Aurora se atrase e acabe sendo reiniciada.
- Correção de um problema em que uma Réplica do Aurora fica muito atrasada com volumes criptografados maiores que 1 terabyte (TB).
- Melhoria na detecção de travas inoperantes da Réplica do Aurora com o uso de uma melhor maneira de ler o tempo do relógio do sistema.
- Correção de um problema em que uma Réplica do Aurora pode ser reiniciada duas vezes em vez de uma após o cancelamento do registro pelo gravador.
- Correção de um problema de performance de consultas lentas em Réplicas do Aurora que ocorre quando estatísticas transitórias causam discrepância em colunas de índice não exclusivas.
- Correção de um problema em que uma Réplica do Aurora pode falhar quando uma instrução DDL é replicada na Réplica do Aurora ao mesmo tempo que a Réplica do Aurora está processando uma consulta relacionada.
- Alteração nas melhorias do pipeline de replicação que foram introduzidas na versão 1.10 de habilitadas por padrão para desabilitadas por padrão. Essas melhorias foram apresentadas para aplicar atualizações de stream de log ao cache de buffer de uma Réplica do Aurora e, embora esse recurso ajude a melhorar a performance e a estabilidade em Réplicas do Aurora, ele aumenta o atraso dessas réplicas em determinadas workloads.
- Correção de um problema em que a ocorrência simultânea de uma instrução DDL contínua e da Leitura antecipada paralela pendente na mesma tabela causa uma falha de asserção durante a fase de confirmação da transação DDL.
- Melhoria do log geral e do log de consultas lentas para sobreviver ao reinício do cluster de banco de dados.
- Foi corrigido um out-of-memory problema para determinadas consultas de longa duração ao reduzir o consumo de memória no módulo ACL.
- Correção de um problema de reinício que ocorre quando uma tabela possui índices não espaciais, quando existem predicados espaciais na consulta, quando o planejador opta por usar um índice não espacial e quando o planejador envia incorretamente a condição espacial ao índice.
- Corrigido um problema em que o cluster de banco de dados reinicia quando há uma exclusão, atualização ou limpeza de objetos geoespaciais muito grandes que são armazenados externamente (como). LOBs
- Correção de um problema em que a simulação de falhas usando ALTER SYSTEM SIMULATE... FOR INTERVAL não está funcionando corretamente.

- Correção de um problema de estabilidade causado por uma declaração inválida em um invariante incorreto no gerenciador de bloqueios.
- Desabilitação das duas melhorias a seguir para a pesquisa de texto completo do InnoDB que foram introduzidas na versão 1.10, pois introduzem problemas de estabilidade para algumas workloads exigentes:
 - Atualização do cache somente após uma solicitação de leitura para uma Réplica do Aurora a fim de melhorar a velocidade de replicação do cache de índice de pesquisa de texto completo.
 - Descarregando a tarefa de sincronização de cache em um thread separado, assim que o tamanho do cache atingir 10% do tamanho total, para evitar que consultas MySQL fiquem paradas por muito tempo durante a sincronização do cache de FTS com o disco. (Bugs nº 22516559, nº 73816).

Integração de correções de bugs do MySQL

- A execução da tabela ALTER da chave estrangeira DROP simultaneamente com outra operação DROP faz com que a tabela desapareça. (Bug nº 16095573)
- Algumas consultas INFORMATION_SCHEMA que usaram ORDER BY não usaram uma otimização de classificação de arquivo como fizeram anteriormente. (Bug nº 16423536)
- FOUND_ROWS () retorna a contagem incorreta de linhas em uma tabela. (Bug nº 68458)
- O servidor falha em vez de indicar um erro quando muitas tabelas temporárias estão abertas. (Bug nº 18948649)

Atualizações do mecanismo de banco de dados do Aurora MySQL: 2017-01-12 (versão 1.10.1) (obsoleta)

Versão: 1.10.1

A Versão 1.10.1 do Aurora MySQL é uma versão de assinatura e não é usada para aplicar patches em suas instâncias de bancos de dados. Ela está disponível para criar novas instâncias do Aurora e para atualizar instâncias existentes. Você pode aplicar o patch escolhendo um cluster no console do [Amazon RDS](#), escolhendo Cluster Actions (Ações do cluster) e depois selecionando Upgrade Now (Atualizar agora). O patch requer uma reinicialização do banco de dados com um tempo de inatividade normalmente durando de 5 a 30 segundos, após o qual você pode continuar usando seus clusters de banco de dados. Esse patch está usando um modelo de patches de cluster em que todos os nós em um cluster do Aurora recebem patches ao mesmo tempo.

Novos recursos

- Auditoria avançada: o Aurora MySQL fornece um recurso de auditoria avançada de alta performance que você pode usar para auditar a atividade do banco de dados. Para obter mais informações sobre como habilitar e usar a auditoria, consulte [Como utilizar a auditoria avançada em um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

- Correção de um problema com a indexação espacial ao criar uma coluna e adicionar um índice a ela na mesma instrução.
- Correção de um problema em que as estatísticas espaciais não são persistentes no reinício do cluster de banco de dados.

Atualizações do mecanismo de banco de dados do Aurora MySQL: 2016-12-14 (versão 1.10) (obsoleta)

Versão: 1.10

Novos recursos

- Patch com tempo de inatividade zero – Esse recurso permite que uma instância de banco de dados seja corrigida sem qualquer tempo de inatividade. Ou seja, as atualizações do banco de dados são realizadas sem desconectar aplicativos cliente ou reiniciar o banco de dados. Essa abordagem aumenta a disponibilidade dos seus clusters de bancos de dados Aurora durante a janela de manutenção. Observe que dados temporários como esse no esquema de performance são reiniciados durante o processo de atualização. Esse recurso aplica-se aos patches entregues no serviço durante uma janela de manutenção, bem como patches iniciados pelo usuário.

Quando um patch é iniciado, o serviço garante que não haja bloqueios abertos, transações ou tabelas temporárias e, em seguida, aguarda uma janela adequada durante a qual o banco de dados pode ser corrigido e reiniciado. As sessões de aplicativos são preservadas, embora haja uma queda no throughput enquanto o patch está em andamento (por aproximadamente 5 segundos). Se nenhuma janela adequada for encontrada, então a aplicação de patch assumirá o comportamento padrão de aplicação de patch.

A aplicação de patch com tempo de inatividade zero ocorre em uma base de melhor esforço, sujeito a certas limitações, conforme descrito a seguir:

- Esse recurso é atualmente aplicável para aplicação de patch em clusters de banco de um único nó ou em instâncias de gravador em clusters de banco de dados com vários nós.
- Conexões SSL não têm suporte em conjunto com esse recurso. Se houver conexões SSL ativas, o Amazon Aurora MySQL não executará um patch com tempo de inatividade zero e, em vez disso, tentará periodicamente verificar se as conexões SSL foram encerradas. Se tiverem, a aplicação de patch com tempo de inatividade zero continuará. Se as conexões SSL persistirem após mais de alguns segundos, a aplicação de patch com tempo de inatividade padrão continuará.
- O recurso está disponível no Aurora versão 1.10 e posterior. Identificaremos todos os lançamentos ou patches que não poderem ser aplicados usando a aplicação de patch com tempo de inatividade zero.
- Esse recurso não será aplicável se a replicação com base em registro em log binário estiver ativa.
- Indexação espacial: a indexação espacial melhora a performance das consultas em conjuntos de dados grandes para consultas que utilizam dados espaciais. Para obter mais informações sobre o uso da indexação espacial, consulte [Amazon Aurora MySQL e dados espaciais](#) no Guia do usuário do Amazon Aurora.

Esse recurso está desabilitado por padrão e pode ser ativado habilitando o modo de laboratório Aurora. Para obter informações, consulte [Modo de laboratório do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

- Melhorias no pipeline de replicação – agora, o Aurora MySQL usa um mecanismo melhorado para aplicar atualizações de stream de log ao cache de buffer de uma Réplica do Aurora. Esse recurso melhora a performance e a estabilidade de leitura nas Réplicas do Aurora quando há uma carga de gravação pesada no mestre, bem como uma carga de leitura significativa na Réplica. Esse recurso está habilitado por padrão.
- Melhoria do throughput para workloads com leituras em cache – agora, o Aurora MySQL usa um algoritmo simultâneo sem trava para implementar visualizações de leitura, o que gera um melhor throughput para consultas de leitura servidas pelo cache de buffer. Como resultado dessa e de outras melhorias, o Amazon Aurora MySQL pode atingir uma taxa de transferência de até 625 mil leituras por segundo, em comparação com 164 mil leituras por segundo do MySQL 5.7 para uma carga de trabalho somente do Select. SysBench

- Melhoria da produtividade para workloads com disputa de linhas dinâmica: o Aurora MySQL usa um novo algoritmo de liberação de bloqueio que melhora a performance, especialmente quando existe uma disputa de páginas dinâmica (ou seja, muitas transações disputando as linhas na mesma página). Em testes com o parâmetro de comparação TPC-C, isso pode resultar em uma melhoria de throughput de até 16x em transações por minuto em relação ao MySQL 5.7. Esse recurso está desabilitado por padrão e pode ser ativado habilitando o modo de laboratório Aurora. Para obter informações, consulte [Modo de laboratório do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

- A velocidade de replicação do cache de índice de pesquisa de texto completo foi melhorada atualizando o cache somente após uma solicitação de leitura para uma Réplica do Aurora. Essa abordagem evita qualquer leitura do disco pelo thread de replicação.
- Correção de um problema em que a invalidação do cache do dicionário não funciona em uma Réplica do Aurora para tabelas que tenham um caractere especial no nome do banco de dados ou no nome da tabela.
- Correção de um problema de STUCK IO durante a migração de dados para nós de armazenamento distribuídos quando o gerenciamento dinâmico de armazenamento está habilitado.
- Correção de um problema no gerenciador de bloqueio em que uma verificação de declaração falha para o thread de espera de bloqueio de transação ao se preparar para reverter ou confirmar uma transação.
- Correção de um problema ao abrir uma tabela de dicionário corrompida atualizando corretamente a contagem de referência para as entradas da tabela de dicionário.
- Correção de um erro em que o ponto de leitura mínimo do cluster do banco de dados pode ser mantido por Réplicas do Aurora lentas.
- Correção de uma possível perda de memória no cache de consultas.
- Correção de um erro em que uma Replicação do Aurora coloca um bloqueio em nível de linha em uma tabela quando uma consulta é usada em uma instrução IF de um procedimento armazenado.

Integração de correções de bugs do MySQL

- UNION de tabelas derivadas retorna resultados incorretos com cláusulas '1=0/false'. (Bug nº 69471)
- O servidor falha em ITEM_FUNC_GROUP_CONCAT::FIX_FIELDS na segunda execução do procedimento armazenado. (Bug nº 20755389)
- Evite que consultas MySQL fiquem paradas por muito tempo durante a sincronização do cache de FTS com o disco, descarregando a tarefa de sincronização de cache em um thread separado, assim que o tamanho do cache atingir 10% do tamanho total. (Bug nº 22516559, Nº 73816)

Atualizações do mecanismo de banco de dados do Aurora MySQL: 2016-11-10 (versão 1.9.0, 1.9.1) (obsoleta)

Versão: 1.9.0, 1.9.1

Novos recursos

- Melhor compilação de índice – A implementação da construção de índices secundários agora funciona criando o índice de baixo para cima, o que elimina as divisões de página desnecessárias. Isso pode reduzir o tempo necessário para criar um índice ou reconstruir uma tabela em até 75% (com base em uma classe de instância de banco de dados db.r3.8xlarge). Esse recurso estava no modo de laboratório no Aurora MySQL versão 1.7 e está habilitado por padrão no Aurora versão 1.9 e posterior. Para obter informações, consulte [Modo de laboratório do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Compactação de bloqueio (modo de laboratório) – Essa implementação reduz significativamente a quantidade de memória que o gerenciador de bloqueio consome em até 66%. O gerenciador de bloqueios pode adquirir mais bloqueios de linha sem encontrar uma out-of-memory exceção. Esse recurso está desabilitado por padrão e pode ser ativado habilitando o modo de laboratório Aurora. Para obter informações, consulte [Modo de laboratório do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Esquema de performance – o Aurora MySQL agora inclui suporte para o esquema de performance com impacto mínimo sobre a performance. Em nossos testes usando SysBench, habilitar o esquema de desempenho pode degradar o desempenho do MySQL em até 60%.

SysBench o teste de um cluster de banco de dados Aurora mostrou um impacto no desempenho que é 4 vezes menor do que o MySQL. A execução da classe de db.r3.8xlarge instância

de banco de dados resultou em 100 mil SQL writes/sec e mais de 550 mil leituras de SQL por segundo, mesmo com o esquema de desempenho ativado.

- Melhoria na disputa de linhas dinâmica – esse recurso reduz a utilização da CPU e aumenta o throughput quando um pequeno número de linhas quentes são acessadas por um grande número de conexões. Esse recurso também elimina `error 188` quando existe uma disputa de linhas dinâmica.
- out-of-memory Tratamento aprimorado — Quando instruções SQL de bloqueio não essenciais são executadas e o pool de memória reservada é violado, o Aurora força a reversão dessas instruções SQL. Esse recurso libera memória e evita falhas no motor devido a out-of-memory exceções.
- Seletor de leitura inteligente — Essa implementação melhora a latência de leitura escolhendo o segmento de armazenamento ideal entre diferentes segmentos para cada leitura, resultando em melhor taxa de transferência de leitura. SysBench os testes mostraram um aumento de desempenho de até 27% para cargas de trabalho de gravação.

Melhorias

- Correção de um problema em que uma Réplica do Aurora encontra um bloqueio compartilhado durante a inicialização do mecanismo.
- Correção de uma possível falha em uma Réplica do Aurora quando o ponteiro de visualização lida no sistema de depuração é NULL.

Atualizações do mecanismo de banco de dados do Aurora MySQL: 2016-10-26 (versão 1.8.1) (obsoleta)

Versão: 1.8.1

Melhorias

- Correção de um problema em que inserções em massa que utilizam triggers que invocam procedimento do AWS Lambda falham.
- Correção de um problema em que a migração do catálogo falha quando a confirmação automática está desligada globalmente.
- Resolveu uma falha de conexão com a Aurora ao usar SSL e aprimorou o grupo Diffie-Hellman para lidar com ataques. LogJam

Integração de correções de bugs do MySQL

- O OpenSSL alterou os parâmetros de comprimento da chave Diffie-Hellman devido ao problema. LogJam (Bug nº 18367167)

Atualizações do mecanismo de banco de dados do Aurora MySQL: 2016-10-18 (versão 1.8) (obsoleta)

Versão: 1.8

Novos recursos

- AWS Lambda integração — agora você pode invocar de forma assíncrona uma função AWS Lambda de um cluster de banco de dados Aurora usando o procedimento. `mysql.lambda_async` Para obter mais informações, consulte [Invocar uma função do Lambda a partir de um cluster de banco de dados do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Carregar dados do Amazon S3 – Agora, você pode carregar arquivos de texto ou XML de um bucket do Amazon S3 no seu cluster de bancos de dados Aurora usando os comandos `LOAD DATA FROM S3` ou `LOAD XML FROM S3`. Para obter mais informações, consulte [Carregar dados em um cluster de banco de dados do Amazon Aurora MySQL a partir de arquivos de texto em um bucket do Amazon S3](#) no Guia do usuário do Amazon Aurora.
- Migração do catálogo – Agora, o Aurora persiste os metadados do catálogo no volume do cluster para dar suporte ao versionamento. Isso permite uma migração de catálogo contínua entre versões e restaurações.
- Manutenção e aplicação de patches em nível de cluster – agora, o Aurora gerencia atualizações de manutenção para um cluster de banco de dados inteiro. Para obter mais informações, consulte [Manutenção de um cluster de banco de dados do Amazon Aurora](#) no Guia do usuário do Amazon Aurora.

Melhorias

- Correção de um problema em que uma Réplica do Aurora falha ao não conceder um bloqueio de metadados a uma tabela DDL dinâmica.

- Réplica do Aurora com permissão para modificar tabelas não InnoDB para facilitar o rodízio dos arquivos de log CSV lentos e gerais, em que `log_output=TABLE`.
- Correção de um atraso ao atualizar estatísticas da instância primária para uma Réplica do Aurora. Sem essa correção, as estatísticas da Réplica do Aurora podem ficar fora de sincronia com as estatísticas da instância primária e resultar em um plano de consulta diferente (e possivelmente com performance insuficiente) em uma Réplica do Aurora.
- Correção de uma condição de corrida que garante que uma Réplica do Aurora não adquira bloqueios.
- Correção de um cenário raro em que uma Réplica do Aurora que efetua ou cancela o registro na instância primária pode falhar.
- Correção de uma condição de corrida que pode provocar um deadlock em instâncias `db.r3.large` ao abrir ou fechar um volume.
- Foi corrigido um out-of-memory problema que pode ocorrer devido a uma combinação de uma grande carga de trabalho de gravação e falhas no serviço de armazenamento distribuído Aurora.
- Correção de um problema com alto consumo de CPU devido ao giro do thread de limpeza na presença de uma transação de longa execução.
- Correção de um problema ao executar consultas de esquema de informações para obter informações sobre bloqueios em carga pesada.
- Correção de um problema com um processo de diagnóstico que, em raros casos, pode fazer com que gravações do Aurora em nós de armazenamento fiquem obsoletas e sejam reiniciadas/façam failover.
- Corrigida uma condição em que uma tabela criada com êxito pode ser excluída durante a recuperação de uma falha caso ela tenha ocorrido enquanto uma instrução `CREATE TABLE [if not exists]` estava sendo processada.
- Correção de um caso em que o procedimento de rotação do log está interrompido quando o log geral e o log lento não são armazenados no disco usando a mitigação de catálogo.
- Correção de uma falha quando um usuário cria uma tabela temporária dentro de uma função definida pelo usuário e, em seguida, usa essa função na lista de seleção da consulta.
- Correção de um falha que ocorreu ao reproduzir eventos GTID. O GTID não é compatível com o Aurora MySQL.

Integração de correções de bugs do MySQL:

- Ao descartar todos os índices em uma coluna com vários índices, o InnoDB falhou ao bloquear uma operação DROP INDEX quando uma restrição de chave estrangeira requer um índice. (Bug nº 16896810)
- Resolvida falha de restrição de chave estrangeira adicionada. (Bug nº 16413976)
- Correção de uma falha ao buscar um cursor em um procedimento armazenado e analisar ou liberar a tabela ao mesmo tempo. (Bug nº 18158639)
- Correção de um bug de incremento automático quando um usuário altera uma tabela para alterar o valor de AUTO_INCREMENT para menos do que o valor de coluna de incremento automático máximo. (Bug nº 16310273)

Atualizações do mecanismo de banco de dados do Aurora MySQL: 2016-09-20 (versão 1.7.1) (obsoleta)

Versão: 1.7.1

Melhorias

- Corrige um problema em que uma Réplica do Aurora falha se o cache de pesquisa de texto completo do InnoDB está cheio.
- Corrige um problema em que o mecanismo de banco de dados falha se um thread de trabalho no pool de threads aguarda ele mesmo.
- Corrige um problema em que uma Réplica do Aurora trava se um bloqueio de metadados em uma tabela causa um deadlock.
- Corrige um problema em que o mecanismo de banco de dados falha devido a uma condição de corrida entre dois segmentos de trabalho no pool de threads.
- Corrige um problema em que ocorre um failover desnecessário em condições de carga pesada quando o agente de monitoramento não detecta o avanço de operações de gravação no subsistema de armazenamento distribuído.

Atualizações do mecanismo de banco de dados do Aurora MySQL: 2016-08-30 (versão 1.7.0) (obsoleta)

Versão: 1.7.0

Novos recursos

- Programador com reconhecimento para NUMA – O programador de tarefas do mecanismo do Aurora MySQL agora tem reconhecimento para NUMA (acesso não uniforme a memória). Isso minimiza a contenção de soquetes entre CPUs, resultando em um throughput de performance melhorado para a classe de instância de banco de dados `db.r3.8xlarge`.
- A leitura antecipada em paralelo funciona de forma assíncrona em segundo plano: a leitura antecipada em paralelo foi revisada para melhorar a performance usando um thread dedicado para reduzir a contenção de threads.
- Melhor compilação de índice (modo de laboratório) – A implementação da construção de índices secundários agora funciona criando o índice de baixo para cima, o que elimina as divisões de página desnecessárias. Isso pode reduzir o tempo necessário para criar um índice ou reconstruir uma tabela. Esse recurso está desabilitado por padrão e pode ser ativado habilitando o modo de laboratório Aurora. Para obter informações, consulte [Modo de laboratório do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Melhorias

- Correção de um problema em que o estabelecimento de uma conexão demoraria muito se houvesse um aumento no número de conexões solicitadas para uma instância.
- Correção de um problema em que ocorria uma falha se `ALTER TABLE` fosse executada em uma tabela particionada que não utilizasse o InnoDB.
- Correção de um problema em que uma workload de gravação pesada poderia causar um failover.
- Correção de uma declaração incorreta que causaria uma falha se `RENAME TABLE` fosse executada em uma tabela particionada.
- Maior estabilidade ao reverter uma transação durante workloads de inserção pesadas.
- Correção de um problema em que os índices de pesquisa de texto completo não eram viáveis em uma Réplica do Aurora.

