



Versionshinweise für Aurora MySQL

# Amazon Aurora



# Amazon Aurora: Versionshinweise für Aurora MySQL

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

Die Handelsmarken und Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

---

# Table of Contents

Aurora-MySQL-Versionshinweise .....	1
Aurora MySQL-Veröffentlichungskalender .....	2
Veröffentlichungskalender für Aurora MySQL-Hauptversionen .....	2
Veröffentlichungskalender für Aurora MySQL-Nebenversionen .....	3
Aurora-MySQL-Version 3 .....	5
Aurora MySQL-Updates: 2024-06-04 (Version 3.07.0, kompatibel mit MySQL 8.0.36) .....	6
Verbesserungen .....	7
Integration von MySQL-Fehlerbehebungen (Community Edition) .....	11
Aurora MySQL-Updates: 2024-03-07 (Version 3.06.0, kompatibel mit MySQL 8.0.34) .....	12
Neue Features .....	12
Verbesserungen .....	14
Integration von MySQL-Fehlerbehebungen (Community Edition) .....	17
Aurora MySQL-Updates: 2024-01-31 (Version 3.05.2, kompatibel mit MySQL 8.0.32)	
Standard .....	17
Verbesserungen .....	18
Integration von MySQL-Fehlerbehebungen (Community Edition) .....	20
Aurora MySQL-Updates: 2023-11-21 (Version 3.05.1, kompatibel mit MySQL 8.0.32) .....	20
Verbesserungen .....	20
Integration von MySQL-Fehlerbehebungen (Community Edition) .....	21
Aurora-MySQL-Updates: 30.10.2023 (Version 3.05.0.1, kompatibel mit MySQL 8.0.32) Beta .....	21
Verbesserungen .....	22
Aurora-MySQL-Updates: 25.10.2023 (Version 3.05.0, kompatibel mit MySQL 8.0.32) .....	22
Verbesserungen .....	23
Integration von MySQL-Fehlerbehebungen (Community Edition) .....	27
Aurora MySQL-Updates: 2024-03-15 (Version 3.04.2, kompatibel mit MySQL 8.0.28) .....	28
Verbesserungen .....	29
Integration von MySQL-Fehlerbehebungen (Community Edition) .....	31
Aurora MySQL-Updates: 2023-11-13 (Version 3.04.1, kompatibel mit MySQL 8.0.28)	
Standard .....	31
Verbesserungen .....	33
Integration von MySQL-Fehlerbehebungen (Community Edition) .....	34
Aurora MySQL-Updates: 2023-07-31 (Version 3.04.0, kompatibel mit MySQL 8.0.28) .....	35
Verbesserungen .....	36
Integration von MySQL-Fehlerbehebungen (Community Edition) .....	42

Aurora-MySQL-Updates: 08.12.2023 (Version 3.03.3, kompatibel mit MySQL 8.0.26) .....	43
Verbesserungen .....	44
Integration von MySQL-Fehlerbehebungen (Community Edition) .....	45
Aurora-MySQL-Updates: 29.08.2023 (Version 3.03.2, kompatibel mit MySQL 8.0.26) .....	45
Verbesserungen .....	46
Integration von MySQL-Fehlerbehebungen (Community Edition) .....	48
Aurora-MySQL-Updates: 11.05.2023 (Version 3.03.1, kompatibel mit MySQL 8.0.26) .....	48
Verbesserungen .....	49
Integration von MySQL-Fehlerbehebungen (Community Edition) .....	51
Aurora-MySQL-Updates 01.03.2023 (Version 3.03.0, kompatibel mit MySQL 8.0.26) Upgrades auf diese Version werden nicht unterstützt. ....	52
Verbesserungen .....	52
Integration von MySQL-Fehlerbehebungen (Community Edition) .....	55
Aurora-MySQL-Updates 17.04.2022 (Version 3.02.3, kompatibel mit MySQL 8.0.23). Ende des Standard-Supports 15. Januar 2024. ....	56
Verbesserungen .....	57
Aurora-MySQL-Updates 18.11.2022 (Version 3.02.2, kompatibel mit MySQL 8.0.23). Ende des Standard-Supports 15. Januar 2024. ....	58
Verbesserungen .....	59
Integration von MySQL-Fehlerbehebungen (Community Edition) .....	55
Aurora-MySQL-Updates 07.09.2022 (Version 3.02.1, kompatibel mit MySQL 8.0.23). Ende des Standard-Supports 15. Januar 2024. Upgrades auf diese Version werden nicht unterstützt. ....	62
Verbesserungen .....	63
Aurora-MySQL-Updates 20.04.2022 (Version 3.02.0, kompatibel mit MySQL 8.0.23). Ende des Standard-Supports 15. Januar 2024. Upgrades auf diese Version werden nicht unterstützt. ....	64
Verbesserungen .....	65
Integration von MySQL-Fehlerbehebungen (Community Edition): .....	67
Aurora-MySQL-Updates 15.04.2022 (Version 3.01.1, kompatibel mit MySQL 8.0.23). Ende des Standard-Supports 15. Januar 2024. Upgrades auf diese Version werden nicht unterstützt. ....	67
Verbesserungen .....	68
Integration von MySQL-Fehlerbehebungen (Community Edition): .....	70
Aurora-MySQL-Updates 18.11.2021 (Version 3.01.0, kompatibel mit MySQL 8.0.23). Ende des Standard-Supports 15. Januar 2024. Upgrades auf diese Version werden nicht unterstützt. ....	70
Verbesserungen .....	71
Aurora-MySQL-Version 2 .....	72
Aurora MySQL-Updates: 2024-03-19 (Version 2.12.2, kompatibel mit MySQL 5.7.44) .....	74

Verbesserungen .....	75
Integration von MySQL-Fehlerbehebungen (Community Edition) .....	76
In Aurora-MySQL-Version 2 nicht unterstützte Funktionen .....	76
Kompatibilität mit MySQL 5.7 .....	76
Aurora-MySQL-Updates: 28.12.2023 (Version 2.12.1, kompatibel mit MySQL 5.7.40) .....	77
Verbesserungen .....	78
Integration von MySQL-Fehlerbehebungen (Community Edition) .....	81
In Aurora-MySQL-Version 2 nicht unterstützte Funktionen .....	81
Kompatibilität mit MySQL 5.7 .....	81
Aurora-MySQL-Updates: 25.10.2023 (Version 2.12.0.1, kompatibel mit MySQL 5.7.40) .....	82
Verbesserungen .....	83
Aurora-MySQL-Updates: 25.07.2023 (Version 2.12.0, kompatibel mit MySQL 5.7.40) .....	83
Verbesserungen .....	84
Integration von MySQL-Fehlerbehebungen (Community Edition) .....	87
In Aurora-MySQL-Version 2 nicht unterstützte Funktionen .....	87
Kompatibilität mit MySQL 5.7 .....	88
Aurora MySQL-Updates: 2024-03-26 (Version 2.11.5, kompatibel mit MySQL 5.7.12)	
Standard .....	88
Verbesserungen .....	89
In Aurora-MySQL-Version 2 nicht unterstützte Funktionen .....	91
Kompatibilität mit MySQL 5.7 .....	91
Aurora-MySQL-Updates: 17.10.2023 (Version 2.11.4, kompatibel mit MySQL 5.7.12) .....	92
Verbesserungen .....	93
Integration von MySQL-Fehlerbehebungen (Community Edition) .....	95
In Aurora-MySQL-Version 2 nicht unterstützte Funktionen .....	96
Kompatibilität mit MySQL 5.7 .....	96
Aurora-MySQL-Updates: 09.06.2023 (Version 2.11.3, kompatibel mit MySQL 5.7.12) .....	96
Verbesserungen .....	98
In Aurora-MySQL-Version 2 nicht unterstützte Funktionen .....	99
Kompatibilität mit MySQL 5.7 .....	99
Aurora-MySQL-Updates: 24.03.2023 (Version 2.11.2, kompatibel mit MySQL 5.7.12) .....	100
Verbesserungen .....	101
In Aurora-MySQL-Version 2 nicht unterstützte Funktionen .....	101
Kompatibilität mit MySQL 5.7 .....	102
Aurora-MySQL-Updates: 14.02.2023 (Version 2.11.1, kompatibel mit MySQL 5.7.12) .....	102
Verbesserungen .....	103

Vergleich mit Aurora MySQL Version 1 .....	104
Kompatibilität mit MySQL 5.7 .....	105
Aurora-MySQL-Updates 25.10.2022 (Version 2.11.0, kompatibel mit MySQL 5.7.12). Diese Version ist für neue Erstellungen nicht verfügbar. ....	106
Verbesserungen .....	106
Integration von MySQL-Fehlerbehebungen (Community Edition) .....	111
Vergleich mit Aurora MySQL Version 1 .....	112
Kompatibilität mit MySQL 5.7 .....	113
Aurora-MySQL-Updates: 01.11.2022 (Version 2.10.3) (veraltet) .....	113
Verbesserungen .....	114
Integration von MySQL-Fehlerbehebungen (Community Edition) .....	115
Vergleich mit Aurora MySQL Version 1 .....	116
Kompatibilität mit MySQL 5.7 .....	116
Aurora-MySQL-Updates: 26.01.2022 (Version 2.10.2) (veraltet) .....	117
Verbesserungen .....	118
Integration von MySQL-Fehlerbehebungen (Community Edition) .....	55
Vergleich mit Aurora MySQL Version 1 .....	122
Kompatibilität mit MySQL 5.7 .....	122
Aurora-MySQL-Updates: 21.10.2021 (Version 2.10.1) (veraltet) .....	123
Verbesserungen .....	124
Integration von MySQL-Fehlerbehebungen (Community Edition): .....	125
Vergleich mit Aurora MySQL Version 1 .....	125
Kompatibilität mit MySQL 5.7 .....	126
Aurora-MySQL-Updates: 25.05.2021 (Version 2.10.0) (veraltet) .....	126
Verbesserungen .....	127
Integration von MySQL-Fehlerbehebungen (Community Edition): .....	130
Vergleich mit Aurora MySQL Version 1 .....	134
Kompatibilität mit MySQL 5.7 .....	134
Aurora-MySQL-Updates: 12.11.2021 (Version 2.09.3) (veraltet) .....	135
Verbesserungen .....	136
Integration von MySQL-Fehlerbehebungen (Community Edition): .....	139
Vergleich mit Aurora MySQL Version 1 .....	139
Kompatibilität mit MySQL 5.7 .....	140
Aurora-MySQL-Updates: 26.02.2021 (Version 2.09.2) (veraltet) .....	140
Verbesserungen .....	141
Vergleich mit Aurora MySQL Version 1 .....	142

Kompatibilität mit MySQL 5.7 .....	143
Aurora-MySQL-Updates: 11.12.2020 (Version 2.09.1) (veraltet) .....	144
Verbesserungen .....	144
Integration von MySQL-Fehlerbehebungen (Community Edition): .....	146
Vergleich mit Aurora MySQL Version 1 .....	146
Kompatibilität mit MySQL 5.7 .....	147
Aurora-MySQL-Updates: 17.09.2020 (Version 2.09.0) (veraltet) .....	147
Verbesserungen .....	148
Integration von MySQL-Fehlerbehebungen (Community Edition): .....	154
Vergleich mit Aurora MySQL Version 1 .....	155
Kompatibilität mit MySQL 5.7 .....	156
Aurora-MySQL-Updates: 06.01.2022 (Version 2.08.4) (veraltet) .....	157
Verbesserungen .....	157
Vergleich mit Aurora MySQL Version 1 .....	158
Kompatibilität mit MySQL 5.7 .....	158
Aurora-MySQL-Updates: 12.11.2020 (Version 2.08.3) (veraltet) .....	159
Verbesserungen .....	160
Integration von MySQL-Fehlerbehebungen (Community Edition): .....	161
Vergleich mit Aurora MySQL Version 1 .....	161
Kompatibilität mit MySQL 5.7 .....	161
Aurora-MySQL-Updates: 28.08.2020 (Version 2.08.2) (veraltet) .....	162
Verbesserungen .....	163
Vergleich mit Aurora MySQL Version 1 .....	163
Kompatibilität mit MySQL 5.7 .....	164
Aurora-MySQL-Updates: 18.06.2020 (Version 2.08.1) (veraltet) .....	165
Verbesserungen .....	165
Vergleich mit Aurora MySQL Version 1 .....	166
Kompatibilität mit MySQL 5.7 .....	166
Aurora-MySQL-Updates: 02.06.2020 (Version 2.08.0) (veraltet) .....	167
Verbesserungen .....	168
Integration von MySQL-Fehlerbehebungen (Community Edition): .....	171
Vergleich mit Aurora MySQL Version 1 .....	171
Kompatibilität mit MySQL 5.7 .....	172
Aurora-MySQL-Update: 15.08.2023 (Version 2.07.10, kompatibel mit MySQL 5.7.12) .....	173
Verbesserungen .....	174
In Aurora-MySQL-Version 2 nicht unterstützte Funktionen .....	175

Kompatibilität mit MySQL 5.7 .....	175
Aurora-MySQL-Update: 04.05.2023 (Version 2.07.9, kompatibel mit MySQL 5.7.12) .....	176
Verbesserungen .....	177
In Aurora-MySQL-Version 2 nicht unterstützte Funktionen .....	177
Kompatibilität mit MySQL 5.7 .....	178
Aurora-MySQL-Updates: 16.06.2022 (Version 2.07.8) (veraltet) .....	178
Verbesserungen .....	179
Integration von MySQL-Fehlerbehebungen (Community Edition): .....	180
Vergleich mit Aurora MySQL Version 1 .....	180
Kompatibilität mit MySQL 5.7 .....	180
Aurora-MySQL-Updates: 24.11.2024 (Version 2.07.7) (veraltet) .....	181
Verbesserungen .....	182
Vergleich mit Aurora MySQL Version 1 .....	183
Kompatibilität mit MySQL 5.7 .....	184
Aurora-MySQL-Updates: 02.09.2021 (Version 2.07.6) (veraltet) .....	184
Integration von MySQL-Fehlerbehebungen (Community Edition): .....	185
Vergleich mit Aurora MySQL Version 1 .....	185
Kompatibilität mit MySQL 5.7 .....	186
Aurora-MySQL-Updates: 06.07.2021 (Version 2.07.5) (veraltet) .....	186
Verbesserungen .....	187
Vergleich mit Aurora MySQL Version 1 .....	187
Kompatibilität mit MySQL 5.7 .....	188
Aurora-MySQL-Updates: 04.03.2021 (Version 2.07.4) (veraltet) .....	189
Verbesserungen .....	189
Integration von MySQL-Fehlerbehebungen (Community Edition): .....	190
Vergleich mit Aurora MySQL Version 1 .....	190
Kompatibilität mit MySQL 5.7 .....	191
Aurora-MySQL-Updates: 10.11.2020 (Version 2.07.3) (veraltet) .....	192
Verbesserungen .....	192
Integration von MySQL-Fehlerbehebungen (Community Edition): .....	194
Vergleich mit Aurora MySQL Version 1 .....	195
Kompatibilität mit MySQL 5.7 .....	196
Aurora-MySQL-Updates: 17.04.2020 (Version 2.07.2) (veraltet) .....	197
Verbesserungen .....	197
Integration von MySQL-Fehlerbehebungen (Community Edition): .....	198
Vergleich mit Aurora MySQL Version 1 .....	198



Kompatibilität mit MySQL 5.7 .....	199
Aurora-MySQL-Updates: 23.12.2019 (Version 2.07.1) (veraltet) .....	200
Verbesserungen .....	201
Vergleich mit Aurora MySQL Version 1 .....	201
Kompatibilität mit MySQL 5.7 .....	201
Aurora-MySQL-Updates: 25.11.2019 (Version 2.07.0) (veraltet) .....	202
Verbesserungen .....	203
Integration von MySQL-Fehlerbehebungen (Community Edition): .....	204
Vergleich mit Aurora MySQL Version 1 .....	205
Kompatibilität mit MySQL 5.7 .....	205
Aurora-MySQL-Updates: 22.11.2019 (Version 2.06.0) (veraltet) .....	206
Verbesserungen .....	207
Vergleich mit Aurora MySQL Version 1 .....	209
Kompatibilität mit MySQL 5.7 .....	210
Aurora-MySQL-Updates: 11.11.2019 (Version 2.05.0) (veraltet) .....	210
Verbesserungen .....	211
Integration von MySQL-Fehlerbehebungen .....	212
Vergleich mit Aurora MySQL Version 1 .....	212
Kompatibilität mit MySQL 5.7 .....	213
Aurora-MySQL-Updates: 14.08.2020 (Version 2.04.9) (veraltet) .....	214
Verbesserungen .....	214
Integration von MySQL-Fehlerbehebungen .....	217
Vergleich mit Aurora MySQL Version 1 .....	218
Kompatibilität mit MySQL 5.7 .....	219
Aurora-MySQL-Updates: 20.11.2019 (Version 2.04.8) (veraltet) .....	219
Verbesserungen .....	220
Vergleich mit Aurora MySQL Version 1 .....	221
Kompatibilität mit MySQL 5.7 .....	221
Aurora-MySQL-Updates: 14.11.2019 (Version 2.04.7) (veraltet) .....	222
Verbesserungen .....	223
Vergleich mit Aurora MySQL Version 1 .....	224
Kompatibilität mit MySQL 5.7 .....	224
Aurora-MySQL-Updates: 19.09.2019 (Version 2.04.6) (veraltet) .....	225
Verbesserungen .....	226
Integration von MySQL-Fehlerbehebungen .....	226
Vergleich mit Aurora MySQL Version 1 .....	226

Kompatibilität mit MySQL 5.7 .....	227
Aurora-MySQL-Updates: 08.07.2019 (Version 2.04.5) (veraltet) .....	228
Verbesserungen .....	228
Vergleich mit Aurora MySQL Version 1 .....	229
Kompatibilität mit MySQL 5.7 .....	230
Aurora-MySQL-Updates: 29.05.2019 (Version 2.04.4) (veraltet) .....	230
Verbesserungen .....	231
Vergleich mit Aurora MySQL Version 1 .....	232
Kompatibilität mit MySQL 5.7 .....	232
Aurora-MySQL-Updates: 09.05.2019 (Version 2.04.3) (veraltet) .....	233
Verbesserungen .....	234
Vergleich mit Aurora MySQL Version 1 .....	234
Kompatibilität mit MySQL 5.7 .....	235
Aurora-MySQL-Updates: 02.05.2019 (Version 2.04.2) (veraltet) .....	235
Verbesserungen .....	236
Integration von MySQL-Fehlerbehebungen .....	237
Vergleich mit Aurora MySQL Version 1 .....	237
Kompatibilität mit MySQL 5.7 .....	237
Aurora-MySQL-Updates: 25.03.2019 (Version 2.04.1) (veraltet) .....	238
Verbesserungen .....	239
Vergleich mit Aurora MySQL Version 1 .....	239
Kompatibilität mit MySQL 5.7 .....	240
Aurora-MySQL-Updates: 25.03.2019 (Version 2.04.0) (veraltet) .....	240
Verbesserungen .....	241
Integration von MySQL-Fehlerbehebungen .....	242
Vergleich mit Aurora MySQL Version 1 .....	242
Kompatibilität mit MySQL 5.7 .....	242
Aurora-MySQL-Updates: 07.02.2019 (Version 2.03.4) (veraltet) .....	243
Verbesserungen .....	244
Vergleich mit Aurora MySQL Version 1 .....	244
Kompatibilität mit MySQL 5.7 .....	245
Aurora-MySQL-Updates: 18.01.2019 (Version 2.03.3) (veraltet) .....	245
Verbesserungen .....	246
Integration von MySQL-Fehlerbehebungen .....	247
Vergleich mit Aurora MySQL Version 1 .....	248
Kompatibilität mit MySQL 5.7 .....	248

Aurora-MySQL-Updates: 09.01.2019 (Version 2.03.3) (veraltet) .....	249
Verbesserungen .....	250
Vergleich mit Aurora MySQL Version 1 .....	250
Kompatibilität mit MySQL 5.7 .....	251
Aurora-MySQL-Updates: 24.10.2018 (Version 2.03.1) (veraltet) .....	251
Verbesserungen .....	252
Vergleich mit Aurora MySQL Version 1 .....	252
Kompatibilität mit MySQL 5.7 .....	253
Aurora-MySQL-Updates: 11.10.2018 (Version 2.03) (veraltet) .....	253
Verbesserungen .....	254
Integration von MySQL-Fehlerbehebungen (Community Edition): .....	255
Vergleich mit Aurora MySQL Version 1 .....	255
Kompatibilität mit MySQL 5.7 .....	255
Aurora-MySQL-Updates: 08.10.2018 (Version 2.02.5) (veraltet) .....	256
Verbesserungen .....	257
Vergleich mit Aurora MySQL Version 1 .....	257
Kompatibilität mit MySQL 5.7 .....	258
Aurora-MySQL-Updates: 21.09.2018 (Version 2.02.4) (veraltet) .....	258
Verbesserungen .....	259
Integration von MySQL-Fehlerbehebungen (Community Edition): .....	259
Vergleich mit Aurora MySQL Version 1 .....	260
Kompatibilität mit MySQL 5.7 .....	261
Aurora-MySQL-Updates: 23.08.2018 (Version 2.02.3) (veraltet) .....	261
Vergleich mit Aurora MySQL Version 1 .....	262
Kompatibilität mit MySQL 5.7 .....	263
CLI-Unterschiede zwischen Aurora MySQL 2.x und Aurora MySQL 1.x .....	264
Verbesserungen .....	264
Aurora-MySQL-Updates: 04.06.2018 (Version 2.02.2) (veraltet) .....	264
Verbesserungen .....	265
Vergleich mit Aurora MySQL 5.6 .....	265
Kompatibilität mit MySQL 5.7 .....	266
CLI-Unterschiede zwischen Aurora MySQL 2.x und Aurora MySQL 1.x .....	264
Verbesserungen .....	265
Aurora-MySQL-Updates: 03.05.2018 (Version 2.02) (veraltet) .....	267
Vergleich mit Aurora MySQL 5.6 .....	268
Kompatibilität mit MySQL 5.7 .....	269

CLI-Unterschiede zwischen Aurora MySQL 2.x und Aurora MySQL 1.x .....	264
Verbesserungen .....	270
Integration von MySQL-Fehlerbehebungen .....	270
Aurora-MySQL-Updates: 13.03.2018 (Version 2.01.1) (veraltet) .....	271
Vergleich mit Aurora MySQL 5.6 .....	271
Kompatibilität mit MySQL 5.7 .....	272
CLI-Unterschiede zwischen Aurora MySQL 2.x und Aurora MySQL 1.x .....	273
Verbesserungen .....	273
Aurora-MySQL-Updates: 06.02.2018 (Version 2.01) (veraltet) .....	273
Vergleich mit Aurora MySQL 5.6 .....	274
Kompatibilität mit MySQL 5.7 .....	274
CLI-Unterschiede zwischen Aurora MySQL 2.x und Aurora MySQL 1.x .....	264
Amazon Aurora MySQL Version 1 (veraltet) .....	276
Aurora-MySQL-Updates: 30.09.2021 (Version 1.23.4) (veraltet) .....	278
Verbesserungen .....	278
Aurora-MySQL-Updates: 28.06.2021 (Version 1.23.3) (veraltet) .....	279
Verbesserungen .....	279
Aurora-MySQL-Updates: 18.03.2021 (Version 1.23.2) (veraltet) .....	279
Verbesserungen .....	280
Integration von MySQL-Fehlerbehebungen (Community Edition): .....	282
Aurora-MySQL-Updates: 24.11.2020 (Version 1.23.1) (veraltet) .....	282
Verbesserungen .....	282
Aurora-MySQL-Updates: 02.09.2020 (Version 1.23.0) (veraltet) .....	283
Verbesserungen .....	284
Integration von MySQL-Fehlerbehebungen (Community Edition): .....	288
Aurora-MySQL-Updates: 03.06.2021 (Version 1.22.5) (veraltet) .....	289
Verbesserungen .....	290
Aurora-MySQL-Updates: 04.03.2021 (Version 1.22.4) (veraltet) .....	290
Verbesserungen .....	291
Aurora-MySQL-Updates: 09.11.2020 (Version 1.22.3) (veraltet) .....	292
Verbesserungen .....	293
Integration von MySQL-Fehlerbehebungen (Community Edition): .....	294
Aurora-MySQL-Updates: 05.03.2020 (Version 1.22.2) (veraltet) .....	294
Verbesserungen .....	295
Aurora-MySQL-Updates: 23.12.2019 (Version 1.22.1) (veraltet) .....	296
Verbesserungen .....	297

Aurora-MySQL-Updates: 25.11.2019 (Version 1.22.0) (veraltet) .....	297
Verbesserungen .....	298
Integration von MySQL-Fehlerbehebungen (Community Edition): .....	301
Aurora-MySQL-Updates: 25.11.2019 (Version 1.21.0) (veraltet) .....	302
Verbesserungen .....	303
Integration von MySQL-Fehlerbehebungen (Community Edition): .....	304
Aurora-MySQL-Updates: 05.03.2020 (Version 1.20.1) (veraltet) .....	304
Verbesserungen .....	305
Aurora-MySQL-Updates: 11.11.2019 (Version 1.20.0) (veraltet) .....	305
Verbesserungen .....	306
Integration von MySQL-Fehlerbehebungen (Community Edition): .....	307
Aurora-MySQL-Updates: 05.03.2020 (Version 1.19.6) (veraltet) .....	308
Verbesserungen .....	308
Aurora-MySQL-Updates: 19.09.2019 (Version 1.19.5) (veraltet) .....	309
Verbesserungen .....	310
Integration von MySQL-Fehlerbehebungen (Community Edition): .....	310
Aurora-MySQL-Updates: 05.06.2019 (Version 1.19.2) (veraltet) .....	311
Verbesserungen .....	312
Aurora-MySQL-Updates: 09.05.2019 (Version 1.19.1) (veraltet) .....	312
Verbesserungen .....	313
Aurora-MySQL-Updates: 07.02.2019 (Version 1.19.0) (veraltet) .....	313
Features .....	314
Verbesserungen .....	314
Integration von MySQL-Fehlerbehebungen (Community Edition): .....	316
Aurora-MySQL-Updates: 20.09.2018 (Version 1.18.0) (veraltet) .....	316
Features .....	316
Aurora-MySQL-Updates: 05.03.2020 (Version 1.17.9) (veraltet) .....	318
Verbesserungen .....	318
Aurora-MySQL-Updates: 17.01.2019 (Version 1.17.8) (veraltet) .....	319
Verbesserungen .....	319
Integration von MySQL-Fehlerbehebungen (Community Edition): .....	319
Aurora-MySQL-Updates: 08.10.2018 (Version 1.17.7) (veraltet) .....	320
Verbesserungen .....	320
Integration von MySQL-Fehlerbehebungen (Community Edition): .....	321
Aurora-MySQL-Updates: 06.09.2018 (Version 1.17.6) (veraltet) .....	321
Verbesserungen .....	322