Integração de correções de bugs do MySQL

- Melhoria de escalabilidade ao particionar o bloqueio LOCK_grant. (Port WL #8355)
- Abrir o cursor em SELECT no procedimento armazenado causa segfault. (Port Bug#16499751)
- O MySQL fornece o resultado incorreto com algum uso especial. (Bug nº 11751794)
- Falha em GET_SEL_ARG_FOR_KEYPART – causada pelo patch para o Bug nº 11751794. (Bug nº 16208709)
- Resultados incorretos para uma consulta simples com GROUP BY. (Bug nº 17909656)
- Linhas extras na consulta de semijunção com predicados de intervalo. (Bug nº 16221623)
- A inclusão de uma cláusula ORDER BY após uma subconsulta IN pode fazer com que linhas duplicadas sejam retornadas. (Bug nº 16308085)
- Falha com explicação para uma consulta com varredura solta para GROUP BY, MyISAM. (Bug nº 16222245)
- A varredura de índice solta com o predicado quoted int retorna dados aleatórios. (Bug nº 16394084)
- Se o otimizador estivesse usando uma varredura de índice solta, o servidor poderia ser encerrado ao tentar criar uma tabela temporária. (Bug nº 16436567)
- COUNT(DISTINCT) não deve contar valores NULL, mas eles foram contados quando o otimizador usou a varredura de índice solta. (Bug nº 17222452)
- Se uma consulta teve MIN()/MAX() e aggregate_function(DISTINCT) (por exemplo, SUM(DISTINCT)) e foi executada usando a varredura de índice solta, os valores de resultados de MIN()/MAX() foram definidos incorretamente. (Bug nº 17217128)

Atualizações do mecanismo de banco de dados do Aurora MySQL: 2016-06-01 (versão 1.6.5) (obsoleta)

Versão: 1.6.5

Novos recursos

- Armazenamento eficiente de logs binários – O armazenamento eficiente de logs binários agora está habilitado por padrão para todos clusters de banco de dados Aurora MySQL e não é configurável. O armazenamento eficiente de logs binários foi introduzido na atualização de abril de

2016. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL: 2016-04-06 \(versão 1.6\) \(obsoleta\)](#).

Melhorias

- Melhor estabilidade para Réplicas do Aurora quando a instância primária se depara com uma workload pesada.
- Maior estabilidade para Réplicas do Aurora ao executar consultas em tabelas particionadas e tabelas com caracteres especiais no nome.
- Correção de problemas de conexão ao usar conexões seguras.

Integração de correções de bugs do MySQL

- SLAVE CAN'T CONTINUE REPLICATION AFTER MASTER'S CRASH RECOVERY (Port Bug #17632285)

Atualizações do mecanismo de banco de dados do Aurora MySQL: 2016-04-06 (versão 1.6) (obsoleta)

Versão: 1.6

Esta atualização inclui os seguintes aprimoramentos:

Novos recursos

- Leitura antecipada paralela – Agora, a leitura antecipada paralela está habilitada por padrão para todos os clusters de banco de dados Aurora MySQL e não é configurável. A Leitura antecipada paralela foi introduzida na atualização de dezembro de 2015. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL: 2015-12-03 \(versão 1.4\) \(obsoleta\)](#).

Além de permitir a leitura paralela por padrão, essa versão inclui as seguintes melhorias para a leitura antecipada paralela:

- Melhoria na lógica para que a leitura paralela seja menos agressiva, o que é benéfico quando seu cluster de banco de dados se depara com muitas workloads paralelas.

- Maior estabilidade em tabelas menores.
- Armazenamento eficiente de logs binários (modo laboratório) – Agora, os arquivos de log binários do MySQL são armazenados de forma mais eficiente no Aurora MySQL. A nova implementação de armazenamento permite que os arquivos de log binários sejam excluídos muito mais cedo e melhora a performance do sistema para uma instância em um cluster de banco de dados Aurora MySQL que é um mestre de replicação de log binário.

Para habilitar o armazenamento eficiente de logs binários, defina o parâmetro `aurora_lab_mode` como 1 no parameter group para a sua instância primária ou Réplica do Aurora. O parâmetro `aurora_lab_mode` é um parâmetro em nível de instância que está no parameter group `default.aurora5.6` por padrão. Para obter informações sobre como modificar um grupo de parâmetros de banco de dados, consulte [Modificar parâmetros em um grupo de parâmetros de banco de dados](#) no Guia do usuário do Amazon Aurora. Para obter informações sobre grupos de parâmetros e o Aurora MySQL, consulte [Parâmetros de configuração do Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Basta ativar o armazenamento eficiente de logs binários para instâncias em um cluster de banco de dados Aurora MySQL que sejam instâncias mestres de replicação de logs binários do MySQL.

- Variável de sistema `AURORA_VERSION` – Agora, é possível obter a versão Aurora do seu cluster de banco de dados Aurora MySQL consultando a variável do sistema `AURORA_VERSION`.

Para obter a versão do Aurora, use uma das seguintes consultas:

```
select AURORA_VERSION();
select @@aurora_version;
show variables like '%version';
```

Você também pode ver a versão do Aurora no Console de gerenciamento da AWS ao modificar um cluster de banco de dados ou ao chamar o [describe-db-engine-versions](#) AWS CLI comando ou a operação da API [Descreve DBEngine Versions](#).

- Métrica de uso de memória do gerenciador de bloqueios – Agora, informações sobre o uso da memória do gerenciador de bloqueios estão disponíveis como uma métrica.

Para obter a métrica de uso de memória do gerenciador de bloqueios, use uma das seguintes consultas:

```
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');
```

Melhorias

- Maior estabilidade durante a recuperação de transações XA e logs binários.
- Correção de um problema de memória resultante de um grande número de conexões.
- Maior precisão nas seguintes métricas: Read Throughput, Read IOPS, Read Latency, Write Throughput, Write IOPS, Write Latency e Disk Queue Depth.
- Correção de um problema de estabilidade que causa a inicialização lenta de instâncias grandes após uma falha.
- Maior simultaneidade no dicionário de dados em relação aos mecanismos de sincronização e ao despejo do cache.
- Melhorias de estabilidade e performance para Réplicas do Aurora:
 - Correção de um problema de estabilidade para Réplicas do Aurora durante workloads de gravação pesadas ou intermitentes para a instância primária.
 - Melhoria no atraso das réplicas para as instâncias db.r3.4xlarge e db.r3.8xlarge.
 - Melhoria da performance ao reduzir a disputa entre a aplicação de registros de log e leituras simultâneas em uma Réplica do Aurora.
 - Correção de um problema para atualizar estatísticas em Réplicas do Aurora para estatísticas recém-criadas ou atualizadas.
 - Maior estabilidade para Réplicas do Aurora quando existem várias transações na instância primária e leituras simultâneas nas Réplicas do Aurora entre os mesmos dados.
 - Maior estabilidade para Réplicas do Aurora ao executar instruções UPDATE e DELETE com instruções JOIN.
 - Maior estabilidade para Réplicas do Aurora ao executar instruções INSERT ... SELECT.

Integração de correções de bugs do MySQL

- BACKPORT Bug 18694052 FIX FOR ASSERTION '!M_ORDERED_REC_BUFFER' FAILED TO 5.6 (Bug de porta 18305270)
- SEGV IN MEMCPY(), HA_PARTITION::POSITION (Bug de porta # 18383840)

- WRONG RESULTS WITH PARTITIONING, INDEX_MERGE AND NO PK (Bug de porta # 18167648)
- FLUSH TABLES FOR EXPORT: ASSERTION IN HA_PARTITION::EXTRA (Bug de porta # 16943907)
- SERVER CRASH IN VIRTUAL HA_ROWS HANDLER::MULTI_RANGE_READ_INFO_CONST (Bug de Porta # 16164031)
- RANGE OPTIMIZER CRASHES IN SEL_ARG::RB_INSERT() (Bug de Porta # 16241773)

Atualizações do mecanismo de banco de dados do Aurora MySQL: 2016-01-11 (versão 1.5) (obsoleta)

Versão: 1.5

Esta atualização inclui os seguintes aprimoramentos:

Melhorias

- Correção de uma pausa de 10 segundos de operações de gravação para instâncias inativas durante implementações de armazenamento do Aurora.
- Agora, a leitura antecipada lógica funciona quando `innodb_file_per_table` está definido como No. Para obter mais informações sobre leitura antecipada lógica, consulte [Atualizações do mecanismo de banco de dados do Aurora MySQL: 2015-12-03 \(versão 1.4\) \(obsoleta\)](#).
- Correção de problemas com Réplicas do Aurora que se reconectam à instância primária. Essa melhoria também corrige um problema quando você especifica um valor grande para o parâmetro `quantity` ao testar falhas de Réplicas do Aurora usando consultas de injeção de falhas. Para obter mais informações, consulte [Teste de falha em uma réplica do Aurora](#) no Guia do usuário do Amazon Aurora.
- Monitoramento aprimorado de Réplicas do Aurora que se atrasam e são reiniciadas.
- Correção de um problema que fazia com que uma Réplica do Aurora atrasasse, tivesse seu registro cancelado e reiniciasse.
- Correção de um problema quando você executa o comando `show innodb status` durante um deadlock.
- Correção de um problema com failovers para instâncias maiores durante o alto throughput de gravação.

Integração de correções de bugs do MySQL

- Resolução de uma correção incompleta na pesquisa de texto completa MySQL que afeta tabelas em que o nome do banco de dados começa com um dígito. (Bug nº 17607956 de porta)

Atualizações do mecanismo de banco de dados do Aurora MySQL: 2015-12-03 (versão 1.4) (obsoleta)

Versão: 1.4

Esta atualização inclui os seguintes aprimoramentos:

Novos recursos

- Inserção rápida – Acelera inserções paralelas classificadas por chave primária. Para obter mais informações, consulte [Melhorias de performance do Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Performance de leitura de conjuntos de dados grandes: o Aurora MySQL detecta automaticamente uma workload pesada de E/S e executa mais threads para aumentar a performance do cluster de banco de dados. O programador do Aurora analisa a atividade de E/S e decide ajustar dinamicamente o número ideal de threads no sistema, ajustando rapidamente as workloads com E/S pesada e CPU pesada com baixa sobrecarga.
- Leitura antecipada paralela: melhora a performance de varreduras de Árvores B que são muito grandes para a memória disponível na sua instância primária ou Réplica do Aurora (incluindo consultas de intervalo). A leitura antecipada paralela detecta automaticamente os padrões de leitura da páginas e faz a pré-busca de páginas no cache de buffer antes que estas sejam necessárias. A leitura antecipada paralela trabalha com várias tabelas ao mesmo tempo dentro da mesma transação.

Melhorias:

- Correção de breves problemas de disponibilidade do banco de dados Aurora durante implantações de armazenamento do Aurora.
- Aplicação correta do limite `max_connection`.

- Melhoria da limpeza de logs binários quando o Aurora é o mestre de logs binários e o banco de dados está sendo reiniciado após uma carga de dados pesada.
- Correção de problemas de gerenciamento de memória com o cache da tabela.
- Suporte adicionado para páginas enormes no cache do buffer de memória compartilhada para uma recuperação mais rápida.
- Correção de um problema em que o armazenamento local de threads não é inicializado.
- Permissão de conexões de 16K por padrão.
- Pool de threads dinâmico para workloads com E/S pesada.
- Correção de um problema com a invalidação correta de visualizações que envolvem UNION no cache de consulta.
- Correção de um problema de estabilidade com o thread de estatísticas de dicionário.
- Correção de uma perda de memória no subsistema de dicionário relacionada ao despejo do cache.
- Correção de problema de alta latência de leitura nas Réplicas do Aurora quando há uma carga muito baixa de gravação no mestre.
- Problemas de estabilidade corrigidos em Réplicas do Aurora ao realizar operações em tabelas partidas DDL, como ALTER TABLE ... REORGANIZE PARTITION, no mestre.
- Problemas de estabilidade corrigidos nas Réplicas do Aurora durante o crescimento do volume.
- Problema de performance corrigido para varreduras em índices não clusterizados em Réplicas do Aurora.
- Correção de problema de estabilidade que causam um atraso de Réplicas do Aurora e fazem com que elas acabem tendo o registro cancelado e reiniciadas.

Integração de correções de bugs do MySQL

- SEGV em FTSPARSE(). (Bug nº 16446108)
- O dicionário de dados do InnoDB não é atualizado ao renomear a coluna. (Bug nº 19465984)
- Falha do FTS após a renomeação da tabela para um banco de dados diferente. (Bug nº 16834860)
- Uma falha na preparação do trigger em tabelas truncadas causa o erro 1054. (Bug nº 18596756)
- Alterações de metadados podem causar problemas com a execução de disparadores. (Bug nº 18684393)
- A materialização não é escolhida para o campo UTF8 VARCHAR longo. (Bug nº 17566396)

- Plano de execução insatisfatório quando ORDER BY com limite X. (Bug n.º 16697792)
- Bug de backport #11765744 PARA 5.1, 5.5 E 5.6. (Bug n.º 17083851)
- Problema de mutex no SQL/SQL_SHOW.CC resultando em. SIG6 Origem provavelmente FILL_VARIABLES. (Bug n.º 20788853)
- Bug de backport #18008907 para versões 5.5+. (Bug n.º 18903155)
- Correção de um erro de excedente de pilha no MySQL 5.7. (Bug n.º 19678930)

Atualizações do mecanismo de banco de dados do Aurora MySQL: 2015-10-16 (versão 1.2, 1.3) (obsoleta)

Versões: 1.2, 1.3

Esta atualização inclui os seguintes aprimoramentos:

Correções

- out-of-memory Problema resolvido no novo gerenciador de bloqueios com transações de longa duração
- Resolução de uma vulnerabilidade de segurança ao replicar com bancos de dados que não fossem do RDS for MySQL
- Atualização para garantir que gravações de quorum sejam repetidas corretamente com falhas de armazenamento
- Atualização para informar o atraso de réplicas com mais precisão
- Melhor performance ao reduzir a disputa quando muitas transações simultâneas estão tentando modificar a mesma linha
- Resolução da invalidação do cache de consulta para visualizações que são criadas com a união de duas tabelas
- Cache de consulta desabilitado para transações com isolamento de UNCOMMITTED_READ

Melhorias

- Melhor performance para consultas de catálogo lentas em caches quentes
- Simultaneidade melhorada nas estatísticas do dicionário

- Maior estabilidade para o novo gerenciador de recursos do cache de consulta, gerenciamento de extensões, arquivos armazenados no armazenamento inteligente do Amazon Aurora e gravações em lote de registros de log

Integração de correções de bugs do MySQL

- Eliminar uma consulta dentro de innodb faz com que ela eventualmente falhe com uma declaração. (Bug nº 1608883)
- Devido a uma falha ao criar um novo thread para o programador de eventos, a execução do evento ou uma nova conexão, nenhuma mensagem foi gravada no log de erros. (Bug nº 16865959)
- Se uma conexão alterou seu banco de dados padrão e, simultaneamente, outra conexão executou SHOW PROCESSLIST, a segunda conexão poderia acessar uma memória inválida ao tentar exibir a memória de banco de dados padrão da primeira conexão. (Bug nº 11765252)
- Por padrão, PURGE BINARY LOGS não remove arquivos de log binários em uso ou ativos, mas não forneceu nenhum aviso quando isso ocorreu. (Bug nº 13727933)
- Para algumas instruções, perdas de memória poderão resultar quando o otimizador tiver removido cláusulas de subconsulta desnecessárias. (Bug nº 15875919)
- Durante o encerramento, o servidor poderia tentar bloquear um mutex não inicializado. (Bug nº 16016493)
- Uma declaração preparada que usava GROUP_CONCAT() e uma cláusula ORDER BY que nomeava várias colunas poderia causar o encerramento do servidor. (Bug nº 16075310)
- A instrumentação do esquema de performance estava ausente para threads de operador de réplica. (Bug nº 16083949)
- STOP SLAVE pode causar um deadlock quando emitido simultaneamente com uma instrução como SHOW STATUS que recuperou os valores de uma ou mais das variáveis de status Slave_retried_transactions, Slave_heartbeat_period, Slave_received_heartbeats, Slave_last_heartbeat ou Slave_running. (Bug nº 16088188)
- Uma consulta de texto completo usando o modo booliano poderia retornar zero resultados em alguns casos em que o termo de pesquisa fosse uma expressão entre aspas. (Bug nº 16206253)
- A tentativa do otimizador de remover cláusulas de subconsulta redundantes gerou uma declaração ao executar uma instrução preparada com uma subconsulta na cláusula ON de uma junção em uma subconsulta. (Bug nº 16318585)

- GROUP_CONCAT instável, falha em ITEM_SUM::CLEAN_UP_AFTER_REMOVAL. (Bug nº 16347450)
- A tentativa de substituir a lista de palavras irrelevantes padrão de pesquisa de texto completo (FTS) do InnoDB, criando uma tabela do InnoDB com a mesma estrutura que INFORMATION_SCHEMA.INNODB_FT_DEFAULT_STOPWORD, resultaria em um erro. (Bug nº 16373868)
- Depois que o thread de cliente em um operador realizou FLUSH TABLES WITH READ LOCK e, depois, algumas atualizações no mestre, o operador travou ao executar SHOW SLAVE STATUS. (Bug nº 16387720)
- Ao analisar uma string de pesquisa delimitada, como "abc-def" em uma pesquisa de texto completo, o InnoDB agora usa os mesmos delimitadores de palavras que o MyISAM. (Bug nº 16419661)
- Falha em FTS_AST_TERM_SET_WILDCARD. (Bug nº 16429306)
- SEGFAULT em FTS_AST_VISIT() para teste de FTS RQG. (Bug nº 16435855)
- Para compilações de depuração, quando o otimizador removía um Item_ref apontando para uma subconsulta, ele causava um encerramento do servidor. (Bug nº 16509874)
- A pesquisa de texto completo em tabelas do InnoDB falhava em pesquisas por expressões literais combinadas com operadores + ou -. (Bug nº 16516193)
- START SLAVE falhou quando o servidor foi iniciado com as opções -- master-info-repository=TABLE relay-log-info-repository=TABLE e com a confirmação automática definida como 0, junto com. --skip-slave-start (Bug nº 16533802)
- Resultados de pesquisa de texto completo (FTS) muito grandes do InnoDB podem consumir uma quantidade excessiva de memória. (Bug nº 16625973)
- Em compilações de depuração, uma declaração poderia ocorrer em OPT_CHECK_ORDER_BY ao usar um binário diretamente em uma string de pesquisa, pois binários podem incluir bytes NULL e outros caracteres não significativos. (Bug nº 16766016)
- Para algumas instruções, perdas de memória poderão resultar quando o otimizador tiver removido cláusulas de subconsulta desnecessárias. (Bug nº 16807641)
- Era possível causar um deadlock após a emissão de FLUSH TABLES WITH READ LOCK, emitindo STOP SLAVE em uma nova conexão com o operador e, depois, emitindo SHOW SLAVE STATUS usando a conexão original. (Bug nº 16856735)
- GROUP_CONCAT() com um separador inválido pode causar um encerramento do servidor. (Bug nº 16870783)

- O servidor fez bloqueios excessivos nos mutexes LOCK_active_mi e active_mi->rli->data_lock para qualquer instrução "padrão" SHOW STATUS LIKE, mesmo quando esse padrão não correspondeu às variáveis de status que usam esses mutexes (Slave_heartbeat_period, Slave_last_heartbeat, Slave_received_heartbeats, Slave_retried_transactions, Slave_running). (Bug nº 16904035)
- Uma pesquisa de texto completo usando o modificador IN BOOLEAN MODE resultaria em uma falha de asserção. (Bug #16927092)
- A pesquisa de texto completo em tabelas do InnoDB falhava em pesquisas que usavam o operador booleano +. (Bug #17280122)
- Deadlock em quatro direções: zombies, purging binlogs, show processlist, show binlogs. (Bug #17283409)
- Quando um thread SQL que estava aguardando um bloqueio de confirmação foi eliminado e reiniciado, ele fez com que uma transação fosse ignorada no operador. (Bug #17450876)
- Uma falta de pesquisa de texto completo do InnoDB ocorreria devido a um token "sem fim". A string e o comprimento da string devem ser transmitidos para comparação de strings. (Bug #17659310)
- Um grande número de tabelas particionadas do InnoDB poderia consumir muito mais memória quando usado no MySQL 5.6 ou 5.7 do que a memória usada pelas mesmas tabelas em versões anteriores do MySQL Server. (Bug #17780517)
- Para consultas de texto completo, uma falha ao verificar se num_token é menor que max_proximity_item poderia resultar em uma declaração. (Bug #18233051)
- Determinadas consultas para as tabelas INFORMATION_SCHEMA TABLES e COLUMNS poderiam levar a um uso excessivo de memória quando houvesse um grande número de tabelas do InnoDB vazias. (Bug #18592390)
- Ao confirmar uma transação, um sinalizador agora é usado para verificar se um thread foi criado, em vez de verificar o thread propriamente dito, o que usa mais recursos, particularmente ao executar o servidor com master_info_repository=TABLE. (Bug #18684222)
- Se um thread de cliente em um operador executasse FLUSH TABLES WITH READ LOCK enquanto o mestre executava um DML, executar SHOW SLAVE STATUS no mesmo cliente era bloqueado, causando um deadlock. (Bug #19843808)
- Ordenar por resultado GROUP_CONCAT() poderia causar um encerramento do servidor. (Bug #19880368)

Atualizações do mecanismo de banco de dados do Aurora MySQL: 2015-08-24 (versão 1.1) (obsoleta)

Versão: 1.1

Esta atualização inclui os seguintes aprimoramentos:

- Melhorias na estabilidade da replicação ao replicar com um banco de dados MySQL (replicação de logs binários). Para obter informações sobre a replicação do Aurora MySQL com o MySQL, consulte [Replicação com o Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.
- Um limite de 1 gigabyte (GB) no tamanho dos logs de retransmissão acumulados para um cluster de banco de dados Aurora MySQL que é um operador de replicação. Isso melhora o gerenciamento de arquivos para clusters de bancos de dados Aurora.
- Melhorias de estabilidade nas áreas de leitura antecipada, relações recursivas de chave externa e replicação do Aurora.
- Integração de correções de bugs do MySQL.
 - Bancos de dados InnoDB com nomes que começam com um dígito causam um erro do analisador de pesquisa de texto completo (FTS). (Bug #17607956)
 - Pesquisas de texto completo do InnoDB falham em bancos de dados cujos nomes começavam com um dígito. (Bug #17161372)
 - Para bancos de dados InnoDB no Windows, o ID de objeto de pesquisa de texto completo (FTS) não está no formato hexadecimal esperado. (Bug #16559254)
 - Uma regressão de código introduzida no MySQL 5.6 afetou negativamente a performance de DROP TABLE e ALTER TABLE. Isso pode causar uma queda de performance entre o MySQL Server 5.5.x e 5.6.x. (Bug #16864741)
- Registro em log simplificado para reduzir o tamanho dos arquivos de log e a quantidade de armazenamento que eles exigem.