Integration von MySQL-Fehlerbehebungen (Community Edition): .....	322
Aurora-MySQL-Updates: 14.08.2018 (Version 1.17.5) (veraltet) .....	322
Verbesserungen .....	323
Aurora-MySQL-Updates: 07.08.2018 (Version 1.17.4) (veraltet) .....	323
Verbesserungen .....	324
Aurora-MySQL-Updates: 05.06.2018 (Version 1.17.3) (veraltet) .....	325
Verbesserungen .....	325
Aurora-MySQL-Updates: 27.04.2018 (Version 1.17.2) (veraltet) .....	326
Verbesserungen .....	326
Aurora-MySQL-Updates: 23.03.2018 (Version 1.17.1) (veraltet) .....	327
Verbesserungen .....	327
Aurora-MySQL-Updates: 13.03.2018 (Version 1.17) (veraltet) .....	328
Zero-Downtime-Patching (Patchen ohne Ausfallzeiten) .....	328
Neue Features .....	328
Verbesserungen .....	329
Integration von MySQL-Fehlerbehebungen .....	329
Aurora-MySQL-Updates: 11.12.2019 (Version 1.16) (veraltet) .....	330
Zero-Downtime-Patching (Patchen ohne Ausfallzeiten) .....	330
Neue Features .....	330
Verbesserungen .....	331
Integration von MySQL-Fehlerbehebungen .....	331
Aurora-MySQL-Updates: 20.11.2017 (Version 1.15.1) (veraltet) .....	331
Zero-Downtime-Patching (Patchen ohne Ausfallzeiten) .....	332
Verbesserungen .....	332
Integration von MySQL-Fehlerbehebungen .....	333
Aurora-MySQL-Updates: 24.10.2017 (Version 1.15) (veraltet) .....	333
Zero-Downtime-Patching (Patchen ohne Ausfallzeiten) .....	333
Neue Features .....	333
Verbesserungen .....	334
Integration von MySQL-Fehlerbehebungen .....	333
Aurora-MySQL-Updates: 13.03.2018 (Version 1.14.4) (veraltet) .....	336
Zero-Downtime-Patching (Patchen ohne Ausfallzeiten) .....	336
Neue Features .....	336
Verbesserungen .....	336
Integration von MySQL-Fehlerbehebungen .....	337
Aurora-MySQL-Updates: 22.09.2017 (Version 1.14.1) (veraltet) .....	337

---

Verbesserungen .....	337
Aurora-MySQL-Updates: 07.08.2017 (Version 1.14) (veraltet) .....	338
Zero-Downtime-Patching (Patchen ohne Ausfallzeiten) .....	338
Verbesserungen .....	338
Integration von MySQL-Fehlerbehebungen .....	340
Aurora-MySQL-Updates: 15.05.2017 (Version 1.13) (veraltet) .....	340
Zero-Downtime-Patching (Patchen ohne Ausfallzeiten) .....	340
Neue Funktionen .....	341
Verbesserungen .....	341
Integration von MySQL-Fehlerbehebungen .....	342
Aurora-MySQL-Updates: 05.04.2017 (Version 1.12) (veraltet) .....	343
Neue Features .....	343
Verbesserungen .....	343
Integration von MySQL-Fehlerbehebungen .....	344
Aurora-MySQL-Updates: 23.02.2017 (Version 1.11) (veraltet) .....	345
Neue Funktionen .....	345
Verbesserungen .....	346
Integration von MySQL-Fehlerbehebungen .....	348
Aurora-MySQL-Updates: 12.01.2017 (Version 1.10.1) (veraltet) .....	348
Neue Funktionen .....	349
Verbesserungen .....	349
Aurora-MySQL-Updates: 14.12.2016 (Version 1.10) (veraltet) .....	349
Neue Funktionen .....	349
Verbesserungen .....	351
Integration von MySQL-Fehlerbehebungen .....	351
Aurora-MySQL-Updates vom 10.11.2016 (Versionen 1.9.0, 1.9.1) (veraltet) .....	352
Neue Funktionen .....	352
Verbesserungen .....	353
Aurora-MySQL-Updates: 26.10.2016 (Version 1.8.1) (veraltet) .....	353
Verbesserungen .....	353
Integration von MySQL-Fehlerbehebungen .....	354
Aurora-MySQL-Updates: 18.10.2016 (Version 1.8) (veraltet) .....	354
Neue Features .....	354
Verbesserungen .....	354
Integration von MySQL-Fehlerbehebungen .....	356
Aurora-MySQL-Updates: 20.09.2016 (Version 1.7.1) (veraltet) .....	356

Verbesserungen .....	356
Aurora-MySQL-Updates: 30.08.2016 (Version 1.7.0) (veraltet) .....	357
Neue Funktionen .....	357
Verbesserungen .....	357
Integration von MySQL-Fehlerbehebungen .....	358
Aurora-MySQL-Updates: 01.06.20216 (Version 1.6.5) (veraltet) .....	358
Neue Funktionen .....	358
Verbesserungen .....	359
Integration von MySQL-Fehlerbehebungen .....	359
Aurora-MySQL-Updates: 06.04.2016 (Version 1.6) (veraltet) .....	359
Neue Funktionen .....	359
Verbesserungen .....	361
Integration von MySQL-Fehlerbehebungen .....	361
Aurora-MySQL-Updates: 11.01.2016 (Version 1.5) (veraltet) .....	362
Verbesserungen .....	362
Integration von MySQL-Fehlerbehebungen .....	363
Aurora-MySQL-Updates: 03.12.2015 (Version 1.4) (veraltet) .....	363
Neue Funktionen .....	363
Verbesserungen .....	363
Integration von MySQL-Fehlerbehebungen .....	364
Aurora-MySQL-Updates vom 16.10.2015 (Versionen 1.2, 1.3) (veraltet) .....	365
Behobene Probleme .....	365
Verbesserungen .....	365
Integration von MySQL-Fehlerbehebungen .....	366
Aurora-MySQL-Updates: 24.08.2015 (Version 1.1) (veraltet) .....	369
MySQL-Fehlerbehebungen durch Aurora MySQL-Updates .....	370
MySQL-Fehlerbehebungen durch Aurora-MySQL-3.x-Updates .....	370
MySQL-Fehlerbehebungen durch Aurora MySQL 2.x-Updates .....	389
MySQL-Fehlerbehebungen durch Aurora MySQL 1.x-Updates .....	410
In Aurora MySQL behobene Sicherheitsschwachstellen .....	430
Dokumentverlauf .....	437
.....	cdlii



# Versionshinweise für Amazon Aurora MySQL-Compatible Edition

Versionen der MySQL-kompatiblen Amazon-Aurora-Edition werden regelmäßig aktualisiert. Updates werden auf Aurora-MySQL-DB-Cluster während des Wartungszeitraums angewendet. Der Zeitpunkt, zu dem Updates installiert werden, hängt von der AWS-Region, der Einstellung des Wartungsfensters für den DB-Cluster sowie von der Art des Updates ab.

Amazon Aurora MySQL-Versionen werden im Laufe mehrerer Tage für alle AWS-Regionen verfügbar gemacht. Einige Regionen zeigen möglicherweise vorübergehend eine Engine-Version an, die in einer anderen Region noch nicht verfügbar ist.

Updates werden auf alle Instances in einem DB-Cluster gleichzeitig angewendet. Ein Update erfordert einen Neustart der Datenbank auf allen Instances in einem DB-Cluster. Es kommt daher zu einem 20- bis 30-minütigen Nutzungsausfall, nach dem Sie wieder mit der Verwendung Ihres/r DB-Cluster(s) fortfahren können. Sie können die Einstellungen für den Wartungszeitraum in der Anzeige und ändern [AWS Management Console](#).

## Topics

- [Veröffentlichungskalender für Amazon Aurora MySQL](#)
- [Aktualisierungen der Datenbank-Engine für Amazon-Aurora-MySQL-Version 3](#)
- [Aktualisierungen der Datenbank-Engine für Amazon Aurora MySQL Version 2](#)
- [Aktualisierungen der Datenbank-Engine für Amazon Aurora MySQL Version 1 \(veraltet\)](#)
- [MySQL-Fehlerbehebungen durch Aurora MySQL-Datenbank-Engine-Updates](#)
- [In Aurora MySQL behobene Sicherheitsschwachstellen](#)

# Veröffentlichungskalender für Amazon Aurora MySQL

Die Veröffentlichungskalender auf dieser Seite können Ihnen bei der Planung Ihrer Haupt- und Nebenversions-Upgrades helfen. Weitere Informationen zu Amazon Aurora Aurora-Upgrades, Versionierung und Lebenszyklus finden Sie unter [Amazon Aurora Aurora-Versionen](#).

## Themen

- [Veröffentlichungskalender für Aurora MySQL-Hauptversionen](#)
- [Veröffentlichungskalender für Aurora MySQL-Nebenversionen](#)

## Veröffentlichungskalender für Aurora MySQL-Hauptversionen

Aurora MySQL-Hauptversionen sind im Rahmen des Standardsupports mindestens bis zum Ende der Nutzungsdauer der entsprechenden Community-Version verfügbar. Gegen eine Gebühr können Sie eine Hauptversion auch nach Ablauf des Standard-Supports von Aurora weiter ausführen. Weitere Informationen finden Sie unter [Amazon RDS Extended Support verwenden](#) und [Amazon Aurora Aurora-Preise](#).

Aurora MySQL unterstützt derzeit die folgenden Hauptversionen.

Community - Hauptversion	Aurora-Hauptversion	Datum des Lebensendes der Gemeinschaft	Ende des Standard-Supports für Aurora	RDS: Beginn des Extended Support, Jahr 1, Preisdatum	RDS: Beginn des Extended Support, Jahr 3, Preisdatum	RDS: Ende des Extended Support	Nebenversionen, die für den erweiterten Support in Frage kommen
MySQL 5.6 (veraltet)	Aurora MySQL Version 1 (veraltet)	5. Februar 2021	28. Februar 2023	N/A	–	–	N/A

Community - Hauptversion	Aurora-Hauptversion	Datum des Lebensendes der Gemeinschaft	Ende des Standard-Supports für Aurora	RDS: Beginn des Extended Support, Jahr 1, Preisdatum	RDS: Beginn des Extended Support, Jahr 3, Preisdatum	RDS: Ende des Extended Support	Nebenversionen, die für den erweiterten Support in Frage kommen
MySQL 5.7	Aurora-MySQL-Version 2	Oktober 2026	31. Oktober 2024	1. Dezember 2024	N/A	28. Februar 2027	Aurora MySQL 2.11 und 2.12
MySQL 8.0	Aurora-MySQL-Version 3	April 2026	30. April 2026	1. Mai 2027	N/A	31. Juli 2029	Muss noch entschieden werden

### Note

Amazon RDS Extended Support für Aurora MySQL Version 2 beginnt am 1. November 2024, Ihnen wird jedoch erst am 1. Dezember 2024 eine Gebühr berechnet. Zwischen dem 1. und 30. November 2024 sind alle Aurora MySQL Version 2-DB-Cluster durch Amazon RDS Extended Support abgedeckt.

## Veröffentlichungskalender für Aurora MySQL-Nebenversionen

Aurora MySQL unterstützt derzeit die folgenden Nebenversionen.

Aurora MySQL Version	Veröffentlichungsdatum von Aurora MySQL	Datum des Ablaufs des Standard-Supports für Aurora MySQL
3.07 (kompatibel mit Community MySQL 8.0.36)	4. Juni 2024	4. August 2025
3.06 (kompatibel mit Community MySQL 8.0.34)	7. März 2024	31. Mai 2025
3.05 (kompatibel mit Community MySQL 8.0.32)	25. Oktober 2023	31. Januar 2025
3.04 (kompatibel mit Community MySQL 8.0.28) (LTS)	31. Juli 2023	31. Oktober 2026
3.03 (kompatibel mit Community MySQL 8.0.26)	1. März 2023	15. August 2024
2.12 <sup>1</sup> (kompatibel mit Community MySQL 5.7.40 oder 5.7.44 <sup>2</sup> )	25. Juli 2023	31. Oktober 2024
2.11 <sup>1</sup> (kompatibel mit Community MySQL 5.7.12)	25. Oktober 2022	31. Oktober 2024
2.07 (kompatibel mit Community MySQL 5.7.12)	25. November 2019	30. April 2024

LTS — Aurora MySQL Long-Term Support (LTS) -Versionen. Weitere Informationen finden Sie unter [Aurora-MySQL-Long-Term-Support- \(LTS, Langzeit-Support\) Versionen](#).

<sup>1</sup> Diese Nebenversion ist weiterhin verfügbar, wenn die Hauptversion im Amazon RDS Extended Support enthalten ist. Weitere Informationen finden Sie unter [Amazon Aurora Aurora-Hauptversionen](#).

<sup>2</sup> Die Versionen von Aurora MySQL 2.12 bis 2.12.1 sind mit der MySQL-Version 5.7.40 kompatibel, und die Versionen 2.12.2 und höher sind mit der MySQL-Version 5.7.44 kompatibel.

# Aktualisierungen der Datenbank-Engine für Amazon-Aurora-MySQL-Version 3

Im Folgenden finden Sie Datenbank-Engine-Updates für Amazon-Aurora-MySQL-Version 3.

## Themen

- [Updates der Aurora MySQL-Datenbank-Engine 2024-06-04 \(Version 3.07.0, kompatibel mit MySQL 8.0.36\)](#)
- [Updates der Aurora MySQL-Datenbank-Engine 2024-03-07 \(Version 3.06.0, kompatibel mit MySQL 8.0.34\)](#)
- [Aurora MySQL-Datenbank-Engine-Updates 2024-01-31 \(Version 3.05.2, kompatibel mit MySQL 8.0.32\) Standard](#)
- [Updates der Aurora MySQL-Datenbank-Engine 2023-11-21 \(Version 3.05.1, kompatibel mit MySQL 8.0.32\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 30.10.2023 \(Version 3.05.0.1, kompatibel mit MySQL 8.0.32\) Beta](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 25.10.2023 \(Version 3.05.0, kompatibel mit MySQL 8.0.32\)](#)
- [Updates der Aurora MySQL-Datenbank-Engine 2024-03-15 \(Version 3.04.2, kompatibel mit MySQL 8.0.28\)](#)
- [Updates der Aurora MySQL-Datenbank-Engine 2023-11-13 \(Version 3.04.1, kompatibel mit MySQL 8.0.28\)](#)
- [Updates der Aurora MySQL-Datenbank-Engine 2023-07-31 \(Version 3.04.0, kompatibel mit MySQL 8.0.28\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 08.12.2023 \(Version 3.03.3, kompatibel mit MySQL 8.0.26\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 29.08.2023 \(Version 3.03.2, kompatibel mit MySQL 8.0.26\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 11.05.2023 \(Version 3.03.1, kompatibel mit MySQL 8.0.26\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 01.03.2023 \(Version 3.03.0, kompatibel mit MySQL 8.0.26\) Upgrades auf diese Version werden nicht unterstützt.](#)

- [Aurora-MySQL-Datenbank-Engine-Updates 17.04.2023 \(Version 3.02.3, kompatibel mit MySQL 8.0.23\)](#). Ende des Standard-Supports 15. Januar 2024.
- [Aurora-MySQL-Datenbank-Engine-Updates 18.11.2022 \(Version 3.02.2, kompatibel mit MySQL 8.0.23\)](#) Das Ende des Standard-Supports ist der 15. Januar 2024.
- [Aurora-MySQL-Datenbank-Engine-Updates 07.09.2022 \(Version 3.02.1, kompatibel mit MySQL 8.0.23\)](#). Die Standardunterstützung endet am 15. Januar 2024. Upgrades auf diese Version werden nicht unterstützt.
- [Aurora-MySQL-Datenbank-Engine-Updates 20.04.2022 \(Version 3.02.0, kompatibel mit MySQL 8.0.23\)](#). Ende des Standard-Supports 15. Januar 2024. Upgrades auf diese Version werden nicht unterstützt.
- [Aurora-MySQL-Datenbank-Engine-Updates 15.04.2022 \(Version 3.01.1, kompatibel mit MySQL 8.0.23\)](#). Ende des Standard-Supports 15. Januar 2024. Upgrades auf diese Version werden nicht unterstützt.
- [Aurora-MySQL-Datenbank-Engine-Updates 18.11.2021 \(Version 3.01.0, kompatibel mit MySQL 8.0.23\)](#). Ende des Standard-Supports 15. Januar 2024. Upgrades auf diese Version werden nicht unterstützt.

## Updates der Aurora MySQL-Datenbank-Engine 2024-06-04 (Version 3.07.0, kompatibel mit MySQL 8.0.36)

Version: 3.07.0

Aurora MySQL 3.07.0 ist allgemein verfügbar. Aurora MySQL 3.07-Versionen sind mit MySQL 8.0.36 kompatibel. Weitere Informationen zu Community-Änderungen, die von 8.0.23 zu 8.0.28 vorgenommen wurden, finden Sie in den [Versionshinweisen zu MySQL 8.0](#).

Details zu den neuen Features in Aurora MySQL Version 3 finden Sie unter [Aurora MySQL Version 3, kompatibel mit MySQL 8.0](#). Zu den Unterschieden zwischen Aurora MySQL Version 3 und Aurora MySQL Version 2 siehe [Vergleich von Aurora MySQL Version 2 und Aurora MySQL Version 3](#). Einen Vergleich von Aurora MySQL Version 3 und MySQL 8.0 Community Edition finden Sie unter [Vergleich von Aurora MySQL Version 3 und MySQL 8.0 Community Edition](#) im Amazon Aurora Aurora-Benutzerhandbuch.

Derzeit werden die Aurora MySQL-Versionen 2.07.9, 2.07.10, 2.11.\*, 2.12.\*, 3.03.\*, 3.04.\*, 3.05.\*, 3.05.\*, 3.06.\* und 3.07.\* unterstützt.

Sie können ein direktes Upgrade durchführen, einen Snapshot wiederherstellen oder ein verwaltetes Blue/Green-Upgrade mithilfe von [Amazon RDS Blue/Green Deployments](#) von jedem aktuell unterstützten Aurora MySQL Version 2-Cluster auf einen Aurora MySQL Version 3.07.0-Cluster starten.

Informationen zur Planung eines Upgrades auf Aurora MySQL Version 3 finden Sie unter [Planung eines Hauptversions-Upgrades für einen Aurora MySQL-Cluster](#). Allgemeine Informationen zu Aurora-MySQL-Upgrades finden Sie unter [Upgrade von Amazon-Aurora-MySQL-DB-Clustern](#) im Amazon-Aurora-Benutzerhandbuch.

Informationen zur Fehlerbehebung finden Sie unter [Problembehandlung für das direkte Upgrade von Aurora MySQL](#) im Amazon Aurora Benutzerhandbuch.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Sicherheitsprobleme und CVEs wurden behoben:

- Unterstützung für FIPS-validierte Kryptografie aktiviert, eine vollständig eigene Implementierung. AWS [Weitere Informationen finden Sie im Security Blog unter AWS-LC ist jetzt FIPS 140-3-zertifiziert.AWS](#)

Die folgenden CVE-Korrekturen sind in dieser Version enthalten:

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

Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem behoben, das dazu führen kann, dass eine Reader-DB-Instance neu gestartet wird, wenn eine Tabelle gelesen wird, die in der Writer-DB-Instance geändert oder gelöscht wurde.

- Es wurde ein Problem behoben, das dazu führen kann, dass eine Aurora MySQL-Writer-DB-Instance neu gestartet wird, wenn eine Schreibweiterleitungssitzung geschlossen wird, während eine weitergeleitete Abfrage ausgeführt wird.
- Es wurde ein Problem behoben, das dazu führte, dass eine DB-Instance neu gestartet wurde, wenn große GTID-Sätze auf einer Instance mit aktiviertem Binärprotokoll verarbeitet wurden.
- Es wurde ein Problem bei der Verarbeitung von INSERT Abfragen auf partitionierten InnoDB-Tabellen behoben, das zu einem allmählichen Rückgang des freien Speichers in der Instanz führen kann.
- Es wurde ein Problem behoben, das in seltenen Fällen dazu führen kann, dass die Reader-DB-Instances neu gestartet werden.
- Es wurde ein Problem behoben, das dazu führen kann, dass eine Datenbank-Instance neu gestartet wird, wenn die Anweisungen [SHOW STATUS](#) und [PURGE BINARY LOGS](#) gleichzeitig ausgeführt werden. PURGE BINARY LOGS ist eine verwaltete Anweisung, die ausgeführt wird, um die vom Benutzer konfigurierte Aufbewahrungsfrist für Binlogs einzuhalten.
- Es wurde ein Problem behoben, das dazu führen kann, dass der Server unerwartet geschlossen wird, nachdem Data Manipulation Language (DML) -Anweisungen für eine Tabelle ausgeführt wurden, deren nichtvirtuelle Spalten mit einer Oder-Anweisung neu angeordnet wurden. MODIFY COLUMN CHANGE COLUMN
- Es wurde ein Problem behoben, das beim Neustart einer Datenbankinstanz zu einem zusätzlichen Neustart führen kann.
- Es wurde ein Problem behoben, das dazu führen kann, dass eine Reader-DB-Instance, die Schreibweiterleitung verwendet, neu gestartet wird, wenn eine weitergeleitete [implizite Commit-Anweisung](#) auf einen Fehler stößt.
- Es wurde ein Problem behoben, das in seltenen Fällen dazu führen kann, dass eine Reader-Instance neu gestartet wird, wenn SELECT Abfragen für Tabellen mit einer Fremdschlüsseleinschränkung ausgeführt werden.
- Es wurde ein Problem behoben, bei dem es bei DB-Instances, die Aurora-Cluster-Volumes mit mehreren TB verwenden, während des Neustarts aufgrund von Fehlern bei der Validierung des InnoDB-Pufferpools zu längeren Ausfallzeiten kommen kann.
- Es wurde ein Problem behoben, das dazu führen kann, dass eine Datenbank neu gestartet wird, wenn eine kaskadierende UPDATE oder DELETE Fremdschlüsseleinschränkung für eine Tabelle definiert ist, an der eine virtuelle Spalte entweder als Spalte in der Fremdschlüsseleinschränkung oder als Mitglied der referenzierten Tabelle beteiligt ist.



- Es wurde ein Problem behoben, das die Datenbankwiederherstellung beim Start unterbrechen konnte, wenn der Neustart während der Ausführung umfangreicher Einfügevorgänge mit Spalten erfolgte. `AUTO_INCREMENT`
- Es wurde ein Problem behoben Aurora Serverless v2, das bei der Skalierung zu einem Neustart der Datenbank führen kann.

#### Allgemeine Verbesserungen:

- Reduzierte I/O-Auslastung und verbesserte Leistung für eine Teilmenge von Abfragen zum Scannen von Primärschlüsselbereichen, die parallel Abfragen verwenden.
- [Aurora MySQL Version 3.06.0](#) hat Unterstützung für die Amazon Bedrock-Integration hinzugefügt. In diesem Zusammenhang wurden neue reservierte Schlüsselwörter (`accept`, `aws_bedrock_invoke_model`, `aws_sagemaker_invoke_endpoint_content_type`, `undtimeout_ms`) hinzugefügt. In Aurora MySQL Version 3.07.0 wurden diese Schlüsselwörter in nicht reservierte Schlüsselwörter geändert, die als Bezeichner ohne Anführungszeichen zulässig sind. Weitere Informationen darüber, wie MySQL mit reservierten und nicht reservierten Schlüsselwörtern umgeht, finden Sie in der MySQL-Dokumentation unter [Schlüsselwörter und reservierte Wörter](#).
- Es wurde ein Problem behoben, bei dem nicht eindeutig eine Fehlermeldung an den Client zurückgegeben wurde, wenn der Amazon Bedrock-Service von einem Aurora MySQL-DB-Cluster in einem Land aufgerufen wurde, in AWS-Region dem Amazon Bedrock noch nicht verfügbar ist.
- Es wurde ein Problem behoben, das zu übermäßigem Speicherverbrauch führen kann, wenn BLOB Spalten mit der Aurora-Parallelabfrage abgefragt werden.
- Es wurde Unterstützung für die `connection_memory_chunk_size` Parameter `connection_memory_limit` und hinzugefügt, die auf Sitzungsebene so gesetzt werden müssen, dass sie sich genauso verhalten wie in der MySQL Community Edition. Der `connection_memory_limit` wird verwendet, um die maximale Speichermenge festzulegen, die von einer einzelnen Benutzerverbindung verwendet werden kann. Der `connection_memory_chunk_size` Parameter kann verwendet werden, um die Chunking-Größe für Aktualisierungen des [globalen Speichernutzungszählers](#) festzulegen.
- Es wurde ein Problem behoben, bei dem der Benutzer keine Abfrage unterbrechen oder Sitzungs-Timeouts für Abfragen festlegen konnte. `performance_schema`
- Es wurde ein Problem behoben, bei dem die Replikation von Binärprotokollen (Binlog), die für die Verwendung benutzerdefinierter SSL-Zertifikate ([mysql.rds\\_import\\_binlog\\_ssl\\_material](#)) konfiguriert war, fehlschlagen konnte, wenn die Replikationsinstanz einem Host-Austausch unterzogen wurde.

- Die `Aurora_fts_cache_memory_used` globale Statusvariable wurde hinzugefügt, um die Speichernutzung für das Volltextsuchsystem in allen Tabellen zu verfolgen. Weitere Informationen finden Sie unter [Globale Statusvariablen von Aurora MySQL](#) im Amazon Aurora Aurora-Benutzerhandbuch.
- Es wurde ein Problem behoben, bei dem es bei einem als Zero-ETL-Ziel konfigurierten Amazon Redshift Redshift-Cluster zu einer vorübergehenden Zunahme kommen konnte, [IntegrationLag](#) wenn ein Amazon Aurora MySQL-DB-Cluster als binäres Protokollreplikat konfiguriert wurde und die Enhanced Binlog- und Zero-ETL-Integration aktiviert war.
- Es wurde ein Problem im Zusammenhang mit der Verwaltung von Audit-Protokolldateien behoben, das dazu führen kann, dass auf Protokolldateien für den Download oder die Rotation nicht zugegriffen werden kann und das in einigen Fällen die CPU-Auslastung erhöht.
- Die `AUTO_INCREMENT` Schlüsselwiederherstellung wurde optimiert, um die Abschlusszeit für die Wiederherstellung von Snapshots, die Durchführung der point-in-time Wiederherstellung und das Klonen von DB-Clustern mit einer großen Anzahl von Tabellen in der Datenbank zu verkürzen.
- [Es wurde ein Problem behoben, bei dem das Ereignis `wait/io/redo\_log\_flush` nicht in den Übersichtstabellen für Warteeignisse des Leistungsschemas angezeigt wurde.](#)
- Es wurde ein Problem behoben, das nach einer Snapshot-Wiederherstellung, einem Backtrack oder dem Klonen von Datenbanken zu Fehlern beim Duplizieren von Schlüsseln für `AUTO_INCREMENT` Spalten mit absteigenden Indizes führen kann.
- Es wurde ein Problem behoben, das dazu führen kann, dass eine Writer-DB-Instance neu gestartet wird, wenn eine Reader-DB-Instance, die Schreibweiterleitung verwendet, eine Data Manipulation Language (DML) -Anweisung ausführt, die einen Zeitstempelwert enthält und der `time_zone` Datenbankparameter auf gesetzt ist. UTC
- Es wurde ein Problem behoben, bei dem eine `SELECT` Abfrage auf einer Aurora-Reader-Instance mit dem Fehler „Tabelle existiert nicht“ fehlschlagen konnte, wenn die Tabelle mindestens einen Volltextsuchindex (FTS) hat und eine `TRUNCATE` Anweisung auf der Aurora-Writer-DB-Instance ausgeführt wird.
- Es wurde ein Problem behoben, das in seltenen Fällen dazu führte, dass Zero Downtime Patching (ZDP) fehlschlug.
- Es wurde ein Problem behoben, das zu einem unvollständigen Ergebnissatz führen kann, wenn Abfragen ausgeführt werden, die den Hash-Join-Algorithmus mit parallel Abfrage beinhalten `LEFT JOIN` oder `RIGHT JOIN` Operationen verwenden.

Upgrades und Migrationen:

- Es wurde ein Problem behoben, das dazu führen kann, dass das Upgrade von Aurora MySQL Version 2 auf Aurora MySQL Version 3 fehlschlägt, wenn im Tabellenschema eine benutzerdefinierte FTS\_DOC\_ID Spalte vorhanden ist.
- Es wurde ein Problem behoben, das aufgrund eines Synchronisationsproblems bei der Verarbeitung von InnoDB-Tablespaces zu Upgrade-Fehlern von Aurora MySQL Version 2 auf Aurora MySQL Version 3 führen kann.
- Es wurde ein Problem behoben, das dazu führen kann, dass größere Versionsupgrades auf Aurora MySQL Version 3 fehlschlagen, weil verwaiste Einträge für bereits gelöschte Tablespaces in InnoDB-Systemtabellen in Aurora MySQL Version 2 vorhanden sind.
- Es wurde ein Problem behoben, bei dem der [SERVER\\_ID-Wert](#) nach einem Amazon RDS Blue/Green Deployment-Switchover nicht aktualisiert wurde. Dies führte zu Problemen, bei denen intelligente Treiber wie der [Amazon Web Services \(AWS\) JDBC-Treiber die DB-Cluster-Topologie](#) nach einem Blau/Grün-Switchover nicht erkennen konnten. Mit diesem Fix wird der SERVER\_ID Wert für Aurora-DB-Cluster, die im Rahmen einer RDS Blue/Green-Bereitstellung umbenannt wurden und auf Aurora MySQL Version 3.07 und höher ausgeführt werden, im Rahmen des Switchovers aktualisiert. In früheren Versionen können die DB-Instances in den blauen und grünen Clustern neu gestartet werden, um den Wert zu aktualisieren. SERVER\_ID

## Integration von MySQL-Fehlerbehebungen (Community Edition)

Diese Version enthält alle Community-Bugfixes bis einschließlich 8.0.36, zusätzlich zu den folgenden. Weitere Informationen finden Sie unter [MySQL-Fehlerbehebungen durch Aurora-MySQL-3.x-Datenbank-Engine-Updates](#).

- Es wurde ein Problem behoben, bei dem der Wert der Cache-Zeile falsch berechnet werden konnte, was zu einem Fehler beim Neustart der Datenbank auf Graviton-basierten Instances führte. (Community-Bugfix #35479763)
- Es wurde ein Problem behoben, bei dem einige Instanzen von Unterabfragen in gespeicherten Routinen nicht korrekt behandelt wurden. (Community-Bugfix #35377192)
- Es wurde ein Problem behoben, das aufgrund der Rotation von TLS-Zertifikaten im Hintergrund zu einer höheren CPU-Auslastung führen kann (Community Bug Fix #34284186).
- Es wurde ein Problem behoben, bei dem InnoDB das Hinzufügen von INSTANT Spalten zu Tabellen im MySQL-Systemschema in Aurora MySQL-Versionen unter 3.05 erlaubte, was dazu führen konnte, dass der Server nach dem Upgrade auf Aurora MySQL Version 3.05.0 unerwartet geschlossen wurde (Datenbankinstanz neu gestartet). (Community-Bugfix #35625510).

# Updates der Aurora MySQL-Datenbank-Engine 2024-03-07 (Version 3.06.0, kompatibel mit MySQL 8.0.34)

Version: 3.06.0

Aurora MySQL 3.06.0 ist allgemein verfügbar. Aurora MySQL 3.06-Versionen sind mit MySQL 8.0.34 kompatibel. Weitere Informationen zu Community-Änderungen, die von 8.0.23 zu 8.0.28 vorgenommen wurden, finden Sie in den [Versionshinweisen zu MySQL 8.0](#).

Details zu den neuen Features in Aurora MySQL Version 3 finden Sie unter [Aurora MySQL Version 3, kompatibel mit MySQL 8.0](#). Zu den Unterschieden zwischen Aurora MySQL Version 3 und Aurora MySQL Version 2 siehe [Vergleich von Aurora MySQL Version 2 und Aurora MySQL Version 3](#). Einen Vergleich von Aurora MySQL Version 3 und MySQL 8.0 Community Edition finden Sie unter [Vergleich von Aurora MySQL Version 3 und MySQL 8.0 Community Edition](#) im Amazon Aurora Aurora-Benutzerhandbuch.

Derzeit werden die Aurora MySQL-Versionen 2.07.9, 2.07.10, 2.11.\*, 2.12.\*, 3.03.\*, 3.04.\*, 3.05.\* und 3.06.\* unterstützt.

Sie können ein direktes Upgrade durchführen, einen Snapshot wiederherstellen oder ein verwaltetes Blue/Green-Upgrade mithilfe von [Amazon RDS Blue/Green Deployments](#) von jedem aktuell unterstützten Aurora MySQL Version 2-Cluster auf einen Aurora MySQL Version 3.06.0-Cluster starten.

Informationen zur Planung eines Upgrades auf Aurora MySQL Version 3 finden Sie unter [Planung eines Hauptversions-Upgrades für einen Aurora MySQL-DB-Cluster](#). Allgemeine Informationen zu Aurora-MySQL-Upgrades finden Sie unter [Upgrade von Amazon-Aurora-MySQL-DB-Clustern](#) im Amazon-Aurora-Benutzerhandbuch.

Informationen zur Fehlerbehebung finden Sie unter [Problembehandlung für das direkte Upgrade von Aurora MySQL](#) im Amazon Aurora Aurora-Benutzerhandbuch.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Neue Features

- Aurora MySQL Version 3.06.0 unterstützt die Amazon Bedrock-Integration und führt die neuen reservierten Schlüsselwörter `accept`,,

`aws_bedrock_invoke_modelaws_sagemaker_invoke_endpoint`, `content_type` und ein `timeout_ms`. Überprüfen Sie die Objektdefinitionen auf die Verwendung der neuen reservierten Schlüsselwörter, bevor Sie auf Version 3.06.0 aktualisieren. Um den Konflikt mit den neuen reservierten Schlüsselwörtern zu vermeiden, geben Sie die in den Objektdefinitionen verwendeten reservierten Schlüsselwörter in Anführungszeichen. Weitere Informationen zur Amazon Bedrock-Integration und zum Umgang mit reservierten Schlüsselwörtern finden Sie unter [Was ist Amazon Bedrock?](#) im Amazon Aurora Aurora-Benutzerhandbuch. Weitere Informationen finden Sie in der MySQL-Dokumentation unter [Schlüsselwörter und reservierte Wörter](#), [Die INFORMATION\\_SCHEMA KEYWORDS-Tabelle](#) und [Schemaobjektnamen](#).

- Verbesserte Leistung für binäre Protokollreplikate bei der Replikation von Transaktionen für große Tabellen mit mehr als einem sekundären Index. Diese Funktion führt einen Threadpool ein, um sekundäre Indexänderungen parallel auf ein Binlog-Replikat anzuwenden. Die Funktion wird durch den `aurora_binlog_replication_sec_index_parallel_workers` DB-Cluster-Parameter gesteuert, der die Gesamtzahl der parallel Threads steuert, die für die Anwendung der sekundären Indexänderungen verfügbar sind. Weitere Informationen finden Sie unter [Optimieren der binären Protokollreplikation](#) im Amazon-Aurora-Benutzerhandbuch.
- Es wurde eine neue gespeicherte Prozedur hinzugefügt `mysql.rds_set_read_only`, die es ermöglicht, den Wert der globalen Systemvariablen `read_only` auf Datenbank-Instances in Ihrem Aurora MySQL-Cluster zu ändern. Weitere Informationen finden Sie unter [Replizieren](#) im Amazon Aurora Aurora-Benutzerhandbuch.
- Es wurde eine neue gespeicherte Prozedur hinzugefügt `mysql.rds_set_binlog_source_ssl`, die es ermöglicht, die Verschlüsselung für ein binäres Protokollreplikat festzulegen, indem ein Wert für angegeben wird. `SOURCE_SSL` Weitere Informationen finden Sie unter [Replizieren](#) im Amazon Aurora Aurora-Benutzerhandbuch.
- [Amazon Aurora Machine Learning](#) ist eine optimierte Integration zwischen der Aurora MySQL-Datenbank und den Services für AWS maschinelles Lernen (ML). [Amazon Bedrock](#) wird jetzt unterstützt, sodass Sie Modelle für maschinelles Lernen in Amazon Bedrock direkt von Ihrem Aurora MySQL-DB-Cluster aus mithilfe von SQL aufrufen können. Weitere Informationen zur Verwendung von Amazon Bedrock mit Ihrem Aurora MySQL-DB-Cluster finden Sie [unter Verwenden von Amazon Aurora Machine Learning mit Aurora MySQL](#) im Amazon Aurora Aurora-Benutzerhandbuch.
- Aurora MySQL Version 3.06 bietet Unterstützung für [automatisiertes Undo-Tablespace-Kürzen](#). Diese Optimierung ermöglicht es Ihnen, ungenutzten Speicherplatz in Undo-Tablespaces zurückzugewinnen, nachdem die Undo-Logs gelöscht wurden.

## Verbesserungen

Sicherheitsprobleme und CVEs wurden behoben:

Die folgenden CVE-Fixes sind in dieser Version enthalten:

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

Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem behoben, bei dem eine Read Replica-DB-Instance nicht erfolgreich gestartet werden konnte, wenn in der Writer-DB-Instance eine hohe Arbeitslast herrscht.
- Es wurde ein Problem behoben, bei dem eine Aurora MySQL Writer-DB-Instance aufgrund eines Kommunikationsfehlers mit dem Aurora-Speicher ausfallen kann. Der Fehler ist auf einen Ausfall der Kommunikation zwischen der DB-Instance und dem zugrunde liegenden Speicher nach einem Software-Update der Aurora-Speicher-Instance zurückzuführen.
- Es wurde ein Problem bei der Verarbeitung von INSERT Abfragen auf partitionierten InnoDB-Tabellen behoben, das zu einem allmählichen Rückgang des freien Speichers in der Instanz führen kann.
- Es wurde ein Problem behoben, das dazu führen kann, dass eine Aurora MySQL-DB-Instance aufgrund einer Verringerung des freien Speichers neu gestartet oder ein Failover durchgeführt wird, wenn beim Ausführen von Abfragen ein Hash-Join verwendet wird.
- Es wurde ein Problem behoben, das dazu führen kann, dass eine Datenbank-Instance neu gestartet wird, wenn die Anweisungen [SHOW STATUS](#) und [PURGE BINARY LOGS gleichzeitig](#) ausgeführt werden. PURGE BINARY LOGS ist eine verwaltete Anweisung, die ausgeführt wird, um die vom Benutzer konfigurierte Aufbewahrungsfrist für Binlogs einzuhalten.
- Es wurde ein Problem behoben, das dazu führen kann, dass der Server unerwartet geschlossen wird, nachdem Data Manipulation Language (DML) -Anweisungen für eine Tabelle ausgeführt wurden, deren nichtvirtuelle Spalten mit einer Oder-Anweisung neu angeordnet wurden. MODIFY COLUMN CHANGE COLUMN

- Es wurde ein Problem behoben, das beim Neustart einer Datenbankinstanz zu einem zusätzlichen Neustart führen kann.
- Es wurde ein Problem behoben, das zu einem Neustart der Datenbank führen kann, wenn eine kaskadierende UPDATE oder DELETE Fremdschlüsseleinschränkung für eine Tabelle definiert ist, an der eine virtuelle Spalte entweder als Spalte in der Fremdschlüsseleinschränkung oder als Mitglied der referenzierten Tabelle beteiligt ist.
- In Aurora MySQL 2.10 haben wir Unterstützung für den Neustart eines Aurora-DB-Clusters mit Leseverfügbarkeit hinzugefügt. Mit dieser Funktion können Leser-DB-Instances online bleiben, während eine Writer-DB-Instance neu gestartet wird. Diese Funktion wird jetzt AWS-Regionen in sekundären globalen Aurora MySQL-Datenbanken unterstützt, sodass Sie während eines Neustarts der Writer-Instance auf dem primären Cluster weiterhin Leseanfragen bearbeiten können. Bisher wurden beim Neustart einer Writer-Instance auch alle Reader-Instances in einem sekundären Aurora MySQL-Cluster neu gestartet. Mit dieser Version bearbeiten sekundäre Cluster-Reader-Instances weiterhin Leseanfragen während eines Neustarts der Writer-Instance, wodurch die Leseverfügbarkeit im Cluster verbessert wird. Weitere Informationen finden Sie unter [Neustart eines Aurora-Clusters mit Leseverfügbarkeit](#).
- Es wurde ein Problem behoben, das die Datenbankwiederherstellung beim Start unterbrechen konnte, wenn der Neustart während der Ausführung umfangreicher Einfügevorgänge mit Spalten erfolgte. AUTO\_INCREMENT

#### Allgemeine Verbesserungen:

- Es wurde ein Problem behoben, das dazu führen kann, dass eine parallel Abfrage aufgrund vorübergehender Netzwerkprobleme beim Lesen von Daten aus dem Aurora-Cluster-Volume fehlschlägt.
- Es wurde ein Problem behoben, bei dem der Benutzer keine Abfrage unterbrechen oder Sitzungs-Timeouts für Abfragen festlegen konnte. `performance_schema`
- Es wurde ein Problem behoben, bei dem die Replikation von Binärprotokollen (Binlog), die für die Verwendung benutzerdefinierter SSL-Zertifikate ([mysql.rds\\_import\\_binlog\\_ssl\\_material](#)) konfiguriert war, fehlschlagen konnte, wenn die Replikationsinstanz einem Host-Austausch unterzogen wurde.
- Kleine DB-Instances mit weniger als oder gleich 4 GiB Speicher schließen jetzt die Verbindungen, die am meisten Speicher beanspruchen, wenn die DB-Instance unter Speicherdruck steht. Sie können den Pufferpool auch anpassen, um seine Größe zu verringern. Weitere Informationen finden Sie unter [Amazon Aurora MySQL out-of-memory Issues](#) im Amazon Aurora Aurora-Benutzerhandbuch.

- Die Standardantwort für `aurora_oom_response` wurde für alle DB-Instance-Klassen mit mehr als 4 GiB Speicher von leer auf `geändertprint`. Weitere Informationen finden Sie unter [Amazon Aurora MySQL out-of-memory Issues](#) im Amazon Aurora Aurora-Benutzerhandbuch.
- Es wurde ein Problem im Zusammenhang mit der Verwaltung von Audit-Protokolldateien behoben, das dazu führen kann, dass auf Protokolldateien für den Download oder die Rotation nicht zugegriffen werden kann und das in einigen Fällen die CPU-Auslastung erhöht.
- Die `AUTO_INCREMENT` Schlüsselwiederherstellung wurde optimiert, um die Abschlusszeit für die Wiederherstellung von Snapshots, die Durchführung der point-in-time Wiederherstellung und das Klonen von DB-Clustern mit einer großen Anzahl von Tabellen in der Datenbank zu verkürzen.
- [Es wurde ein Problem behoben, bei dem das Ereignis `wait/io/redo\_log\_flush` nicht in den Übersichtstabellen für Warteereignisse im Leistungsschema angezeigt wurde.](#)
- Die Metriken `aurora_lockmgr_memory_used` und `aurora_lockmgr_buffer_pool_memory_used` wurden hinzugefügt, um die Speichernutzung des Lock-Managers zu verfolgen. Weitere Informationen finden Sie unter [Globale Statusvariablen von Aurora MySQL](#) im Amazon Aurora Aurora-Benutzerhandbuch.
- Es wurde ein Problem behoben, bei dem es bei kleinen Read Replica-Instances nach einem Upgrade von Aurora MySQL-Versionen unter 2.11.\* zu einer erhöhten Replikationsverzögerung kommen kann.
- Es wurde ein Problem behoben, das nach einer Snapshot-Wiederherstellung, einem Backtrack oder dem Klonen von Datenbanken zu Fehlern beim Duplizieren von Schlüsseln für `AUTO_INCREMENT` Spalten mit absteigenden Indizes führen kann.
- Es wurde ein Problem behoben, bei dem eine `SELECT` Abfrage auf einer Aurora-Reader-Instance mit dem Fehler „Tabelle existiert nicht“ fehlschlagen konnte, wenn die Tabelle mindestens einen Volltextsuchindex (FTS) hat und eine `TRUNCATE` Anweisung auf der Aurora-Writer-DB-Instance ausgeführt wird.
- Es wurde ein Problem behoben, das zu einem unvollständigen Ergebnissatz führen kann, wenn Abfragen ausgeführt werden, die den Hash-Join-Algorithmus mit parallel Abfrage beinhalten `LEFT JOIN` oder `RIGHT JOIN` Operationen verwenden.

#### Upgrades und Migrationen:

- Es wurde ein Problem behoben, das dazu führen kann, dass größere Versionsupgrades fehlschlagen, wenn das Tabellenschema eine benutzerdefinierte `FTS_DOC_ID` Spalte enthält.



- Es wurde ein Problem behoben, das aufgrund eines Synchronisationsproblems bei der Verarbeitung von InnoDB-Tablespaces zu Upgrade-Fehlern von Aurora MySQL Version 2 auf Aurora MySQL Version 3 führen kann.
- Es wurde ein Problem behoben, das dazu führen kann, dass größere Versionsupgrades auf Aurora MySQL Version 3 fehlschlagen, weil verwaiste Einträge für bereits gelöschte Tablespaces in InnoDB-Systemtabellen in Aurora MySQL Version 2 vorhanden sind.

## Integration von MySQL-Fehlerbehebungen (Community Edition)

Diese Version enthält alle Community-Bugfixes bis einschließlich 8.0.34, zusätzlich zu den folgenden. Weitere Informationen finden Sie unter [MySQL-Fehlerbehebungen durch Aurora-MySQL-3.x-Datenbank-Engine-Updates](#).

- Es wurde ein Problem behoben, bei dem der Cache-Zeilenvwert falsch berechnet werden konnte, was zu einem Fehler beim Neustart der Datenbank auf Graviton-basierten Instances führte. (Community-Bugfix #35479763)
- Es wurde ein Problem behoben, bei dem einige Instanzen von Unterabfragen in gespeicherten Routinen nicht immer korrekt behandelt wurden. (Community-Bugfix #35377192)
- Es wurde ein Problem behoben, das aufgrund der Rotation von TLS-Zertifikaten im Hintergrund zu einer höheren CPU-Auslastung führen kann (Community Bug Fix #34284186).
- Es wurde ein Problem behoben, bei dem InnoDB das Hinzufügen von INSTANT Spalten zu Tabellen im MySQL-Systemschema in Aurora MySQL-Versionen vor 3.05 erlaubte, was dazu führen konnte, dass der Server nach dem Upgrade auf Aurora MySQL Version 3.05.0 unerwartet geschlossen wurde (Datenbankinstanz neu gestartet). (Community-Bugfix #35625510).

## Aurora MySQL-Datenbank-Engine-Updates 2024-01-31 (Version 3.05.2, kompatibel mit MySQL 8.0.32) Standard

Version: 3.05.2

Aurora MySQL 3.05.2 ist allgemein verfügbar. Aurora-MySQL-3.05-Versionen sind kompatibel mit MySQL 8.0.32. Weitere Informationen zu Community-Änderungen, die von 8.0.23 zu 8.0.28 vorgenommen wurden, finden Sie in den [Versionshinweisen zu MySQL 8.0](#).

Details zu den neuen Features in Aurora MySQL Version 3 finden Sie unter [Aurora MySQL Version 3, kompatibel mit MySQL 8.0](#). Zu den Unterschieden zwischen Aurora MySQL Version 3 und Aurora

MySQL Version 2 siehe [Vergleich von Aurora MySQL Version 2 und Aurora MySQL Version 3](#). Einen Vergleich von Aurora MySQL Version 3 und MySQL 8.0 Community Edition finden Sie unter [Vergleich von Aurora MySQL Version 3 und MySQL 8.0 Community Edition](#) im Amazon Aurora Aurora-Benutzerhandbuch.

Derzeit werden die Aurora-MySQL-Versionen 2.07.9, 2.07.10, 2.11.\*, 2.12.\*, 3.03.\*, 3.04.\* und 3.05.\* unterstützt.

Sie können ein direktes Upgrade durchführen, einen Snapshot wiederherstellen oder ein verwaltetes Blue/Green-Upgrade mithilfe von [Amazon RDS Blue/Green Deployments](#) von jedem aktuell unterstützten Aurora MySQL Version 2-Cluster auf einen Aurora MySQL Version 3.05.2-Cluster starten.

Informationen zur Planung eines Upgrades auf Aurora MySQL Version 3 finden Sie unter [Upgrade-Planung für Aurora MySQL Version 3](#). Allgemeine Informationen zu Aurora-MySQL-Upgrades finden Sie unter [Upgrade von Amazon-Aurora-MySQL-DB-Clustern](#) im Amazon-Aurora-Benutzerhandbuch.

Informationen zur Fehlerbehebung finden Sie unter [Behebung von Upgrade-Problemen mit Aurora MySQL Version 3](#) im Amazon Aurora Aurora-Benutzerhandbuch.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Sicherheitsprobleme und CVEs wurden behoben:

Die folgenden CVE-Fixes sind in dieser Version enthalten:

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

Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem behoben, bei dem die Verarbeitung von INSERT Abfragen in mit InnoDB partitionierten Tabellen zu einem allmählichen Rückgang des freien Speichers in der Instanz führen kann.

- Es wurde ein Problem behoben, das dazu führen kann, dass eine Datenbankinstanz neu gestartet wird, wenn die Anweisungen [SHOW STATUS](#) und [PURGE BINARY LOGS](#) gleichzeitig ausgeführt werden. PURGE BINARY LOGS ist eine verwaltete Anweisung, die ausgeführt wird, um die vom Benutzer konfigurierte Aufbewahrungsfrist für Binlogs einzuhalten.
- Es wurde ein Problem behoben, das dazu führen kann, dass der Server unerwartet geschlossen wird, nachdem Data Manipulation Language (DML) -Anweisungen für eine Tabelle ausgeführt wurden, deren nichtvirtuelle Spalten mit einer Oder-Anweisung neu angeordnet wurden. MODIFY COLUMN CHANGE COLUMN
- Es wurde ein Problem behoben, das beim Neustart einer Datenbankinstanz zu einem zusätzlichen Neustart führen kann.

#### Allgemeine Verbesserungen:

- Es wurde ein Problem behoben, bei dem der Benutzer keine Abfrage unterbrechen oder Sitzungs-Timeouts für performance\_schema Abfragen festlegen konnte.
- Es wurde ein Problem behoben, bei dem die Einrichtung der Replikation von Binärprotokollen (Binlog) mithilfe benutzerdefinierter SSL-Zertifikate ([mysql.rds\\_import\\_binlog\\_ssl\\_material](#)) [fehlschlagen konnte, wenn die Replikationsinstanz](#) einem Host-Austausch unterzogen wurde.
- Es wurde ein Problem im Zusammenhang mit der Verwaltung von Audit-Logdateien behoben, das dazu führen kann, dass auf Protokolldateien für den Download oder die Rotation nicht zugegriffen werden kann und das in einigen Fällen die CPU-Auslastung erhöht.

#### Upgrades und Migrationen:

- Es wurde ein Problem behoben, das dazu führen kann, dass das Upgrade von Aurora MySQL Version 2 auf Aurora MySQL Version 3 fehlschlägt, wenn im Tabellenschema eine benutzerdefinierte FTS\_DOC\_ID Spalte vorhanden ist.
- Es wurde ein Problem behoben, das aufgrund eines Synchronisationsproblems bei der Verarbeitung von InnoDB-Tablespaces zu Upgrade-Fehlern von Aurora MySQL Version 2 auf Aurora MySQL Version 3 führen kann.
- Es wurde ein Problem behoben, das dazu führen kann, dass größere Versionsupgrades auf Aurora MySQL Version 3 fehlschlagen, weil verwaiste Einträge für bereits gelöschte Tablespaces in InnoDB-Systemtabellen in Aurora MySQL Version 2 vorhanden sind.

## Integration von MySQL-Fehlerbehebungen (Community Edition)

Diese Version enthält alle Community-Bugfixes bis einschließlich 8.0.32. Weitere Informationen finden Sie unter [MySQL-Fehlerbehebungen durch Aurora-MySQL-3.x-Datenbank-Engine-Updates](#).

## Updates der Aurora MySQL-Datenbank-Engine 2023-11-21 (Version 3.05.1, kompatibel mit MySQL 8.0.32)

Version: 3.05.1

Aurora MySQL 3.05.1 ist allgemein verfügbar. Aurora-MySQL-3.05-Versionen sind kompatibel mit MySQL 8.0.32. Weitere Informationen finden Sie in den [Versionshinweisen zu MySQL 8.0](#).

Derzeit werden die Aurora-MySQL-Versionen 2.07.\*, 2.11.\*, 2.12.\*, 3.01.\*, 3.02.\*, 3.03.\*, 3.04.\* und 3.05.\* unterstützt.

Sie können ein vorhandenes Aurora MySQL 3.\*-Datenbankcluster auf Aurora MySQL 3.05.1 aktualisieren. Sie können auch einen Snapshot von jeder aktuell unterstützten Aurora MySQL-Version in Aurora MySQL 3.05.1 wiederherstellen.

Wenn Sie eine globale Aurora-MySQL-Datenbank zu Version 3.05.\* aktualisieren, müssen Sie Ihre primären und sekundären DB-Cluster auf exakt dieselbe Version aktualisieren, einschließlich des Patch-Levels. Weitere Informationen zum Upgrade der Nebenversion einer globalen Aurora-Datenbank finden Sie unter [Upgrades für Nebenversionen](#).

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Behobene Sicherheitsprobleme und CVEs sind unten aufgeführt:

Diese Version enthält alle Community-CVEs-Korrekturen bis einschließlich MySQL 8.0.32.

- [CVE-2023-38545](#)

## Integration von MySQL-Fehlerbehebungen (Community Edition)

Diese Version enthält alle Community-Fehlerbehebungen bis einschließlich 8.0.32, zusätzlich zu den unten aufgeführten. Weitere Informationen finden Sie unter [MySQL-Fehlerbehebungen durch Aurora-MySQL-3.x-Datenbank-Engine-Updates](#).

- Es wurde ein Problem in InnoDB behoben, das zwischen den Aurora MySQL-Versionen 3.01 bis Aurora MySQL-Versionen 3.04 und nach dem Upgrade von Aurora MySQL auf Version 3.05.0 bei DMLs für MySQL-Tabellen, denen in einem Systemschema eine INSTANT ADD-Spalte hinzugefügt wird, zu einem unerwarteten Schleifen des Servers führen kann. (Community-Fehlerbehebung #35625510)

## Aurora-MySQL-Datenbank-Engine-Updates 30.10.2023 (Version 3.05.0.1, kompatibel mit MySQL 8.0.32) Beta

Version: 3.05.0.1

Aurora MySQL 3.05.0.1 ist in den folgenden Regionen allgemein verfügbar: USA Ost (Nord-Virginia), USA Ost (Ohio), USA West (Nordkalifornien), USA West (Oregon), AWS GovCloud (US-Ost) und AWS GovCloud (US-West). Dies ist eine frühe Version, die nur für Sicherheitsupdates verfügbar ist. Diese Korrekturen werden mit der nächsten Patch-Version, 3.05.1, in größerem Umfang in allen Regionen verfügbar sein. Aurora-MySQL-3.05-Versionen sind kompatibel mit MySQL 8.0.32. Weitere Informationen zu den eingetretenen Community-Änderungen finden Sie in den [Versionshinweisen zu MySQL 8.0](#).

Derzeit werden die Aurora-MySQL-Versionen 2.07.\*, 2.11.\*, 2.12.\*, 3.01.\*, 3.02.\*, 3.03.\*, 3.04.\* und 3.05.\* unterstützt.

Sie können einen vorhandenen Aurora-MySQL-3.\*-Datenbank-Cluster zu Aurora MySQL 3.05.0.1 aktualisieren. Sie können auch einen Snapshot von jeder derzeit unterstützten Aurora-MySQL-Version zu Aurora MySQL 3.05.0.1 wiederherstellen.

Wenn Sie eine globale Aurora-MySQL-Datenbank zu Version 3.05.\* aktualisieren, müssen Sie Ihre primären und sekundären DB-Cluster auf exakt dieselbe Version aktualisieren, einschließlich des Patch-Levels. Weitere Informationen zum Upgrade der Nebenversion einer globalen Aurora-Datenbank finden Sie unter [Upgrades für Nebenversionen](#).

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Behobene Sicherheitsprobleme und CVEs sind unten aufgeführt:

Diese Version enthält alle Community-CVEs-Korrekturen bis einschließlich MySQL 8.0.32.

- [CVE-2023-38545](#)

## Aurora-MySQL-Datenbank-Engine-Updates 25.10.2023 (Version 3.05.0, kompatibel mit MySQL 8.0.32)

Version: 3.05.0

Aurora MySQL 3.05.0 ist allgemein verfügbar. Aurora-MySQL-3.05-Versionen sind kompatibel mit MySQL 8.0.32. Weitere Informationen zu Community-Änderungen, die von 8.0.23 zu 8.0.28 vorgenommen wurden, finden Sie in den [Versionshinweisen zu MySQL 8.0](#).