Bugs do MySQL corrigidos por atualizações do mecanismo de banco de dados Aurora MySQL

As seções a seguir identificam bugs do MySQL que foram corrigidos pelas atualizações do mecanismo de banco de dados Aurora MySQL.

Tópicos

- [Bugs do MySQL corrigidos por atualizações do mecanismo de banco de dados do Aurora MySQL 3.x](#)
- [Bugs do MySQL corrigidos por atualizações do mecanismo de banco de dados Aurora MySQL 2.x](#)
- [Bugs do MySQL corrigidos por atualizações do mecanismo de banco de dados Aurora MySQL 1.x](#)

Bugs do MySQL corrigidos por atualizações do mecanismo de banco de dados do Aurora MySQL 3.x

O Aurora versão compatível com MySQL 8.0 contém todas as correções de bug do MySQL por meio da versão correspondente de compatibilidade com o MySQL. A tabela a seguir identifica os bugs do MySQL adicionais que foram corrigidos por atualizações do mecanismo de banco de dados Aurora MySQL e em qual atualização eles foram corrigidos.

Atualização do mecanismo de banco de dados	Versão compatível com o MySQL	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados Aurora MySQL em	8.0.39	3.08.0	• Correção de um problema que fazia com que valores NULL fossem omitidos incorretamente do conjunto de resultados para determinadas consultas que têm operações JOIN e UNION. (Correção de bug da comunidade e #114301)

Atualização do mecanismo de banco de dados	Versão compatível com o MySQL	Versão	Erros corrigidos no MySQL
2024-11-18 (versão 3.08.0, compatível com MySQL 8.0.39)			

Atualização do mecanismo de banco de dados	Versão compatível com o MySQL	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados Aurora MySQL 2024-06-04 (versão 3.07.0) (Obsoleto)	8.0.36	3.07.0	• Correção de um problema em que o valor da linha de cache podia ser calculado incorretamente, causando uma falha durante a reinicialização do banco de dados em instâncias baseadas no Graviton. (Correção de bug da comunidade e #35479763) • Correção de um problema em que algumas instâncias de subconsultas em rotinas armazenadas nem sempre eram gerenciadas corretamente. (Correção de bug da comunidade e #35377192) • Correção um problema que podia causar maior uso da CPU devido à rotação do certificado do TLS em segundo plano (Correção de bug n.º 34284186 da comunidade). • Correção de um problema em que o InnoDB permitia a adição de colunas INSTANT às tabelas no esquema do sistema MySQL nas versões do Aurora MySQL anteriores à 3.05, o que podia fazer com que o servidor fechasse

Atualização do mecanismo de banco de dados	Versão compatível com o MySQL	Versão	Erros corrigidos no MySQL
			inesperadamente (reinicia lização da instância do banco de dados) após a atualização para o Aurora MySQL versão 3.05.0. (Correção de bug n.º 35625510 da comunidade).

Atualização do mecanismo de banco de dados	Versão compatível com o MySQL	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados Aurora MySQL 2024-03-07 (versão 3.06.0) (Obsoleto)	8.0.34	3.06.0	• Correção de um problema em que o valor da linha de cache podia ser calculado incorretamente, causando uma falha durante a reinicialização do banco de dados em uma instância baseada no Graviton. (Correção de bug da comunidade e #35479763) • Correção de um problema em que algumas instâncias de subconsultas em rotinas armazenadas nem sempre eram gerenciadas corretamente. (Correção de bug da comunidade e #35377192) • Correção um problema que podia causar maior uso da CPU devido à rotação do certificado do TLS em segundo plano. (Correção de bug nº 34284186 da comunidade) • Correção de um problema em que o InnoDB permitia a adição de colunas INSTANT às tabelas no esquema do sistema MySQL nas versões do Aurora MySQL anteriores à 3.05, o que podia fazer com que o servidor fechasse

Atualização do mecanismo de banco de dados	Versão compatível com o MySQL	Versão	Erros corrigidos no MySQL
			inesperadamente (reinicia a instância do banco de dados) após a atualização para o Aurora MySQL versão 3.05.0. (Correção de bug nº 35625510 da comunidade)
Atualizações do mecanismo de banco de dados Aurora MySQL 2024-01-31 (versão 3.05.2) (Obsoleto)	8.0.32	3.05.2	Correção de um problema em que <code>records_in_range</code> realizava um número excessivo de leituras de disco para operações INSERT, levando a um declínio gradual no desempenho. (Correção de bug da comunidade #34976138)

Atualização do mecanismo de banco de dados	Versão compatível com o MySQL	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados Aurora MySQL 2023-11-21 (versão 3.05.1) (Obsoleto)	8.0.32	3.05.1	Corrigido um problema no InnoDB quando, se uma tabela do MySQL em um esquema do sistema tivesse uma coluna adicionada <code>INSTANT ADD</code> entre as versões 3.01 do Aurora MySQL até as versões 3.04 do Aurora MySQL e após o upgrade do Aurora MySQL para a versão 3.05.0, essas tabelas resultariam no fechamento inesperado do servidor. DMLs (Correção de bug nº 35625510 da comunidade)
Atualizações do mecanismo de banco de dados Aurora MySQL 2023-10-25 (versão 3.05.0) (Obsoleto)	8.0.32	3.05.0	Corrigido um problema que pode causar maior utilização da CPU devido à rotação do certificado TLS em segundo plano (Correção de bug nº 34284186 da comunidade).

Atualização do mecanismo de banco de dados	Versão compatível com o MySQL	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados Aurora MySQL 2024-03-15 (versão 3.04.2, compatível com o MySQL 8.0.28)	8.0.28	3.04.2	- Correção de um problema em que o valor da linha de cache podia ser calculado incorretamente, causando uma falha durante a reinicialização do banco de dados em instâncias baseadas no Graviton. (Correção de bug da comunidade #35479763) - A execução repetida de uma rotina armazenada, tendo como subconsulta uma declaração SELECT contendo várias condições AND, OR ou XOR levava ao consumo excessivo e a uma possível exaustão eventual da memória virtual. (Correção de bug da comunidade #33852530)

Atualização do mecanismo de banco de dados	Versão compatível com o MySQL	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados Aurora MySQL 2023-11-13 (versão 3.04.1, compatível com MySQL 8.0.28)	8.0.28	3.04.1	Corrigido um problema que pode causar maior utilização da CPU devido à rotação do certificado TLS em segundo plano (Correção de bug nº 34284186 da comunidade).

Atualização do mecanismo de banco de dados	Versão compatível com o MySQL	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados Aurora MySQL 2023-07-31 (versão 3.04.0), compatível com o MySQL 8.0.28)	8.0.28	3.04.0	- Corrigido um problema em que um bloco de buffer que contém uma página de tabela temporária intrínseca era realocado durante a passagem da página, causando uma falha de declaração (Bug nº 33715694) - InnoDB: Impeça que operações online de DDL acessem a out-of-bounds memória (Bug# 34750489, Bug# 108925) - Corrigido um problema que às vezes pode produzir resultados de consulta incorretos ao processar instruções SQL complexas que consistem em várias expressões de tabela comuns aninhadas (CTEs) (Bug# 34572040, Bug# 34634469, Bug# 33856374)

Atualização do mecanismo de banco de dados	Versão compatível com o MySQL	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2023-12-08 (versão 3.03.3) (obsoleta)	8.0.26	3.03.3	Corrigido um problema que pode causar maior utilização da CPU devido à rotação do certificado TLS em segundo plano (Correção de bug nº 34284186 da comunidade).

Atualização do mecanismo de banco de dados	Versão compatível com o MySQL	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2023-08-29 (versão 3.03.2) (obsoleta)	8.0.26	3.03.2	- Corrigido um problema que às vezes pode produzir resultados de consulta incorretos ao processar instruções SQL complexas que consistem em várias expressões de tabela comuns aninhadas (CTEs) (Bug #34572040, Bug #34634469, Bug #33856374) - InnoDB: uma condição de corrida entre threads que tenta desinicializar e inicializar estatísticas para a mesma tabela gerou uma falha de declaração (Bug n.º 33135425) - InnoDB: Impeça que operações online de DDL acessem a out-of-bounds memória (Bug #34750489, Bug #108925)

Atualização do mecanismo de banco de dados	Versão compatível com o MySQL	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2023-05-11 (versão 3.03.1) (obsoleta)	8.0.26	3.03.1	Correção de um problema em que um bloco de buffer que contém uma página de tabela temporária intrínseca era realocado durante a passagem da página, causando uma falha de declaração (Bug n.º 33715694).

Atualização do mecanismo de banco de dados	Versão compatível com o MySQL	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2023-03-01 (versão 3.03.0) (obsoleta)	8.0.26	3.03.0	- Corrigido um problema em que alguns tipos de coluna, incluindo JSON e TEXT, às vezes esgotavam o buffer de classificação se seu tamanho não fosse pelo menos 15 vezes maior do que a maior linha na classificação. Agora, o buffer de classificação precisa ser apenas 15 vezes maior que a maior chave de classificação. (Bug nº 103325, Bug nº 105532, Bug nº 32738705, Bug nº 33501541) - Corrigido um problema em que o InnoDB nem sempre manipulava alguns nomes legais para partições de tabela corretamente. (Bug nº 32208630) - Corrigido um problema que, em determinadas condições, pode retornar resultados incorretos devido a um cálculo impreciso da propriedade de nulidade ao executar uma consulta com uma condição OR. (Bug nº 34060289) - Corrigido um problema que, em determinadas condições, pode retornar resultados incorretos

Atualização do mecanismo de banco de dados	Versão compatível com o MySQL	Versão	Erros corrigidos no MySQL
			quando as duas condições a seguir são atendidas: • Uma tabela derivada é mesclada no bloco de consulta externo. • A consulta inclui uma junção esquerda e uma subconsulta IN. (Bug nº 34060289) • Corrigido um problema em que valores de AUTO_INCREMENT incorretos eram gerados quando o valor máximo da coluna de número inteiro era excedido. O erro ocorreu porque o valor máximo da coluna não foi considerado. O valor anterior de AUTO_INCREMENT válido deveria ter sido retornado nesse caso, causando um erro de chave duplicada. (Bug nº 87926, Bug nº 26906787) • Corrigido um problema em que não era possível revogar o privilégio DROP no Performance Schema. (Bug nº 33578113) • Corrigido um problema em que um procedimento armazenado o que contém uma instrução IF

Atualização do mecanismo de banco de dados	Versão compatível com o MySQL	Versão	Erros corrigidos no MySQL
			usando EXISTS, que atuava em uma ou mais tabelas que foram excluídas e recriadas entre as execuções, não era executado corretamente nas invocações subsequentes à primeira. (Bug nº 32855634). Corrigido um problema em que uma consulta que faz referência a uma visualização em uma subconsulta e em um bloco de consulta externo pode causar uma reinicialização inesperada (Bug nº 32324234)

Atualização do mecanismo de banco de dados	Versão compatível com o MySQL	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2022-11-18 (versão 3.02.2) (obsoleta)	8.0.23	3.02.2	- Corrigido um problema que, em determinadas condições, pode retornar resultados incorretos devido a um cálculo impreciso da propriedade de nulidade ao executar uma consulta com uma condição OR. (Bug nº 34060289) - Corrigido um problema que, em determinadas condições, pode retornar resultados incorretos quando as duas condições a seguir são atendidas: - Uma tabela derivada é mesclada no bloco de consulta externo. - A consulta inclui uma junção esquerda e uma subconsulta IN. (Bug nº 34060289) - Corrigido um problema em que não era possível revogar o privilégio DROP no Performance Schema. (Bug nº 33578113) - Corrigido um problema em que um procedimento armazenado que contém uma instrução IF usando EXISTS, que atuava em uma ou mais tabelas que foram excluídas e recriadas entre as

Atualização do mecanismo de banco de dados	Versão compatível com o MySQL	Versão	Erros corrigidos no MySQL
			execuções, não era executado corretamente nas invocações subsequentes à primeira. (Bug nº 32855634 do MySQL). • Valores de AUTO_INCREMENT incorretos foram gerados quando o valor máximo da coluna de número inteiro foi excedido. O erro ocorreu porque o valor máximo da coluna não foi considerado. O valor anterior de AUTO_INCREMENT válido deveria ter sido retornado nesse caso, causando um erro de chave duplicada. (Bug nº 87926, Bug nº 26906787) • Corrigido um problema que pode causar uma falha ao atualizar um cluster de banco de dados Aurora MySQL versão 1 (compatível com o MySQL 5.6) contendo uma tabela criada pelo usuário com determinada tabela. IDs A atribuição dessas tabelas IDs pode resultar em conflito na tabela do dicionário de dados IDs durante a atualização do Aurora MySQL versão 2 (compatível com o MySQL 5.7) para o Aurora MySQL versão 3

Atualização do mecanismo de banco de dados	Versão compatível com o MySQL	Versão	Erros corrigidos no MySQL
			(compatível com o MySQL 8.0). (Bug n.º 33919635)
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2022-04-20 (versão 3.02.0) (obsoleta)	8.0.23	3.02.0	Correção do manuseio incorreto de tabelas temporárias usadas para cursores em procedimentos armazenados que poderia resultar em um comportamento inesperado do servidor. (Bug n.º 32416811)

Atualização do mecanismo de banco de dados	Versão compatível com o MySQL	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2022-04-15 (versão 3.01.1) (obsoleta)	8.0.23	3.01.1	Correção do manuseio incorreto de tabelas temporárias usadas para cursores em procedimentos armazenados que poderia resultar em um comportamento inesperado do servidor. (Bug nº 32416811)

Bugs do MySQL corrigidos por atualizações do mecanismo de banco de dados Aurora MySQL 2.x

A versão Aurora compatível com MySQL 5.7 contém todas as correções de bugs do MySQL por meio do MySQL 5.7.44. A tabela a seguir identifica os bugs do MySQL adicionais que foram corrigidos por atualizações do mecanismo de banco de dados Aurora MySQL e em qual atualização eles foram corrigidos.

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de	2.12.3	Correção de um problema em que tabelas temporárias vinculadas a gatilhos durante a execução de declarações

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
banco de dados do Aurora MySQL de 2024-07-09 (versão 2.12.3, compatível com MySQL 5.7.44): versão do Suporte estendido do RDS		podiam causar uma reinicialização inesperada do mecanismo de banco de dados. Correção de um defeito que podia fazer com que o servidor saísse quando declarações UPDATE e DELETE de tabela única usando expressões indexadas eram executadas como declarações preparadas. (Bug #29257254)
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2023-12-28 (versão 2.12.1, compatível com MySQL 5.7.40): versão do Suporte estendido do RDS	2.12.1	- Correção de um problema que pode fazer com que conexões remotas novas e existentes parem quando executadas simultaneamente com a instrução SHOW PROCESSLIST (Bug n.º 34857411 da comunidade) - Replicação: alguns eventos de logs binários nem sempre foram tratados corretamente (Bug n.º 34617506) - Correção do processamento de tokens de um único caractere por um plug-in de analisador de pesquisa de texto completo (FTS) (Bug n.º 35432973)
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2023-07-25 (versão 2.12.0, compatível com MySQL 5.7.40): versão do Suporte estendido do RDS	2.12.0	Corrigido um problema que pode causar maior utilização da CPU devido à rotação do certificado TLS em segundo plano. (Correção de bug nº 34284186 da comunidade)

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2023-10-17 (versão 2.11.4, compatível com MySQL 5.7.12): versão do Suporte estendido do RDS	2.11.4	• Replicação: alguns eventos de logs binários nem sempre foram tratados corretamente. (Bug nº 34617506) • Corrigido um problema que pode causar maior utilização da CPU devido à rotação do certificado TLS em segundo plano. (Correção de bug nº 34284186 da comunidade) • Em declarações preparadas, alguns tipos de subconsultas poderiam causar uma saída do servidor. (Bug nº 33100586)

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2022-10-25 (versão 2.11.0, compatível com MySQL 5.7.12): versão do Suporte estendido do RDS	2.11.0	- Corrigido um problema em que o código para ler informações do conjunto de caracteres das tabelas de eventos de declarações do Performance Schema (por exemplo, <code>events_statements_current</code>) não impedia a gravação simultânea nas informações desse conjunto de caracteres. Como resultado, o conjunto de caracteres de texto da consulta SQL pode ser inválido, o que pode resultar na saída do servidor. Com essa correção, um conjunto de caracteres inválido causa o truncamento da coluna <code>SQL_TEXT</code> e impede a saída do servidor. (Bug nº 23540008) - InnoDB: reversão de correção para Bug nº 25189192, Bug nº 84038 da comunidade. Corrigido um problema em que, após uma operação <code>RENAME TABLE</code> que movia uma tabela para um esquema diferente, o InnoDB falhava ao atualizar a tabela do dicionário de dados do <code>INNODB_SYS_DATAFILES</code>. Isso resultou em um erro na reinicialização, indicando que não foi possível localizar o arquivo de dados do espaço de tabela. - InnoDB: corrigido um problema em que o servidor descartava um índice de chave externa definido internamente ao adicionar um novo índice e tentava utilizar um índice secundário definido em uma coluna virtual gerada como o índice de chave externa, causando a saída do servidor. Agora, o InnoDB permite que uma restrição de chave externa faça referência a um índice secundário definido em uma coluna virtual gerada. (Bug #23533396) - Corrigido um problema em que duas sessões executavam simultaneamente uma operação <code>INSERT... ON DUPLICATE KEY UPDATE</code> gerava um impasse. Durante a reversão parcial de uma tupla, outra sessão poderia atualizá-la. A correção desse bug reverte as correções do Bug nº 11758237, Bug nº 17604730 e Bug nº 20040791. (Bug nº 25966845)

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
		- Corrigido um problema em que os privilégios EXECUTE e ALTER ROUTINE não eram concedidos corretamente aos criadores da rotina, mesmo com automatic_sp_privileges habilitado. (Bug nº 27407480) - Reversão da correção do Bug nº 24671968 da comunidade: corrigido um problema em que uma consulta poderia produzir resultados incorretos se a cláusula WHERE contivesse uma subconsulta dependente, a tabela tivesse um índice secundário nas colunas na lista de seleção seguido pelas colunas na subconsulta e GROUP BY ou DISTINCT permitisse que a consulta usasse uma varredura de índice solta. - Corrigido um problema em que a replicação era interrompida se uma instrução de exclusão de várias tabelas fosse emitida em várias tabelas com chaves estrangeiras. (Bug nº 80821) - Corrigido um problema em que, em casos especiais, certos erros secundários não eram ignorados, mesmo com slave_skip_errors habilitado. Nos casos em que a abertura e o bloqueio de uma tabela falham ou quando as conversões de campo falham em um servidor que executa a replicação baseada em linhas, o erro é considerado crítico e o estado de slave_skip_errors é ignorado. A correção garante que, com slave_skip_errors habilitado, todos os erros relatados durante a aplicação de uma transação sejam tratados corretamente. (Bug nº 70640, Bug nº 17653275) - Corrigido um problema em que uma declaração SET PASSWORD era replicada de uma entidade principal do MySQL 5.6 para uma entidade secundária do MySQL 5.7 ou de uma entidade principal do MySQL 5.7 com a variável de sistema log_built_in_as_identified_by_password definida como ON para uma entidade secundária do MySQL 5.7. O próprio hash da senha também era codificado antes de ser armazenado na entidade secundária. O problema agora foi corrigido e o