Details zu den neuen Features in Aurora MySQL Version 3 finden Sie unter [Aurora MySQL Version 3, kompatibel mit MySQL 8.0](#). Zu den Unterschieden zwischen Aurora MySQL Version 3 und Aurora MySQL Version 2 siehe [Vergleich von Aurora MySQL Version 2 und Aurora MySQL Version 3](#). Einen Vergleich zwischen Aurora MySQL Version 3 und MySQL 8.0 Community Edition finden Sie unter [Vergleich von Aurora MySQL Version 3 und MySQL 8.0 Community Edition](#).

Derzeit werden die Aurora-MySQL-Versionen 2.07.9, 2.07.10, 2.11.\*, 2.12.\*, 3.03.\*, 3.04.\* und 3.05.\* unterstützt.

Sie können ein direktes Upgrade durchführen, einen Snapshot wiederherstellen oder ein verwaltetes Blau/Grün-Upgrade mithilfe von [Amazon-RDS-Blau/Grün-Bereitstellungen](#) von jedem aktuell unterstützten Aurora-MySQL-Version-2-Cluster auf einen Aurora-MySQL-Version-3.05.0-Cluster starten.

Informationen zum Planen eines Upgrades auf Aurora MySQL Version 3 finden Sie unter [Upgrade-Planung für Aurora MySQL Version 3](#) im Amazon-Aurora-Benutzerhandbuch. Allgemeine Informationen zu Aurora-MySQL-Upgrades finden Sie unter [Upgrade von Amazon-Aurora-MySQL-DB-Clustern](#) im Amazon-Aurora-Benutzerhandbuch.

Informationen zur Fehlerbehebung finden Sie unter [Beheben von Upgrade-Problemen mit Aurora MySQL Version 3](#).

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Neue Funktionen:

- Es wurde Unterstützung für das Speichern von Daten aus einem Aurora-MySQL-Datenbank-Cluster in Textdateien in einem Amazon-S3-Bucket hinzugefügt, der mit einem KMS-Schlüssel (SSE-KMS) verschlüsselt ist. Weitere Informationen finden Sie unter [Speichern von Daten aus einem Amazon-Aurora-MySQL-DB-Cluster in Textdateien in einem Amazon-S3-Bucket](#).
- Es wurde eine neue globale Statusvariable `aurora_tmz_version` eingeführt, die die aktuelle Version der Zeitoneninformationen (TZ) bezeichnet, die von der Engine verwendet werden. Die Werte entsprechen der Version der IANA-Zeitonen-Datenbank und sind als „yyyySuffix“ formatiert, z. B. 2022a und 2023c. Weitere Informationen finden Sie unter [Aurora-MySQL-Global-Statusvariablen](#).

Behobene Sicherheitsprobleme und CVEs sind unten aufgeführt:

Korrekturen und andere Verbesserungen bei der Feinabstimmung der Handhabung in einer verwalteten Umgebung. Weitere CVE-Korrekturen unten:

- [CVE-2022-37434](#)

Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem behoben, bei dem es bei Aurora-MySQL-Datenbank-Instances, die parallele Abfragen verwenden, zu einem Datenbankneustart kommen konnte, wenn eine große Anzahl gleichzeitiger paralleler Abfragen ausgeführt wurde.
- Es wurde ein Problem mit Sperrkonflikten behoben, die durch einen Prüfprotokoll-Thread verursacht wurden und letztendlich zu einer hohen CPU-Auslastung und Timeouts für Client-Anwendungen führten.
- Es wurde ein Problem behoben, das dazu führen kann, dass der ausgeführte GTID-Satz auf einem Replikat-Cluster für Binärprotokolle (Binlog) mit aktiviertem erweitertem Binlog falsch

wiederhergestellt wird, wenn eine Binärprotokoll-Quelle auf `gtid_mode, ON` oder `ON_PERMISSIVE` gesetzt ist. Dieses Problem kann dazu führen, dass die Writer-Instance des Replikat-Clusters während der Wiederherstellung ein weiteres Mal neu gestartet wird, oder es kann zu falschen Ergebnissen führen, wenn der ausgeführte GTID-Satz abgefragt wird.

- Es wurde ein Speicherverwaltungsproblem behoben, das zu einem Neustart einer Aurora-MySQL-Datenbank-Instance oder einem Failover führen kann, da weniger freier Speicher verfügbar ist, wenn das erweiterte Binärprotokoll aktiviert ist.
- Es wurde ein Problem behoben, das dazu führen kann, dass eine Datenbank-Instance neu gestartet wird, wenn versucht wird, eine Datenbankseite zu lesen, die zu einer gelöschten Tabelle gehört.
- Es wurde ein Problem behoben, das dazu führen kann, dass die Reader-Instance neu gestartet wird, wenn die Writer-Instance das Datenbank-Volume auf ein Vielfaches von 160 GB erhöht.
- Es wurde ein Problem behoben, bei dem eine Aurora-MySQL-Datenbank-Instance mit aktiviertem erweitertem Binärprotokoll-Feature während des Startups der Datenbank-Instance hängen bleiben kann, während der Wiederherstellungsprozess für das Binärprotokoll ausgeführt wird.
- Es wurde ein Problem behoben, bei dem es bei einer Aurora-MySQL-Datenbank-Instance während des Instance-Startups zu mehreren Neustarts kommen kann, während große Rollback-Segmente initialisiert werden.
- Es wurde ein Problem behoben, das beim Patchen ohne Ausfallzeiten zu einem Neustart der Instance führt, was bewirkt, dass Datenbankverbindungen unerwartet geschlossen werden.
- Es wurde ein Problem behoben, das dazu führen kann, dass eine Datenbank-Instance aufgrund eines Deadlatches neu gestartet wird, wenn die Anweisungen [SHOW STATUS](#) und [PURGE BINARY LOGS](#) gleichzeitig ausgeführt werden. Bei den Bereinigungsbinärprotokollen handelt es sich um eine verwaltete Anweisung, die so ausgeführt wird, dass die vom Benutzer konfigurierte Aufbewahrungsfrist für Binärprotokolle eingehalten wird.
- Es wurde ein Problem behoben, das dazu führen kann, dass Datenbank-Cluster nicht verfügbar sind, wenn die Writer-Instance neu gestartet wird, während die Datenbank Auslöser für interne Systemtabellen erstellt oder löscht.
- Es wurde ein Problem behoben, das dazu führen kann, dass eine Datenbank-Instance aufgrund einer langen Semaphor-Wartezeit neu gestartet wurde, wenn das erweiterte Binärprotokoll-Feature auf einem Cluster mit einem Aurora-Replikat verwendet wurde.
- Es wurde ein Problem behoben, das dazu führen kann, dass eine Datenbank-Instance neu gestartet wird, während eine Abfrage ausgeführt wird, die auf eine Aggregatfunktion verweist.



- Es wurde ein Problem behoben, das unter seltenen Bedingungen dazu führen kann, dass die Datenbank-Instance neu gestartet wird, wenn Aurora Serverless v2 fälschlicherweise versucht, den Tabellen-Cache während der Skalierung zu aktualisieren.
- Es wurde ein Problem behoben, bei dem beim Materialisieren temporärer Zwischentabellen nicht unterstützte Index-Scan-Zugriffsmethoden für Common Table Expressions (CTE) berücksichtigt wurden, was zu unerwünschtem Verhalten wie Datenbankneustarts oder falschen Abfrageergebnissen führen kann. Wir haben dieses Problem behoben, indem wir die Verwendung solcher nicht unterstützter Indexscan-Zugriffsmethoden für Tabellen vermieden haben, die die TempTable Speicher-Engine verwenden.

#### Allgemeine Verbesserungen:

- Es wurde ein Problem behoben, das dazu führen kann, dass die Datenbank nicht verfügbar ist, wenn das erweiterte Binlog auf einem Aurora Serverless v2-Datenbank-Cluster aktiviert ist, der auf Aurora MySQL 3.04.0 ausgeführt wird.
- Wenn das erweiterte Binärprotokoll-Feature aktiviert ist, werden ungenutzte Speichermetadaten vor dem Schreiben in den Aurora-Speicher entfernt. Dadurch werden bestimmte Szenarien vermieden, in denen ein Neustart oder ein Failover der Datenbank aufgrund einer erhöhten Schreiblatenz durch die erhöhte Anzahl von über das Netzwerk übertragenen Byte auftreten kann.
- Mit der Hinzufügung der Tabellen `malloc_stats` und `malloc_stats_totals` im `performance_schema` wurden drei erweiterte Systemvariablen hinzugefügt, um das Verhalten von Jemalloc, einem internen Speicherzuweiser, zu steuern:
  - `aurora_jemalloc_background_thread`.
  - `aurora_jemalloc_dirty_decay_ms`.
  - `aurora_jemalloc_tcache_enabled`.
- Es wurde ein Problem behoben, bei dem Aurora-spezifische Leistungsschematabellen bei einem Upgrade oder einer Migration nicht erstellt wurden.
- Eine neue Systemvariable, `aurora_use_vector_instructions`, wurde hinzugefügt. Wenn dieser Parameter aktiviert ist, verwendet Aurora MySQL optimierte Vektorverarbeitungsanweisungen, um die Leistung bei E/A-intensiven Workloads zu verbessern. Diese Einstellung ist in Aurora MySQL 3.05 und höher standardmäßig auf ON gesetzt. Weitere Informationen finden Sie unter [Aurora-MySQL-Konfigurationsparameter](#).
- Es wurde ein Problem behoben, das dazu führen kann, dass die `NumBinaryLogFiles` Metriken falsche Ergebnisse anzeigen, wenn das erweiterte Binlog aktiviert ist. CloudWatch

- Das Anforderungs-Timeout für [Aurora-MySQL-Machine-Learning](#)-Operationen an Amazon Sagemaker wurde von 3 auf 30 Sekunden erhöht. Dies hilft bei der Lösung eines Problems, bei dem Kunden eine erhöhte Anzahl von Wiederholungsversuchen oder Fehlschlägen bei Anfragen von Aurora MySQL Machine Learning an Amazon Sagemaker feststellen können, wenn sie höhere Batch-Größen verwenden.
- Unterstützung für `malloc_stats`- und `malloc_stats_totals`-Tabellen in der `performance_schema`-Datenbank hinzugefügt.
- Das `FROM`-Schlüsselwort im `LOAD DATA FROM S3`-Befehl wurde aktualisiert und ist jetzt optional. Weitere Informationen finden Sie unter [Laden von Daten in einen Amazon-Aurora-MySQL-DB-Cluster aus Textdateien in einen Amazon-S3-Bucket](#).
- Es wurde Unterstützung für den Parameter `innodb_aurora_instant_alter_column_allowed` hinzugefügt, der steuert, ob der `INSTANT`-Algorithmus für `ALTER COLUMN`-Operationen verwendet werden kann. Weitere Informationen finden Sie unter [Parameter auf Cluster-Ebene](#).
- Es wurde ein Problem behoben, das verhindern kann, dass neue Client-Verbindungen zur Datenbank hergestellt werden, wenn die Schreibweiterleitung aktiviert ist.
- Es wurde ein Problem behoben, das dazu führen kann, dass die Änderung des `table_open_cache`-Datenbankparameters erst wirksam wird, wenn die Datenbank-Instance neu gestartet wird.
- Es wurde ein Problem behoben, das nach einer Snapshot-Wiederherstellung, einem Backtrack oder einem Datenbank-Klonvorgang zu Fehlern beim Duplizieren von Schlüsseln für `AUTO_INCREMENT`-Spalten mit absteigenden Indizes führen kann.
- Es wurde ein Problem mit Index-Scans behoben, bei dem möglicherweise ein falsches Ergebnis zurückgegeben wurde, wenn eine `SELECT`-Abfrage ausgeführt wurde, bei der die `GROUP BY`-Klausel verwendet und der `aurora_parallel_query`-Parameter zu `ON` geändert wurde.
- Es wurde ein Problem behoben, das dazu führen kann, dass bei der Ausführung von Abfragen für die `INFORMATION_SCHEMA INNODB_TABLESPACES`-Tabelle der verfügbare Speicher aufgebraucht wird.
- Es wurde ein Problem behoben, bei dem die Reader-Instance eine Tabelle nicht öffnen konnte, mit FEHLER 1146. Dieses Problem tritt auf, wenn bestimmte Typen von Online-DDL (Data Definition Language) ausgeführt werden, während der `INPLACE`-Algorithmus auf der Writer-Instance verwendet wird.

- Es wurde ein Problem behoben, um zu verhindern, dass eine Instance während der Aurora Serverless v2-Skalierung neu gestartet wird, wenn der interne Überwachungsprozess fälschlicherweise doppelte Skalierungsanforderungen sendet.
- Es wurde ein Problem behoben, das einen Datenbank-Neustart verursachen kann, wenn verbundene Binärprotokoll-Benutzer (Binlog) doppelte Binlog-Replikationsserver-IDs verwenden.
- Es wurde ein [In-Memory-Relay-Log-Cache](#) für von Aurora MySQL verwaltete binäre Protokollreplikate eingeführt. Diese Verbesserung kann dazu beitragen, den Durchsatz der binären Protokollreplikation um bis zu 40 % zu erhöhen. Diese Erweiterung wird automatisch aktiviert, wenn die binäre Protokollreplikation mit einem Thread oder die Multithread-Replikation mit aktivierter [GTID-Autopositionierung](#) verwendet wird.

#### Upgrades und Migrationen:

- Ein Upgrade von MySQL 5.7 zu MySQL 8.0 mit einer sehr großen Anzahl von Tabellen in einer einzigen Datenbank führte dazu, dass der Server übermäßig viel Speicher beanspruchte. Es wurde festgestellt, dass wir bei der Überprüfung, ob Tabellen aktualisiert werden konnten, alle Datenwörterbuch-Table-Objekte im Voraus abgerufen, jedes verarbeitet und dessen Namen abgerufen und dann die [Versionskompatibilität für die Liste überprüft](#) haben. Das vorherige Abrufen aller Objekte war in diesem Fall nicht erforderlich und trug erheblich zum Speicherverbrauch bei. Um dieses Problem zu beheben, rufen wir in solchen Fällen jetzt ein Table-Objekt nach dem anderen ab, führen alle erforderlichen Prüfungen durch, rufen seinen Namen ab und geben das Objekt frei, bevor wir mit dem nächsten fortfahren. (Fehler #34526001)
- Die Leistung von Hauptversions-Upgrades von Aurora MySQL Version 2 zu Version 3 wurde verbessert, indem Tablespace-Checks parallel unter Verwendung aller verfügbaren vCPUs auf der Datenbank-Instance ausgeführt werden.

## Integration von MySQL-Fehlerbehebungen (Community Edition)

Diese Version enthält alle Community-Fehlerbehebungen bis einschließlich 8.0.32, zusätzlich zu den unten aufgeführten. Weitere Informationen finden Sie unter [MySQL-Fehlerbehebungen durch Aurora-MySQL-3.x-Datenbank-Engine-Updates](#).

- Es wurde ein Problem behoben, das aufgrund der Rotation von TLS-Zertifikaten im Hintergrund zu einer höheren CPU-Auslastung führen kann. (Community-Fehlerbehebung #34284186)

# Updates der Aurora MySQL-Datenbank-Engine 2024-03-15 (Version 3.04.2, kompatibel mit MySQL 8.0.28)

Version: 3.04.2

Aurora MySQL 3.04.2 ist allgemein verfügbar. Aurora MySQL 3.04-Versionen sind mit MySQL 8.0.28 kompatibel. Weitere Informationen zu Community-Änderungen, die eingetreten sind, finden Sie in den [Versionshinweisen zu MySQL 8.0](#).

Details zu den neuen Features in Aurora MySQL Version 3 finden Sie unter [Aurora MySQL Version 3, kompatibel mit MySQL 8.0](#). Zu den Unterschieden zwischen Aurora MySQL Version 3 und Aurora MySQL Version 2 siehe [Vergleich von Aurora MySQL Version 2 und Aurora MySQL Version 3](#). Einen Vergleich zwischen Aurora MySQL Version 3 und MySQL 8.0 Community Edition finden Sie unter [Vergleich von Aurora MySQL Version 3 und MySQL 8.0 Community Edition](#).

## Note

Diese Version ist als Long-Term Support- (LTS, Langzeit-Support)-Version ausgewiesen. Weitere Informationen finden Sie unter [Aurora-MySQL-Long-Term-Support \(LTS, Langzeit-Support\)-Versionen](#) im Amazon-Aurora-Benutzerhandbuch.

Wir empfehlen, den `AutoMinorVersionUpgrade` Parameter für LTS-Versionen nicht auf `true` zu setzen (oder die automatische Aktualisierung kleinerer Versionen in der zu aktivieren AWS Management Console). Dies könnte dazu führen, dass Ihr DB-Cluster auf eine Nicht-LTS-Version wie 3.05.2 aktualisiert wird.

Derzeit werden die Aurora MySQL-Versionen 2.07.9, 2.7.10, 2.11.\*, 2.12.\*, 3.03.\*, 3.04.\*, 3.05.\* und 3.06.\* unterstützt.

Sie können ein direktes Upgrade durchführen, einen Snapshot wiederherstellen oder ein verwaltetes Blue/Green-Upgrade mit [Amazon RDS Blue/Green Deployments](#) von jedem derzeit verfügbaren Aurora MySQL Version 2-Cluster in einen Aurora MySQL-Cluster der Version 3.04.2 initiieren.

Informationen zum Planen eines Upgrades auf Aurora MySQL Version 3 finden Sie unter [Upgrade-Planung für Aurora MySQL Version 3](#) im Amazon-Aurora-Benutzerhandbuch. Allgemeine Informationen zu Aurora-MySQL-Upgrades finden Sie unter [Upgrade von Amazon-Aurora-MySQL-DB-Clustern](#) im Amazon-Aurora-Benutzerhandbuch.

Informationen zur Fehlerbehebung finden Sie unter [Beheben von Upgrade-Problemen mit Aurora MySQL Version 3](#).

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Sicherheitsprobleme und CVEs wurden behoben:

Die folgenden CVE-Fixes sind in dieser Version enthalten:

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

Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem behoben, bei dem eine Read Replica-DB-Instance nicht erfolgreich gestartet werden konnte, wenn in der Writer-DB-Instance eine hohe Arbeitslast herrscht.
- Es wurde ein Problem behoben, bei dem eine Aurora MySQL Writer-DB-Instance aufgrund eines Defekts in der Komponente, die mit dem Aurora-Speicher kommuniziert, ein Failover durchführen kann. Der Fehler ist auf einen Ausfall der Kommunikation zwischen der DB-Instance und dem zugrunde liegenden Speicher nach einem Softwareupdate zurückzuführen.
- Es wurde ein Problem behoben, das dazu führen kann, dass eine Datenbankinstanz neu gestartet wird, wenn die Anweisungen [SHOW STATUS](#) und [PURGE BINARY LOGS](#) gleichzeitig ausgeführt werden. `PURGE BINARY LOGS` ist eine verwaltete Anweisung, die ausgeführt wird, um die vom Benutzer konfigurierte Aufbewahrungsfrist für Binlogs einzuhalten.
- Es wurde ein Problem behoben, das beim Neustart einer Datenbankinstanz zu einem zusätzlichen Neustart führen kann.
- Es wurde ein Problem mit Sperrkonflikten behoben, die durch einen Prüfprotokoll-Thread verursacht wurden und zu einer hohen CPU-Auslastung und Timeouts für Client-Anwendungen führen können.

- Es wurde ein Problem behoben, bei dem es bei einer Aurora MySQL-Datenbank-Instance während des Instance-Starts zu mehreren Neustarts kommen kann, während große Rollback-Segmente initialisiert wurden.
- Es wurde ein Problem behoben, das dazu führen kann, dass eine DB-Instance neu gestartet wird, während eine Abfrage ausgeführt wird, die auf eine Aggregatfunktion verweist.

#### Allgemeine Verbesserungen:

- Es wurde ein Problem behoben, das dazu führen kann, dass eine parallel Abfrage aufgrund vorübergehender Netzwerkprobleme beim Lesen von Daten aus dem Aurora-DB-Cluster-Volume fehlschlägt
- Es wurde ein Problem behoben, bei dem der Benutzer keine Abfrage unterbrechen oder Sitzungs-Timeouts für Abfragen festlegen konnte. `performance_schema`
- Es wurde ein Problem behoben, bei dem die Replikation von Binärprotokollen (Binlog), die für die Verwendung benutzerdefinierter SSL-Zertifikate ([mysql.rds\\_import\\_binlog\\_ssl\\_material](#)) konfiguriert war, fehlschlagen konnte, wenn die Replikationsinstanz einem Host-Austausch unterzogen wurde.
- Es wurde ein Problem im Zusammenhang mit der Verwaltung von Audit-Protokolldateien behoben, das dazu führen kann, dass auf Protokolldateien für den Download oder die Rotation nicht zugegriffen werden kann und das in einigen Fällen die CPU-Auslastung erhöht.
- Die `AUTO_INCREMENT` Schlüsselwiederherstellung wurde optimiert, um die Abschlusszeit für die Wiederherstellung von Snapshots, die Durchführung der point-in-time Wiederherstellung und das Klonen von DB-Clustern mit einer großen Anzahl von Tabellen in der Datenbank zu verkürzen.
- Es wurde ein Problem behoben, bei dem SQL-Anweisungen, die sich auf einige `performance_schema` Tabellen beziehen, einen Fehler zurückgeben können, weil diese Tabellen nach der Migration von Community MySQL auf die Aurora MySQL-Versionen 3.04.0 und 3.04.1 fehlen.
- Es wurde ein Problem behoben, bei dem es bei kleinen Read Replica-Instances nach einem Upgrade von Aurora MySQL-Versionen unter 2.11.\* zu einer erhöhten Replikationsverzögerung kommen kann.
- Es wurde ein Problem behoben, das nach einer Snapshot-Wiederherstellung, einem Backtrack oder dem Klonen von Datenbanken zu Fehlern beim Duplizieren von Schlüsseln für `AUTO_INCREMENT` Spalten mit absteigenden Indizes führen kann.
- Es wurde ein Problem behoben, das dazu führen kann, dass Änderungen des `table_open_cache` Datenbankparameters erst wirksam werden, wenn die DB-Instance neu gestartet wird.

- Es wurde ein Problem behoben, bei dem die Reader-DB-Instance eine Tabelle nicht öffnen konnte (FEHLER 1146). Dieses Problem tritt auf, wenn bestimmte Arten von Online-DDL-Anweisungen (Data Definition Language) ausgeführt werden, während der INPLACE Algorithmus auf der Writer-DB-Instance verwendet wird.
- Es wurde ein Problem behoben, um zu verhindern, dass eine Instance während der Aurora Serverless v2-Skalierung neu gestartet wird, wenn der interne Überwachungsprozess fälschlicherweise doppelte Skalierungsanforderungen sendet.
- Es wurde ein Problem behoben, das zu einem Datenbankneustart führen kann, wenn verbundene Benutzer von Binärprotokollen (Binlog) doppelte Binlog-Replikationsserver-IDs verwenden.

#### Upgrades und Migrationen:

- Es wurde ein Problem behoben, das dazu führen kann, dass größere Versionsupgrades auf Aurora MySQL Version 3 fehlschlagen, weil verwaiste Einträge für bereits gelöschte Tablespaces in InnoDB-Systemtabellen in Aurora MySQL Version 2 vorhanden sind.

## Integration von MySQL-Fehlerbehebungen (Community Edition)

Diese Version enthält alle Community-Bugfixes bis einschließlich 8.0.28, zusätzlich zu den folgenden. Weitere Informationen finden Sie unter [MySQL-Fehlerbehebungen durch Aurora-MySQL-3.x-Datenbank-Engine-Updates](#).

- Es wurde ein Problem behoben, bei dem der Wert der Cache-Zeile falsch berechnet werden konnte, was zu einem Fehler beim Neustart der Datenbank auf Graviton-basierten Instances führte. (Community-Bugfix #35479763)
- Wiederholtes Ausführen einer gespeicherten Routine, bei der als Unterabfrage eine SELECT-Anweisung verwendet wurde, die mehrere AND, OR oder XOR -Bedingungen enthielt, führte zu übermäßigem Verbrauch und möglicherweise schließlich zur Erschöpfung des virtuellen Speichers. (Community-Bugfix #33852530)

## Updates der Aurora MySQL-Datenbank-Engine 2023-11-13 (Version 3.04.1, kompatibel mit MySQL 8.0.28)

Version: 3.04.1

Aurora MySQL 3.04.1 ist allgemein verfügbar. Aurora MySQL 3.04-Versionen sind mit MySQL 8.0.28 kompatibel. Weitere Informationen zu Community-Änderungen, die eingetreten sind, finden Sie in den [Versionshinweisen zu MySQL 8.0](#).

#### Note

Diese Version ist als Long-Term Support- (LTS, Langzeit-Support)-Version ausgewiesen. Weitere Informationen finden Sie unter [Aurora-MySQL-Long-Term-Support \(LTS, Langzeit-Support\)-Versionen](#) im Amazon-Aurora-Benutzerhandbuch.

Wir empfehlen, den `AutoMinorVersionUpgrade` Parameter für LTS-Versionen nicht auf `true` zu setzen (oder die automatische Aktualisierung kleinerer Versionen in der zu aktivieren AWS Management Console). Dies könnte dazu führen, dass Ihr DB-Cluster auf eine Nicht-LTS-Version wie 3.05.2 aktualisiert wird.

Details zu den neuen Features in Aurora MySQL Version 3 finden Sie unter [Aurora MySQL Version 3, kompatibel mit MySQL 8.0](#). Zu den Unterschieden zwischen Aurora MySQL Version 3 und Aurora MySQL Version 2 siehe [Vergleich von Aurora MySQL Version 2 und Aurora MySQL Version 3](#). Einen Vergleich zwischen Aurora MySQL Version 3 und MySQL 8.0 Community Edition finden Sie unter [Vergleich von Aurora MySQL Version 3 und MySQL 8.0 Community Edition](#).

Derzeit werden die Aurora MySQL-Versionen 2.07.9, 2.7.10, 2.11.\*, 2.12.\*, 3.01.\*, 3.02.\*, 3.03.\*, 3.04.\* und 3.05.\* unterstützt.

Sie können ein direktes Upgrade durchführen, einen Snapshot wiederherstellen oder ein verwaltetes Blue/Green-Upgrade mithilfe von [Amazon RDS Blue/Green Deployments](#) von jedem derzeit verfügbaren Aurora MySQL Version 2-Cluster auf einen Aurora MySQL Version 3.04.1-Cluster starten.

Informationen zum Planen eines Upgrades auf Aurora MySQL Version 3 finden Sie unter [Upgrade-Planung für Aurora MySQL Version 3](#) im Amazon-Aurora-Benutzerhandbuch. Allgemeine Informationen zu Aurora-MySQL-Upgrades finden Sie unter [Upgrade von Amazon-Aurora-MySQL-DB-Clustern](#) im Amazon-Aurora-Benutzerhandbuch.

Informationen zur Fehlerbehebung finden Sie unter [Beheben von Upgrade-Problemen mit Aurora MySQL Version 3](#).



Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem behoben, bei dem es bei Aurora-MySQL-Datenbank-Instances, die parallele Abfragen verwenden, zu einem Datenbankneustart kommen konnte, wenn eine große Anzahl gleichzeitiger paralleler Abfragen ausgeführt wurde.
- Es wurde ein Problem behoben, das dazu führen kann, dass der ausgeführte GTID-Satz auf einem Replikat-Cluster für Binärprotokolle (Binlog) mit aktiviertem erweitertem Binlog falsch wiederhergestellt wurde, wenn eine Binlog-Quelle auf oder gesetzt war. `gtid_mode ON ON_PERMISSIVE` Dieses Problem kann dazu führen, dass die Writer-Instance des Replikat-Clusters während der Wiederherstellung ein weiteres Mal neu gestartet wird, oder es kann zu falschen Ergebnissen führen, wenn der ausgeführte GTID-Satz abgefragt wird.
- Es wurde ein Speicherverwaltungsproblem behoben, das zu einem Neustart einer Aurora-MySQL-Datenbank-Instance oder einem Failover führen kann, da weniger freier Speicher verfügbar ist, wenn das erweiterte Binärprotokoll aktiviert ist.
- Es wurde ein Problem behoben, das dazu führen kann, dass die Reader-Instance neu gestartet wird, wenn die Writer-Instance das Datenbank-Volumen auf ein Vielfaches von 160 GB erhöht.
- Es wurde ein Problem behoben, bei dem eine Aurora-MySQL-Datenbank-Instance mit aktiviertem erweitertem Binärprotokoll-Feature während des Startups der Datenbank-Instance hängen bleiben kann, während der Wiederherstellungsprozess für das Binärprotokoll ausgeführt wird.
- Es wurde ein Problem behoben, das dazu führen kann, dass eine Datenbank-Instance aufgrund eines Deadlatches neu gestartet wird, wenn die Anweisungen [SHOW STATUS](#) und [PURGE BINARY LOGS](#) gleichzeitig ausgeführt werden. Bei den Bereinigungsbinärprotokollen handelt es sich um eine verwaltete Anweisung, die so ausgeführt wird, dass die vom Benutzer konfigurierte Aufbewahrungsfrist für Binärprotokolle eingehalten wird.
- Es wurde ein Problem behoben, das dazu führen kann, dass Datenbank-Cluster nicht verfügbar sind, wenn die Writer-Instance neu gestartet wird, während die Datenbank Auslöser für interne Systemtabellen erstellt oder löscht.
- Es wurde ein Problem behoben, das dazu führen kann, dass eine Datenbank-Instance aufgrund einer langen Semaphor-Wartezeit neu gestartet wurde, wenn das erweiterte Binärprotokoll-Feature auf einem Cluster mit einem Aurora-Replikat verwendet wurde.

## Allgemeine Verbesserungen:

- Es wurde ein Problem behoben, das dazu führen kann, dass die Datenbank nicht verfügbar ist, wenn das erweiterte Binlog auf einem Aurora Serverless v2-Datenbank-Cluster aktiviert ist, der auf Aurora MySQL 3.04.0 ausgeführt wird.
- Unbenutzte Speichermetadaten wurden vor dem Schreiben in Aurora Storage entfernt, wenn die erweiterte Binlog-Funktion aktiviert ist. Dadurch werden bestimmte Szenarien vermieden, in denen ein Neustart oder ein Failover der Datenbank aufgrund einer erhöhten Schreiblatenz durch die erhöhte Anzahl von über das Netzwerk übertragenen Byte auftreten kann.
- Es wurde ein Problem behoben, bei dem Aurora-spezifische Leistungsschematabellen bei einem Upgrade oder einer Migration nicht erstellt wurden.
- Es wurde ein Problem behoben, das dazu führen kann, dass bei NumBinaryLogFiles eingeschalteten CloudWatch Metriken falsche Ergebnisse angezeigt werden, wenn das erweiterte Binlog aktiviert ist.

## Upgrades und Migrationen:

- Ein Upgrade von MySQL 5.7 zu MySQL 8.0 mit einer sehr großen Anzahl von Tabellen in einer einzigen Datenbank führte dazu, dass der Server übermäßig viel Speicher beanspruchte. Es wurde festgestellt, dass wir bei der Überprüfung, ob Tabellen aktualisiert werden konnten, alle `Table` Datenwörterbuchobjekte im Voraus abgerufen, jedes einzelne verarbeitet und seinen Namen abgerufen und dann anhand der Liste ausgeführt wurden [CHECK TABLE ... FOR UPGRADE](#). Das vorherige Abrufen aller Objekte war in diesem Fall nicht erforderlich und trug erheblich zum Speicherverbrauch bei. Um dieses Problem zu beheben, rufen wir in solchen Fällen jetzt ein `Table`-Objekt nach dem anderen ab, führen alle erforderlichen Prüfungen durch, rufen seinen Namen ab und geben das Objekt frei, bevor wir mit dem nächsten fortfahren. (Fehler #34526001)

## Integration von MySQL-Fehlerbehebungen (Community Edition)

Diese Version enthält alle Community-Fehlerbehebungen bis einschließlich 8.0.28, zusätzlich zu den unten aufgeführten. Weitere Informationen finden Sie unter [MySQL-Fehlerbehebungen durch Aurora-MySQL-3.x-Datenbank-Engine-Updates](#).

- Es wurde ein Problem behoben, das aufgrund der Rotation von TLS-Zertifikaten im Hintergrund zu einer höheren CPU-Auslastung führen kann (Community-Fehlerbehebung #34284186)

# Updates der Aurora MySQL-Datenbank-Engine 2023-07-31 (Version 3.04.0, kompatibel mit MySQL 8.0.28)

Version: 3.04.0

Aurora MySQL 3.04.0 ist allgemein verfügbar. Aurora-MySQL-3.04-Versionen sind mit MySQL 8.0.28 kompatibel, Aurora-MySQL-3.03-Versionen sind kompatibel mit MySQL 8.0.26 und Aurora-MySQL-3.02-Versionen sind kompatibel mit MySQL 8.0.23. Weitere Informationen zu Community-Änderungen, die von 8.0.23 zu 8.0.28 vorgenommen wurden, finden Sie in den [Versionshinweisen zu MySQL 8.0](#).

## Note

Diese Version ist als Long-Term Support- (LTS, Langzeit-Support)-Version ausgewiesen. Weitere Informationen finden Sie unter [Aurora-MySQL-Long-Term-Support \(LTS, Langzeit-Support\)-Versionen](#) im Amazon-Aurora-Benutzerhandbuch.

Wir empfehlen, den `AutoMinorVersionUpgrade` Parameter für LTS-Versionen nicht auf `true` zu setzen (oder die automatische Aktualisierung kleinerer Versionen in zu aktivieren). AWS Management Console Dies könnte dazu führen, dass Ihr DB-Cluster auf eine Nicht-LTS-Version wie 3.05.2 aktualisiert wird.

Details zu den neuen Features in Aurora MySQL Version 3 finden Sie unter [Aurora MySQL Version 3, kompatibel mit MySQL 8.0](#). Zu den Unterschieden zwischen Aurora MySQL Version 3 und Aurora MySQL Version 2 siehe [Vergleich von Aurora MySQL Version 2 und Aurora MySQL Version 3](#). Einen Vergleich zwischen Aurora MySQL Version 3 und MySQL 8.0 Community Edition finden Sie unter [Vergleich von Aurora MySQL Version 3 und MySQL 8.0 Community Edition](#).

Derzeit werden die Aurora-MySQL-Versionen 2.07.9, 2.11.1, 2.11.2, 3.01.\*, 3.02.\*, 3.03.\* und 3.04.0 unterstützt.

Sie können ein direktes Upgrade durchführen, einen Snapshot wiederherstellen oder ein verwaltetes Blau/Grün-Upgrade mit [Amazon-RDS-Blau/Grün-Bereitstellungen](#) von jedem aktuell unterstützten Aurora-MySQL-Version-2-Cluster auf einen Aurora-MySQL-Version-3.04.0-Cluster starten.

Informationen zum Planen eines Upgrades auf Aurora MySQL Version 3 finden Sie unter [Upgrade-Planung für Aurora MySQL Version 3](#) im Amazon-Aurora-Benutzerhandbuch. Allgemeine

Informationen zu Aurora-MySQL-Upgrades finden Sie unter [Upgrade von Amazon-Aurora-MySQL-DB-Clustern](#) im Amazon-Aurora-Benutzerhandbuch.

Informationen zur Fehlerbehebung finden Sie unter [Beheben von Upgrade-Problemen mit Aurora MySQL Version 3](#).

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

#### Note

Das verbesserte Aurora-MySQL-Binärprotokoll (binlog) wird derzeit für die Datenbank-Instance von Aurora Serverless v2 in Aurora MySQL Version 3.04.0 nicht unterstützt. Die Aktivierung dieses Features kann dazu führen, dass die Datenbank nicht verfügbar ist. Wenn Sie die Verwendung eines erweiterten Binärprotokolls auf Aurora MySQL Version 3.04.0 benötigen, empfehlen wir, eine [Nicht-Serverless-Datenbank-Instance-Klasse](#) zu verwenden oder die minimale und maximale ACU der Serverless-v2-Datenbank-Instance auf denselben Wert zu setzen.

Weitere Informationen zur verbesserten Binärprotokollierung in Aurora MySQL finden Sie im [Aurora-Benutzerhandbuch](#).

## Verbesserungen

Neue Funktionen:

- Die Leistung von Abfragen, die InnoDB-Volltextindizes verwenden, um Phrasen im [natürlichen Sprachmodus](#) zu suchen, wurde verbessert. Weitere Informationen zu Volltextsuchen in MySQL finden Sie unter [Funktionen für die Volltextsuche](#).
- Amazon Aurora MySQL unterstützt die lokale (Cluster-interne) Schreibweiterleitung. Sie können jetzt Schreibvorgänge von einer Reader-DB-Instance an eine Writer-DB-Instance innerhalb eines DB-Clusters von Aurora MySQL weiterleiten. Weitere Informationen finden Sie unter [Verwendung der lokalen Schreibweiterleitung in einem DB-Cluster von Amazon Aurora MySQL](#).
- Es wurde die Möglichkeit hinzugefügt, den Wert des `aurora_replica_read_consistency`-Parameters für das Feature [Verwendung der Schreibweiterleitung in einer globalen Amazon-Aurora-Datenbank](#) in Sitzungen zu ändern, in denen `autocommit` deaktiviert ist. Weitere Informationen finden Sie unter [Konfigurationsparameter für die Schreibweiterleitung](#).

- Ab Aurora MySQL 3.04 können Sie für das [globale Datenbank-Schreibweiterleitungs-Feature](#) jetzt den Wert des `aurora_replica_read_consistency`-Parameters über die Parametergruppen Datenbank-Cluster und Datenbank-Instance festlegen. Vor Aurora MySQL Version 3.04 konnte der Wert dieses Parameters nur auf Sitzungsebene konfiguriert werden.

Sicherheitsprobleme und CVEs wurden behoben:

- Der SSL/TLS-Anbieter wurde von OpenSSL zu [AWS-LC](#) geändert. Dies bringt eine Reihe von Änderungen mit sich, darunter:
  - Datenbankverbindungen, die SSL verwenden, können jetzt durch Zero Downtime Restart und Zero Downtime Patching wiederhergestellt werden, wenn von Aurora MySQL Version 3.04.0 auf eine höhere Version aktualisiert wird.
  - Unterstützung für TLSv1.3, einschließlich Unterstützung für `TLS_AES_128_GCM_SHA256`, `TLS_AES_256_GCM_SHA384` und `TLS_CHACHA20_POLY1305_SHA256` SSL-Verschlüsselungen.
  - Entfernung der Unterstützung für weniger sichere DHE-RSA-\*-Verschlüsselungen.

Weitere Informationen finden Sie unter [Verwenden von TLS mit DB-Clustern von Aurora MySQL](#).

- `SHOW_ROUTINE` wurde das dynamische Recht auf die `rds_superuser_role` hinzugefügt, das den Zugriff auf Definitionen und Eigenschaften aller gespeicherten Routinen ermöglicht, wie z. B. gespeicherte Prozeduren und Funktionen. Weitere Informationen finden Sie unter [SHOW\\_ROUTINE](#).
- Es wurde ein Problem behoben, das dazu führen kann, dass im Prüfprotokoll Ereignisse während der Rotation der Prüfprotokolldatei übersehen werden.
- Die Unterstützung für das sichere und leistungsstarke Transport Layer Security (TLS) 1.3-Protokoll unter Beibehaltung der Kompatibilität mit der Version TLS 1.2 wurde aktiviert.
- Die TLS-Versionen TLSv1 und TLSv1.1 wurden in Community MySQL 8.0.26 und entsprechend in Aurora MySQL 3.03 eingestellt. Diese Protokolle wurden jetzt in Community MySQL 8.0.28 und entsprechend in Aurora MySQL 3.04 entfernt. Standardmäßig werden alle sicheren Clients, die nicht über TLS 1.2 oder höher kommunizieren können, abgelehnt. Weitere Informationen zur Verbindung mit Ihren Datenbank-Instances mithilfe von TLS finden Sie unter [Sicherheit mit Amazon Aurora MySQL](#).

Die folgenden CVE-Fixes sind in dieser Version enthalten:

- [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](#)

#### Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem behoben, das bei der Wiederherstellung langer Transaktionen zu Datenbankneustarts führen kann.
- Es wurde ein Problem bei der Verschlüsselung von Datenbankaktivitätsströmen behoben, das zu Datenbankneustarts führen kann.
- Es wurde ein Problem mit der Speicherverwaltung behoben, das aufgrund von Fehlern aufgrund von Speichermangel beim Startup des InnoDB-Pufferpools während des Startups oder während der Skalierung in Aurora Serverless v2 aufgetreten ist. Dieses Problem hat möglicherweise zu Neustarts von Datenbankinstanzen oder zu Leistungseinbußen, einschließlich Durchsatzreduzierung oder erhöhter Latenz, geführt.
- Es wurde ein Problem behoben, das dazu führen kann, dass eine Aurora MySQL-Reader-Instance neu gestartet wird, während eine Abfrage ausgeführt wird, die einen parallel Aurora MySQL-Abfrageausführungsplan verwendet.
- Es wurde ein Problem behoben, das in bestimmten Situationen dazu führen konnte, dass Aurora-Reader-Instances während einer Reichweitenschätzung neu gestartet wurden.
- Es wurde ein Problem behoben, das die Datenbankwiederherstellung beim Start unterbrechen konnte, wenn der Neustart während der Ausführung umfangreicher Einfügevorgänge mit automatisch inkrementierten Spalten erfolgte.
- Es wurde ein Problem mit Aurora Advanced Auditing behoben, das zu einer übermäßigen Protokollierung von Informationsmeldungen im Aurora-MySQL-Fehlerprotokoll führt, wenn die

Servervariable `server_audit_events` auf ALL oder QUERY gesetzt ist. Dieses Problem kann zu einem Neustart der Datenbankinstanz führen.

- Es wurde ein Problem behoben, das dazu führen kann, dass die Datenbank während des Rollbacks einer INSERT Anweisung neu gestartet wird, wenn die parallel Abfrage aktiviert ist.
- Es wurde ein Problem behoben, das dazu führen kann, dass die Datenbankinstanz neu gestartet wird, wenn das EXPLAIN ANALYZE Profilerstellungstool für eine Abfrage ausgeführt wird, die die Ausgabe `all select tables were optimized away` in der EXTRA Informationsspalte zurückgegeben hat. Weitere Informationen finden Sie unter [EXPLAIN-Ausgabeformat](#).
- Es wurde ein Problem behoben, das dazu führen kann, dass eine Reader-Instance der sekundären Aurora-Datenbank, die globale Schreibweiterleitung verwendet, neu gestartet wird, wenn eine weitergeleitete [implizite Commit-Anweisung](#) auf einen Fehler stößt.
- Es wurde ein Problem behoben, das dazu führen kann, dass die Writer-Instance in einer primären Aurora-Datenbankregion neu gestartet wird, wenn eine SELECT FOR UPDATE Abfrage mithilfe der globalen Schreibweiterleitung von einer sekundären Aurora-Datenbankregion ausgeführt wird.

#### Allgemeine Verbesserungen:

- Es wurde eine neue gespeicherte Prozedur `mysql.rds_gtid_purged` hinzugefügt, damit Kunden die GTID\_PURGED-Systemvariable festlegen können. Weitere Informationen finden Sie unter [mysql.rds\\_gtid\\_purged](#).
- Es wurden zwei neue gespeicherte Prozeduren `mysql.rds_start_replication_until` und `mysql.rds_start_replication_until_gtid` hinzugefügt, die es Kunden ermöglichen, einen Speicherort zu konfigurieren, um die Replikation von Binärprotokollen zu beenden. Weitere Informationen zur Konfiguration eines Stopp-Speicherorts für die binäre Protokollreplikation in Aurora MySQL finden Sie unter [mysql.rds\\_start\\_replication\\_until](#).
- Es wurde ein Problem behoben, das die [gespeicherten Prozeduren der Aurora-MySQL-Replikationskontrolle](#) daran hinderte, die `sql_log_bin`-Variable zu ändern, wenn sie aus einer Sitzung mit deaktiviertem Autocommit-Modus aufgerufen wurden.
- Unterstützung für logische Replikation wurde für die folgenden DCL-Anweisungen (Data Control Language) hinzugefügt: GRANT/REVOKE und CREATE/DROP/ALTER/RENAME USER.
- Es wurde ein Problem behoben, das verhindert hat, dass InnoDB-Statistiken veralten, was manchmal einen suboptimalen Abfrageausführungsplan verursacht, der zu einer Verlängerung der Abfrageausführungszeit führen kann.
- Es wurden zwei neue Systemansichten hinzugefügt, `information_schema.aurora_global_db_instance_status` und

`information_schema.aurora_global_db_status`. Diese Ansichten können verwendet werden, um den Status und die Topologie der primären und sekundären Ressourcen in einem globalen Aurora-MySQL-Datenbank-Cluster anzuzeigen. Die Details der beiden Systemansichten finden Sie unter [Aurora-MySQL-spezifische information\\_schema-Tabellen](#).

- Es wurde ein Problem behoben, bei dem ein Benutzer mit einem Platzhalterzeichen im Datenbanknamen nicht auf die Datenbank zugreifen konnte, nachdem er die `SET ROLE`-Anweisung mit einem maskierten Platzhalterzeichen ausgeführt hat.
- Es wurde ein Problem behoben, bei dem Ereignisse, die während der Verarbeitung von Prüfprotokoll-Rotationen gemeldet wurden, möglicherweise nicht in das Prüfprotokoll geschrieben wurden.
- Es wurde ein Problem behoben, bei dem das Erstellen einer internen temporären Tabelle über eine TRIGGER-Ausführung dazu führen kann, dass eine Writer-Datenbank-Instance neu gestartet wird.
- Eine neue Systemvariable, `innodb_aurora_max_partitions_for_range`, wurde hinzugefügt. In einigen Fällen, in denen persistente Statistiken nicht verfügbar sind, können Sie diesen Parameter verwenden, um die Ausführungszeit von Schätzungen der Zeilenanzahl in partitionierten Tabellen zu verbessern. Weitere Informationen finden Sie in der Dokumentation [Aurora-MySQL-Konfigurationsparameter](#).
- Es wurde ein Problem behoben, das es Kunden fälschlicherweise ermöglichte, `ROW_FORMAT` beim Erstellen partitionierter Tabellen auf `COMPRESSED` zu setzen. Tabellen werden implizit in das `COMPACT`-Format mit einer Warnung konvertiert, die darauf hinweist, dass Aurora MySQL komprimierte Tabellen nicht unterstützt.
- Es wurde ein Problem behoben, das dazu führen kann, dass die Replikation von Binärprotokollen mit mehreren Threads beendet wird, wenn die `replica_parallel_type`-Variable auf `LOGICAL_CLOCK` und die `replica_preserve_commit_order`-Variable auf `ON` gesetzt wird. Dieses Problem kann auftreten, wenn eine Transaktion, die größer als 500 MB ist, auf der Quelle ausgeführt wird.
- Es wurde ein Problem behoben, bei dem das [globale Datenbank-Schreibweiterleitungs-Feature](#) aktiviert ist. Dies kann dazu führen, dass Änderungen an der `performance_schema`-Konfiguration in den Reader-Instances in den sekundären Regionen unbeabsichtigt an die Writer-Instance in der primären Region weitergeleitet werden.
- Es wurde ein Problem behoben, bei dem die Serverstatusvariable `innodb_buffer_pool_reads` möglicherweise nicht aktualisiert wird, nachdem eine Datenseite aus dem Aurora-Speicherdateisystem gelesen wurde.



- Die parallele Aurora-MySQL-Abfrage wird bei Auswahl der Cluster-Konfiguration Aurora I/O-Optimized nicht unterstützt. Weitere Informationen finden Sie unter [Einschränkungen der parallelen Abfrage für Amazon Aurora MySQL](#).
- Es wurde ein Problem behoben, das dazu führte, dass der Abfrageplanoptimierer bei aktivierter paralleler Abfrage für bestimmte SELECT-Abfragen, die von einem primären oder sekundären Index profitieren, einen ineffizienten Ausführungsplan auswählte.
- Die Zeitzonendefinitionen wurden auf die Version IANA 2023c aktualisiert.
- Es wurden Leistungsoptimierungen bei der Dateiverwaltung für Binärprotokoll-Replikate eingeführt, um Konflikte beim Schreiben in Relay-Protokolldateien zu reduzieren.
- Es wurde ein Problem behoben, bei dem die RPO\_LAG\_IN\_MILLISECONDS Spalte in der `information_schema.aurora_global_db_status` Tabelle und der AuroraGlobalDBRPOlag CloudWatch Metrik unabhängig von der Benutzerauslastung immer Null anzeigte.
- Ein neuer Parameter `aurora_tmptable_enable_per_table_limit` wurde eingeführt. Wenn dieser Parameter aktiviert ist, definiert die `tmp_table_size` Variable die maximale Größe der einzelnen internen temporären Tabelle im Arbeitsspeicher, die von der TempTable Speicher-Engine erstellt wurde. Weitere Details finden Sie im Abschnitt [Speicher-Engine für interne \(implizite\) temporäre Tabellen](#).
- Es wurde ein Problem behoben, bei dem eine zusätzliche Verbindung hergestellt wurde, wenn das [globale Datenbank-Schreibweiterleitungs-Feature](#) aktiviert ist. Das Problem tritt auf, wenn schreibgeschützte Transaktionen auf einer Reader-Instance fälschlicherweise ein implizites Commit an den Writer weiterleiten.
- Es wurde ein Problem behoben, bei dem die Felder `PROCESSLIST_USER` und `PROCESSLIST_HOST` in der `performance_schema.threads`-Tabelle auf dem Writer in der primären Region für Verbindungen, die das Feature zur [globalen Datenbank-Schreibweiterleitung](#) verwenden, nicht aufgefüllt wurden. Weitere Informationen zu dieser Tabelle und dem Leistungsschema finden Sie im MySQL-Referenzhandbuch, in [der Threads-Tabelle](#) und im Amazon-Aurora-Benutzerhandbuch unter [Überblick über das Leistungsschema](#).
- Es wurde ein Problem behoben, bei dem die CommitLatency-Cloudwatch-Metrik falsche Werte für Reader-Instances in sekundären Regionen anzeigt, wenn das [globale Datenbank-Schreibweiterleitungs-Feature](#) verwendet wird. Um die Latenz der weitergeleiteten DML-Anweisungen auf sekundären Datenbank-Clustern zu überwachen, wird empfohlen, die Metriken `ForwardingReplicaDMLLatency` und `ForwardingWriterDMLLatency` zu verwenden. Die Commit-Latenz kann auch anhand der `CommitLatency`-Metrik auf der Writer-

Instance der primären Region beobachtet werden. Weitere Informationen finden Sie im Aurora-Benutzerhandbuch unter [CloudWatch Amazon-Metriken für die Weiterleitung von E-Mails](#).

- Es wurde ein Problem behoben, bei dem die [gespeicherten Prozeduren der Aurora-MySQL-Replikationssteuerung](#), die zur Verwaltung und Konfiguration der binären Protokollreplikation verwendet wurden, fälschlicherweise Fehler meldeten, wenn die Multi-Thread-Binärprotokollreplikation konfiguriert wurde, indem die Variable [replica\\_parallel\\_workers](#) auf einen Wert größer als 0 gesetzt wurde.
- Es wurde ein Problem behoben, das zu einem hohen CPU-Verbrauch führen kann, wenn mehrere Sitzungen versuchen, auf eine Seite zuzugreifen, die nicht im Speicher vorhanden ist.

#### Upgrades und Migrationen:

- Informationen zur Durchführung eines kleineren Versions-Upgrades für eine globale Aurora-Datenbank von Aurora MySQL Version 3.01, 3.02 oder 3.03 zu Aurora MySQL Version 3.04 oder höher finden Sie unter [Aktualisieren von Aurora MySQL durch Ändern der Engine-Version](#).
- Es wurde ein Problem behoben, das aufgrund von Schemainkonsistenzfehlern, die für die Tabellen `mysql.general_log_backup`, `mysql.general_log`, `mysql.slow_log_backup` und `mysql.slow_log` gemeldet wurden, beim Upgrade von Aurora MySQL 2 auf Aurora MySQL 3 zu Fehlern bei der Upgrade-Vorabprüfung führen kann. Weitere Informationen zur Fehlerbehebung bei Upgrades finden Sie unter [Behebung von Upgrade-Problemen mit Aurora MySQL Version 3](#).
- Es wurde ein Problem behoben, das zu schwerwiegenden Fehlern beim Upgrade auf Hauptversionen von Aurora MySQL 3 führen kann, wenn eine Auslöserdefinition ein reserviertes Schlüsselwort enthält, das nicht in Anführungszeichen steht.

## Integration von MySQL-Fehlerbehebungen (Community Edition)

Diese Version enthält alle Community-Fehlerbehebungen bis einschließlich 8.0.28, zusätzlich zu den unten aufgeführten. Weitere Informationen finden Sie unter [MySQL-Fehlerbehebungen durch Aurora-MySQL-3.x-Datenbank-Engine-Updates](#).

- Es wurde ein Problem behoben, bei dem ein Pufferblock mit einer systemeigenen temporären Tabellenseite während des Seitendurchlaufs verschoben wurde, was zu einem Assert-Fehler führte (Fehler #33715694)
- InnoDB: Verhindern Sie, dass Online-DDL-Operationen auf den out-of-bounds Speicher zugreifen (Bug# 34750489, Bug# 108925)

- Es wurde ein Problem behoben, das bei der Verarbeitung komplexer SQL-Anweisungen, die aus mehreren verschachtelten Common Table Expressions (CTEs) bestehen, manchmal zu falschen Abfrageergebnissen führen kann (Fehler-Nr. 34572040, Fehler-Nr. 34634469, Fehler-Nr. 33856374)

## Aurora-MySQL-Datenbank-Engine-Updates 08.12.2023 (Version 3.03.3, kompatibel mit MySQL 8.0.26)

Version: 3.03.3

Aurora MySQL 3.03.3 ist allgemein verfügbar. Aurora-MySQL-3.03-Versionen sind kompatibel mit MySQL 8.0.32. Weitere Informationen zu Community-Änderungen, die von 8.0.23 zu 8.0.28 vorgenommen wurden, finden Sie in den [Versionshinweisen zu MySQL 8.0](#).

Details zu den neuen Features in Aurora MySQL Version 3 finden Sie unter [Aurora MySQL Version 3, kompatibel mit MySQL 8.0](#). Zu den Unterschieden zwischen Aurora MySQL Version 3 und Aurora MySQL Version 2 siehe [Vergleich von Aurora MySQL Version 2 und Aurora MySQL Version 3](#). Einen Vergleich zwischen Aurora MySQL Version 3 und MySQL 8.0 Community Edition finden Sie unter [Vergleich von Aurora MySQL Version 3 und MySQL 8.0 Community Edition](#).

Derzeit sind die Aurora-MySQL-Versionen 2.07.9, 2.07.10, 2.11.\*, 2.12.\*, 3.01.\*, 3.02.\*, 3.03.\*, 3.04.\* und 3.05.\* verfügbar.

Sie können ein direktes Upgrade durchführen, einen Snapshot wiederherstellen oder ein verwaltetes Blau/Grün-Upgrade mithilfe von [Amazon-RDS-Blau/Grün-Bereitstellungen](#) von jedem derzeit verfügbaren Aurora-MySQL-Version-2-Cluster auf einen Aurora-MySQL-Version-3.03.3-Cluster initiieren.

Informationen zum Planen eines Upgrades auf Aurora MySQL Version 3 finden Sie unter [Upgrade-Planung für Aurora MySQL Version 3](#) im Amazon-Aurora-Benutzerhandbuch. Allgemeine Informationen zu Aurora-MySQL-Upgrades finden Sie unter [Upgrade von Amazon-Aurora-MySQL-DB-Clustern](#) im Amazon-Aurora-Benutzerhandbuch.