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
		hash da senha replicada é armazenado conforme passado originalmente para a entidade secundária. (Bug nº 24687073) - Corrigido um problema em que a serialização de um valor JSON que consiste em um grande subdocumento encapsula do em vários níveis de matrizes ou objetos JSON, ou ambos, às vezes exigia muito tempo para ser concluída. (Bug nº 23031146) - Declarações que não podem ser analisadas (devido, por exemplo, a erros de sintaxe) não são mais gravadas no log de consultas lentas. (Bug nº 33732907)
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2022-11-01 (versão 2.10.3) (obsoleta)	2.10.3	- Corrigido um problema em que o código para ler informações do conjunto de caracteres das tabelas de eventos de declarações do Performance Schema (por exemplo, <code>events_statements_current</code>) não impedia a gravação simultânea nas informações desse conjunto de caracteres. Como resultado, o conjunto de caracteres de texto da consulta SQL pode ser inválido, o que pode resultar na saída do servidor. Com essa correção, um conjunto de caracteres inválido causa o truncamento da coluna <code>SQL_TEXT</code> e impede a saída do servidor. (Bug nº 23540008) - Corrigido um problema quando <code>UPDATE</code> exigia uma tabela temporária com uma chave primária maior que 1024 bytes e essa tabela era criada usando o InnoDB, o servidor poderia sair. (Bug nº 25153670)

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2022-01-26 (versão 2.10.2) (obsoleta)	2.10.2	- Correção de um problema no InnoDB em que um erro no código relacionado às estatísticas da tabela gerava uma afirmação no arquivo fonte dict0stats.cc (http://dict0stats.cc/). (Bug #24585978) - Um índice secundário sobre uma coluna virtual ficou corrompido quando o índice foi construído online. Para instruções UPDATE (https://dev.mysql.com/doc/refman/5.7/en/update.html), corrigimos isso da seguinte forma: Se o valor da coluna virtual do registro do índice estiver definido como NULL, geraremos esse valor a partir do registro do índice do cluster. (Bug #30556595) - ASSERTION "IOOTHER_LOCK" IN LOCK_REC_ADD_TO_QUEUE (Bug #29195848) - HANDLE_FATAL_SIGNAL (SIG=11) EM __STRCHR_ (Bug #28653104) SSE2 - Correção de um problema que uma interrupção de consulta durante uma espera de bloqueio pode causar um erro no InnoDB. (Bug #28068293) - As transações intercaladas poderiam, por vezes, impedir o aplicador de réplica quando o nível de isolamento da transação foi definido como LEITURA REPETIDA. (Erro nº 25040331) - Correção de um problema que pode fazer com que as réplicas de log binário parassem devido ao tempo limite de espera de bloqueio. (Bug #27189701)

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-10-21 (versão 2.10.1) (obsoleta)	2.10.1	CURRENT_TIMESTAMP PRODUCES ZEROS IN TRIGGER. (Bug #25209512)

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-05-25 (versão 2.10.0) (obsoleta)	2.10.0	- As transações intercaladas poderiam, por vezes, impedir o aplicador de réplica quando o nível de isolamento da transação foi definido como LEITURA REPETIDA. (Erro n.º 25040331) - Quando um procedimento armazenado continha uma instrução referente a uma visão que, por sua vez, se referia a outra visão, não era possível invocar o procedimento com sucesso mais de uma vez. (Erro n.º 87858, erro n.º 26864199) - Para consultas com muitas condições OR, o otimizador agora é mais eficiente em memória e menos provável exceder o limite de memória imposto pela variável de sistema range_optimizer_max_mem_size. Além disso, o valor padrão dessa variável foi aumentado de 1.536.000 para 8.388.608. (Erro n.º 79450, erro n.º 22283790) - Replicação: na função <code>next_event()</code>, que é chamada por um processo SQL de uma réplica para ler o próximo evento do log de retransmissão, o processo SQL não liberava o <code>relaylog.log_lock</code> adquirido quando ocorria um erro (por exemplo, devido a um log de retransmissão fechado), fazendo com que todos os outros processos ficassem esperando para adquirir um bloqueio no log de retransmissão para continuar. Com essa correção, o bloqueio é liberado antes que o processo SQL deixe a função sob a situação. (Bug n.º 21697821) - Corrigindo uma corrupção de memória em <code>ALTER TABLE</code> com coluna virtual. (Erro n.º 24961167, erro n.º 24960450) - Replicação: réplicas de vários processos não poderiam ser configuradas com pequenos tamanhos de fila usando slave_pending_jobs_size_max se precisassem processar transações maiores que esse tamanho. Qualquer pacote maior que slave_pending_jobs_size_max era rejeitado com

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
		o erro ER_MTS_EVENT_BIGGER_PENDING_JOBS_SIZE_MAX, mesmo que o pacote fosse menor que o limite definido por slave_max_allowed_packet. Com essa correção, slave_pending_jobs_size_max se torna um limite flexível em vez de um limite rígido. Se o tamanho de um pacote exceder slave_pending_jobs_size_max, mas for menor que slave_max_allowed_packet, a transação será retida até que todos os operadores de réplica tenham filas vazias e, em seguida, é processada. Todas as transações subsequentes são retidas até que a grande transação seja concluída. O tamanho da fila para operadores de réplica pode, portanto, ser limitado e ainda permitir transações maiores ocasionais. (Erro n.º 21280753, erro n.º 77406) • Replicação: ao usar uma réplica de vários processos, os erros do aplicador exibiram dados de ID do trabalhador que eram inconsistentes com os dados externalizados nas tabelas de replicação do Performance Schema. (Erro n.º 25231367) • Replicação: em uma réplica de replicação baseada em GTID executada com -GTID-mode=ON, -log-bin=OFF e usando -, quando um erro que deveria ser ignorado não estava sendo atualizado corretamente slave-skip-errors, causando perda de sincronia com. Exec_Master_Log_Pos Exec_Master_Log_Pos Read_master_log_pos Se um GTID_NEXT não estava especificado, a réplica nunca atualizaria seu estado GTID ao reverter de uma única transação de instrução. O Exec_Master_Log_Pos não seria atualizado porque, mesmo que a transação tivesse sido concluída, seu estado GTID mostraria o contrário. A correção remove a restrição da atualização do estado GTID quando uma transação é revertida somente se GTID_NEXT estiver especificado. (Erro n.º 22268777)

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
		• Replicação: uma instrução com falha parcial não estava consumindo corretamente um GTID gerado ou especificado automaticamente quando o registro em log binário era desabilitado. A correção garante que um DROP TABLE, DROP USER ou DROP VIEW com falha parcial consuma respectivamente o GTID relevante e o salve em @@GLOBAL.GTID_EXECUTED e na tabela mysql.gtid_executed quando o registro em log binário está desabilitado. (Erro nº 21686749) • Replicação: as réplicas que executam o MySQL 5.7 não conseguiam se conectar a uma origem do MySQL 5.5 devido a um erro ao recuperar o server_uuid, que não faz parte do MySQL 5.5.0 Isso foi causado por mudanças no método de recuperação do server_uuid. (Erro nº 22748612) • Replicação de log binário: o mecanismo de ignorar transações GTID não estava funcionando corretamente para a transação XA antes dessa correção. O servidor tem um mecanismo para ignorar (silenciosamente) uma transação GTID se já tiver sido executada essa transação específica no passado. (Erro nº 25041920) • Declarações XA ROLLBACK, que falharam porque foi fornecido o ID incorreto de uma transação, poderiam ser registradas no log binário com o ID correto da transação e, portanto, poderiam ser acionadas por réplicas de replicação. Agora é feita uma verificação da situação de erro antes que o registro em log binário ocorra, e ROLLBACK as instruções XA com falha não são registradas em log. (Erro nº 26618925) • Replicação: Se uma réplica foi configurada usando uma instrução CHANGE MASTER TO que não especificou o nome do arquivo de log de origem e a posição do log de origem, desligue antes da emissão de START SLAVE e reinicie com a opção -relay-log-recovery set, a replicação não foi iniciada.

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
		Isso aconteceu porque o processo do receptor não havia sido iniciado antes da tentativa de recuperação do log de retransmissão, portanto, nenhum evento de rotação de log estava disponível no log de retransmissão para fornecer o nome de arquivo e a posição do log de origem. Nessa situação, a réplica agora ignora a recuperação do log de retransmissão e registra um aviso. Em seguida, prossegue para iniciar a replicação. (Erro n.º 28996606, erro n.º 93397) - Replicação: na replicação baseada em linha, uma mensagem que exibía incorretamente comprimentos de campo era retornada ao replicar de uma tabela com uma coluna <code>utf8mb3</code> para uma tabela da mesma definição em que a coluna foi definida com um conjunto de caracteres <code>utf8mb4</code>. (Erro n.º 25135304, erro n.º 83918) - Replicação: quando uma instrução RESET SLAVE foi emitida em uma réplica de replicação GTIDs em uso, os arquivos de log de retransmissão existentes foram eliminados, mas o novo arquivo de log de retransmissão substituto foi gerado antes que o conjunto de arquivos recebidos GTIDs para o canal fosse apagado. Portanto, o antigo conjunto de GTID foi gravado no novo arquivo de log de retransmissão como o <code>PREVIOUS_GTIDS</code> evento, causando um erro fatal na replicação, afirmando que a réplica tinha GTIDs mais do que a origem, mesmo que o conjunto <code>gtid_execute</code> para ambos os servidores estivesse vazio. Agora, quando <code>RESET SLAVE</code> é emitido, o conjunto de recebidos GTIDs é apagado antes que o novo arquivo de log de retransmissão seja gerado, para que essa situação não ocorra. (Erro n.º 27411175) - Replicação: Com GTIDs o uso para replicação, transações incluindo instruções que causaram um erro de análise (ER_PARSE_ERROR) não podiam ser ignoradas manualmente pelo método recomendado de injetar uma transação vazia

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
		ou de substituição com o mesmo GTID. Essa ação deve resultar na réplica que identifica o GTID como já usado e, portanto, ignorando a transação indesejada que compartilhou seu GTID. No entanto, no caso de um erro de análise, como a instrução era analisada antes que o GTID fosse verificado para ver se ele precisava ser ignorado, o processo do aplicador de replicação parava devido ao erro de análise, mesmo que a intenção fosse ignorar a transação de qualquer maneira. Com essa correção, o processo do aplicador de replicação agora ignora erros de análise se a transação em questão precisar ser ignorada porque o GTID já foi usado. Observe que essa alteração de comportamento não se aplica no caso de workloads que consistem em saída de log binário produzida por <code>mysqlbinlog</code>. Nessa situação, haveria o risco de que uma transação com um erro de análise que viesse imediatamente após uma transação ignorada também fosse ignorada silenciosamente, quando deveria gerar um erro. (Erro nº 27638268) • Replicação: habilite o processo SQL para que o GTID ignore uma transação parcial. (Erro nº 25800025) • Replicação: quando um parâmetro de tempo limite negativo ou fracionário era fornecido a <code>WAIT_UNTIL_SQL_THREAFTER_GTIDS()</code>, o servidor se comportava de maneiras inesperadas. Com esta correção: • Um valor de tempo limite fracionário é lido como está, sem arredondamento. • Um valor de tempo limite negativo é rejeitado com um erro se o servidor estiver em um modo SQL estrito. Se o servidor não estiver em um modo SQL estrito, o valor fará com que a função retorne NULL imediatamente sem qualquer espera e, em seguida, emita um aviso. (Erro n.º 24976304, erro n.º 83537)

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
		- Replicação: se a função <code>WAIT_FOR_EXECUTED_GTID_SET()</code> fosse usada com um valor de tempo limite incluindo uma parte fracionada (por exemplo, 1,5), um erro na lógica de transmissão significava que o tempo limite foi arredondado para o segundo inteiro mais próximo e para zero para valores menores que 1 segundo (por exemplo, 0,1). Agora a lógica de transmissão foi corrigida para que o valor do tempo limite seja aplicado conforme especificado originalmente, sem arredondamento. Agradecemos a Dirkjan Bussink pela contribuição. (Erro n.º 29324564, erro n.º 94247) - Quando GTIDs ativado, o XA COMMIT em uma transação XA desconectada em uma transação de várias declarações gerou uma afirmação. (Erro n.º 22173903) - Replicação: uma afirmação era gerada em compilações de depuração se uma instrução XA ROLLBACK fosse emitida para um identificador de transação desconhecido quando o valor de <code>gtid_next</code> tivesse sido definido manualmente. Agora, o servidor não tenta atualizar o estado de GTID se uma instrução XA ROLLBACK falhar com um erro. (Erro n.º 27928837, erro n.º 90640) - Corrigido o problema de ordem de classificação incorreta quando várias funções CASE são usadas na cláusula ORDER BY. (Bug n.º 22810883) - Algumas consultas que usavam ordenação podiam acessar uma coluna não inicializada durante a otimização e causar a saída do servidor. (Bug n.º 27389294)

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-11-12 (versão 2.09.3) (obsoleta)	2.09.3	• ASSERTION !M_PREBUILT->TRX->CHECK_FOREIGNS. (Bug #23533396) • Replicação: * um problema de bloqueio na função WAIT_FOR_EXECUTED_GTID_SET() poderia fazer com que o servidor travasse em certas circunstâncias. O problema já foi corrigido. (Bug #29550513)
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-12-11 (versão 2.09.1) (obsoleta)	2.09.1	• Replicação: As transações intercaladas poderiam, por vezes, impedir o aplicador escravo quando o nível de isolamento da transação foi definido como LEITURA REPETIDA. (Erro nº 25040331) • Para uma tabela com uma coluna TIMESTAMP ou DATETIME cujo valor padrão é CURRENT_TIMESTAMP, a coluna podia ser inicializada em 0000-00-00 00:00:00 caso a tabela tivesse um acionador BEFORE INSERT. (Erro nº 25209512, erro nº 84077) • Para uma instrução INSERIR para a qual a VALUES lista produziu valores para a segunda linha ou linha posterior usando uma subconsulta contendo uma junção, o servidor pode sair depois de não resolver os privilégios necessários. (Erro nº 23762382)
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-11-12 (versão 2.08.3) (obsoleta)	2.08.3	• Erro #23762382 - INSERIR CONSULTA DE VALORES COM JUNÇÃO EM UMA SELEÇÃO QUE CAUSA COMPORTAMENTO INCORRETO. • Erro #25209512 - CURRENT_TIMESTAMP PRODUZ ZEROS NO GATILHO.

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-06-02 (versão 2.08.0) (obsoleta)	2.08.0	• Bug #25289359: um bloqueio de cache de texto completo obtido quando os dados foram sincronizados não foi liberado se o tamanho de cache de texto completo excedeu o limite respectivo. • Bug #29138644: alterar manualmente a hora do sistema enquanto o servidor MySQL estava sendo executado causou atrasos no thread do agente de limpeza de página. • Bug #25222337: um nome de campo de coluna virtual NULL em um índice virtual causou uma saída do servidor durante uma comparação de nome de campo que ocorre ao preencher colunas virtuais afetadas por uma restrição de chave externa. • Bug #25053286: executar um procedimento armazenado que contém uma consulta que acessou uma visualização pode alocar memória que não foi liberada até o término da sessão. • Bug #25586773: executar um procedimento armazenado que contém uma instrução que criou uma tabela com base no conteúdo de determinadas instruções SELECT pode resultar em um vazamento de memória. • Bug #28834208: durante o aplicativo de log, após uma operação OPTIMIZE TABLE, o InnoDB não preencheu as colunas virtuais antes de verificar se há atualizações de índice de coluna virtual. • Bug n.º 26666274: loop infinito no contêiner de buffer de esquema de performance devido a excesso de inteiros não assinados de 32 bits.

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2022-06-16 (versão 2.07.8) (obsoleta)	2.07.8	Quando UPDATE exigia uma tabela temporária com uma chave primária maior que 1024 bytes e essa tabela era criada usando o InnoDB, o servidor poderia sair. (Bug nº 25153670)
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-09-02 (versão 2.07.6) (obsoleta)	2.07.6	• INSERIR REGISTROS COM 64K É MUITO DEMORADO. (Bug n.º 23031146)
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-03-04 (versão 2.07.4) (obsoleta)	2.07.4	• Correção de um problema no analisador de ngram de texto completo ao lidar com tokens contendo ' ' (espaço), '%', ou ','. Os clientes devem reconstruir seus índices FTS se estiverem usando o analisador ngram. (Bug n.º 25873310) • Correção de um problema que poderia causar a reinicialização do mecanismo durante a execução da consulta com visualizações SQL aninhadas. (Bug n.º 27214153, bug n.º 26864199)

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-11-10 (versão 2.07.3) (obsoleta)	2.07.3	- InnoDB: transações XA simultâneas que foram executadas com sucesso para o estágio de preparação XA no mestre em conflito quando repetidas no escravo, o que gera um tempo limite de espera de bloqueio no applier thread. O conflito foi devido ao intervalo de bloqueio GAP que diferiu quando as transações foram repetidas em série no escravo. Para evitar esse tipo de conflito, os bloqueios do GAP realizados pelas transações XA no nível de isolamento LEITURA CONFIRMADA agora são liberados (e não mais herdados) quando as transações XA atingem o estágio de preparação. (Erro nº 27189701, erro nº 25866046) - InnoDB: Um bloqueio de gap foi realizado desnecessariamente durante a validação de chave estrangeira ao usar o nível de isolamento LEITURA CONFIRMADA. (Erro nº 25082593) - Replicação: Ao usar transações XA, se um tempo limite de espera de bloqueio ou um deadlock ocorreu para o thread do aplicador (SQL) em um escravo de replicação, a repetição automática não funcionou. A causa era que, mesmo que o thread SQL fizesse uma reversão, ele não reverteria a transação XA de volta. Isso significava que, quando a transação foi repetida, o primeiro evento era XA START, que era inválido, pois a transação XA já estava em andamento, levando a um erro XAER_RMFAIL. (Erro nº 24764800) - Replicação: As transações intercaladas poderiam, por vezes, impedir o aplicador escravo quando o nível de isolamento da transação foi definido como LEITURA REPETIDA. (Erro nº 25040331) - Replicação: O valor retornado por uma instrução MOSTRAR STATUS ESCRAVO para o tamanho total combinado de todos os arquivos de log de retransmissão existentes (Relay_Log_Space) pode se tornar muito maior do que

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
		o espaço em disco real usado pelos arquivos de log de retransmissão. O I/O encadeamento não bloqueou a variável enquanto atualizava o valor, portanto, o encadeamento SQL poderia excluir automaticamente um arquivo de log de retransmissão e gravar um valor reduzido antes que o I/O encadeamento terminasse de atualizar o valor. O thread de E/S gravou seu cálculo de tamanho original, ignorando a atualização do thread SQL e, portanto, adicionando de volta o espaço para o arquivo excluído. O valor Relay_Log_Space agora está bloqueado durante as atualizações para evitar atualizações simultâneas e garantir um cálculo preciso. (Erro nº 26997096, erro nº 87832) • Para uma instrução INSERIR para a qual a lista VALORES produziu valores para a segunda linha ou linha posterior com uma subconsulta contendo uma junção, o servidor pode sair depois de não resolver os privilégios necessários. (Erro nº 23762382) • Para uma tabela com uma coluna TIMESTAMP ou DATETIME cujo valor padrão é CURRENT_TIMESTAMP, a coluna podia ser inicializada em 0000-00-00 00:00:00 caso a tabela tivesse um acionador BEFORE INSERT. (Erro nº 25209512, erro nº 84077) • Uma saída do servidor pode gerar tentativas simultâneas de vários threads para registrar e cancelar o registro de objetos do Performance Schema de metadados. (Erro nº 26502135) • Executar um procedimento armazenado que contém uma instrução que criou uma tabela com base no conteúdo de determinadas instruções SELECIONAR que pode resultar em um vazamento de memória. (Erro nº 25586773) • Executar um procedimento armazenado que contém uma consulta que acessou uma visualização poderia alocar

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
		memória que não foi liberada até o término da sessão. (Erro nº 25053286) • Certos casos de materialização da subconsulta poderiam causar uma saída do servidor. Essas consultas agora produzem um erro sugerindo que a materialização seja desabilitada. (Erro nº 26402045) • As consultas com muitas junções à esquerda ficavam lentas se o buffer de junção fosse usado (por exemplo, usando o algoritmo de loop aninhado em bloco). (Erro nº 18898433, erro nº 72854) • O otimizador ignorou a segunda coluna em um índice composto ao executar uma junção interna com uma LIKE cláusula contra a segunda coluna. (Erro nº 28086754)
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-04-17 (versão 2.07.2) (obsoleta)	2.07.2	• Erro nº 23104498: corrigido um problema no esquema de performance ao relatar a memória total usada. (https://github.com/mysql/mysql-server/commit/20b6840df5452f47313c6f9a6ca075bfbc00a96b) • Erro nº 22551677: corrigido um problema no esquema de performance que poderia fazer com que o mecanismo de banco de dados falhasse ao tentar colocá-lo offline. (https://github.com/mysql/mysql-server/commit/05e2386eccd32b6b444b900c9f8a87a1d8d531e9) • Erro nº 23550835, erro nº 23298025, erro nº 81464: correção de um problema no esquema de performance que causava uma falha no mecanismo de banco de dados devido porque excedia a capacidade de um buffer interno. (https://github.com/mysql/mysql-server/commit/b4287f93857bf2f99b18fd06f555bbe5b12debfc, https://github.com/mysql/mysql-server/commit/b4287f93857bf2f99b18fd06f555bbe5b12debfc)