Informationen zur Fehlerbehebung finden Sie unter [Beheben von Upgrade-Problemen mit Aurora MySQL Version 3](#).

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Behobene Sicherheitsprobleme und CVEs sind unten aufgeführt:

Korrekturen und andere Verbesserungen bei der Feinabstimmung der Handhabung in einer verwalteten Umgebung. Weitere CVE-Korrekturen unten:

- [CVE-2023-38545](#)

Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem behoben, bei dem es bei Aurora-MySQL-Datenbank-Instances, die parallele Abfragen verwenden, zu einem Datenbankneustart kommen konnte, wenn eine große Anzahl gleichzeitiger paralleler Abfragen ausgeführt wurde.
- Es wurde ein Problem behoben, das dazu führen kann, dass der ausgeführte GTID-Satz auf einem Replikat-Cluster für Binärprotokolle (Binlog) mit aktiviertem erweitertem Binlog falsch wiederhergestellt wird, wenn eine Binärprotokoll-Quelle auf `gtid_mode, ON` oder `ON_PERMISSIVE` gesetzt ist. Dieses Problem kann dazu führen, dass die Writer-Instance des Replikat-Clusters während der Wiederherstellung ein weiteres Mal neu gestartet wird, oder es kann zu falschen Ergebnissen führen, wenn der ausgeführte GTID-Satz abgefragt wird.
- Es wurde ein Speicherverwaltungsproblem behoben, das zu einem Neustart einer Aurora-MySQL-Datenbank-Instance oder einem Failover führen kann, da weniger freier Speicher verfügbar ist, wenn das erweiterte Binärprotokoll aktiviert ist.
- Es wurde ein Problem behoben, das dazu führen kann, dass die Reader-Instance neu gestartet wird, wenn die Writer-Instance das Datenbank-Volumen auf ein Vielfaches von 160 GB erhöht.
- Es wurde ein Problem behoben, bei dem eine Aurora-MySQL-Datenbank-Instance mit aktiviertem erweitertem Binärprotokoll-Feature während des Startups der Datenbank-Instance hängen bleiben kann, während der Wiederherstellungsprozess für das Binärprotokoll ausgeführt wird.
- Es wurde ein Problem behoben, das beim Patchen ohne Ausfallzeiten zu einem Neustart der Instance führt, was bewirkt, dass die Datenbankverbindungen unerwartet geschlossen werden.
- Es wurde ein Problem behoben, das dazu führen kann, dass eine Datenbank-Instance aufgrund eines Deadlatches neu gestartet wird, wenn die Anweisungen [SHOW STATUS](#) und [PURGE BINARY LOGS](#) gleichzeitig ausgeführt werden. Bei den Bereinigungsbinärprotokollen handelt es sich um eine verwaltete Anweisung, die so ausgeführt wird, dass die vom Benutzer konfigurierte Aufbewahrungsfrist für Binärprotokolle eingehalten wird.

- Es wurde ein Problem behoben, das dazu führen kann, dass eine Datenbank-Instance aufgrund einer langen Semaphor-Wartezeit neu gestartet wurde, wenn das erweiterte Binärprotokoll-Feature auf einem Cluster mit einem Aurora-Replikat verwendet wurde.

#### Allgemeine Verbesserungen:

- Wenn das erweiterte Binärprotokoll-Feature aktiviert ist, werden ungenutzte Speichermetadaten vor dem Schreiben in den Aurora-Speicher entfernt. Dadurch werden bestimmte Szenarien vermieden, in denen ein Neustart oder ein Failover der Datenbank aufgrund einer erhöhten Schreiblatenz durch die erhöhte Anzahl von über das Netzwerk übertragenen Byte auftreten kann.
- Es wurde ein Problem behoben, das dazu führen kann, dass die NumBinaryLogFiles Metriken falsche Ergebnisse anzeigen, wenn das erweiterte Binlog aktiviert ist. CloudWatch
- Es wurde ein Problem behoben, das dazu führen kann, dass die Änderung des table\_open\_cache-Datenbankparameters erst wirksam wird, wenn die Datenbank-Instance neu gestartet wird.
- Es wurde ein Problem behoben, das einen Datenbank-Neustart verursachen kann, wenn verbundene Binärprotokoll-Benutzer (Binlog) doppelte Binlog-Replikationsserver-IDs verwenden.

## Integration von MySQL-Fehlerbehebungen (Community Edition)

Diese Version enthält alle Community-Fehlerbehebungen bis einschließlich 8.0.26, zusätzlich zu den unten aufgeführten. Weitere Informationen finden Sie unter [MySQL-Fehlerbehebungen durch Aurora-MySQL-3.x-Datenbank-Engine-Updates](#).

- Es wurde ein Problem behoben, das aufgrund der Rotation von TLS-Zertifikaten im Hintergrund zu einer höheren CPU-Auslastung führen kann (Community-Fehlerbehebung #34284186)

## Aurora-MySQL-Datenbank-Engine-Updates 29.08.2023 (Version 3.03.2, kompatibel mit MySQL 8.0.26)

Version: 3.03.2

Aurora MySQL 3.03.2 ist allgemein verfügbar. Aurora-MySQL-3.04-Versionen sind mit MySQL 8.0.28 kompatibel, Aurora-MySQL-3.03-Versionen sind kompatibel mit MySQL 8.0.26 und Aurora-MySQL-3.02-Versionen sind kompatibel mit MySQL 8.0.23. Weitere Informationen zu Community-

Änderungen, die von 8.0.23 zu 8.0.28 vorgenommen wurden, finden Sie in den [Versionshinweisen zu MySQL 8.0](#).

Details zu den neuen Features in Aurora MySQL Version 3 finden Sie unter [Aurora MySQL Version 3, kompatibel mit MySQL 8.0](#). Zu den Unterschieden zwischen Aurora MySQL Version 3 und Aurora MySQL Version 2 siehe [Vergleich von Aurora MySQL Version 2 und Aurora MySQL Version 3](#). Einen Vergleich zwischen Aurora MySQL Version 3 und MySQL 8.0 Community Edition finden Sie unter [Vergleich von Aurora MySQL Version 3 und MySQL 8.0 Community Edition](#).

Derzeit sind die Aurora-MySQL-Versionen 2.07.9, 2.07.10, 2.11.\*, 3.01.\*, 3.02.\*, 3.03.\* und 3.04.\* verfügbar.

Sie können ein direktes Upgrade durchführen, einen Snapshot wiederherstellen oder ein verwaltetes Blau/Grün-Upgrade mithilfe von [Amazon-RDS-Blau/Grün-Bereitstellungen](#) von jedem derzeit verfügbaren Aurora-MySQL-Version-2-Cluster auf einen Aurora-MySQL-Version-3.03.2-Cluster initiieren.

Informationen zum Planen eines Upgrades auf Aurora MySQL Version 3 finden Sie unter [Upgrade-Planung für Aurora MySQL Version 3](#) im Amazon-Aurora-Benutzerhandbuch. Allgemeine Informationen zu Aurora-MySQL-Upgrades finden Sie unter [Upgrade von Amazon-Aurora-MySQL-DB-Clustern](#) im Amazon-Aurora-Benutzerhandbuch.

Informationen zur Fehlerbehebung finden Sie unter [Beheben von Upgrade-Problemen mit Aurora MySQL Version 3](#).

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Sicherheitsprobleme und CVEs wurden behoben:

- Es wurde ein Problem behoben, das dazu führen konnte, dass im Auditprotokoll Ereignisse während der Rotation der Audit-Protokolldatei übersehen wurden.

Die folgenden CVE-Korrekturen sind in dieser Version enthalten:

- [CVE-2023-21963](#)

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

#### Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem behoben, das bei der Wiederherstellung langer Transaktionen zu Datenbankneustarts führen kann.
- Es wurde ein Problem behoben, das dazu führen kann, dass der Datenbank-Cluster nicht mehr verfügbar ist, wenn die Writer-Instanz neu gestartet wird, während die Datenbank Trigger für interne Systemtabellen erstellt oder gelöscht hat.
- Es wurde ein Problem behoben, das dazu führen kann, dass eine Datenbankinstanz neu gestartet wird, während eine Abfrage ausgeführt wird, die auf eine Aggregatfunktion verweist.
- Es wurde ein Problem behoben, das dazu führen kann, dass die Datenbank während des Rollbacks einer INSERT Anweisung neu gestartet wird, wenn die parallel Abfrage aktiviert ist.
- Fast Insert ist nur für reguläre InnoDB-Tabellen in Aurora MySQL Version 3.03.2 und höher aktiviert. Diese Optimierung funktioniert nicht für temporäre InnoDB-Tabellen. Weitere Informationen zur Fast-Insert-Optimierung finden Sie unter [Amazon Aurora MySQL Performance Enhancements](#).

#### Allgemeine Verbesserungen:

- Es wurde ein Problem behoben, bei dem die Reader-Instance keine Tabelle öffnen konnte, mit ERROR 1146. Dieses Problem tritt auf, wenn bestimmte Typen von Online-DDL (Data Definition Language) ausgeführt werden, während der INPLACE-Algorithmus auf der Writer-Instance verwendet wird.
- Es wurden Leistungsoptimierungen bei der Dateiverwaltung für Binärprotokoll-Replikate eingeführt, um Konflikte beim Schreiben in Relay-Protokolldateien zu reduzieren.
- Es wurde ein Problem behoben, das bei aktivierter paralleler Abfrage dazu führte, dass der Abfrageplanoptimierer für bestimmte SELECT-Abfragen, die von einem primären oder sekundären Index profitieren, einen ineffizienten Ausführungsplan auswählte.
- Unterstützung für logische Replikation wurde für die folgenden DCL-Anweisungen (Data Control Language) hinzugefügt: GRANT/REVOKE und CREATE/DROP/ALTER/RENAME USER.

- Parallele Abfragen für Amazon Aurora MySQL werden nicht unterstützt, wenn Sie die Aurora-E/A-optimierte Cluster-Konfiguration wählen. Weitere Informationen finden Sie unter [Einschränkungen der parallelen Abfrage für Aurora MySQL](#).

#### Upgrades und Migrationen:

- Zur Durchführung eines Nebenversions-Upgrades für eine globale Aurora-Datenbank von Aurora MySQL Version 3.01 oder 3.02 zu Aurora MySQL Version 3.03 oder höher siehe [Aktualisieren von Aurora MySQL durch Modifizierung der Engine-Version](#).
- Es wurde ein Problem behoben, das zu schwerwiegenden Fehlern beim Upgrade auf Aurora MySQL Version 3 führen kann, wenn eine Auslöserdefinition ein reserviertes Schlüsselwort enthält, das nicht in Anführungszeichen eingeschlossen ist.

## Integration von MySQL-Fehlerbehebungen (Community Edition)

Diese Version enthält alle Community-Fehlerbehebungen bis einschließlich 8.0.26, zusätzlich zu den unten aufgeführten. Weitere Informationen finden Sie unter [MySQL-Fehlerbehebungen durch Aurora-MySQL-3.x-Datenbank-Engine-Updates](#).

- Es wurde ein Problem behoben, das bei der Verarbeitung komplexer SQL-Anweisungen, die aus mehreren verschachtelten Common Table Expressions (CTEs) bestehen, manchmal zu falschen Abfrageergebnissen führen kann. (Fehler #34572040, Fehler #34634469, Fehler #33856374)
- InnoDB: Eine Race-Bedingung zwischen Threads, die versuchten, Statistiken für dieselbe Tabelle zu deinitialisieren und zu initialisieren, führte zu einem Assert-Fehler. (Fehler #33135425)
- InnoDB: Verhindert, dass Online-DDL-Operationen auf den Speicher zugreifen out-of-bounds . (Fehler #34750489, Fehler #108925)

## Aurora-MySQL-Datenbank-Engine-Updates 11.05.2023 (Version 3.03.1, kompatibel mit MySQL 8.0.26)

Version: 3.03.1

Aurora MySQL 3.03.1 ist allgemein verfügbar. Aurora-MySQL-3.03-Versionen sind mit MySQL 8.0.26 kompatibel. Aurora-MySQL-3.02-Versionen sind mit MySQL 8.0.23 kompatibel. Weitere Informationen zu Community-Änderungen, die von 8.0.23 zu 8.0.26 vorgenommen wurden, finden Sie in den [Versionshinweisen zu MySQL 8.0](#).



Details zu den neuen Features in Aurora MySQL Version 3 finden Sie unter [Aurora MySQL Version 3, kompatibel mit MySQL 8.0](#). Zu den Unterschieden zwischen Aurora MySQL Version 3 und Aurora MySQL Version 2 siehe [Vergleich von Aurora MySQL Version 2 und Aurora MySQL Version 3](#). Einen Vergleich zwischen Aurora MySQL Version 3 und MySQL 8.0 Community Edition finden Sie unter [Vergleich von Aurora MySQL Version 3 und MySQL 8.0 Community Edition](#).

Derzeit werden die Aurora-MySQL-Versionen 2.07.9, 2.11.1, 2.11.2, 3.01.\*, 3.02.\* und 3.03.\* unterstützt.

Sie können einen Snapshot von jedem derzeit unterstützten Aurora-MySQL-Version-2-Cluster in Aurora MySQL 3.03.1 wiederherstellen.

Informationen zum Planen eines Upgrades auf Aurora MySQL Version 3 finden Sie unter [Upgrade-Planung für Aurora MySQL Version 3](#) im Amazon-Aurora-Benutzerhandbuch. Allgemeine Informationen zu Aurora-MySQL-Upgrades finden Sie unter [Upgrade von Amazon-Aurora-MySQL-DB-Clustern](#) im Amazon-Aurora-Benutzerhandbuch.

Informationen zur Fehlerbehebung finden Sie unter [Beheben von Upgrade-Problemen mit Aurora MySQL Version 3](#).

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Neue Funktionen:

- Die Aurora-optimierte E/A-Speicherkonfiguration ist ab Version 3.03.1 verfügbar. Weitere Informationen finden Sie unter [Speicherkonfigurationen für DB-Cluster von Amazon Aurora](#).
- Eine neue Systemvariable, `innodb_aurora_max_partitions_for_range`, wurde hinzugefügt. In einigen Fällen, in denen persistente Statistiken nicht verfügbar sind, können Sie diesen Parameter verwenden, um die Ausführungszeit von Schätzungen der Zeilenanzahl in partitionierten Tabellen zu verbessern. Weitere Informationen finden Sie in der Dokumentation [Aurora-MySQL-Konfigurationsparameter](#).

Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem behoben, das dazu führen kann, dass die Datenbank-Instance aufgrund eines falschen Zugriffs auf ungültigen Speicher neu gestartet wurde, wenn eine Verbindung unmittelbar nach dem Commit einer Transaktion geschlossen wurde.
- Es wurde ein Problem mit Aurora Advanced Auditing behoben, das zu einer übermäßigen Protokollierung von Informationsmeldungen im Aurora-MySQL-Fehlerprotokoll führte, wenn die Servervariable `server_audit_events` auf ALL oder QUERY gesetzt ist. Dieses Problem kann zu einem Neustart der Datenbank-Instance führen.
- Es wurde ein Problem behoben, das in bestimmten Situationen dazu führen kann, dass Aurora-Reader-Instances neu gestartet wurden, wenn versucht wurde, eine Seite zu lesen, auf die während einer Reichweitenschätzung nicht mehr zugegriffen werden kann.
- Es wurde ein Problem behoben, das dazu führen kann, dass eine Aurora-MySQL-Reader-Instance neu gestartet wird, während eine Abfrage ausgeführt wird, die einen parallelen Aurora-Abfrageausführungsplan verwendet.
- Es wurde ein Problem behoben, bei dem es bei Datenbank-Instances, die binäre Protokollreplikation verwenden, zu einer erhöhten CPU-Auslastung und zu Verbindungsausfällen kommen kann, wenn mehrere Benutzer für die binäre Protokollreplikation angeschlossen sind.
- Es wurde ein Problem behoben, bei dem beim Materialisieren temporärer Zwischentabellen nicht unterstützte Index-Scan-Zugriffsmethoden für Common Table Expressions (CTE) berücksichtigt wurden, was zu unerwünschtem Verhalten wie Datenbankneustarts oder falschen Abfrageergebnissen führen kann. Wir beheben dieses Problem, indem wir die Verwendung solcher nicht unterstützter Indexscan-Zugriffsmethoden für Tabellen vermeiden, die die TempTable Speicher-Engine verwenden.
- Das schnelle Einfügen ist in dieser Aurora MySQL-Version nicht aktiviert, da ein Problem zu Inkonsistenzen beim Ausführen von Abfragen wie `INSERT INTOSELECT`, und führen kann. `FROM` Weitere Informationen zur Fast-Insert-Optimierung finden Sie unter [Amazon Aurora MySQL Performance Enhancements](#).

#### Allgemeine Verbesserungen:

- Es wurde ein Problem behoben, das längere Ausführungszeiten für die Anweisung `SHOW BINARY LOGS` verursachen kann. Dies könnte zu einem Rückgang des Commit-Durchsatzes der Datenbank führen.
- Es wurde ein Problem behoben, das dazu führen kann, dass der parallele Export von Benutzertabellen fehlschlägt, denen mithilfe der Funktion `ADD COLUMN` Spalten hinzugefügt wurden.

- Es wurde ein Problem behoben, bei dem Ereignisse, die während der Verarbeitung von Prüfprotokoll-Rotationen gemeldet wurden, möglicherweise nicht in das Prüfprotokoll geschrieben wurden.
- Es wurde ein Problem behoben, das bei der Ausführung von Abfragen für die Tabelle `INFORMATION_SCHEMA INNODB_TABLESPACES` zu einer Erschöpfung des verfügbaren Speichers führen kann.
- Es wurde ein Problem behoben, das es Kunden fälschlicherweise ermöglichte, `ROW_FORMAT` beim Erstellen partitionierter Tabellen auf `COMPRESSED` zu setzen. Tabellen werden implizit in das `COMPACT`-Format konvertiert, wobei eine Warnung angezeigt wird, dass Aurora MySQL komprimierte Tabellen nicht unterstützt.

#### Upgrades und Migrationen:

- Zur Durchführung eines Nebenversions-Upgrades für eine globale Aurora-Datenbank von Aurora MySQL Version 3.01 oder 3.02 zu Aurora MySQL Version 3.03 oder höher siehe [Aktualisieren von Aurora MySQL durch Modifizierung der Engine-Version](#).
- Es wurde ein Problem behoben, das aufgrund von Schemainkonsistenzfehlern, die für die Tabellen `mysql.general_log_backup`, `mysql.general_log`, `mysql.slow_log_backup` und `mysql.slow_log` gemeldet wurden, beim Upgrade von Aurora MySQL 2 auf Aurora MySQL 3 zu Fehlern bei der Upgrade-Vorabprüfung führen kann. Weitere Informationen zur Fehlerbehebung bei Upgrades finden Sie unter [Behebung von Upgrade-Problemen mit Aurora MySQL Version 3](#).

## Integration von MySQL-Fehlerbehebungen (Community Edition)

Diese Version enthält alle Community-Fehlerbehebungen bis einschließlich 8.0.26, zusätzlich zu den unten aufgeführten. Weitere Informationen finden Sie unter [MySQL-Fehlerbehebungen durch Aurora-MySQL-3.x-Datenbank-Engine-Updates](#).

- Es wurde ein Problem behoben, bei dem ein Pufferblock mit einer systemeigenen temporären Tabellenseite während des Seitendurchlaufs verschoben wurde, was zu einem Assert-Fehler führte. (Fehler #33715694)

# Aurora-MySQL-Datenbank-Engine-Updates 01.03.2023 (Version 3.03.0, kompatibel mit MySQL 8.0.26) Upgrades auf diese Version werden nicht unterstützt.

Version: 3.03.0

Aurora MySQL 3.03.0 ist allgemein verfügbar. Aurora-MySQL-3.03-Versionen sind mit MySQL 8.0.26 kompatibel. Aurora-MySQL-3.02-Versionen sind mit MySQL 8.0.23 kompatibel. Weitere Informationen zu Community-Änderungen, die von 8.0.23 zu 8.0.26 vorgenommen wurden, finden Sie in den [Versionshinweisen zu MySQL 8.0](#).

Details zu den neuen Features in Aurora MySQL Version 3 finden Sie unter [Aurora MySQL Version 3, kompatibel mit MySQL 8.0](#). Zu den Unterschieden zwischen Aurora MySQL Version 3 und Aurora MySQL Version 2 siehe [Vergleich von Aurora MySQL Version 2 und Aurora MySQL Version 3](#). Einen Vergleich zwischen Aurora MySQL Version 3 und MySQL 8.0 Community Edition finden Sie unter [Vergleich von Aurora MySQL Version 3 und MySQL 8.0 Community Edition](#).

Derzeit werden die Aurora-MySQL-Versionen 2.07.\*, 2.11.\*, 3.01.\*, 3.02.\* und 3.03.\* unterstützt.

Sie können einen Snapshot von jedem derzeit unterstützten Aurora-MySQL-Version-2-Cluster in Aurora MySQL 3.03.0 wiederherstellen.

Informationen zum Planen eines Upgrades auf Aurora MySQL Version 3 finden Sie unter [Upgrade-Planung für Aurora MySQL Version 3](#) im Amazon-Aurora-Benutzerhandbuch. Allgemeine Informationen zu Aurora-MySQL-Upgrades finden Sie unter [Upgrade von Amazon-Aurora-MySQL-DB-Clustern](#) im Amazon-Aurora-Benutzerhandbuch.

Informationen zur Fehlerbehebung finden Sie unter [Beheben von Upgrade-Problemen mit Aurora MySQL Version 3](#).

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Behobene Sicherheitsprobleme und CVEs sind unten aufgeführt:

Korrekturen und andere Verbesserungen bei der Feinabstimmung der Handhabung in einer verwalteten Umgebung. Weitere CVE-Korrekturen unten:

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

#### Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem behoben, bei dem es bei größeren DB-Instance-Klassen zu Problemen beim Neustart kommen konnte, weil die Initialisierung des Pufferpools länger als erwartet dauerte.
- Es wurde ein Problem behoben, durch das die DB-Instance während des Datenbank-Wiederherstellungsprozesses neu gestartet wurde, wenn die binäre Protokollierung aktiviert war.
- Es wurde ein Problem behoben, das zu Verbindungsfehlern in Reader-Instances führen kann, wenn DCL-Anweisungen (Data Control Language) ausgeführt werden, z. B. GRANT und REVOKE, oder beim Herstellen neuer Verbindungen in der Writer-Instance.
- Es wurde ein Problem behoben, bei dem parallele Abfragen fälschlicherweise für DML-Operationen (Data Manipulation Language) verwendet wurden, wie z. B. die Anweisungen DELETE und UPDATE, die derzeit nicht unterstützt werden, und zu einem Neustart der DB-Instance führten. Weitere Informationen zu Operationen, die in parallelen Abfragen unterstützt werden, finden Sie unter [Einschränkungen für parallele Abfragen in Aurora MySQL](#).
- Es wurde ein Problem behoben, das in seltenen Fällen dazu führen kann, dass ein Aurora-Replikat neu gestartet wird, wenn große Aktualisierungsvorgänge oder Data Definition Language (DDL)-Workloads auf der Writer-Instance und Lesevorgänge für denselben Tabellensatz auf dem Aurora-Replikat gleichzeitig ausgeführt werden.
- Es wurde ein Problem mit dem Herunterskalieren der Aurora-Serverless-v2-Reader-Instance behoben, das dazu führen kann, dass diese Reader-Instance neu gestartet wird und in einigen seltenen Fällen zu Dateninkonsistenzen führt.
- Es wurde ein Problem behoben, das dazu führen kann, dass die Datenbank-Instance aufgrund eines falschen Zugriffs auf ungültigen Speicher neu gestartet wird, wenn eine Verbindung unmittelbar nach dem Commit einer Transaktion geschlossen wurde.
- Es wurde ein Problem behoben, das unter seltenen Bedingungen dazu führen kann, dass die DB-Instance neu gestartet wird, während eine Abfrage mit einer GROUP BY-Klausel verarbeitet wird, die eine Dezimalspalte auf null Dezimalstellen kürzt.
- Es wurde ein Problem behoben, das dazu führen kann, dass eine DB-Instance neu gestartet wird, weil bei der Ausführung einer Bereichsabfrage mit einem räumlichen Index falsch auf einen Datensatz zugegriffen wurde.

- Es wurde ein Problem behoben, das zu einem Neustart der DB-Instance auf Aurora-MySQL-Replikat-Instances führen kann, wenn interne temporäre Tabellen die standardmäßigen oder vom Kunden konfigurierten Speicher- oder MMAP-Werte überschreiten.
- Es wurde ein Problem behoben, bei dem die Protokollrotation von Advanced Audit zu Problemen bei der Speicherverwaltung führen kann.
- Das schnelle Einfügen ist in dieser Aurora MySQL-Version nicht aktiviert, da ein Problem zu Inkonsistenzen beim Ausführen von Abfragen wie `INSERT INTOSELECT`, und führen kann. `FROM` Weitere Informationen zur Fast-Insert-Optimierung finden Sie unter [Amazon Aurora MySQL Performance Enhancements](#).

#### Allgemeine Verbesserungen:

- Die Latenz bei Leseabfragen in globalen Datenbank-Schreibweiterleitungssitzungen wurde mithilfe der Lesekonsistenzeneinstellung `GLOBAL` verbessert.
- Es wurde ein Problem behoben, bei dem der `wait_timeout`-Parameterwert nicht berücksichtigt wurde, nachdem eine Clientsitzung die Befehle `reset_connection` oder `change_user` ausgeführt hatte.
- Es wurde ein Problem behoben, bei dem Anwendungen beim Herstellen einer Verbindung zu einer DB-Instance eine erhöhte Latenz aufweisen konnten, wenn die Instance plötzlich mehr eingehende Verbindungen verzeichnete. Zwei neue CloudWatch Metriken, `AuroraSlowHandshakeCount` und `AuroraSlowConnectionHandleCount`, wurden eingeführt, um Verzögerungen beim Verbindungsaufbau für Aurora MySQL-DB-Instances zu beheben. Weitere Informationen zu diesen Metriken finden Sie in der Dokumentation zu CloudWatch Aurora-Metrikdefinitionen, [CloudWatch Amazon-Metriken für Amazon Aurora](#).
- `Dertemptable_use_mmap`-Parameter ist veraltet und es wird erwartet, dass die Unterstützung dafür in einer künftigen MySQL-Version eingestellt wird. Weitere Informationen finden Sie unter [Speicher-Engine für interne \(implizite\) temporäre Tabellen](#).
- Es wurde ein Problem behoben, das längere Ausführungszeiten für die Anweisung `SHOW BINARY LOGS` verursachen kann. Dieses könnte zu einem Rückgang des Commit-Durchsatzes der Datenbank führen.

#### Upgrades und Migrationen:

- Zur Durchführung eines Nebenversions-Upgrades für eine globale Aurora-Datenbank von Aurora MySQL Version 3.01 oder 3.02 zu Aurora MySQL Version 3.03 oder höher siehe [Aktualisieren von Aurora MySQL durch Modifizierung der Engine-Version](#).
- Es wurde ein Problem behoben, das dazu führen kann, dass Hauptversions-Upgrades von Aurora MySQL Version 2 zu Aurora MySQL Version 3 fehlschlagen, wenn sich eine große Anzahl von Tabellen (über 750 000) im Cluster befindet.
- Es wurde ein Problem behoben, das dazu führen kann, dass Hauptversions-Upgrades von Aurora MySQL Version 2 zu Aurora MySQL Version 3 fehlschlagen, wenn die Migration der Tabellen `mysql.innodb_table_stats` und `mysql.innodb_index_stats` länger als erwartet dauert. Dieses Problem betraf hauptsächlich DB-Cluster mit Millionen von Tabellen.
- Es wurde ein Problem behoben, das aufgrund von Schemainkonsistenzfehlern beim Upgrade von Aurora MySQL 2 zu Aurora MySQL 3 zu Fehlern führen kann. Diese Fehler werden vom Upgrade-Pre-Checker für die Tabellen `mysql.general_log_template` und `mysql.slow_log_template` gemeldet. Weitere Informationen zur Fehlerbehebung bei Upgrades finden Sie unter [Behebung von Upgrade-Problemen mit Aurora MySQL Version 3](#).
- Es wurde ein Problem behoben, das aufgrund des `schemaInconsistencyCheck`-Fehlers beim Upgrade von Aurora MySQL 2 zu Aurora MySQL 3 zu Fehlern führen kann. Dieser Fehler wird durch Schemainkonsistenzen innerhalb der `mysql.table_migration_index_info`-Tabelle verursacht, wie von der `upgrade-prechecks.log` gemeldet. Weitere Informationen zur Fehlerbehebung bei Upgrades zu Aurora MySQL Version 3 finden Sie unter [Behebung von Upgrade-Problemen mit Aurora MySQL Version 3](#).

## Integration von MySQL-Fehlerbehebungen (Community Edition)

Diese Version enthält alle Community-Fehlerbehebungen bis einschließlich 8.0.26, zusätzlich zu den unten aufgeführten. Weitere Informationen finden Sie unter [MySQL-Fehlerbehebungen durch Aurora-MySQL-3.x-Datenbank-Engine-Updates](#).

- Es wurde ein Problem behoben, bei dem einige Spaltentypen, einschließlich JSON und TEXT, manchmal den Sortierpuffer erschöpften, wenn seine Größe nicht mindestens das 15-fache der Größe der größten Zeile in der Sortierung betrug. Jetzt muss der Sortierpuffer nur noch 15-mal so groß sein wie der größte Sortierschlüssel. (Fehler #103325, Fehler #105532, Fehler #32738705, Fehler #33501541)
- Es wurde ein Problem behoben, bei dem InnoDB einige legale Namen für Tabellenpartitionen nicht immer korrekt behandelte. (Fehler #32208630)

- Es wurde ein Problem behoben, das unter bestimmten Bedingungen aufgrund einer ungenauen Berechnung der NULL-Eigenschaft bei der Ausführung einer Abfrage mit einer OR-Bedingung zu falschen Ergebnissen führen kann. (Fehler #34060289)
- Es wurde ein Problem behoben, das unter bestimmten Bedingungen zu falschen Ergebnissen führen kann, wenn die folgenden beiden Bedingungen erfüllt sind:
  - Eine abgeleitete Tabelle wird mit dem äußeren Abfrageblock zusammengeführt
  - Die Abfrage umfasst einen Left Join und eine IN-Unterabfrage

(Fehler #34060289)

- Es wurde ein Problem behoben, bei dem falsche AUTO\_INCREMENT-Werte generiert wurden, wenn der maximale ganzzahlige Spaltenwert überschritten wurde. Der Fehler war darauf zurückzuführen, dass der maximale Spaltenwert nicht berücksichtigt wurde. In diesem Fall hätte der vorherige gültige AUTO\_INCREMENT-Wert zurückgegeben werden müssen, was zu einem Fehler beim doppelten Schlüssel geführt hat. (Fehler #87926, Fehler #26906787)
- Es wurde ein Problem behoben, bei dem es nicht möglich war, die DROP-Berechtigung für das Performance-Schema zu widerrufen. (Fehler #33578113)
- Es wurde ein Problem behoben, bei dem eine gespeicherte Prozedur, die eine IF-Anweisung mit EXISTS, die auf eine oder mehrere Tabellen reagierte, die zwischen den Ausführungen gelöscht und neu erstellt wurden, bei den nachfolgenden Aufrufen nach dem ersten nicht korrekt ausgeführt wurde. (Fehler #32855634).
- Es wurde ein Problem behoben, bei dem eine Abfrage, die auf eine Ansicht in einer Unterabfrage und einem äußeren Abfrageblock verweist, zu einem unerwarteten Neustart führen kann. (Fehler #32324234)

## Aurora-MySQL-Datenbank-Engine-Updates 17.04.2023 (Version 3.02.3, kompatibel mit MySQL 8.0.23). Ende des Standard-Supports 15. Januar 2024.

Version: 3.02.3

Aurora MySQL 3.02.3 ist allgemein verfügbar. Aurora-MySQL-3.02-Versionen sind mit MySQL 8.0.23 kompatibel und Aurora-MySQL-2.x-Versionen sind mit MySQL 5.7 kompatibel.

Details zu den neuen Features in Aurora MySQL Version 3 und zu den Unterschieden zwischen Aurora MySQL Version 3 und Aurora MySQL Version 2 oder Community MySQL 8.0 finden Sie



unter [Vergleich von Aurora MySQL Version 2 und Aurora MySQL Version 3](#) im Amazon-Aurora-Benutzerhandbuch.

Derzeit werden die Aurora-MySQL-Versionen 2.07.\*, 2.11.1, 2.11.2, 3.01.\*, 3.02.\* und 3.03.\* unterstützt.

Sie können einen Snapshot von jedem derzeit unterstützten Aurora-MySQL-Version-2-Cluster in Aurora MySQL 3.02.3 wiederherstellen.

Informationen zum Planen eines Upgrades auf Aurora MySQL Version 3 finden Sie unter [Upgrade-Planung für Aurora MySQL Version 3](#) im Amazon-Aurora-Benutzerhandbuch. Informationen zum Upgrade-Verfahren selbst finden Sie unter [Upgrade zu Aurora MySQL Version 3](#) im Amazon-Aurora-Benutzerhandbuch. Allgemeine Informationen zu Aurora-MySQL-Upgrades finden Sie unter [Upgrade von Amazon-Aurora-MySQL-DB-Clustern](#) im Amazon-Aurora-Benutzerhandbuch.

Informationen zur Fehlerbehebung finden Sie unter [Beheben von Upgrade-Problemen mit Aurora MySQL Version 3](#).

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem behoben, das dazu führen kann, dass die Datenbank-Instance aufgrund eines falschen Zugriffs auf ungültigen Speicher neu gestartet wurde, wenn eine Verbindung unmittelbar nach dem Commit einer Transaktion geschlossen wurde.
- Das schnelle Einfügen ist in dieser Aurora MySQL-Version nicht aktiviert, da ein Problem zu Inkonsistenzen beim Ausführen von Abfragen wie `INSERT INTOSELECT`, und führen kann. `FROM` Weitere Informationen zur Fast-Insert-Optimierung finden Sie unter [Amazon Aurora MySQL Performance Enhancements](#).

Allgemeine Verbesserungen:

- Es wurde ein Problem behoben, bei dem beim Materialisieren temporärer Zwischentabellen nicht unterstützte Index-Scan-Zugriffsmethoden für Common Table Expressions (CTE) berücksichtigt wurden, was zu unerwünschtem Verhalten wie Datenbankneustarts oder falschen Abfrageergebnissen führen kann. Dieses Problem wurde behoben, indem die Verwendung solcher

nicht unterstützter Index-Scan-Zugriffsmethoden für Tabellen, die die TempTable Speicher-Engine verwenden, vermieden wurde.

- Es wurde ein Problem behoben, das in seltenen Fällen dazu führen kann, dass eine Aurora-Reader-Instance neu gestartet wird, wenn auf eine Tabelle zugegriffen wurde, in der umfangreiche Update- oder Data Definition Language (DDL)-Operationen gleichzeitig auf der Writer-Instance ausgeführt werden.
- Es wurde ein Problem behoben, das in bestimmten Situationen dazu führen kann, dass Aurora-MySQL-Reader-Instances neu gestartet werden, wenn versucht wird, eine Seite zu lesen, auf die während einer Reichweitenschätzung nicht mehr zugegriffen werden konnte.
- Es wurde ein Problem behoben, bei dem es bei Datenbank-Instances, die binäre Protokollreplikation verwenden, zu einer erhöhten CPU-Auslastung und zu Verbindungsausfällen kommen kann, wenn mehrere Benutzer für die binäre Protokollreplikation angeschlossen sind.
- Es wurde ein Problem behoben, das dazu führen kann, dass eine Aurora-MySQL-Reader-Instance neu gestartet wird, während eine Abfrage ausgeführt wird, die einen parallelen Aurora-Abfrageausführungsplan verwendet.

## Aurora-MySQL-Datenbank-Engine-Updates 18.11.2022 (Version 3.02.2, kompatibel mit MySQL 8.0.23) Das Ende des Standard-Supports ist der 15. Januar 2024.

Version: 3.02.2

Aurora MySQL 3.02.2 ist allgemein verfügbar. Aurora MySQL 3.02-Versionen sind mit MySQL 8.0.23 kompatibel, Aurora-MySQL 2.x-Versionen sind kompatibel mit MySQL 5.7 und Aurora-MySQL-1.x-Versionen sind kompatibel mit MySQL 5.6.

Details zu den neuen Features in Aurora MySQL Version 3 und den Unterschieden zwischen Aurora MySQL Version 3 und Aurora MySQL Version 2 oder Community MySQL 8.0 finden Sie unter [Vergleich von Aurora MySQL Version 2 und Aurora MySQL Version 3](#) im Amazon-Aurora-Benutzerhandbuch.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Sie können einen Snapshot von jedem derzeit unterstützten Aurora-MySQL-Version-2-Cluster in Aurora MySQL 3.02.2 wiederherstellen.

Informationen zum Planen eines Upgrades auf Aurora MySQL Version 3 finden Sie unter [Upgrade-Planung für Aurora MySQL Version 3](#) im Amazon-Aurora-Benutzerhandbuch. Informationen zum Upgrade-Verfahren selbst finden Sie unter [Upgrade zu Aurora MySQL Version 3](#) im Amazon-Aurora-Benutzerhandbuch. Allgemeine Informationen zu Aurora-MySQL-Upgrades finden Sie unter [Upgrade von Amazon-Aurora-MySQL-DB-Clustern](#) im Amazon-Aurora-Benutzerhandbuch.

Informationen zur Fehlerbehebung finden Sie unter [Beheben von Upgrade-Problemen mit Aurora MySQL Version 3](#).

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Aurora MySQL Version 3.02.2 ist im Allgemeinen mit Community MySQL 8.0.23 kompatibel.

Behobene Sicherheitsprobleme und CVEs sind unten aufgeführt:

Korrekturen und andere Verbesserungen bei der Feinabstimmung der Handhabung in einer verwalteten Umgebung. Weitere CVE-Korrekturen unten:

- [CVE-2022-21451](#)
- [CVE-2021-36222](#)
- [CVE-2021-22926](#)
  
- [CVE-2022-21444](#)

Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem behoben, das dazu führen kann, dass die Datenbank-Instance aufgrund eines irrtümlichen Zugriffs auf ungültigen Speicher neu gestartet wird, wenn eine Verbindung unmittelbar nach dem Commit einer Transaktion explizit oder implizit geschlossen wurde.
- Es wurde ein Problem behoben, das dazu führen kann, dass das Datenbank-Startup bei größeren Instance-Klassen wiederholt unterbrochen wird, wenn die Initialisierung des Pufferpools länger als erwartet dauert.

- Es wurde ein Problem behoben, das unter seltenen Bedingungen dazu führen kann, dass die Datenbank-Instance neu gestartet wird, wenn Aurora Serverless v2 fälschlicherweise versucht, den Tabellen-Cache während der Skalierung zu aktualisieren.
- Es wurde ein Problem behoben, das unter seltenen Bedingungen dazu führen könnte, dass die DB-Instance neu gestartet wird, während eine Abfrage mit einer GROUP BY-Klausel verarbeitet wird, die eine Dezimalspalte auf null Dezimalstellen kürzt.
- Das schnelle Einfügen ist in dieser Aurora MySQL-Version nicht aktiviert, da ein Problem zu Inkonsistenzen beim Ausführen von Abfragen wie `INSERT INTOSELECT`, und führen kann. Weitere Informationen zur Fast-Insert-Optimierung finden Sie unter [Amazon Aurora MySQL Performance Enhancements](#).

#### Allgemeine Verbesserungen:

- Es wurde ein Problem behoben, das dazu führen kann, dass das Upgrade von Aurora MySQL Version 2 (mit MySQL 5.7 kompatibel) zu Aurora MySQL Version 3 (mit MySQL 8.0 kompatibel) aufgrund von Metadaten-Inkonsistenzen in der `mysql.host`-Tabelle fehlschlägt.
- Es wurden Leistungsverbesserungen hinzugefügt, um die Upgrade-Zeit von Aurora MySQL Version 2 (mit MySQL 5.7 kompatibel) zu Aurora MySQL Version 3 (mit MySQL 8.0 kompatibel) zu reduzieren. Durch die Parallelisierung bestimmter Upgrade-Schritte wird die Zeit bei Verwendung größerer Instance-Klassen wie `db.r6g.16xlarge` oder `db.r5.24xlarge` weiter verkürzt.
- Es wurde Unterstützung für die Anzeige aller Fehler beim Upgrade von Aurora MySQL Version 2 (mit MySQL 5.7 kompatibel) zu Aurora MySQL Version 3 (mit MySQL 8.0 kompatibel) hinzugefügt, wohingegen frühere Versionen auf die Anzeige von nur 50 Fehlern beschränkt waren.
- Es wurde ein Problem behoben, das unter seltenen Bedingungen dazu führen kann, dass die Zähler für die automatische Inkrementierung nach dem Upgrade von Aurora MySQL Version 2 (kompatibel mit MySQL 5.7) zu Aurora MySQL Version 3 (mit MySQL 8.0 kompatibel) inkorrekt sind.
- Es wurde ein Problem behoben, das dazu führen kann, dass Hauptversions-Upgrades von Aurora MySQL Version 2 zu Aurora MySQL Version 3 fehlschlagen, wenn die Migration der Tabellen ``mysql.innodb_table_stats`` und ``mysql.innodb_index_stats`` länger dauert als erwartet. Dieses Problem betraf hauptsächlich Datenbank-Cluster mit einer großen Anzahl von Tabellen (> 1,5 Millionen).
- Es wurde ein Problem behoben, das dazu führen kann, dass Hauptversions-Upgrades von Aurora MySQL Version 2 zu Aurora MySQL Version 3 aufgrund eines Fehlers im AMS-8.0-Engine-Upgrade-Workflow fehlschlagen, wodurch die Protokolldatensätze auf dem Aurora-Speicher-

Cluster-Volumen gesammelt und normale Schreibvorgänge gestoppt werden. Dieses Problem betraf hauptsächlich Datenbank-Cluster mit einer großen Anzahl von Tabellen (> 750 000).

- Es wurde ein Problem behoben, das verhinderte, dass ungenutzte Aurora-MySQL-Serverless-v2-Instances auf 0,5 ACUs herunterskaliert wurden, da die MySQL-Bereinigungs-Threads fälschlicherweise aktiv gehalten wurden.
- Es wurde ein Problem behoben, bei dem Anwendungen beim Herstellen einer Verbindung zu einer DB-Instance eine erhöhte Latenz aufweisen konnten, wenn die Instance plötzlich deutlich mehr eingehende Verbindungen verzeichnete.
- Zwei neue CloudWatch Amazon-Metriken wurden eingeführt, um Verzögerungen beim Verbindungsaufbau für Aurora MySQL-Datenbank-Instances zu beheben. Weitere Informationen zu AuroraSlowHandshakeCount und AuroraSlowConnectionHandleCount Metriken finden Sie in den [CloudWatch Aurora-Metrikdefinitionen](#).

## Integration von MySQL-Fehlerbehebungen (Community Edition)

Diese Version enthält alle Community-Fehlerbehebungen bis einschließlich 8.0.23, zusätzlich zu den unten aufgeführten. Weitere Informationen finden Sie unter [MySQL-Fehlerbehebungen durch Aurora-MySQL-3.x-Datenbank-Engine-Updates](#).

- Es wurde ein Problem behoben, das unter bestimmten Bedingungen aufgrund einer ungenauen Berechnung der NULL-Eigenschaft bei der Ausführung einer Abfrage mit einer OR-Bedingung zu falschen Ergebnissen führen kann. (Fehler #34060289)
- Es wurde ein Problem behoben, das unter bestimmten Bedingungen zu falschen Ergebnissen führen kann, wenn die folgenden beiden Bedingungen erfüllt sind:
  - Eine abgeleitete Tabelle wird mit dem äußeren Abfrageblock zusammengeführt.
  - Die Abfrage umfasst einen Left Join und eine IN-Unterabfrage. (Fehler #34060289)
- Es wurde ein Problem behoben, bei dem es nicht möglich war, die DROP-Berechtigung für das Performance-Schema zu widerrufen. (Fehler #33578113)
- Es wurde ein Problem behoben, bei dem eine gespeicherte Prozedur, die eine IF-Anweisung mit EXISTS, die auf eine oder mehrere Tabellen wirkte, die zwischen den Ausführungen gelöscht und neu erstellt wurden, bei den nachfolgenden Aufrufen nach dem ersten nicht korrekt ausgeführt wurde. (MySQL-Fehler #32855634).
- Es wurden inkorrekte AUTO\_INCREMENT-Werte generiert, wenn der maximale ganzzahlige Spaltenwert überschritten wurde. Der Fehler war darauf zurückzuführen, dass der maximale Spaltenwert nicht berücksichtigt wurde. Es hätte der vorherige gültige AUTO\_INCREMENT-Wert

zurückgegeben werden müssen, was zu einem Fehler mit doppeltem Schlüssel führte. (Fehler #87926, Fehler #26906787)

- Es wurde ein Problem behoben, das zu einem Fehler beim Upgrade eines Aurora-MySQL-Datenbank-Clusters der Version 1 (kompatibel mit MySQL 5.6) führen kann, der vom Benutzer erstellte Tabellen mit bestimmten Tabellen-IDs enthält. Die Zuweisung dieser Tabellen-IDs kann beim Upgrade von Aurora MySQL Version 2 (kompatibel mit MySQL 5.7) zu Aurora MySQL Version 3 (kompatibel mit MySQL 8.0) dazu führen, dass Datenwörterbuch-Tabellen-IDs in Konflikt geraten (Fehler #33919635)

**Aurora-MySQL-Datenbank-Engine-Updates 07.09.2022 (Version 3.02.1, kompatibel mit MySQL 8.0.23).** Die Standardunterstützung endet am 15. Januar 2024. Upgrades auf diese Version werden nicht unterstützt.

Version: 3.02.1

Aurora MySQL 3.02.1 ist allgemein verfügbar. Aurora-MySQL-3.02-Versionen sind mit MySQL 8.0.23 kompatibel, Aurora-MySQL-2.x-Versionen sind kompatibel mit MySQL 5.7 und Aurora-MySQL-1.x-Versionen sind kompatibel mit MySQL 5.6.

Details zu den neuen Features in Aurora MySQL Version 3 und den Unterschieden zwischen Aurora MySQL Version 3 und Aurora MySQL Version 2 oder Community MySQL 8.0 finden Sie unter [Vergleich von Aurora MySQL Version 2 und Aurora MySQL Version 3](#) im Amazon-Aurora-Benutzerhandbuch.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Informationen zum Planen eines Upgrades auf Aurora MySQL Version 3 finden Sie unter [Upgrade-Planung für Aurora MySQL Version 3](#) im Amazon-Aurora-Benutzerhandbuch. Informationen zum Upgrade-Verfahren selbst finden Sie unter [Upgrade zu Aurora MySQL Version 3](#) im Amazon-Aurora-Benutzerhandbuch. Allgemeine Informationen zu Aurora-MySQL-Updates finden Sie unter [Upgrade von Amazon-Aurora-MySQL-DB-Clustern](#) im Amazon-Aurora-Benutzerhandbuch.

Informationen zur Fehlerbehebung finden Sie unter [Beheben von Upgrade-Problemen mit Aurora MySQL Version 3](#).

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Aurora MySQL Version 3.02.1 ist im Allgemeinen mit Community MySQL 8.0.23 kompatibel.

Behobene Sicherheitsprobleme und CVEs sind unten aufgeführt:

Korrekturen und andere Verbesserungen bei der Feinabstimmung der Handhabung in einer verwalteten Umgebung. Weitere CVE-Korrekturen unten:

- [CVE-2022-0778](#)

Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem behoben, das zu Verbindungsfehlern und hoher Latenz führen kann, wenn mehrere MySQL-Binärprotokollreplikate (Binlog) an einen Aurora-Writer-Knoten angefügt sind oder wenn eine große Anzahl gleichzeitiger Abfragen mit langer Laufzeit in Verbindung mit einem Anstieg neuer Verbindungsanfragen vorliegen.
- Es wurde ein Problem behoben, das zu einem Datenbankneustart führt, wenn die erweiterte Überwachung von CONNECT-Ereignissen aktiviert ist.
- Es wurde ein Problem behoben, das zu einem Neustart der DB-Instance auf Aurora-MySQL-Replikat-Instances führen kann, wenn interne temporäre Tabellen die standardmäßigen oder vom Kunden konfigurierten Speicher- oder MMAP-Werte überschreiten.
- Es wurde ein Problem behoben, das dazu führen kann, dass ein Lesereplikat während gleichzeitiger DDL-Operationen mit gespeicherten Prozeduren wiederholt neu gestartet wird.
- Das schnelle Einfügen ist in dieser Aurora MySQL-Version nicht aktiviert, da ein Problem zu Inkonsistenzen beim Ausführen von Abfragen wie `INSERT INTOSELECT`, und führen kann. `FROM` Weitere Informationen zur Fast-Insert-Optimierung finden Sie unter [Amazon Aurora MySQL Performance Enhancements](#).

Allgemeine Verbesserungen:

- Es werden jetzt R6i-Instances unterstützt.

## Zusätzliche Informationen:

- Aurora MySQL Version 3.02.1 unterstützt keine Unterversions-Upgrades direkt von Aurora MySQL Version 2 (kompatibel mit MySQL 5.7). Um ein Hauptversions-Upgrade auf diese Version durchzuführen, führen Sie zunächst ein Hauptversions-Upgrade auf Aurora MySQL Version 3.02.0 durch und dann ein direktes Upgrade der Nebenversion auf Aurora MySQL Version 3.02.1.

**Aurora-MySQL-Datenbank-Engine-Updates 20.04.2022 (Version 3.02.0, kompatibel mit MySQL 8.0.23). Ende des Standard-Supports 15. Januar 2024. Upgrades auf diese Version werden nicht unterstützt.**

Version: 3.02.0

Aurora MySQL 3.02.0 ist allgemein verfügbar. Aurora-MySQL-3.02-Versionen sind mit MySQL 8.0.23 kompatibel, Aurora-MySQL-2.x-Versionen sind kompatibel mit MySQL 5.7 und Aurora-MySQL-1.x-Versionen sind kompatibel mit MySQL 5.6.

Details zu den neuen Features in Aurora MySQL Version 3 und den Unterschieden zwischen Aurora MySQL Version 3 und Aurora MySQL Version 2 oder Community MySQL 8.0 finden Sie unter [Vergleich von Aurora MySQL Version 2 und Aurora MySQL Version 3](#) im Amazon-Aurora-Benutzerhandbuch.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Sie können einen Snapshot von jedem derzeit unterstützten Aurora-MySQL-Version-2-Cluster in Aurora MySQL 3.02.0 wiederherstellen.

Informationen zum Planen eines Upgrades auf Aurora MySQL Version 3 finden Sie unter [Upgrade-Planung für Aurora MySQL Version 3](#) im Amazon-Aurora-Benutzerhandbuch. Informationen zum Upgrade-Verfahren selbst finden Sie unter [Upgrade zu Aurora MySQL Version 3](#) im Amazon-Aurora-Benutzerhandbuch. Allgemeine Informationen zu Aurora-MySQL-Upgrades finden Sie unter [Upgrade von Amazon-Aurora-MySQL-DB-Clustern](#) im Amazon-Aurora-Benutzerhandbuch.

Informationen zur Fehlerbehebung finden Sie unter [Beheben von Upgrade-Problemen mit Aurora MySQL Version 3](#).



Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Aurora MySQL Version 3.02.0 ist im Allgemeinen mit Community MySQL 8.0.23 kompatibel.

Behobene Sicherheitsprobleme und CVEs sind unten aufgeführt:

Korrekturen und andere Verbesserungen bei der Feinabstimmung der Handhabung in einer verwalteten Umgebung. Weitere CVE-Korrekturen unten:

- [CVE-2021-22946](#)

Neue Funktionen:

- Amazon Aurora Serverless v2 ist allgemein verfügbar. Weitere Informationen finden Sie in der Übersicht über [Amazon Aurora Serverless](#), im [Blog](#) und in der Dokumentation [Verwendung von Aurora Serverless v2](#). Beginnen Sie noch heute mit der Erstellung einer Aurora Serverless v2-Datenbank in nur wenigen Schritten in der AWS Management Console.

Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem behoben, das dazu führen kann, dass der Server beim Löschen eines Datensatzes oder beim Löschen einer Tabelle mit zwei oder mehr Spalten variabler Länge (Typen VARCHAR, VARBINARY, BLOB und TEXT) in eine Neustartschleife gerät und nicht verfügbar ist. Weitere Informationen zu Spaltentypen finden Sie unter [innodb-row-format](#).
- Es wurde ein Problem behoben, bei dem das Zeitlimit für bestehende Verbindungen überschritten und neue Verbindungen auf einem Cluster mit aktiviertem Binärprotokoll und mindestens einem Binärprotokoll-Consumer angeschlossen waren, nicht hergestellt werden konnten, was zu Ressourcenkonflikten zwischen der Anwendung und dem/den Verbraucher(n) führte.
- Der freie Speicherplatz wird anhand der `FreeableMemory` CloudWatch Metrik angegeben. Weitere Informationen finden Sie unter [CloudWatch Amazon-Metriken für Amazon Aurora](#).
  - Es wurde ein Problem behoben, das bei aktivierter binärer Protokollreplikation zu einem Neustart oder einem Failover der DB-Instance führen kann, wenn der freie Speicher abnimmt.
  - Es wurde ein Problem behoben, das beim Einstellen von Sitzungsvariablen zu einem Neustart oder einem Failover der DB-Instance führen kann, wenn der freie Speicher abnimmt.

- Es wurde ein Problem behoben, das zu einem Neustart oder einem Failover der DB-Instance führen kann, wenn der freie Speicher abnimmt, während die Datenbank eine vorhandene Datei verarbeitet.
- Es wurde ein Problem behoben, das unter seltenen Bedingungen zu einem Fehler mit doppelten Einträgen führen kann, wenn neue Zeilen in eine Tabelle eingefügt werden, die eine AUTO\_INCREMENT-Spalte in einem aus einem Snapshot wiederhergestellten Cluster enthält.
- Das schnelle Einfügen ist in dieser Aurora MySQL-Version nicht aktiviert, da ein Problem zu Inkonsistenzen beim Ausführen von Abfragen wie INSERT INTOSELECT, und führen kann. Weitere Informationen zur Fast-Insert-Optimierung finden Sie unter [Amazon Aurora MySQL Performance Enhancements](#).

#### Allgemeine Verbesserungen:

- Es wurde ein Problem behoben, bei dem der Volume-Status bei Verwendung des SHOW VOLUME STATUS-Befehls nicht angezeigt wurde. Weitere Informationen finden Sie unter [AuroraMysql.Managing. VolumeStatus](#).
- Es wurde ein Problem behoben, das dazu führte, dass Aufrufe von [mysql\\_rds\\_import\\_binlog\\_ssl\\_material](#) mit dem [MySQL-Serverfehler 3512](#) fehlschlugen.
- Es wurde ein Problem behoben, bei dem die Aurora-Replikatzögerung fälschlicherweise für gelöschte Aurora-Reader-Instances gemeldet wurde.

#### Upgrades/Migration:

- Es wurde ein Problem behoben, das aufgrund eines Problems beim Kopieren von ibdata-Dateien und -Tablespaces in den Aurora-Speicher zu Migrationsfehlern von MySQL-8.0.x-Datenbanken zu Aurora MySQL Version 3 führen kann.
- Es wurde ein Problem behoben, das dazu führen kann, dass Upgrades von Clustern von Aurora MySQL Version 2 zu Aurora MySQL Version 3 fehlschlagen, wenn Datenbanktabellen eine große Datenmenge enthalten.
- Es wurde ein Problem behoben, das bei der Wiederherstellung von Clustern von Aurora MySQL Version 2 zu Aurora MySQL Version 3 aufgrund eines Fehlers bei der Erstellung [serialisierter Datenwörterbuchinformationen](#) (SDI) für eine Tabelle zu Fehlern führen kann.
- Es wurde ein Problem behoben, das aufgrund von Schemainkonsistenzfehlern, die für RDS-Systemtabellen von Upgrade-Vorabprüfungen gemeldet wurden, beim Upgrade von Aurora MySQL 2 zu Aurora MySQL 3 zu Fehlern führen kann.