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-11-25 (versão 2.07.0) (obsoleta)	2.07.0	• Bug nº 26251621: INCORRECT BEHAVIOR WITH TRIGGER AND GCOL • Bug nº 22574695: ASSERTION `!TABLE (!TABLE->READ_SET BITMAP_IS_SET(TABLE->READ_SET, FIEL • Bug nº 25966845: INSERT ON DUPLICATE KEY GENERATE A DEADLOCK • Bug nº 23070734: CONCURRENT TRUNCATE TABLES CAUSE STALL • Bug nº 26191879: FOREIGN KEY CASCADES USE EXCESSIVE MEMORY • Bug nº 20989615: INNODB AUTO_INCREMENT PRODUCES SAME VALUE TWICE
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-11-11 (versão 2.05.0) (obsoleta)	2.05.0	• Bug #23054591: PURGE BINARY LOGS TO está lendo todo o arquivo de log binário e causando MySql a paralisação

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-08-14 (versão 2.04.9) (obsoleta)	2.04.9	- Bug #23070734, Bug #80060: TRUNCATE simultâneo causa paralisações TABLEs - Erro nº 23103937: PS_TRUNCATE_ALL_TABLES() NÃO FUNCIONA NO MODO SUPER_READ_ONLY - Erro nº 22551677: ao colocar o servidor offline, uma condição de corrida dentro do esquema de performance poderia levar a uma saída do servidor. - Erro nº 27082268: sincronização de FTS inválida. - ERRO nº 12589870: corrigido um problema que causa uma reinicialização com instrução de várias consultas quando o cache de consulta está habilitado. - Erro nº 26402045: certos casos de materialização da subconsulta poderiam causar uma saída do servidor. Essas consultas agora produzem um erro sugerindo que a materialização seja desabilitada. - Erro nº 18898433: consultas com muitas junções à esquerda ficavam lentas se o buffer de junção fosse usado (por exemplo, usando o algoritmo de loop aninhado em bloco). - Erro nº 25222337: um nome de campo de coluna virtual NULL em um índice virtual causava uma saída do servidor durante uma comparação de nome de campo que ocorre ao preencher colunas virtuais afetadas por uma restrição de chave externa. (https://github.com/mysql/mysql-server/commit/273d5c9d7072c63b6c47dbef6963d7dc491d5131) - Erro nº 25053286: executar um procedimento armazenado o que contém uma consulta que acessou uma visualização poderia alocar memória que não foi liberada até o término da sessão. (https://github.com/mysql/mysql-server/commit/d7b37d4d141a95f577916448650c429f0d6e193d) - Bug #25586773: Executar um procedimento armazenado o contendo uma instrução que criou uma tabela a partir

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
		do conteúdo de determinadas instruções SELECT (https://dev.mysql.com/doc/refman/5.7/en/select.html) pode resultar em um vazamento de memória. (https://github.com/mysql/mysql-server/commit/88301e5adab65f6750f66af284be410c4369d0c1) • Erro nº 26666274: LOOP INFINITO NO CONTÊINER DE BUFFER DO ESQUEMA DE PERFORMANCE. • Erro nº 23550835, erro nº 23298025, erro nº 81464: tabelas do esquema de performance SELECT quando um buffer interno estava cheio poderia causar uma saída do servidor.
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-09-19 (versão 2.04.6) (obsoleta)	2.04.6	• Bug #23054591: PURGE BINARY LOGS TO está lendo todo o arquivo de log binário e causando MySql a paralisação
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-05-02 (versão 2.04.2) (obsoleta)	2.04.2	Bug nº 24829050 - A OTIMIZAÇÃO DE INDEX_MERGE_INTERSECTION PROVOCA RESULTADOS INCORRETOS NA CONSULTA

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-10-1 1 (versão 2.03) (obsoleta)	2.03	• VARREDURA REVERSA EM UMA TABELA PARTICIONADA REALIZA ICP - PEDIDO DE DESC (Bug nº 24929748). • JSON_OBJECT CRIA CÓDIGO JSON INVÁLIDO (Bug nº 26867509). • INSERIR DADOS JSON GRANDES DEMORA MUITO (Bug nº 22843444). • TABELAS PARTICIONADAS USAM MAIS MEMÓRIA EM 5.7 DO QUE EM 5.6 (Bug nº 25080442).

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-09-21 (versão 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(!VIC TIM_TRX->READ_ONLY) • BUG#21983865 UNEXPECTED DEADLOCK WITH INNODB_AUTOINC_LOCK_MODE=0 • BUG#22679185 INVALID INNODB FTS DOC ID DURING INSERT • BUG#22899305 GCOLS: ASSERTION: !(COL->PR TYPE & 256). • BUG#22956469 MEMORY LEAK INTRODUCED IN 5.7.8 IN MEMORY/INNODB/OS0FILE • BUG#22996488 CRASH IN FTS_SYNC_INDEX WHEN DOING DDL IN A LOOP • BUG#23014521 GCOL:INNODB: ASSERTION: !IS_V • BUG#23021168 REPLICATION STOPS AFTER TRX IS ROLLED BACK ASYNC • BUG#23052231 ASSERTION: ADD_AUTOINC < DICT_TABLE_GET_N_USER_COLS • BUG#23149683 ROTATE INNODB MASTER KEY WITH KEYRING_OKV_CONF_DIR MISSING: SIGSEGV; SIGNAL 11 • BUG#23762382 INSERT VALUES QUERY WITH JOIN IN A SELECT CAUSES INCORRECT BEHAVIOR • BUG#25209512 CURRENT_TIMESTAMP PRODUCES ZEROS IN TRIGGER

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
		• 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
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-05-03 (versão 2.02) (obsoleta)	2.02.0	A junção à esquerda retorna resultados incorretos na parte externa (Bug N° 22833364)

Bugs do MySQL corrigidos por atualizações do mecanismo de banco de dados Aurora MySQL 1.x

O Aurora da versão compatível com o MySQL 5.6 contém todas as correções de bug do MySQL por meio do MySQL 5.6.10. A tabela a seguir identifica os bugs do MySQL adicionais que foram corrigidos por atualizações do mecanismo de banco de dados Aurora MySQL e em qual atualização eles foram corrigidos.

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2021-03-18	1.23.2	• Replicação: Enquanto uma instrução SHOW BINLOG EVENTS estava sendo executada, todas as transações paralelas foram bloqueadas. A correção garante que o processo SHOW BINLOG EVENTS agora só adquira um bloqueio durante o cálculo da posição final do arquivo, portanto, as transações

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
(versão 1.23.2) (obsoleta)		paralelas não são bloqueadas por longas durações. (Erro n.º 76618, erro n.º 20928790)

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-09-02 (versão 1.23.0) (obsoleta)	1.23.0	• Eventos de binlog com ALTER TABLE ADD COLUMN ALGORITHM=QUICK serão regravados ALGORITHM=DEFAULT para serem compatíveis com a edição da comunidade. • ERRO nº 22350047: SE O CLIENTE FOI ENCERRADO APÓS A REVERSÃO PARA OS STMTS ANTERIORES DO PONTO DE SALVAMENTO CONFIRMADO • Erro nº 29915479: EXECUÇÃO COM_REGISTER_SLAVE SEM COM_BINLOG_DUMP PODE RESULTAR EM SAÍDA DO SERVIDOR • Erro nº 30441969: BUG #29723340: TRAVAMENTO DO SERVIDOR MYSQL APÓS CONSULTA SQL COM DADOS ? AST • Erro nº 30628268: TRAVAMENTO POR FALTA DE MEMÓRIA • Erro nº 27081349: COMPORTAMENTO INESPERADO AO EXCLUIR COM FUNÇÃO ESPACIAL • Erro nº 27230859: COMPORTAMENTO INESPERADO AO LIDAR COM POLÍGONO INVÁLIDO" • Erro nº 27081349: COMPORTAMENTO INESPERADO AO EXCLUIR COM ESPACIAL" • Erro nº 26935001: ALTERAR TABELA AUTO_INCREMENT TENTA LER ÍNDICE DO TABLESPACE DESCARTADO • Erro nº 29770705: SERVIDOR TRAVOU AO EXECUTAR SELEÇÃO COM CLÁUSULA WHERE ESPECÍFICA • Erro nº 27659490: SELECIONAR USANDO INTERVALO DINÂMICO E MESCLAGEM DE ÍNDICE USA MUITA MEMÓRIA (OOM) • Erro nº 24786290: A REPLICAÇÃO É INTERROMPIDA APÓS O Erro nº 74145 OCORRER NO MASTER

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
		• Erro nº27703912: USO EXCESSIVO DE MEMÓRIA COM MUITA PREPARAÇÃO • Bug #20527363: TRUNCATE TEMPORARY TABLE CRASH:!! DICT_ TF2 _FLAG_IS_SET (TABELA, DICT_ _TEMPORARY) TF2 • Erro nº 23103937: PS_TRUNCATE_ALL_TABLES() NÃO FUNCIONA NO MODO SUPER_READ_ONLY • Erro nº 25053286: USAR VISUALIZAÇÃO COM CONDIÇÃO NO PROCEDIMENTO CAUSA COMPORTAMENTO INCORRETO (corrigido em 5.6.36) • Erro nº 25586773: COMPORTAMENTO INCORRETO PARA CRIAR TABELA SELECT EM UM LOOP EM SP (corrigido em 5.6.39) • Erro nº 27407480: AUTOMATIC_SP_PRIVILEGES REQUER OS PRIVILÉGIOS DE INSERÇÃO PARA A TABELA MYSQL.USER • Erro nº 26997096: o valor relay_log_space não é atualizado de forma sincronizada, de modo que seu valor é às vezes muito maior do que o espaço em disco real usado pelos logs de retransmissão. • Erro nº 15831300 SLAVE_TYPE_CONVERSIONS=ALL_NON_LOSSY NÃO ESTÁ FUNCIONANDO CONFORME O ESPERADO • Erro de backport SSL nº 17087862, erro nº 20551271 • Erro nº 16894092: REGRESSÃO DE PERFORMANCE EM 5.6.6+ PARA INSERIR EM... SELECT ... DE (corrigido em 5.6.15). • Transfira uma correção de erro relacionada a SLAVE_TYPE_CONVERSIONS .

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2020-11-09 (versão 1.22.3) (obsoleta)	1.22.3	- Bug #26654685: Um ID de índice corrompido encontrado durante uma verificação de chave estrangeira gerou uma declaração - Erro nº 15831300: Por padrão, ao promover inteiros de um tipo menor no mestre para um tipo maior no escravo (por exemplo, de uma coluna SMALLINT no mestre para uma coluna BIGINT no escravo), os valores promovidos são tratados como se fossem assinados. Agora, nesses casos, é possível modificar ou substituir esse comportamento usando um ou ambos de <code>ALL_SIGNED</code>, <code>ALL_UNSIGNED</code> no conjunto de valores especificados para a variável de sistema de servidor slave_type_conversions. Para obter mais informações, consulte Replicação baseada em linha: promoção e rebaixamento de atributos, bem como a descrição da variável. - Erro nº 17449901: Com <code>foreign_key_checks=0</code>, o InnoDB permitiu que um índice exigido por uma restrição de chave estrangeira fosse descartado, colocando a tabela em um inconsistente e fazendo com que a falha na verificação de chave estrangeira ocorresse na carga da tabela. O InnoDB agora evita descartar um índice exigido por uma restrição de chave estrangeira, mesmo com <code>foreign_key_checks=0</code>. A restrição de chave estrangeira deve ser removida antes de soltar o índice de chave estrangeira. - ERRO N.º 20768847: Um ALTER TABLE... A operação DROP INDEX em uma tabela com dependências de chave estrangeira levantou uma declaração.

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-11-25 (versão 1.22.0) (obsoleta)	1.22.0	- Bug nº 16346241 - SERVER CRASH IN ITEM_PARA M::QUERY_VAL_STR - Bug nº 17733850 - NAME_CONST() CRASH IN ITEM_NAME _CONST::ITEM_NAME_CONST() - Bug nº 20989615 - INNODB AUTO_INCREMENT PRODUCES SAME VALUE TWICE - Erro nº 20181776 – O CONTROLE DE ACESSO NÃO CORRESPONDE AO HOST MAIS ESPECÍFICO QUANDO CONTÉM CARACTERES CURINGA - Bug #27326796 - FALHA NO MYSQL COM FALHA DE ASSERÇÃO DO INNODB NO ARQUIVO PARS.CC PARS0 - Bug nº 20590013 - IF YOU HAVE A FULLTEXT INDEX AND DROP IT YOU CAN NO LONGER PERFORM ONLINE DDL
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-11-25 (versão 1.21.0) (obsoleta)	1.21.0	- Bug #19929406: HANDLE_FATAL_SIGNAL (SIG=11) EM __MEMMOVE__BACK FROM STRING: :COPY SSSE3 - Bug #17059925: para instruções UNION, o valor das linhas examinadas era calculado de maneira incorreta. Isso era manifestado como valores grandes demais para a coluna ROWS_EXAMINED das tabelas de instrução do Performance Schema (como events_statements_current). - Bug #11827369: algumas consultas com subconsultas aninhadas SELECT . . . FROM DUAL geravam uma afirmação. - Bug nº 16311231: resultados incorretos eram retornados se uma consulta contivesse uma subconsulta em uma cláusula IN, que contivesse uma operação XOR na cláusula WHERE.

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-11-11 (versão 1.20.0) (obsoleta)	1.20.0	• Bug #19929406: <code>HANDLE_FATAL_SIGNAL (SIG=11) EM __MEMMOVE__BACK FROM STRING: :COPY SSSE3</code> • Bug #17059925: para instruções UNION, o valor das linhas examinadas era calculado de maneira incorreta. Isso era manifestado como valores grandes demais para a coluna <code>ROWS_EXAMINED</code> das tabelas de instrução do Performance Schema (como events_statements_current). • Bug #11827369: algumas consultas com subconsultas aninhadas <code>SELECT . . . FROM DUAL</code> geravam uma afirmação. • Bug nº 16311231: resultados incorretos eram retornados se uma consulta contivesse uma subconsulta em uma cláusula <code>IN</code>, que contivesse uma operação XOR na cláusula <code>WHERE</code>.
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-09-19 (versão 1.19.5) (obsoleta)	1.19.5	• CVE-2018-2696 • CVE-2015-4737 • Bug #19929406: <code>HANDLE_FATAL_SIGNAL (SIG=11) EM __MEMMOVE__BACK FROM STRING: :COPY SSSE3</code> • Bug #17059925: para instruções UNION, o valor das linhas examinadas era calculado de maneira incorreta. Isso era manifestado como valores grandes demais para a coluna <code>ROWS_EXAMINED</code> das tabelas de instrução do Performance Schema (como events_statements_current). • Bug #11827369: algumas consultas com subconsultas aninhadas <code>SELECT . . . FROM DUAL</code> geravam uma afirmação. • Bug #16311231: resultados incorretos eram retornados se uma consulta contivesse uma subconsulta em uma cláusula <code>IN</code>, que contivesse uma operação XOR na cláusula <code>WHERE</code>.

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-02-07 (versão 1.19.0) (obsoleta)	1.19.0	• BUG N° 32917: DETECTAR ARQUIVOS DE GRUPOS TEMPORÁRIOS ÓRFÃOS E TRATAR COM TRANQUILIDADE • BUG N° 63144: BLOQUEIO DE METADADOS DO COMANDO CREATE TABLE IF NOT EXISTS É MUITO RESTRITIVO
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2019-01-17 (versão 1.17.8) (obsoleta)	1.17.8	• BUG N° 13418638: CREATE TABLE IF NOT EXISTS METADATA LOCK IS TOO RESTRICTIVE
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-10-08 (versão 1.17.7) (obsoleta)	1.17.7	• Índice perdido em uma coluna de chave estranha causa perda de tabela. (Bug #16208542) • Vazamento da memória em add_derived_key(). (Bug #76349) • Em tabelas particionadas, as consultas podem retornar resultados diferentes, dependendo do uso de mesclagem de índice. (Bug #16862316) • Consultas que usam a otimização de mesclagem de índice (consulte Otimização de mesclagem de índice) podem retornar resultados inválidos quando executadas em tabelas que foram particionadas por HASH. (Bug #17588348)

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-09-06 (versão 1.17.6) (obsoleta)	1.17.6	- Para uma instrução ALTER TABLE que renomeou ou alterou o valor padrão de uma coluna BINARY, a alteração foi feita usando uma cópia da tabela e não está no local. (Bug nº 67141, Bug nº 14735373, Bug nº 69580, Bug nº 17024290) - Uma união externa entre uma tabela regular e uma tabela derivada que agrupa implicitamente pode causar uma saída do servidor. (Bug nº 16177639)
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-03-13 (versão 1.17) (obsoleta)	1.17.0	- LAST_INSERT_ID é replicado incorretamente se os filtros de replicação forem usados (Bug 69861) - A consulta retorna resultados diferentes dependendo da configuração INDEX_MERGE (Bug 16862316) - Nova execução de procedimento de consulta de rotina armazenada, plano de consulta ineficiente (Bug 16346367) - InnoDB FTS : declaração em FTS_CACHE_APPEND_D ELETED_DOC_IDS (Bug 18079671) - Declaração RBT_EMPTY(INDEX_CACHE->WORDS) na COLUNA DE ALTERAÇÃO DE ALTER TABLE (Bug 17536995) - Pesquisa InnoDB fulltext não encontra registros quando pontos de gravação estão envolvidos (bug 70333, bug 17458835)
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-11-20 (versão 1.15.1) (obsoleta)	1.15.1	- Revertido — Instância do MySQL para ao "fazer um índice SYNC" (bug 73816) - Revertido — Declaração RBT_EMPTY(INDEX_CACHE->WORDS) na COLUNA DE ALTERAÇÃO DE ALTER TABLE (Bug 17536995) - Revertido — Pesquisa InnoDB Fulltext não encontra registros quando pontos de gravação estão envolvidos (bug 70333)

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2017-10-24 (versão 1.15) (obsoleta)	1.15.0	• CREATE USER aceita plugin e hash de senha, mas ignora o hash de senha (erro #78033) • O mecanismo de particionamento adiciona campos ao conjunto de bits de leitura capazes de retornar entradas classificadas de um índice particionado. Isso leva o buffer de junção a tentar ler campos desnecessários. Corrigido não adicionando todos os campos de particionamento ao read_set, mas em vez de somente classificar nos campos de prefixo definidos no read_set. Adicionado um DBUG_ASSERT que, se key_cmp, pelo menos o primeiro campo deverá ser lido (Bug 16367691). • Instância do MySQL paralisada ao "fazer um índice SYNC" (Bug nº 73816) • Declaração RBT_EMPTY(INDEX_CACHE->WORDS) na COLUNA DE ALTERAÇÃO DE ALTER TABLE (Bug 17536995) • Pesquisa de texto completo de InnoDB não encontra registros quando pontos de salvamento estão envolvidos (erro #70333)
Atualizações do mecanismo de banco de dados do Aurora MySQL de 2018-03-13 (versão 1.14.4) (obsoleta)	1.14.4	• Eventos ignoráveis não funcionam e não são testados (bug N° 74683) • NOVO->FALHA EM ASSERT ANTIGA "GTID_MODE > 0" (erro #20436436)

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL: 2017-08-07 (versão 1.14) (obsoleta)	1.14.0	Uma pesquisa de texto completo combinada com tabelas derivadas (subconsultas na cláusula FROM) causou uma saída do servidor. Agora, se uma operação de texto completa depender de uma tabela derivada, o servidor produzirá um erro indicando que uma pesquisa de texto completa não pode ser realizada em uma tabela materializada. (Bug nº 68751, Bug nº 16539903)
Atualizações do mecanismo de banco de dados do Aurora MySQL: 2017-05-15 (versão 1.13) (obsoleta)	1.13.0	• Recarregar uma tabela que foi removida enquanto vazia fez com que um valor AUTO_INCREMENT fosse redefinido. (Bug nº 21454472, Bug nº 77743) • Um registro de índice não foi encontrado na reversão devido a inconsistências na estrutura de purge_node_t. A inconsistência resultou em avisos e mensagens de erro, como "erro na atualização da entrada do índice sec", "impossível limpar um registro" e "tentativa de limpar a entrada do índice sec não marcada para exclusão". (Bug nº 19138298, Bug nº 70214, Bug nº 21126772, Bug nº 21065746) • O cálculo incorreto do tamanho da pilha para a operação qsort resulta em excedente de pilha. (Bug nº 73979) • Registro não encontrado em um índice após reversão. (Bug nº 70214, Bug nº 72419) • ALTER TABLE adicionar coluna TIMESTAMP na atualização CURRENT_TIMESTAMP insere ZERO-datas (Bug nº 17392)

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL: 2017-04-05 (versão 1.12) (obsoleta)	1.12.0	- Recarregar uma tabela que foi removida enquanto vazia fez com que um valor AUTO_INCREMENT fosse redefinido. (Bug nº 21454472, Bug nº 77743) - Um registro de índice não foi encontrado na reversão devido a inconsistências na estrutura de purge_node_t. A inconsistência resultou em avisos e mensagens de erro, como "erro na atualização da entrada do índice sec", "impossível limpar um registro" e "tentativa de limpar a entrada do índice sec não marcada para exclusão". (Bug nº 19138298, Bug nº 70214, Bug nº 21126772, Bug nº 21065746) - O cálculo incorreto do tamanho da pilha para a operação qsort resulta em excedente de pilha. (Bug nº 73979) - Registro não encontrado em um índice após reversão. (Bug nº 70214, Bug nº 72419) - ALTER TABLE adicionar coluna TIMESTAMP na atualização CURRENT_TIMESTAMP insere ZERO-datas (Bug nº 17392)
Atualizações do mecanismo de banco de dados do Aurora MySQL: 2017-02-23 (versão 1.11) (obsoleta)	1.11.0	- A execução da tabela ALTER da chave estrangeira DROP simultaneamente com outra operação DROP faz com que a tabela desapareça. (Bug nº 16095573) - Algumas consultas INFORMATION_SCHEMA que usam ORDER BY não usaram uma otimização de classificação de arquivo como fizeram anteriormente. (Bug nº 16423536) - FOUND_ROWS () retorna a contagem incorreta de linhas em uma tabela. (Bug nº 68458) - O servidor falha em vez de indicar um erro quando muitas tabelas temporárias estão abertas. (Bug nº 18948649)

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL: 2016-12-14 (versão 1.10) (obsoleta)	1.10.0	• UNION de tabelas derivadas retorna resultados incorretos com cláusulas '1=0/false'. (Bug nº 69471) • O servidor falha em ITEM_FUNC_GROUP_CONCAT::FIX_FIELDS na segunda execução do procedimento armazenado. (Bug nº 20755389) • Evite que consultas MySQL fiquem paradas por muito tempo durante a sincronização do cache de FTS com o disco, descarregando a tarefa de sincronização de cache em um thread separado, assim que o tamanho do cache atingir 10% do tamanho total. (Bugs #22516559, #73816)
Atualizações do mecanismo de banco de dados do Aurora MySQL: 2016-10-26 (versão 1.8.1) (obsoleta)	1.8.1	• O OpenSSL alterou os parâmetros de comprimento da chave Diffie-Hellman devido ao problema. LogJam (Bug nº 18367167)
Atualizações do mecanismo de banco de dados do Aurora MySQL: 2016-10-18 (versão 1.8) (obsoleta)	1.8.0	• Ao descartar todos os índices em uma coluna com vários índices, o InnoDB falhou ao bloquear uma operação DROP INDEX quando uma restrição de chave estrangeira requer um índice. (Bug nº 16896810) • Resolvida falha de restrição de chave estrangeira adicionada. (Bug nº 16413976) • Correção de uma falha ao buscar um cursor em um procedimento armazenado e analisar ou liberar a tabela ao mesmo tempo. (Bug nº 18158639) • Correção de um bug de incremento automático quando um usuário altera uma tabela para alterar o valor de AUTO_INCREMENT para menos do que o valor de coluna de incremento automático máximo. (Bug nº 16310273)

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL: 2016-08-30 (versão 1.7.0) (obsoleta)	1.7.0	• Melhoria de escalabilidade ao particionar o bloqueio LOCK_grant. (Port WL #8355) • Abrir o cursor em SELECT no procedimento armazenado causa segfault. (Port Bug#16499751) • O MySQL fornece o resultado incorreto com algum uso especial. (Bug nº 11751794) • Falha em GET_SEL_ARG_FOR_KEYPART – causada pelo patch para o Bug nº 11751794. (Bug nº 16208709) • Resultados incorretos para uma consulta simples com GROUP BY. (Bug nº 17909656) • Linhas extras na consulta de semijunção com predicados de intervalo. (Bug nº 16221623) • A inclusão de uma cláusula ORDER BY após uma subconsulta IN pode fazer com que linhas duplicadas sejam retornadas. (Bug nº 16308085) • Falha com explicação para uma consulta com varredura solta para GROUP BY, MyISAM. (Bug nº 16222245) • A varredura de índice solta com o predicado quoted int retorna dados aleatórios. (Bug nº 16394084) • Se o otimizador estivesse usando uma varredura de índice solta, o servidor poderia ser encerrado ao tentar criar uma tabela temporária. (Bug nº 16436567) • COUNT(DISTINCT) não deve contar valores NULL, mas eles foram contados quando o otimizador usou a varredura de índice solta. (Bug nº 17222452) • Se uma consulta teve MIN()/MAX() e aggregate_function (DISTINCT) (por exemplo, SUM(DISTINCT)) e foi executada usando a varredura de índice solta, os valores de resultados de MIN()/MAX() foram definidos incorretamente. (Bug nº 17217128)

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL: 2016-06-01 (versão 1.6.5) (obsoleta)	1.6.5	• SLAVE CAN'T CONTINUE REPLICATION AFTER MASTER'S CRASH RECOVERY (Port Bug #17632285)
Atualizações do mecanismo de banco de dados do Aurora MySQL: 2016-04-06 (versão 1.6) (obsoleta)	1.6.0	• BACKPORT Bug 18694052 FIX FOR ASSERTION '! M_ORDERED_REC_BUFFER' FAILED TO 5.6 (Bug de porta 18305270) • SEGV IN MEMCPY(), HA_PARTITION::POSITION (Bug de porta # 18383840) • WRONG RESULTS WITH PARTITIONING, INDEX_MERGE AND NO PK (Bug de porta # 18167648) • FLUSH TABLES FOR EXPORT: ASSERTION IN HA_PARTITION::EXTRA (Bug de porta # 16943907) • SERVER CRASH IN VIRTUAL HA_ROWS HANDLER::MULTI_RANGE_READ_INFO_CONST (Bug de Porta # 16164031) • RANGE OPTIMIZER CRASHES IN SEL_ARG::RB_INSERT() (Bug de Porta # 16241773)
Atualizações do mecanismo de banco de dados do Aurora MySQL: 2016-01-11 (versão 1.5) (obsoleta)	1.5.0	• Resolução de uma correção incompleta na pesquisa de texto completa MySQL que afeta tabelas em que o nome do banco de dados começa com um dígito. (Bug nº 17607956 de porta)

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL: 2015-12-03 (versão 1.4) (obsoleta)	1.4	• SEGV em FTSPARSE(). (Bug nº 16446108) • O dicionário de dados do InnoDB não é atualizado ao renomear a coluna. (Bug nº 19465984) • Falha do FTS após a renomeação da tabela para um banco de dados diferente. (Bug nº 16834860) • Uma falha na preparação do trigger em tabelas truncadas causa o erro 1054. (Bug nº 18596756) • Alterações de metadados podem causar problemas com a execução de disparadores. (Bug nº 18684393) • A materialização não é escolhida para o campo UTF8 VARCHAR longo. (Bug nº 17566396) • Plano de execução insatisfatório quando ORDER BY com limite X. (Bug nº 16697792) • Bug de backport #11765744 PARA 5.1, 5.5 E 5.6. (Bug nº 17083851) • Problema de mutex no SQL/SQL_SHOW.CC resultando em. SIG6 Origem provavelmente FILL_VARIABLES. (Bug nº 20788853) • Bug de backport #18008907 para versões 5.5+. (Bug nº 18903155) • Correção de um erro de excedente de pilha no MySQL 5.7. (Bug nº 19678930)

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
Atualizações do mecanismo de banco de dados do Aurora MySQL: 2015-10-16 (versão 1.2, 1.3) (obsoleta)	1.2, 1.3	• Eliminar uma consulta dentro de innodb faz com que ela eventualmente falhe com uma declaração. (Bug nº 1608883) • Devido a uma falha ao criar um novo thread para o programador de eventos, a execução do evento ou uma nova conexão, nenhuma mensagem foi gravada no log de erros. (Bug nº 16865959) • Se uma conexão alterou seu banco de dados padrão e, simultaneamente, outra conexão executou SHOW PROCESSLIST, a segunda conexão poderia acessar uma memória inválida ao tentar exibir a memória de banco de dados padrão da primeira conexão. (Bug nº 11765252) • Por padrão, PURGE BINARY LOGS não remove arquivos de log binários em uso ou ativos, mas não forneceu nenhum aviso quando isso ocorreu. (Bug nº 13727933) • Para algumas instruções, perdas de memória poderão resultar quando o otimizador tiver removido cláusulas de subconsulta desnecessárias. (Bug nº 15875919) • Durante o encerramento, o servidor poderia tentar bloquear um mutex não inicializado. (Bug nº 16016493) • Uma declaração preparada que usava GROUP_CONCAT() e uma cláusula ORDER BY que nomeava várias colunas poderia causar o encerramento do servidor. (Bug nº 16075310) • A instrumentação do esquema de performance estava ausente para threads de operador de réplica. (Bug nº 16083949) • STOP SLAVE pode causar um deadlock quando emitido simultaneamente com uma instrução como SHOW STATUS que recuperou os valores de uma ou mais das variáveis de status Slave_retried_transactions, Slave_heartbeat_period, Slave_received_heartbeats,

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
		<code>Slave_last_heartbeat</code> ou <code>Slave_running</code> . (Bug nº 16088188) • Uma consulta de texto completo usando o modo booliano poderia retornar zero resultados em alguns casos em que o termo de pesquisa fosse uma expressão entre aspas. (Bug nº 16206253) • A tentativa do otimizador de remover cláusulas de subconsulta redundantes gerou uma declaração ao executar uma instrução preparada com uma subconsulta na cláusula ON de uma junção em uma subconsulta. (Bug nº 16318585) • GROUP_CONCAT instável, falha em ITEM_SUM::CLEAN_UP_AFTER_REMOVAL. (Bug nº 16347450) • A tentativa de substituir a lista de palavras irrelevantes padrão de pesquisa de texto completo (FTS) do InnoDB, criando uma tabela do InnoDB com a mesma estrutura que INFORMATION_SCHEMA.INNODB_FT_DEFAULT_STOPWORD, resultaria em um erro. (Bug nº 16373868) • Depois que o thread de cliente em um operador realizou FLUSH TABLES WITH READ LOCK e, depois, algumas atualizações no mestre, o operador travou ao executar SHOW SLAVE STATUS. (Bug nº 16387720) • Ao analisar uma string de pesquisa delimitada, como "abc-def" em uma pesquisa de texto completo, o InnoDB agora usa os mesmos delimitadores de palavras que o MyISAM. (Bug nº 16419661) • Falha em FTS_AST_TERM_SET_WILDCARD. (Bug nº 16429306) • SEGFAULT em FTS_AST_VISIT() para teste de FTS RQG. (Bug nº 16435855)

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
		• Para compilações de depuração, quando o otimizador removia um Item_ref apontando para uma subconsulta, ele causava um encerramento do servidor. (Bug nº 16509874) • A pesquisa de texto completo em tabelas do InnoDB falhava em pesquisas por expressões literais combinadas com operadores + ou -. (Bug nº 16516193) • START SLAVE falhou quando o servidor foi iniciado com as opções --master-info-repository=TABLE relay-log-info-repository=TABLE e com a confirmação automática definida como 0, junto com. --skip-slave-start (Bug nº 16533802) • Resultados de pesquisa de texto completo (FTS) muito grandes do InnoDB podem consumir uma quantidade excessiva de memória. (Bug nº 16625973) • Em compilações de depuração, uma declaração poderia ocorrer em OPT_CHECK_ORDER_BY ao usar um binário diretamente em uma string de pesquisa, pois binários podem incluir bytes NULL e outros caracteres não significativos. (Bug nº 16766016) • Para algumas instruções, perdas de memória poderão resultar quando o otimizador tiver removido cláusulas de subconsulta desnecessárias. (Bug nº 16807641) • Era possível causar um deadlock após a emissão de FLUSH TABLES WITH READ LOCK, emitindo STOP SLAVE em uma nova conexão com o operador e, depois, emitindo SHOW SLAVE STATUS usando a conexão original. (Bug nº 16856735) • GROUP_CONCAT() com um separador inválido pode causar um encerramento do servidor. (Bug nº 16870783) • O servidor fez bloqueios excessivos nos mutexes LOCK_active_mi e active_mi->rli->data_lock para qualquer instrução

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
		"padrão" SHOW STATUS LIKE, mesmo quando esse padrão não correspondeu às variáveis de status que usam esses mutexes (Slave_heartbeat_period , Slave_last_heartbeat , Slave_received_heartbeats , Slave_retried_transactions , Slave_running). (Bug nº 16904035) • Uma pesquisa de texto completo usando o modificador IN BOOLEAN MODE resultaria em uma falha de asserção. (Bug #16927092) • A pesquisa de texto completo em tabelas do InnoDB falhava em pesquisas que usavam o operador booleano +. (Bug #17280122) • Deadlock em quatro direções: zombies, purging binlogs, show processlist, show binlogs. (Bug #17283409) • Quando um thread SQL que estava aguardando um bloqueio de confirmação foi eliminado e reiniciado, ele fez com que uma transação fosse ignorada no operador. (Bug #17450876) • Uma falta de pesquisa de texto completo do InnoDB ocorreria devido a um token "sem fim". A string e o comprimento da string devem ser transmitidos para comparação de strings. (Bug #17659310) • Um grande número de tabelas particionadas do InnoDB poderia consumir muito mais memória quando usado no MySQL 5.6 ou 5.7 do que a memória usada pelas mesmas tabelas em versões anteriores do MySQL Server. (Bug #17780517) • Para consultas de texto completo, uma falha ao verificar se num_token é menor que max_proximity_item poderia resultar em uma declaração. (Bug #18233051) • Determinadas consultas para as tabelas INFORMATION_SCHEMA TABLES e COLUMNS poderiam levar a um

Atualização do mecanismo de banco de dados	Versão	Erros corrigidos no MySQL
		uso excessivo de memória quando houvesse um grande número de tabelas do InnoDB vazias. (Bug #18592390) • Ao confirmar uma transação, um sinalizador agora é usado para verificar se um thread foi criado, em vez de verificar o thread propriamente dito, o que usa mais recursos, particularmente ao executar o servidor com master_info_repository=TABLE. (Bug #18684222) • Se um thread de cliente em um operador executasse FLUSH TABLES WITH READ LOCK enquanto o mestre executava um DML, executar SHOW SLAVE STATUS no mesmo cliente era bloqueado, causando um deadlock. (Bug #19843808) • Ordenar por resultado GROUP_CONCAT() poderia causar um encerramento do servidor. (Bug #19880368)
Atualizações do mecanismo de banco de dados do Aurora MySQL: 2015-08-24 (versão 1.1) (obsoleta)	1.1	• Bancos de dados InnoDB com nomes que começam com um dígito causam um erro do analisador de pesquisa de texto completo (FTS). (Bug #17607956) • Pesquisas de texto completo do InnoDB falham em bancos de dados cujos nomes começavam com um dígito. (Bug #17161372) • Para bancos de dados InnoDB no Windows, o ID de objeto de pesquisa de texto completo (FTS) não está no formato hexadecimal esperado. (Bug #16559254) • Uma regressão de código introduzida no MySQL 5.6 afetou negativamente a performance de DROP TABLE e ALTER TABLE. Isso pode causar uma queda de performance entre o MySQL Server 5.5.x e 5.6.x. (Bug #16864741)

Vulnerabilidades de segurança corrigidas no Aurora MySQL

A lista de Vulnerabilidades e exposições comuns (CVE) é uma lista de vulnerabilidades de segurança cibernética publicamente conhecidas. Cada entrada contém um número de identificação, uma descrição e pelo menos uma referência pública.

Encontre nesta página uma lista de vulnerabilidades de segurança corrigidas no Aurora MySQL. Para obter informações gerais sobre a segurança do Aurora, consulte [Segurança no Amazon Aurora](#) no Guia do usuário do Amazon Aurora. Para obter informações adicionais sobre segurança do Aurora MySQL, consulte [Segurança com o Amazon Aurora MySQL](#) no Guia do usuário do Amazon Aurora.

Recomendamos que você sempre atualize para a versão mais recente do Aurora para estar protegido contra vulnerabilidades conhecidas. É possível utilizar esta página para verificar se uma determinada versão do Aurora MySQL tem uma correção para uma vulnerabilidade de segurança específica. Se o cluster não tiver a correção de segurança, será possível ver qual versão do Aurora MySQL é necessário atualizar para essa correção.

CVEs corrigidos no Aurora MySQL versão 1, 2 e 3 também estão listados nas notas de lançamento dessa versão:

- [Atualizações feitas no mecanismo de banco de dados do Amazon Aurora MySQL versão 3](#)
- [Atualizações feitas no mecanismo de banco de dados do Amazon Aurora MySQL versão 2](#)
- [Atualizações feitas no mecanismo de banco de dados do Amazon Aurora MySQL versão 1 \(obsoleta\)](#)

Note

O release inicial do Aurora MySQL versão 3 inclui todos os CVEs corrigidos até o MySQL 8.0.23 edição da comunidade. Para futuros CVEs corrigidos, procure-os aqui e nas notas de release do Aurora MySQL versão 3.

CVEs e versões mínimas do Aurora MySQL corrigidas

- [CVE-2024-37371](#): [3.08.0](#)

- [CVE-2024-25062](#): [3.08.0](#), [2.12.4](#)
- [CVE-2024-21185](#): [3.08.0](#)
- [CVE-2024-21179](#): [3.08.0](#)
- [CVE-2024-21177](#): [3.08.0](#), [2.12.4](#)
- [CVE-2024-21173](#): [3.08.0](#)
- [CVE-2024-21166](#): [3.08.0](#)
- [CVE-2024-21165](#): [3.08.0](#)
- [CVE-2024-21163](#): [3.08.0](#)
- [CVE-2024-21162](#): [3.08.0](#)
- [CVE-2024-21160](#): [3.08.0](#)
- [CVE-2024-21159](#): [3.08.0](#)
- [CVE-2024-21157](#): [3.08.0](#)
- [CVE-2024-21142](#): [3.08.0](#), [2.12.4](#)
- [CVE-2024-21137](#): [3.08.0](#)
- [CVE-2024-21135](#): [3.08.0](#)
- [CVE-2024-21134](#): [3.08.0](#)
- [CVE-2024-21130](#): [3.08.0](#)
- [CVE-2024-21129](#): [3.08.0](#)
- [CVE-2024-21127](#): [3.08.0](#)
- [CVE-2024-21125](#): [3.08.0](#)
- [CVE-2024-21097](#): [2.12.3](#), [2.11.6](#)
- [CVE-2024-21096](#): [2.12.3](#), [2.11.6](#)
- [CVE-2024-21069](#): [2.12.3](#), [2.11.6](#)
- [CVE-2024-21062](#): [2.12.3](#), [2.11.6](#)
- [CVE-2024-21057](#): [2.12.3](#), [2.11.6](#)
- [CVE-2024-21055](#): [2.12.3](#), [2.11.6](#)
- [CVE-2024-21054](#): [2.12.3](#), [2.11.6](#)
- [CVE-2024-21047](#): [2.12.3](#), [2.11.6](#)
- [CVE-2024-21013](#): [3.08.0](#), [2.12.3](#), [2.11.6](#)
- [CVE-2024-21009](#): [2.12.3](#), [2.11.6](#)

- [CVE-2024-21008](#): [2.12.3](#), [2.11.6](#)
- [CVE-2024-21000](#): [3.08.0](#)
- [CVE-2024-20998](#): [2.12.3](#), [2.11.6](#)
- [CVE-2024-20996](#): [3.08.0](#)
- [CVE-2024-20993](#): [2.12.3](#), [2.11.6](#)
- [CVE-2024-20963](#): [2.12.2](#), [2.11.6](#), [2.11.5](#)
- [CVE-2024-0853](#): [3.08.0](#), [3.06.1](#), [3.04.3](#), [2.12.3](#), [2.11.6](#)
- [CVE-2023-44487](#): [3.08.0](#), [3.06.1](#), [2.12.4](#), [2.12.3](#), [2.11.6](#)
- [CVE-2023-39975](#): [3.07.0](#), [3.06.0](#), [3.05.2](#), [3.04.2](#), [2.12.2](#), [2.11.6](#), [2.11.5](#)
- [CVE-2023-38546](#): [3.07.0](#), [3.06.0](#), [3.04.2](#), [2.11.5](#)
- [CVE-2023-38545](#): [3.07.0](#), [3.06.0](#), [3.05.2](#), [3.05.1](#), [3.05.0.1](#), [3.04.2](#), [3.03.3](#), [2.12.2](#), [2.12.1](#), [2.12.0.1](#), [2.11.6](#), [2.11.5](#)
- [CVE-2023-22084](#): [2.11.5](#)
- [CVE-2023-22053](#): [2.12.1](#)
- [CVE-2023-22028](#): [2.12.1](#), [2.11.5](#)
- [CVE-2023-22026](#): [2.12.1](#), [2.11.5](#)
- [CVE-2023-22015](#): [2.12.1](#), [2.11.5](#)
- [CVE-2023-21963](#): [3.04.0](#), [3.03.2](#), [2.12.0](#), [2.11.3](#)
- [CVE-2023-21912](#): [3.04.0](#), [3.03.2](#), [2.12.3](#), [2.12.0](#), [2.11.6](#), [2.11.3](#)
- [CVE-2023-21840](#): [2.12.0](#)
- [CVE-2023-0215](#): [3.04.0](#), [3.03.2](#), [2.12.0](#), [2.11.3](#)
- [CVE-2022-43551](#): [3.04.0](#), [3.03.2](#), [2.12.0](#), [2.11.3](#)
- [CVE-2022-37434](#): [3.05.0](#), [3.04.0](#), [3.03.2](#), [2.12.0](#), [2.11.3](#)
- [CVE-2022-32221](#): [3.03.0](#), [2.12.0](#), [2.11.1](#), [2.07.9](#)
- [CVE-2022-24407](#): [2.12.1](#), [2.11.4](#)
- [CVE-2022-21635](#): [3.04.0](#)
- [CVE-2022-21556](#): [3.04.0](#)
- [CVE-2022-21460](#): [2.11.0](#)
- [CVE-2022-21451](#): [3.03.0](#), [3.02.2](#), [2.11.0](#)
- [CVE-2022-21444](#): [3.03.0](#), [3.02.2](#), [2.11.0](#), [2.10.3](#)
- [CVE-2022-21417](#): [2.11.0](#)

- [CVE-2022-21352](#): [3.04.0](#)
- [CVE-2022-21344](#): [2.10.3](#)
- [CVE-2022-21304](#): [2.11.0](#), [2.10.3](#)
- [CVE-2022-21303](#): [2.11.0](#)
- [CVE-2022-21245](#): [2.11.0](#), [2.10.3](#), [2.07.8](#)
- [CVE-2022-0778](#): [3.02.1](#), [2.11.0](#)
- [CVE-2021-36222](#): [3.02.2](#), [3.01.1](#), [2.12.0](#), [2.11.1](#), [2.11.0](#), [2.10.3](#), [2.10.2](#), [2.07.8](#)
- [CVE-2021-35630](#): [3.04.0](#)
- [CVE-2021-35624](#): [3.04.0](#), [2.10.2](#)
- [CVE-2021-35604](#): [2.10.2](#)
- [CVE-2021-28196](#): [2.11.0](#)
- [CVE-2021-23841](#): [2.11.0](#), [2.10.0](#), [2.09.3](#), [1.23.3](#)
- [CVE-2021-22946](#): [3.02.0](#), [3.01.1](#), [2.12.0](#)
- [CVE-2021-22926](#): [3.02.2](#), [3.01.1](#), [2.11.1](#), [2.11.0](#), [2.10.3](#), [2.10.22.07.8](#)
- [CVE-2021-3712](#): [2.09.3](#)
- [CVE-2021-3449](#): [2.11.0](#), [2.10.0](#), [2.09.3](#), [1.23.3](#)
- [CVE-2021-2390](#): [2.10.2](#)
- [CVE-2021-2389](#): [2.10.2](#)
- [CVE-2021-2385](#): [2.10.2](#)
- [CVE-2021-2356](#): [2.10.2](#)
- [CVE-2021-2307](#): [2.11.0](#), [2.10.1](#), [2.09.3](#), [1.23.4](#)
- [CVE-2021-2226](#): [2.11.0](#), [2.10.1](#), [2.09.3](#), [1.23.4](#)
- [CVE-2021-2202](#): [2.11.0](#)
- [CVE-2021-2194](#): [2.11.0](#), [2.10.1](#)
- [CVE-2021-2179](#): [2.11.0](#)
- [CVE-2021-2178](#): [2.11.0](#)
- [CVE-2021-2174](#): [2.11.0](#), [2.10.1](#), [2.09.3](#)
- [CVE-2021-2171](#): [2.11.0](#), [2.10.1](#), [2.09.3](#)
- [CVE-2021-2169](#): [2.12.0](#), [2.11.1](#), [2.11.0](#), [2.10.1](#), [2.09.3](#)
- [CVE-2021-2166](#): [2.11.0](#), [2.10.1](#), [2.09.3](#)

- [CVE-2021-2160](#): [2.11.0](#), [2.10.1](#), [1.23.4](#)
- [CVE-2021-2154](#): [2.11.0](#), [2.10.1](#), [2.09.3](#), [1.23.4](#)
- [CVE-2021-2144](#): [2.07.3](#)
- [CVE-2021-2060](#): [2.10.1](#), [2.09.3](#), [1.23.4](#)
- [CVE-2021-2032](#): [2.10.1](#), [2.09.3](#), [1.23.4](#)
- [CVE-2021-2001](#): [2.10.1](#), [2.09.3](#), [1.23.4](#)
- [CVE-2020-28196](#): [2.10.0](#), [2.09.3](#), [1.23.3](#)
- [CVE-2020-14867](#): [1.23.2](#), [1.22.4](#)
- [CVE-2020-14812](#): [2.09.2](#), [2.07.4](#), [1.23.2](#), [1.22.4](#)
- [CVE-2020-14793](#): [2.09.2](#), [2.07.4](#), [1.23.2](#), [1.22.4](#)
- [CVE-2020-14790](#): [2.10.0](#), [2.09.2](#), [2.07.4](#)
- [CVE-2020-14776](#): [2.10.0](#)
- [CVE-2020-14775](#): [2.09.2](#), [2.07.4](#)
- [CVE-2020-14769](#): [2.09.3](#), [2.09.2](#), [2.07.4](#), [1.23.2](#), [1.22.4](#)
- [CVE-2020-14765](#): [2.09.2](#), [2.07.4](#), [1.23.2](#), [1.22.4](#)
- [CVE-2020-14760](#): [2.09.2](#), [2.07.4](#)
- [CVE-2020-14672](#): [2.09.2](#), [2.07.4](#), [1.23.2](#), [1.22.4](#)
- [CVE-2020-14567](#): [2.10.0](#), [2.09.1](#), [2.08.3](#), [2.07.3](#)
- [CVE-2020-14559](#): [2.10.0](#), [2.09.1](#), [2.08.3](#), [2.07.3](#), [1.23.1](#), [1.22.3](#)
- [CVE-2020-14553](#): [2.10.0](#), [2.09.1](#), [2.08.3](#), [2.07.3](#)
- [CVE-2020-14547](#): [2.10.0](#), [2.09.1](#), [2.08.3](#), [2.07.3](#)
- [CVE-2020-14540](#): [2.10.0](#), [2.09.1](#), [2.08.3](#), [2.07.3](#)
- [CVE-2020-14539](#): [2.10.0](#), [1.23.1](#), [1.22.3](#)
- [CVE-2020-11105](#): [3.07.0](#), [3.06.0](#), [3.05.2](#), [3.04.2](#), [2.12.1](#), [2.11.5](#)
- [CVE-2020-11104](#): [3.07.0](#), [3.06.0](#), [3.05.2](#), [3.04.2](#), [2.12.1](#), [2.11.5](#)
- [CVE-2020-2812](#): [2.09.1](#), [2.08.3](#), [2.07.3](#), [1.22.3](#)
- [CVE-2020-2806](#): [2.09.1](#), [2.08.3](#), [2.07.3](#)
- [CVE-2020-2780](#): [2.09.1](#), [2.08.3](#), [2.07.3](#), [1.22.3](#)
- [CVE-2020-2765](#): [2.09.1](#), [2.08.3](#), [2.07.3](#)
- [CVE-2020-2763](#): [2.09.1](#), [2.08.3](#), [2.07.3](#), [1.22.3](#)

- [CVE-2020-2760](#): [2.09.1](#), [2.08.3](#), [2.07.3](#), [2.04.9](#)
- [CVE-2020-2579](#): [2.09.1](#), [2.08.3](#), [2.07.3](#), [1.22.3](#)
- [CVE-2020-1971](#): [2.09.2](#), [2.07.4](#), [1.23.2](#), [1.22.4](#)
- [CVE-2019-17543](#): [2.10.2](#), [2.09.3](#), [2.07.7](#)
- [CVE-2019-5443](#): [2.08.0](#), [2.04.9](#)
- [CVE-2019-3822](#): [2.08.0](#), [2.04.9](#)
- [CVE-2019-2960](#): [2.10.2](#), [2.09.3](#), [2.07.7](#)
- [CVE-2019-2948](#): [2.09.0](#)
- [CVE-2019-2924](#): [2.07.0](#), [2.04.9](#), [1.22.0](#)
- [CVE-2019-2923](#): [2.07.0](#), [2.04.9](#), [1.22.0](#)
- [CVE-2019-2922](#): [2.07.0](#), [2.04.9](#), [1.22.0](#)
- [CVE-2019-2911](#): [2.09.0](#), [2.04.9](#), [1.23.0](#)
- [CVE-2019-2910](#): [2.07.0](#), [2.04.9](#), [1.22.0](#)
- [CVE-2019-2805](#): [2.06.0](#), [2.04.9](#), [1.22.0](#)
- [CVE-2019-2791](#): [2.06.0](#), [2.04.9](#)
- [CVE-2019-2778](#): [2.06.0](#), [2.04.9](#)
- [CVE-2019-2758](#): [2.06.0](#), [2.04.9](#)
- [CVE-2019-2740](#): [2.07.3](#), [2.04.9](#), [1.22.0](#)
- [CVE-2019-2739](#): [2.06.0](#), [2.04.9](#)
- [CVE-2019-2731](#): [2.09.0](#)
- [CVE-2019-2730](#): [2.06.0](#), [2.04.9](#), [1.22.0](#)
- [CVE-2019-2628](#): [2.04.9](#)
- [CVE-2019-2581](#): [2.09.0](#)
- [CVE-2019-2537](#): [2.09.0](#), [1.23.0](#)
- [CVE-2019-2534](#): [2.05.0](#), [2.04.3](#), [1.21.0](#), [1.20.0](#), [1.19.1](#)
- [CVE-2019-2482](#): [2.09.0](#)
- [CVE-2019-2434](#): [2.09.0](#)
- [CVE-2019-2420](#): [2.09.0](#)
- [CVE-2018-3284](#): [2.09.0](#)
- [CVE-2018-3251](#): [2.10.0](#)

- [CVE-2018-3156](#): [2.10.0](#)
- [CVE-2018-3155](#): [2.05.0](#), [2.04.3](#)
- [CVE-2018-3143](#): [2.10.0](#), [1.23.2](#)
- [CVE-2018-3065](#): [2.09.0](#)
- [CVE-2018-3064](#): [2.06.0](#), [2.04.9](#), [1.22.0](#)
- [CVE-2018-3058](#): [2.06.0](#), [2.04.9](#), [1.22.0](#)
- [CVE-2018-3056](#): [2.05.0](#), [2.04.4](#)
- [CVE-2018-2813](#): [2.04.9](#)
- [CVE-2018-2787](#): [2.09.0](#), [1.23.0](#)
- [CVE-2018-2786](#): [2.06.0](#), [2.04.9](#)
- [CVE-2018-2784](#): [2.09.0](#), [1.23.0](#)
- [CVE-2018-2696](#): [2.05.0](#), [2.04.5](#), [1.21.0](#), [1.20.0](#), [1.19.5](#)
- [CVE-2018-2645](#): [2.09.0](#), [1.23.0](#)
- [CVE-2018-2640](#): [2.09.0](#), [1.23.0](#)
- [CVE-2018-2612](#): [2.05.0](#), [2.04.3](#), [1.21.0](#), [1.20.0](#), [1.19.1](#)
- [CVE-2018-2562](#): [2.05.0](#), [2.04.4](#), [1.21.0](#), [1.20.0](#), [1.19.2](#)
- [CVE-2018-0734](#): [2.05.0](#), [2.04.3](#), [1.21.0](#), [1.20.0](#), [1.19.1](#)
- [CVE-2017-3653](#): [2.06.0](#), [2.04.9](#), [1.22.0](#)
- [CVE-2017-3599](#): [2.05.0](#), [2.04.3](#), [1.21.0](#), [1.20.0](#), [1.19.1](#)
- [CVE-2017-3465](#): [2.06.0](#), [2.04.9](#)
- [CVE-2017-3464](#): [1.22.0](#), [2.04.9](#)
- [CVE-2017-3455](#): [2.06.0](#), [2.04.9](#)
- [CVE-2017-3329](#): [2.05.0](#), [2.04.4](#), [1.21.0](#), [1.20.0](#), [1.19.2](#)
- [CVE-2017-3244](#): [2.06.0](#), [2.04.9](#), [1.22.0](#)
- [CVE-2016-8287](#): [2.07.2](#)
- [CVE-2016-5634](#): [2.07.2](#)
- [CVE-2016-5612](#): [2.06.0](#), [2.04.9](#), [1.22.0](#)
- [CVE-2016-5440](#): [2.10.0](#)
- [CVE-2016-5439](#): [1.22.0](#), [2.03.3](#)
- [CVE-2016-5436](#): [2.04.9](#), [2.03.3](#)

- [CVE-2016-3518](#): [2.04.5](#)
- [CVE-2016-3495](#): [2.03.2](#)
- [CVE-2016-3486](#): [2.02.2](#)
- [CVE-2016-0606](#): [1.22.0](#)
- [CVE-2015-4904](#): [1.22.0](#)
- [CVE-2015-4879](#): [1.22.0](#)
- [CVE-2015-4864](#): [1.22.0](#)
- [CVE-2015-4830](#): [1.22.0](#)
- [CVE-2015-4826](#): [1.22.0](#)
- [CVE-2015-4737](#): [1.21.0](#), [1.20.0](#), [1.19.5](#)
- [CVE-2015-2620](#): [1.22.0](#)
- [CVE-2015-0382](#): [1.22.0](#)
- [CVE-2015-0381](#): [1.22.0](#)
- [CVE-2014-6555](#): [1.22.0](#)
- [CVE-2014-6489](#): [1.22.0](#)
- [CVE-2014-4260](#): [1.22.0](#)
- [CVE-2014-4258](#): [1.22.0](#)
- [CVE-2014-2444](#): [1.22.0](#)
- [CVE-2014-2436](#): [1.22.0](#)
- [CVE-2014-0393](#): [1.22.0](#)
- [CVE-2013-5908](#): [1.22.0](#)
- [CVE-2013-5881](#): [1.22.0](#)
- [CVE-2013-5807](#): [1.22.0](#)
- [CVE-2013-3811](#): [1.22.0](#)
- [CVE-2013-3807](#): [1.22.0](#)
- [CVE-2013-3806](#): [1.22.0](#)
- [CVE-2013-3804](#): [1.22.0](#)
- [CVE-2013-2381](#): [1.22.0](#)
- [CVE-2013-2378](#): [1.22.0](#)
- [CVE-2013-2375](#): [1.22.0](#)

- [CVE-2013-1523](#): [1.22.0](#)
- [CVE-2012-5615](#): [1.22.0](#)

Histórico do documento das Notas de lançamento do Aurora MySQL

A tabela a seguir descreve as versões de documentação das Notas de lançamento do Aurora MySQL.

Alteração	Descrição	Data
Aurora MySQL versão 3.12.0, compatível com o MySQL 8.0.44	O Aurora MySQL versão 3.12.0 está disponível. Esta versão é compatível com o MySQL 8.0.44. Para obter mais detalhes, consulte o tópico Aurora MySQL versão 3 compatível com o MySQL 8.0 .	17 de fevereiro de 2026
Aurora MySQL versão 3.10.3, compatível com o MySQL 8.0.42	O Aurora MySQL versão 3.10.3 está disponível. Esta versão é compatível com o MySQL 8.0.42. Para obter mais informações, consulte as atualizações do mecanismo de banco de dados Aurora MySQL 2026-01-02 (versão 3.10.3) , compatível com o MySQL 8.0.42).	2 de janeiro de 2026
Aurora MySQL versão 3.04.6, compatível com o MySQL 8.0.28	O Aurora MySQL versão 3.04.6 está disponível. Essa versão é compatível com o MySQL 8.0.28. Para obter mais detalhes, consulte o tópico Aurora MySQL versão 3 compatível com o MySQL 8.0 .	2 de janeiro de 2026

[Aurora MySQL versão 3.11.1,
compatível com o MySQL
8.0.43](#)

O Aurora MySQL versão 3.11.1 está disponível. Esta versão é compatível com o MySQL 8.0.43. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados Aurora MySQL 2025-12-16 \(versão 3.11.1, compatível com o MySQL 8.0.43\)](#).

16 de dezembro de 2025

[Aurora MySQL versão 3.10.2,
compatível com o MySQL
8.0.42](#)

O Aurora MySQL versão 3.10.2 está disponível. Esta versão é compatível com o MySQL 8.0.42. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados Aurora MySQL 2025-11-20 \(versão 3.10.2, compatível com o MySQL 8.0.42\)](#).

20 de novembro de 2025

[Aurora MySQL versão 3.11.0,
compatível com o MySQL
8.0.43](#)

O Aurora MySQL versão 3.11.0 está disponível. Esta versão é compatível com o MySQL 8.0.43. Para obter mais informações, consulte [Atualizações do mecanismo de banco de dados Aurora MySQL 2025-11-13 \(versão 3.11.0, compatível com o MySQL 8.0.43\)](#).

13 de novembro de 2025

[Aurora MySQL versão 3.10.1,
compatível com o MySQL
8.0.42](#)

O Aurora MySQL versão 3.10.1 está disponível. Esta versão é compatível com o MySQL 8.0.42. Para obter mais detalhes, consulte o tópico [Aurora MySQL versão 3 compatível com o MySQL 8.0.](#)

30 de setembro de 2025

[Aurora MySQL versão 3.10.0,
compatível com o MySQL
8.0.42](#)

O Aurora MySQL versão 3.10.0 está disponível. Esta versão é compatível com o MySQL 8.0.42. Para obter mais detalhes, consulte o tópico [Aurora MySQL versão 3 compatível com o MySQL 8.0.](#)

31 de julho de 2025

[Aurora MySQL versão 3.09.0,
compatível com o MySQL
8.0.40](#)

O Aurora MySQL versão 3.09.0 está disponível. Esta versão é compatível com o MySQL 8.0.40. Para obter mais detalhes, consulte o tópico [Aurora MySQL versão 3 compatível com o MySQL 8.0.](#)

14 de maio de 2025

[Aurora MySQL versão 3.04.4,
compatível com o MySQL
8.0.28](#)

O Aurora MySQL versão 3.04.4 está disponível. Essa versão é compatível com o MySQL 8.0.28. Para obter mais detalhes, consulte o tópico [Aurora MySQL versão 3 compatível com o MySQL 8.0.](#)

5 de maio de 2025

[Aurora MySQL versão 2.12.5, compatível com o MySQL 5.7.44](#)

O Aurora MySQL versão 2.12.5 está disponível. Esta versão é compatível com o MySQL 5.7.44. Para obter mais detalhes, consulte [Aurora MySQL versão 2 compatível com o MySQL 5.7](#).

9 de abril de 2025

[Aurora MySQL versão 3.08.2, compatível com o MySQL 8.0.39](#)

O Aurora MySQL versão 3.08.2 está disponível. Esta versão é compatível com o MySQL 8.0.39. Para obter mais detalhes, consulte o tópico [Aurora MySQL versão 3 compatível com o MySQL 8.0](#).

17 de fevereiro de 2025

[Aurora MySQL versão 3.08.1, compatível com o MySQL 8.0.39](#)

O Aurora MySQL versão 3.08.1 está disponível. Esta versão é compatível com o MySQL 8.0.39. Para obter mais detalhes, consulte o tópico [Aurora MySQL versão 3 compatível com o MySQL 8.0](#).

17 de fevereiro de 2025

[Aurora MySQL versão 3.08.0, compatível com o MySQL 8.0.39](#)

O Aurora MySQL versão 3.08.0 está disponível. Esta versão é compatível com o MySQL 8.0.39. Para obter mais detalhes, consulte o tópico [Aurora MySQL versão 3 compatível com o MySQL 8.0](#).

18 de novembro de 2024

[Aurora MySQL versão 2.12.4,
compatível com o MySQL
5.7.44](#)

O Aurora MySQL versão 2.12.4 está disponível. Esta versão é compatível com o MySQL 5.7.44. Para obter mais detalhes, consulte [Aurora MySQL versão 2 compatível com o MySQL 5.7](#).

23 de outubro de 2024

[Aurora MySQL versão 3.07.1,
compatível com o MySQL
8.0.36](#)

O Aurora MySQL versão 3.07.1 está disponível. Esta versão é compatível com o MySQL 8.0.36. Para obter mais detalhes, consulte o tópico [Aurora MySQL versão 3 compatível com o MySQL 8.0](#).

23 de julho de 2024

[Aurora MySQL versão 2.11.6,
compatível com o MySQL
5.7.12](#)

O Aurora MySQL versão 2.11.6 está disponível. Essa versão é compatível com o MySQL 5.7.12. Para obter mais detalhes, consulte [Aurora MySQL versão 2 compatível com o MySQL 5.7](#).

19 de julho de 2024

[Aurora MySQL versão 2.12.3,
compatível com o MySQL
5.7.44](#)

O Aurora MySQL versão 2.12.3 está disponível. Esta versão é compatível com o MySQL 5.7.44. Para obter mais detalhes, consulte [Aurora MySQL versão 2 compatível com o MySQL 5.7](#).

9 de julho de 2024

[Aurora MySQL versão 3.06.1, compatível com o MySQL 8.0.34](#)

O Aurora MySQL versão 3.06.1 está disponível. Esta versão é compatível com o MySQL 8.0.34. Para obter mais detalhes, consulte o tópico [Aurora MySQL versão 3 compatível com o MySQL 8.0.](#)

26 de junho de 2024

[Aurora MySQL versão 3.04.3, compatível com o MySQL 8.0.28](#)

O Aurora MySQL versão 3.04.3 está disponível. Essa versão é compatível com o MySQL 8.0.28. Para obter mais detalhes, consulte o tópico [Aurora MySQL versão 3 compatível com o MySQL 8.0.](#)

26 de junho de 2024

[Aurora MySQL versão 3.07.0, compatível com o MySQL 8.0.36](#)

O Aurora MySQL versão 3.07.0 está disponível. Esta versão é compatível com o MySQL 8.0.36. Para obter mais detalhes, consulte o tópico [Aurora MySQL versão 3 compatível com o MySQL 8.0.](#)

4 de junho de 2024

[Aurora MySQL versão 2.11.5, compatível com o MySQL 5.7.12](#)

O Aurora MySQL versão 2.11.5 está disponível. Essa versão é compatível com o MySQL 5.7.12. Para obter mais detalhes, consulte [Aurora MySQL versão 2 compatível com o MySQL 5.7.](#)

26 de março de 2024

[Aurora MySQL versão 2.12.2, compatível com o MySQL 5.7.44](#)

O Aurora MySQL versão 2.12.2 está disponível. Esta versão é compatível com o MySQL 5.7.44. Para obter mais detalhes, consulte [Aurora MySQL versão 2 compatível com o MySQL 5.7.](#)

19 de março de 2024

[Aurora MySQL versão 3.04.2, compatível com o MySQL 8.0.28](#)

A versão 3.04.2 do Aurora MySQL está disponível. Essa versão é compatível com o MySQL 8.0.28. Para obter mais detalhes, consulte o tópico [Aurora MySQL versão 3 compatível com o MySQL 8.0.](#)

15 de março de 2024

[Aurora MySQL versão 3.06.0, compatível com o MySQL 8.0.34](#)

O Aurora MySQL versão 3.06.0 está disponível. Esta versão é compatível com o MySQL 8.0.34. Para obter mais detalhes, consulte o tópico [Aurora MySQL versão 3 compatível com o MySQL 8.0.](#)

7 de março de 2024

[Aurora MySQL versão 3.05.2, compatível com o MySQL 8.0.32](#)

O Aurora MySQL versão 3.05.2 está disponível. Essa versão é compatível com o MySQL 8.0.32. Para obter mais detalhes, consulte o tópico [Aurora MySQL versão 3 compatível com o MySQL 8.0.](#)

31 de janeiro de 2024

[Aurora MySQL versão 2.12.1, compatível com o MySQL 5.7.40](#)

O Aurora MySQL versão 2.12.1 está disponível. Essa versão é compatível com o MySQL 5.7.40. Para obter mais detalhes, consulte [Aurora MySQL versão 2 compatível com o MySQL 5.7.](#)

28 de dezembro de 2023

[Aurora MySQL versão 3.03.3, compatível com o MySQL 8.0.26](#)

O Aurora MySQL versão 3.03.3 está disponível. Essa versão é compatível com o MySQL 8.0.26. Para obter mais detalhes, consulte o tópico [Aurora MySQL versão 3 compatível com o MySQL 8.0.](#)

8 de dezembro de 2023

[Aurora MySQL versão 3.05.1, compatível com o MySQL 8.0.32](#)

O Aurora MySQL versão 3.05.1 está disponível. Essa versão é compatível com o MySQL 8.0.32. Para obter mais detalhes, consulte o tópico [Aurora MySQL versão 3 compatível com o MySQL 8.0.](#)

21 de novembro de 2023

[Aurora MySQL versão 3.04.1, compatível com o MySQL 8.0.28](#)

O Aurora MySQL versão 3.04.1 está disponível. Essa versão é compatível com o MySQL 8.0.28. Para obter mais detalhes, consulte o tópico [Aurora MySQL versão 3 compatível com o MySQL 8.0.](#)

13 de novembro de 2023

[Aurora MySQL versão 3.05.0.1, compatível com o MySQL 8.0.32 Beta](#)

O Aurora MySQL versão 3.05.0.1 está disponível. Essa versão é compatível com o MySQL 8.0.32. Para obter mais detalhes, consulte o tópico [Aurora MySQL versão 3 compatível com o MySQL 8.0.](#)

30 de outubro de 2023

[Aurora MySQL versão 3.05.0, compatível com o MySQL 8.0.32](#)

O Aurora MySQL versão 3.05.0 está disponível. Essa versão é compatível com o MySQL 8.0.32. Para obter mais detalhes, consulte o tópico [Aurora MySQL versão 3 compatível com o MySQL 8.0.](#)

25 de outubro de 2023

[Aurora MySQL versão 2.12.0.1, compatível com o MySQL 5.7.40 Beta](#)

O Aurora MySQL versão 2.12.0.1 Beta está disponível. Essa versão é compatível com o MySQL 5.7.40. Para obter mais detalhes, consulte [Aurora MySQL versão 2 compatível com o MySQL 5.7.](#)

25 de outubro de 2023

[Aurora MySQL versão 2.11.4, compatível com o MySQL 5.7.12](#)

O Aurora MySQL versão 2.11.4 está disponível. Essa versão é compatível com o MySQL 5.7.12. Para obter mais detalhes, consulte [Aurora MySQL versão 2 compatível com o MySQL 5.7.](#)

17 de outubro de 2023

[Aurora MySQL versão 3.03.2,
compatível com o MySQL
8.0.26](#)

O Aurora MySQL versão 3.03.2 está disponível. Essa versão é compatível com o MySQL 8.0.26. Para obter mais detalhes, consulte o tópico [Aurora MySQL versão 3 compatível com o MySQL 8.0.](#)

29 de agosto de 2023

[Aurora MySQL versão
2.07.10, compatível com o
MySQL 5.7.12](#)

O Aurora MySQL versão 2.07.10 está disponível. Essa versão é compatível com o MySQL 5.7.12. Para obter mais detalhes, consulte [Aurora MySQL versão 2 compatível com o MySQL 5.7.](#)

15 de agosto de 2023

[Aurora MySQL versão 3.04.0,
compatível com o MySQL
8.0.28](#)

O Aurora MySQL versão 3.04.0 está disponível. Essa versão é compatível com o MySQL 8.0.28. Para obter mais detalhes, consulte o tópico [Aurora MySQL versão 3 compatível com o MySQL 8.0.](#)

31 de julho de 2023

[Aurora MySQL versão 2.12.0,
compatível com o MySQL
5.7.40](#)

O Aurora MySQL versão 2.12.0 está disponível. Essa versão é compatível com o MySQL 5.7.40. Para obter mais detalhes, consulte [Aurora MySQL versão 2 compatível com o MySQL 5.7.](#)

25 de julho de 2023

[Aurora MySQL versão 2.11.3,
compatível com o MySQL
5.7.12](#)

O Aurora MySQL versão 2.11.3 está disponível. Essa versão é compatível com o MySQL 5.7.12. Para obter mais detalhes, consulte [Aurora MySQL versão 2 compatível com o MySQL 5.7.](#)

9 de junho de 2023

[Aurora MySQL versão 3.03.1,
compatível com o MySQL
8.0.26](#)

O Aurora MySQL versão 3.03.1 está disponível. Essa versão é compatível com o MySQL 8.0.26. Para obter mais detalhes, consulte o tópico [Aurora MySQL versão 3 compatível com o MySQL 8.0.](#)

11 de maio de 2023

[Aurora MySQL versão 2.07.9,
compatível com o MySQL
5.7.12.](#)

O Aurora MySQL versão 2.07.9 está disponível. Essa versão é compatível com o MySQL 5.7.12. Para obter mais detalhes, consulte [Aurora MySQL versão 2 compatível com o MySQL 5.7.](#)

4 de maio de 2023

[Aurora MySQL versão 3.02.3,
compatível com o MySQL
8.0.23](#)

O Aurora MySQL versão 3.02.3 está disponível. Essa versão tem compatibilidade com o MySQL 8.0.23. Para obter mais detalhes, consulte o tópico [Aurora MySQL versão 3 compatível com o MySQL 8.0.](#)

17 de abril de 2023

[Aurora MySQL versão 2.11.2, compatível com o MySQL 5.7.12](#)

O Aurora MySQL versão 2.11.2 está disponível. Essa versão é compatível com o MySQL 5.7.12. Para obter mais detalhes, consulte [Aurora MySQL versão 2 compatível com o MySQL 5.7](#).

24 de março de 2023

[Aurora MySQL versão 3.03.0, compatível com o MySQL 8.0.26](#)

O Aurora MySQL versão 3.03.0 está disponível. Essa versão é compatível com o MySQL 8.0.26. Para obter mais detalhes, consulte o tópico [Aurora MySQL versão 3 compatível com o MySQL 8.0](#).

1 de março de 2023

[Aurora MySQL versão 2.11.1, compatível com o MySQL 5.7.12](#)

O Aurora MySQL versão 2.11.1 está disponível. Essa versão é compatível com o MySQL 5.7.12. Para obter mais detalhes, consulte [Aurora MySQL versão 2 compatível com o MySQL 5.7](#).

14 de fevereiro de 2023

[Aurora MySQL versão 3.02.2, compatível com o MySQL 8.0.23](#)

O Aurora MySQL versão 3.02.2 está disponível. Essa versão tem compatibilidade com o MySQL 8.0.23. Para obter mais detalhes, consulte o tópico [Aurora MySQL versão 3 compatível com o MySQL 8.0](#).

18 de novembro de 2022

[Aurora MySQL versão 2.10.3, compatível com o MySQL 5.7](#)

O Aurora MySQL versão 2.10.3 está disponível. Essa versão é compatível com o MySQL 5.7. Para obter mais detalhes, consulte [Aurora MySQL versão 2 compatível com o MySQL 5.7](#).

1º de novembro de 2022

[Aurora MySQL versão 2.11.0, compatível com o MySQL 5.7.12](#)

O Aurora MySQL versão 2.11.0 está disponível. Essa versão é compatível com o MySQL 5.7.12. Para obter mais detalhes, consulte [Aurora MySQL versão 2 compatível com o MySQL 5.7](#).

25 de outubro de 2022

[Aurora MySQL versão 3.02.1, compatível com o MySQL 8.0.23](#)

O Aurora MySQL versão 3.02.1 está disponível. Essa versão tem compatibilidade com o MySQL 8.0.23. Para obter mais detalhes, consulte o tópico [Aurora MySQL versão 3 compatível com o MySQL 8.0](#).

7 de setembro de 2022

[Atualização no local do Aurora Serverless v1 compatível com MySQL 5.6](#)

Você pode realizar uma atualização no local de um cluster do Aurora Serverless v1 compatível com MySQL 5.6 a fim de alterar um cluster existente para um cluster do Aurora Serverless v1 compatível com MySQL 5.7. Para obter mais informações, consulte [Aurora MySQL Serverless 5.7 engine updates 2020-06-18 \(version 2.07.1\)](#).

16 de junho de 2022

[Atualização no local do Aurora Serverless v1 compatível com MySQL 5.6](#)

Você pode realizar uma atualização no local de um cluster do Aurora Serverless v1 compatível com MySQL 5.6 a fim de alterar um cluster existente para um cluster do Aurora Serverless v1 compatível com MySQL 5.7. Para obter mais informações, consulte [Aurora MySQL Serverless 5.7 engine updates 2020-06-18 \(version 2.07.1\)](#).

16 de junho de 2022

[Atualizações do mecanismo de banco de dados do Aurora MySQL de 2022-06-16 \(versão 2.07.8\) estão disponíveis.](#)

O Aurora MySQL versão 2.07.8 está disponível.

16 de junho de 2022

[Aurora MySQL versão 3.02.0, compatível com o MySQL 8.0.23](#)

O Aurora MySQL versão 3.02.0 está disponível. Essa versão tem compatibilidade com o MySQL 8.0.23. Para obter mais detalhes, consulte o tópico [Aurora MySQL versão 3 compatível com o MySQL 8.0](#).

20 de abril de 2022

[Aurora MySQL versão 3.01.1, compatível com o MySQL 8.0.23](#)

O Aurora MySQL versão 3.01.1 está disponível. Essa versão tem compatibilidade com o MySQL 8.0.23. Para obter mais detalhes, consulte o tópico [Aurora MySQL versão 3 compatível com o MySQL 8.0](#).

15 de abril de 2022

Lançamento inicial	Versão inicial das Notas de lançamento do Aurora MySQL.	22 de março de 2022
Aurora MySQL versão 2.10.2	O Aurora MySQL versão 2.10.2 está disponível.	26 de janeiro de 2022
Aurora MySQL versão 2.08.4	O Aurora MySQL versão 2.08.4 está disponível.	6 de janeiro de 2022
Aurora MySQL versão 2.07.7	O Aurora MySQL versão 2.07.7 está disponível.	24 de novembro de 2021
Aurora MySQL versão 3.01.0, compatível com o MySQL 8.0.23	O Aurora MySQL versão 3.01.0 está disponível. Essa versão tem compatibilidade com o MySQL 8.0.23. Para obter mais detalhes, consulte o tópico Aurora MySQL versão 3 compatível com o MySQL 8.0 .	18 de novembro de 2021
Aurora MySQL versão 2.09.3	O Aurora MySQL versão 2.09.3 está disponível.	12 de novembro de 2021
Aurora MySQL versão 2.10.1	O Aurora MySQL versão 2.10.1 está disponível.	21 de outubro de 2021
Versão 1.23.4 do Aurora MySQL	A versão 1.23.4 do Aurora MySQL está disponível.	30 de setembro de 2021
Versão 2.07.6 do Aurora MySQL	A versão 2.07.6 do Aurora MySQL está disponível.	2 de setembro de 2021
Aurora MySQL versão 2.07.5	O Aurora MySQL versão 2.07.5 está disponível.	6 de julho de 2021
Aurora MySQL versão 1.23.3	O Aurora MySQL versão 1.23.3 está disponível.	28 de junho de 2021

Aurora MySQL versão 1.22.5	A versão 1.22.5 do Aurora MySQL está disponível.	3 de junho de 2021
Aurora MySQL versão 2.10.0	O Aurora MySQL versão 2.10.0 está disponível. Alguns dos destaques incluem maior disponibilidade de instâncias de leitura durante a reinicialização do gravador, melhorias na correção de tempo de inatividade zero (ZDP), melhorias na reinicialização com tempo de inatividade zero (ZDR) e otimização do cache de log binário. I/O	25 de maio de 2021
Aurora MySQL versão 1.23.2	A versão 1.23.2 do Aurora MySQL está disponível.	18 de março de 2021
Aurora MySQL versão 2.07.4	A versão 2.07.4 do Aurora MySQL está disponível.	4 de março de 2021
Aurora MySQL versão 1.22.4	A versão 1.22.4 do Aurora MySQL está disponível.	4 de março de 2021
Aurora MySQL versão 2.09.2	A versão 2.09.2 do Aurora MySQL está disponível.	26 de fevereiro de 2021
Aurora MySQL versão 2.09.1	O Aurora MySQL versão 2.09.1 está disponível.	11 de dezembro de 2020
Aurora MySQL versão 1.23.1	O Aurora MySQL versão 1.23.1 está disponível.	24 de novembro de 2020
Aurora MySQL versão 2.08.3	O Aurora MySQL versão 2.08.3 está disponível.	12 de novembro de 2020
Aurora MySQL versão 2.07.3	O Aurora MySQL versão 2.07.3 está disponível.	10 de novembro de 2020

Aurora MySQL versão 1.22.3	O Aurora MySQL versão 1.22.3 está disponível.	9 de novembro de 2020
Aurora MySQL versão 2.09.0	Aurora MySQL versão 2.09.0 está disponível.	17 de setembro de 2020
Aurora MySQL versão 1.23.0	O Aurora MySQL versão 1.23.0 está disponível.	2 de setembro de 2020
Aurora MySQL versão 2.08.2	O Aurora MySQL versão 2.08.2 está disponível.	28 de agosto de 2020
Aurora MySQL versão 2.04.9	Aurora MySQL versão 2.04.9 está disponível.	14 de agosto de 2020
Aurora MySQL versão 2.08.1	O Aurora MySQL versão 2.08.1 está disponível.	18 de junho de 2020
Aurora MySQL versão 1.22.2 para clusters de consulta paralela	O Aurora MySQL versão 1.22.2 está disponível quando você cria um cluster de consulta paralela.	18 de junho de 2020
Aurora MySQL versão 1.20.1 para clusters de consulta paralela	O Aurora MySQL versão 1.20.1 está disponível quando você cria um cluster de consulta paralela.	11 de junho de 2020
Aurora MySQL versão 2.08.0	O Aurora MySQL versão 2.08.0 está disponível.	2 de junho de 2020
Aurora MySQL versão 1.19.6 para clusters de consulta paralela	O Aurora MySQL versão 1.19.6 está disponível quando você cria um cluster de consulta paralela.	2 de junho de 2020
Aurora MySQL versão 2.07.2	O Aurora MySQL versão 2.07.2 está disponível.	17 de abril de 2020

Aurora MySQL versão 1.22.2	O Aurora MySQL versão 1.22.2 está disponível.	5 de março de 2020
Aurora MySQL versão 1.20.1	O Aurora MySQL versão 1.20.1 está disponível.	5 de março de 2020
Aurora MySQL versão 1.19.6	O Aurora MySQL versão 1.19.6 está disponível.	5 de março de 2020
Aurora MySQL versão 1.17.9	O Aurora MySQL versão 1.17.9 está disponível.	5 de março de 2020
Aurora MySQL versão 2.07.1	O Aurora MySQL versão 2.07.1 está disponível.	23 de dezembro de 2019
Aurora MySQL versão 1.22.1	O Aurora MySQL versão 1.22.1 está disponível.	23 de dezembro de 2019
Aurora MySQL versão 2.07.0	O Aurora MySQL versão 2.07.0 está disponível.	25 de novembro de 2019
Aurora MySQL versão 1.22.0	O Aurora MySQL versão 1.22.0 está disponível.	25 de novembro de 2019
Aurora MySQL versão 1.21.0	O Aurora MySQL versão 1.21.0 está disponível.	25 de novembro de 2019
Aurora MySQL versão 2.06.0	O Aurora MySQL versão 2.06.0 está disponível.	22 de novembro de 2019
Aurora MySQL versão 2.04.8	O Aurora MySQL versão 2.04.8 está disponível.	20 de novembro de 2019
Aurora MySQL versão 2.04.7	O Aurora MySQL versão 2.04.7 está disponível.	14 de novembro de 2019
Aurora MySQL versão 2.05.0	O Aurora MySQL versão 2.05.0 está disponível.	11 de novembro de 2019

Aurora MySQL versão 1.20.0	O Aurora MySQL versão 1.20.0 está disponível.	11 de novembro de 2019
Aurora MySQL versão 2.04.6	O Aurora MySQL versão 2.04.6 está disponível.	19 de setembro de 2019
Aurora MySQL versão 1.19.5	O Aurora MySQL versão 1.19.5 está disponível.	19 de setembro de 2019
Aurora MySQL versão 2.04.5	O Aurora MySQL versão 2.04.5 está disponível.	8 de julho de 2019
Aurora MySQL versão 1.19.2	O Aurora MySQL versão 1.19.2 está disponível.	5 de junho de 2019
Aurora MySQL versão 2.04.4	O Aurora MySQL versão 2.04.4 está disponível.	29 de maio de 2019
Aurora MySQL versão 2.04.3	O Aurora MySQL versão 2.04.3 está disponível.	9 de maio de 2019
Aurora MySQL versão 1.19.1	O Aurora MySQL versão 1.19.1 está disponível.	9 de maio de 2019
Aurora MySQL versão 2.04.2	O Aurora MySQL versão 2.04.2 está disponível.	2 de maio de 2019
Aurora MySQL versão 2.04.1	O Aurora MySQL versão 2.04.1 está disponível.	25 de março de 2019
Aurora MySQL versão 2.04	O Aurora MySQL versão 2.04 está disponível.	25 de março de 2019
Aurora MySQL versão 2.03.4	O Aurora MySQL versão 2.03.4 está disponível.	7 de fevereiro de 2019
Aurora MySQL versão 1.19.0	O Aurora MySQL versão 1.19.0 está disponível.	7 de fevereiro de 2019

Aurora MySQL versão 2.03.3	O Aurora MySQL versão 2.03.3 está disponível.	18 de janeiro de 2019
Aurora MySQL versão 1.17.8	O Aurora MySQL versão 1.17.8 está disponível.	17 de janeiro de 2019
Aurora MySQL versão 2.03.2	O Aurora MySQL versão 2.03.2 está disponível.	9 de janeiro de 2019
Aurora MySQL versão 2.03.1	O Aurora MySQL versão 2.03.1 está disponível.	24 de outubro de 2018
Aurora MySQL versão 2.03	O Aurora MySQL versão 2.03 está disponível.	11 de outubro de 2018
Aurora MySQL versão 2.02.5	O Aurora MySQL versão 2.02.5 está disponível.	8 de outubro de 2018
Aurora MySQL versão 1.17.7	O Aurora MySQL versão 1.17.7 está disponível.	8 de outubro de 2018
Aurora MySQL versão 2.02.4	O Aurora MySQL versão 2.02.4 está disponível.	21 de setembro de 2018
Aurora MySQL versão 1.18.0	O Aurora MySQL versão 1.18.0 está disponível.	20 de setembro de 2018
Aurora MySQL versão 1.17.6	O Aurora MySQL versão 1.17.6 está disponível.	6 de setembro de 2018

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.