- Es wurde ein Problem behoben, das bei der Migration oder Wiederherstellung von RDS-für-MySQL 8.0- oder Aurora-MySQL-Version-2- zu Aurora-MySQL-Version-3-Datenbanken aufgrund einer ungültigen Syntax in einer von RDS verwalteten gespeicherten Prozedur zu Fehlern führen kann.
- Es wurde ein Problem behoben, das aufgrund von Schemainkonsistenzfehlern, die für die Tabellen [Allgemeines Protokoll](#) und [Slow-Protokoll](#) gemeldet wurden, beim Upgrade von Aurora MySQL 2 zu Aurora MySQL 3 zu Fehlern bei der Upgrade-Vorabprüfung führen kann.

## Integration von MySQL-Fehlerbehebungen (Community Edition):

Diese Version enthält alle Community-Fehlerbehebungen bis einschließlich 8.0.23, zusätzlich zu den unten aufgeführten. Weitere Informationen finden Sie unter [MySQL-Fehlerbehebungen durch Aurora-MySQL-3.x-Datenbank-Engine-Updates](#).

- Die unsachgemäße Behandlung von temporären Tabellen, die für Cursor in gespeicherten Prozeduren verwendet werden, was zu unerwartetem Serververhalten führen könnte, wurde behoben, [mysqld-8-0-24-bug](#). (Fehler #32416811)

**Aurora-MySQL-Datenbank-Engine-Updates 15.04.2022 (Version 3.01.1, kompatibel mit MySQL 8.0.23). Ende des Standard-Supports 15. Januar 2024. Upgrades auf diese Version werden nicht unterstützt.**

Version: 3.01.1

Aurora MySQL 3.01.1 ist allgemein verfügbar. Aurora MySQL 3.01-Versionen sind mit MySQL 8.0.23 kompatibel, Aurora-MySQL 2.x-Versionen sind kompatibel mit MySQL 5.7 und Aurora-MySQL-1.x-Versionen sind kompatibel mit MySQL 5.6.

Details zu den neuen Features in Aurora MySQL Version 3 und den Unterschieden zwischen Aurora MySQL Version 3 und Aurora MySQL Version 2 oder Community MySQL 8.0 finden Sie unter [Vergleich von Aurora MySQL Version 2 und Aurora MySQL Version 3](#) im Amazon-Aurora-Benutzerhandbuch.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Sie können einen Snapshot von jedem derzeit unterstützten Aurora-MySQL-Version-2-Cluster in Aurora MySQL 3.01.1 wiederherstellen.

Informationen zum Planen eines Upgrades auf Aurora MySQL Version 3 finden Sie unter [Upgrade-Planung für Aurora MySQL Version 3](#) im Amazon-Aurora-Benutzerhandbuch. Informationen zum Upgrade-Verfahren selbst finden Sie unter [Upgrade zu Aurora MySQL Version 3](#) im Amazon-Aurora-Benutzerhandbuch. Allgemeine Informationen zu Aurora-MySQL-Updates finden Sie unter [Upgrade von Amazon-Aurora-MySQL-DB-Clustern](#) im Amazon-Aurora-Benutzerhandbuch.

Informationen zur Fehlerbehebung finden Sie unter [Beheben von Upgrade-Problemen mit Aurora MySQL Version 3](#).

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Aurora MySQL Version 3.01.1 ist im Allgemeinen mit Community MySQL 8.0.23 kompatibel.

Aurora MySQL Version 3.01.1 wird für Upgrades und Migrationen auf eine MySQL-8.0-kompatible Aurora-Datenbank empfohlen.

Behobene Sicherheitsprobleme und CVEs sind unten aufgeführt:

Korrekturen und andere Verbesserungen bei der Feinabstimmung der Handhabung in einer verwalteten Umgebung. Weitere CVE-Korrekturen unten:

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

Verbesserungen der Verfügbarkeit:

- Der freie Speicherplatz wird anhand der FreeableMemory CloudWatch Metrik angegeben. Weitere Informationen finden Sie unter [CloudWatch Amazon-Metriken für Amazon Aurora](#).
- Es wurde ein Problem behoben, das bei aktivierter binärer Protokollreplikation zu einem Neustart oder einem Failover der DB-Instance führen kann, wenn der freie Speicher abnimmt.

- Es wurde ein Problem behoben, das beim Einstellen von Sitzungsvariablen zu einem Neustart oder einem Failover der DB-Instance führen kann, wenn der freie Speicher abnimmt.
- Es wurde ein Problem behoben, das zu einem Neustart oder einem Failover der DB-Instance führen kann, wenn der freie Speicher abnimmt, während die Datenbank eine vorhandene Datei verarbeitet.
- Es wurde ein Problem behoben, das unter seltenen Bedingungen zu einem Fehler mit doppelten Einträgen führen kann, wenn neue Zeilen in eine Tabelle eingefügt werden, die eine AUTO\_INCREMENT-Spalte in einem aus einem Snapshot wiederhergestellten Cluster enthält.
- Das schnelle Einfügen ist in dieser Aurora MySQL-Version nicht aktiviert, da ein Problem zu Inkonsistenzen beim Ausführen von Abfragen wie INSERT INTOSELECT, und führen kann. FROM Weitere Informationen zur Fast-Insert-Optimierung finden Sie unter [Amazon Aurora MySQL Performance Enhancements](#).

#### Allgemeine Verbesserungen:

- Es wurde ein Problem behoben, bei dem der Volume-Status bei Verwendung des SHOW VOLUME STATUS-Befehls nicht angezeigt wurde. Weitere Informationen finden Sie unter [AuroraMysql.Managing. VolumeStatus](#).
- Es wurde ein Problem behoben, das dazu führte, dass Aufrufe von [mysql\\_rds\\_import\\_binlog\\_ssl\\_material](#) mit dem [MySQL-Serverfehler 3512](#) fehlschlugen.
- Es wurde ein Problem behoben, bei dem die Aurora-Replikatzögerung fälschlicherweise für gelöschte Aurora-Reader-Instances gemeldet wurde.

#### Upgrades/Migration:

- Es wurde ein Problem behoben, das aufgrund eines Problems beim Kopieren von ibdata-Dateien und -Tablespaces in den Aurora-Speicher zu Migrationsfehlern von MySQL-8.0.x-Datenbanken zu Aurora MySQL Version 3 führen kann.
- Es wurde ein Problem behoben, das dazu führen kann, dass Upgrades von Clustern von Aurora MySQL Version 2 zu Aurora MySQL Version 3 fehlschlagen, wenn Datenbanktabellen eine große Datenmenge enthalten.
- Es wurde ein Problem behoben, das bei der Wiederherstellung von Clustern von Aurora MySQL Version 2 zu Aurora MySQL Version 3 aufgrund eines Fehlers bei der Erstellung [serialisierter Datenwörterbuchinformationen](#) (SDI) für eine Tabelle zu Fehlern führen kann.

- Es wurde ein Problem behoben, das aufgrund von Schemainkonsistenzfehlern, die für RDS-Systemtabellen von Upgrade-Vorabprüfungen gemeldet wurden, beim Upgrade von Aurora MySQL 2 zu Aurora MySQL 3 zu Fehlern führen kann.
- Es wurde ein Problem behoben, das bei der Migration oder Wiederherstellung von RDS-für-MySQL 8.0- oder Aurora-MySQL-Version-2- zu Aurora-MySQL-Version-3-Datenbanken aufgrund einer ungültigen Syntax in einer von RDS verwalteten gespeicherten Prozedur zu Fehlern führen kann.
- Es wurde ein Problem behoben, das aufgrund von Schemainkonsistenzfehlern, die für die Tabellen [Allgemeines Protokoll](#) und [Slow-Protokoll](#) gemeldet wurden, beim Upgrade von Aurora MySQL 2 zu Aurora MySQL 3 zu Fehlern bei der Upgrade-Vorabprüfung führen kann.

## Integration von MySQL-Fehlerbehebungen (Community Edition):

Diese Version enthält alle Community-Fehlerbehebungen bis einschließlich 8.0.23, zusätzlich zu den unten aufgeführten. Weitere Informationen finden Sie unter [MySQL-Fehlerbehebungen durch Aurora-MySQL-3.x-Datenbank-Engine-Updates](#).

- Die unsachgemäße Behandlung von temporären Tabellen, die für Cursor in gespeicherten Prozeduren verwendet werden, was zu unerwartetem Serververhalten führen könnte, wurde behoben, [mysqld-8-0-24-bug](#). (Fehler #32416811)

**Aurora-MySQL-Datenbank-Engine-Updates 18.11.2021 (Version 3.01.0, kompatibel mit MySQL 8.0.23). Ende des Standard-Supports 15. Januar 2024. Upgrades auf diese Version werden nicht unterstützt.**

Version: 3.01.0

Aurora MySQL 3.01.0 ist allgemein verfügbar. Aurora MySQL 3.01-Versionen sind mit MySQL 8.0.23 kompatibel, Aurora-MySQL 2.x-Versionen sind kompatibel mit MySQL 5.7 und Aurora-MySQL-1.x-Versionen sind kompatibel mit MySQL 5.6.

Details zu den neuen Features in Aurora MySQL Version 3 und den Unterschieden zwischen Aurora MySQL Version 3 und Aurora MySQL Version 2 oder Community MySQL 8.0 finden Sie unter [Vergleich von Aurora MySQL Version 2 und Aurora MySQL Version 3](#) im Amazon-Aurora-Benutzerhandbuch.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Sie können einen Snapshot von jedem derzeit unterstützten Aurora-MySQL-Version 2-Cluster in Aurora MySQL 3.01.0 wiederherstellen.

Informationen zum Planen eines Upgrades auf Aurora MySQL Version 3 finden Sie unter [Upgrade-Planung für Aurora MySQL Version 3](#) im Amazon-Aurora-Benutzerhandbuch. Informationen zum Upgrade-Verfahren selbst finden Sie unter [Upgrade zu Aurora MySQL Version 3](#) im Amazon-Aurora-Benutzerhandbuch. Allgemeine Informationen zu Aurora-MySQL-Upgrades finden Sie unter [Upgrade von Amazon-Aurora-MySQL-DB-Clustern](#) im Amazon-Aurora-Benutzerhandbuch.

Informationen zur Fehlerbehebung finden Sie unter [Beheben von Upgrade-Problemen mit Aurora MySQL Version 3](#).

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Aurora MySQL Version 3.01.0 ist im Allgemeinen mit der Community-MySQL 8.0.23 kompatibel. Diese Version enthält die Sicherheitsfixes für Probleme mit allgemeinen Sicherheitslücken und Expositionen (CVE) ab der Community-MySQL 8.0.23.

Aurora-MySQL-Version 3.01.0 enthält alle Aurora-spezifischen Bugfixes durch Aurora-MySQL-Version 2.10.0.

Details zu den neuen Features in Aurora MySQL Version 3 finden Sie unter [Features aus Community MySQL 8.0](#) und [Neue Optimierungen für parallele Abfragen](#) im Amazon-Aurora-Benutzerhandbuch.

Verbesserungen der Verfügbarkeit:

- Das schnelle Einfügen ist in dieser Aurora MySQL-Version nicht aktiviert, da ein Problem zu Inkonsistenzen beim Ausführen von Abfragen wie `INSERT INTOSELECT`, und führen kann. `FROM` Weitere Informationen zur Fast-Insert-Optimierung finden Sie unter [Amazon Aurora MySQL Performance Enhancements](#).

# Aktualisierungen der Datenbank-Engine für Amazon Aurora MySQL Version 2

Im Folgenden finden Sie Datenbank-Engine-Updates für Amazon Aurora MySQL Version 2.

- [Updates der Aurora MySQL-Datenbank-Engine 2024-03-19 \(Version 2.12.2, kompatibel mit MySQL 5.7.44\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 28.12.2023 \(Version 2.12.1, kompatibel mit MySQL 5.7.40\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 25.10.2023 \(Version 2.12.0.1, kompatibel mit MySQL 5.7.40\) Beta](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 25.07.2023 \(Version 2.12.0, kompatibel mit MySQL 5.7.40\)](#)
- [Aurora MySQL-Datenbank-Engine-Updates 2024-03-26 \(Version 2.11.5, kompatibel mit MySQL 5.7.12\) Standard](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 17.10.2023 \(Version 2.11.4, kompatibel mit MySQL 5.7.12\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 09.06.2023 \(Version 2.11.3, kompatibel mit MySQL 5.7.12\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 24.03.2023 \(Version 2.11.2, kompatibel mit MySQL 5.7.12\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 14.02.2023 \(Version 2.11.1, kompatibel mit MySQL 5.7.12\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 25.10.2022 \(Version 2.11.0, kompatibel mit MySQL 5.7.12\). Diese Version ist für neue Erstellungen nicht verfügbar.](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 01.11.2022 \(Version 2.10.3\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 26.01.2022 \(Version 2.10.2\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 21.10.2021 \(Version 2.10.1\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 25.05.2021 \(Version 2.10.0\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 12.11.2021 \(Version 2.09.3\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 26.02.2021 \(Version 2.09.2\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2020 \(Version 2.09.1\) \(veraltet\)](#)



- [Aurora-MySQL-Datenbank-Engine-Updates 17.09.2020 \(Version 2.09.0\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 06.01.2022 \(Version 2.08.4\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 12.11.2020 \(Version 2.08.3\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 28.08.2020 \(Version 2.08.2\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 18.06.2020 \(Version 2.08.1\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 02.06.2020 \(Version 2.08.0\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 15.08.2023 \(Version 2.07.10, kompatibel mit MySQL 5.7.12\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 04.05.2023 \(Version 2.07.9, kompatibel mit MySQL 5.7.12\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 16.06.2022 \(Version 2.07.8\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 24.11.2021 \(Version 2.07.7\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 02.09.2021 \(Version 2.07.6\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 06.07.2021 \(Version 2.07.5\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 04.03.2021 \(Version 2.07.4\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 10.11.2020 \(Version 2.07.3\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 17.04.2020 \(Version 2.07.2\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 23.12.2019 \(Version 2.07.1\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 25.11.2019 \(Version 2.07.0\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 22.11.2019 \(Version 2.06.0\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 11.11.2019 \(Version 2.05.0\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 14.08.2018 \(Version 2.04.9\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 20.11.2019 \(Version 2.04.8\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 14.11.2019 \(Version 2.04.7\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 19.09.2019 \(Version 2.04.6\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 08.07.2019 \(Version 2.04.5\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 29.05.2019 \(Version 2.04.4\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 09.05.2019 \(Version 2.04.3\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 02.05.2019 \(Version 2.04.2\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 25.03.2019 \(Version 2.04.1\) \(veraltet\)](#)

- [Aurora-MySQL-Datenbank-Engine-Updates 25.03.2019 \(Version 2.04.0\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 07.02.2019 \(Version 2.03.4\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 18.01.2019 \(Version 2.03.3\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 09.01.2019 \(Version 2.03.2\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 24.10.2018 \(Version 2.03.1\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 11.10.2018 \(Version 2.03\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 08.10.2018 \(Version 2.02.5\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 21.09.2018 \(Version 2.02.4\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 23.08.2018 \(Version 2.02.3\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 04.06.2018 \(Version 2.02.2\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 03.05.2018 \(Version 2.02\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 13.03.2018 \(Version 2.01.1\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 06.02.2018 \(Version 2.01\) \(veraltet\)](#)

## Updates der Aurora MySQL-Datenbank-Engine 2024-03-19 (Version 2.12.2, kompatibel mit MySQL 5.7.44)

Version: 2.12.2

Aurora MySQL 2.12.2 ist allgemein verfügbar. Aurora MySQL 2.12-Versionen sind bis zu MySQL 5.7.44 kompatibel. Weitere Informationen zu Community-Änderungen finden Sie unter [Änderungen in MySQL 5.7.44 \(11.10.2022, allgemeine Verfügbarkeit\)](#).

Derzeit werden die Aurora MySQL-Versionen 2.07.9, 2.07.10, 2.11.\*, 2.12.\*, 3.03.\*, 3.04.\*, 3.05.\* und 3.06.\* unterstützt.

Sie können einen vorhandenen Aurora MySQL 2.\*-Datenbankcluster auf Aurora MySQL 2.12.2 aktualisieren. Sie können auch einen Snapshot von jeder aktuell unterstützten Aurora MySQL-Version in Aurora MySQL 2.12.2 wiederherstellen.

Wenn Sie eine globale Aurora-MySQL-Datenbank zu Version 2.12.\* aktualisieren und die Schreibweiterleitung aktiviert haben, müssen Sie Ihre primären und sekundären DB-Cluster auf exakt dieselbe Version aktualisieren, einschließlich des Patch-Levels, um die Schreibweiterleitung weiterhin verwenden zu können. Weitere Informationen zum Upgrade der Nebenversion einer globalen Aurora-Datenbank finden Sie unter [Upgrades für Nebenversionen](#).

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

### Note

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Behobene Sicherheitsprobleme und CVEs sind unten aufgeführt:

Diese Version enthält alle Community-CVE-Fixes bis einschließlich MySQL 5.7.44. Die folgenden CVE-Korrekturen sind enthalten:

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

Sicherheitsprobleme:

- Es wurde ein Fix hinzugefügt, der sicherstellt, dass binäre Protokollreplikate standardmäßig SSL/TLS verwenden, wenn die Quelle verschlüsselte Verbindungen unterstützt, unabhängig von der Einstellung. MASTER\_SSL

Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem behoben, das verhindern kann, dass eine Read Replica-Instanz erfolgreich gestartet wird, wenn die Writer-Instance stark ausgelastet ist.
- Es wurde ein Problem behoben, das dazu führen kann, dass eine Aurora MySQL-Datenbank-Writer-Instance aufgrund eines Defekts in der Komponente, die mit dem Aurora-Speicher kommuniziert, zu einem Failover führt. Der Fehler ist auf einen Ausfall der Kommunikation zwischen der Datenbank-Instance und dem zugrunde liegenden Speicher nach einem Software-Update der Aurora-Speicher-Instance zurückzuführen.

- Es wurde ein Problem behoben, das in seltenen Fällen dazu führen kann, dass die Reader-Instances neu gestartet werden.
- Es wurde ein Problem behoben, bei dem ein privilegierter Benutzer die dem Benutzer [rdsadmin](#) zugewiesenen [Ressourcenlimits](#) ändern kann. Wenn diese Ressourcenlimits falsch festgelegt werden, können sie die Fähigkeit des RDS-Monitoring-Agenten beeinträchtigen, den Zustand der Datenbankinstanz zu überwachen, was zur Nichtverfügbarkeit der Datenbank führen kann.

#### Upgrades und Migrationen:

- Es wurde ein Problem behoben, das auftrat, wenn versucht wurde, die binäre Protokollreplikation für Aurora MySQL-Cluster zu starten, die von Amazon RDS MySQL 5.7 migriert wurden und nicht unterstützte gespeicherte Prozeduren enthielten.
- Der Datenbank-Event-Scheduler wurde während eines Hauptversions-Upgrades auf Aurora MySQL Version 3 deaktiviert. Dieses Update trägt dazu bei, dass während des Upgrades der Hauptversion keine Änderungen an der Datenbank durch die Ausführung des Ereignisses auftreten.

## Integration von MySQL-Fehlerbehebungen (Community Edition)

Diese Version enthält alle Community-Bugfixes bis einschließlich 5.7.44. Weitere Informationen finden Sie unter [MySQL-Fehlerbehebungen durch Aurora-MySQL-2.x-Datenbank-Engine-Updates](#).

## In Aurora-MySQL-Version 2 nicht unterstützte Funktionen

Die folgenden Features werden derzeit in Aurora MySQL Version 2 nicht unterstützt (kompatibel mit MySQL 5.7):

- Scan-Batching

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Die SQL-Anweisung CREATE TABLESPACE
- Plugin für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- X Protocol

## Aurora-MySQL-Datenbank-Engine-Updates 28.12.2023 (Version 2.12.1, kompatibel mit MySQL 5.7.40)

Version: 2.12.1

Aurora MySQL 2.12.1 ist allgemein verfügbar. Aurora-MySQL 2.12-Versionen sind bis zu MySQL 5.7.40 kompatibel. Weitere Informationen zu Community-Änderungen finden Sie unter [Änderungen in MySQL 5.7.40 \(11.10.22, Allgemeine Verfügbarkeit\)](#).

Derzeit werden die Aurora-MySQL-Versionen 2.07.\*, 2.11.\*, 2.12.\*, 3.01.\*, 3.02.\*, 3.03.\*, 3.04.\* und 3.05.\* unterstützt.

Sie können einen vorhandenen Aurora-MySQL-2.\*-Datenbank-Cluster zu Aurora MySQL 2.12.1 aktualisieren. Sie können auch einen Snapshot von jeder derzeit unterstützten Aurora-MySQL-Version in Aurora MySQL 2.12.1 wiederherstellen.

Wenn Sie eine globale Aurora-MySQL-Datenbank zu Version 2.12.\* aktualisieren und die Schreibweiterleitung aktiviert haben, müssen Sie Ihre primären und sekundären DB-Cluster auf exakt dieselbe Version aktualisieren, einschließlich des Patch-Levels, um die Schreibweiterleitung weiterhin verwenden zu können. Weitere Informationen zum Upgrade der Nebenversion einer globalen Aurora-Datenbank finden Sie unter [Upgrades für Nebenversionen](#).

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

### Note

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Behobene Sicherheitsprobleme und CVEs sind unten aufgeführt:

Diese Version enthält alle Community-CVEs-Fixes bis einschließlich MySQL 5.7.44.

Korrekturen und andere Verbesserungen bei der Feinabstimmung der Handhabung in einer verwalteten Umgebung. Weitere CVE-Korrekturen unten:

- [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](#)
- Die Verarbeitung von Einzelzeichen-Token durch ein Parser-Plugin für die Volltextsuche (FTS) wurde behoben (Fehler 35432973)
- Es wurde ein Problem behoben, bei dem Ereignisse, die während der Verarbeitung der Audit-Protokollrotationen gemeldet wurden, möglicherweise nicht in das Audit-Protokoll geschrieben werden.

Neue Funktionen:

- Unterstützung für die Replikation von Binärprotokollen (Binlog) mit mehreren Threads wurde hinzugefügt. Dabei wendet der SQL-Thread auf dem Binärprotokollreplikant nach Möglichkeit binäre Protokollereignisse parallel an. Weitere Informationen zu den Konfigurationsoptionen zur Feinabstimmung Ihrer Multithread-Replikation finden Sie im [Aurora-Benutzerhandbuch](#).

#### Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem behoben, bei dem es bei Aurora-MySQL-Datenbank-Instances, die parallele Abfragen verwenden, zu einem Datenbankneustart kommen konnte, wenn eine große Anzahl gleichzeitiger paralleler Abfragen ausgeführt wurde.
- Es wurde ein Problem mit Sperrkonflikten behoben, die durch einen Prüfprotokoll-Thread verursacht wurden und zu einer hohen CPU-Auslastung und Timeouts für Client-Anwendungen führen können.
- Es wurde ein Problem behoben, das dazu führen kann, dass eine Datenbank-Instance neu gestartet wird, wenn versucht wird, eine Datenbankseite zu lesen, die zu einer gelöschten Tabelle gehört.
- Es wurde ein Problem behoben, das dazu führen kann, dass die Reader-Instance neu gestartet wird, wenn die Writer-Instance das Datenbank-Volumen auf ein Vielfaches von 160 GB erhöht.
- Es wurde ein Problem im Lock-Manager behoben, das zu einem Neustart oder Failover führen könnte, wenn zweiphasige Commits verarbeitet werden, bei denen die Isolationsstufe auf READ\_COMMITTED oder READ\_UNCOMMITTED festgelegt ist und entweder XA-Transaktionen verwendet werden oder das Binärprotokoll (Binlog) aktiviert ist.
- Es wurde ein Problem behoben, das dazu führen kann, dass Datenbank-Cluster nicht verfügbar sind, wenn die Writer-Instance neu gestartet wird, während die Datenbank Auslöser für interne Systemtabellen erstellt oder löscht.
- Es wurde ein Problem behoben, das dazu führen kann, dass die Datenbank-Instance neu gestartet wird, wenn sich die Anzahl der Datenbankverbindungen dem durch den `max_connections`-Parameter festgelegten Wert nähert.
- Es wurde ein Problem behoben, das dazu führen kann, dass die Aurora-Reader-Instance neu gestartet wird, wenn DML-Abfragen (Data Manipulation Language) in einer Tabelle ausgeführt werden, die einen Volltextindex enthält.
- Schnelles Einfügen ist in dieser Aurora MySQL-Version aufgrund eines Problems, das zu Inkonsistenzen bei der Ausführung von Abfragen wie `INSERT INTO`, und führen kann `SELECT`, nicht aktiviert `FROM`. Weitere Informationen zur Optimierung schneller Einfügungen finden Sie unter [Leistungsverbesserungen für Amazon Aurora MySQL](#).

## Allgemeine Verbesserungen:

- Es wurde ein Problem behoben, das dazu führen kann, dass eine parallele Abfrage aufgrund vorübergehender Netzwerkprobleme beim Lesen von Daten aus dem Aurora-Cluster-Volume fehlschlägt.
- Es wurde ein Problem im Zusammenhang mit der Verwaltung von Audit-Protokolldateien behoben, das dazu führen könnte, dass auf Protokolldateien für den Download oder die Rotation nicht zugegriffen werden kann und das in einigen Fällen die CPU-Auslastung erhöht.
- Es wurde ein Problem behoben, bei dem es bei kleinen Lesereplikat-Instances nach einem Upgrade von Versionen unter 2.11.\* zu einer erhöhten Replikationsverzögerung kommen kann.
- Es wurde ein Problem behoben, das zu einer übermäßigen Anzahl von Protokollmeldungen führen kann, wenn die [procs\\_priv-Grant-Tabelle](#) zur Überprüfung von Anfragen konsultiert wird, die gespeicherte Routinen beinhalten.
- Es wurde ein Speicherverwaltungsproblem behoben, das dazu führen kann, dass die Datenbank-Instance bei der Ausführung von Abfragen mithilfe der Hash-Join-Optimierung zu viel Speicher beansprucht.
- Es wurde ein Problem behoben, das bei Verwendung der Schreibweiterleitung in den globalen Statustabellen `information_schema` und `performance_schema` zu einem falschen Wert der Variablen `Threads_running` führen konnte.
- Es wurde ein Problem behoben, das zu einem Neustart der Datenbank führte, wenn SELECT-Anweisungen mit partitionierten Tabellen (erstellt in einer Version von MySQL, die den alten `ha_partition`-Partitions-Handler unterstützt) ausgeführt wurden und die parallele Abfrage vom Abfrageplaner ausgewählt wurde.
- Es wurde ein Problem behoben, das verhindern kann, dass neue Client-Verbindungen zur Datenbank hergestellt werden, wenn die Schreibweiterleitung aktiviert ist.
- Reduzierte Verzögerung bei der Replikation von Binärprotokollen (Binlog), wenn ein Aurora-MySQL-Binlog-Replikat QUERY-Ereignisse ausführt, die in die Binlog-Datei der Quelle geschrieben werden, ohne dass eine durch den Befehl `USE` definierte Standarddatenbank vorhanden ist.
- Es wurde ein Problem behoben, das dazu führen kann, dass die `CommitLatency CloudWatch` Metrik falsch gemeldet wird, wenn der `innodb_flush_log_at_trx_commit` Parameter nicht auf 1 gesetzt ist.
- Es wurde ein Problem behoben, das dazu führen kann, dass Datenbankverbindungen geschlossen werden, bevor sie hergestellt wurden. Dieses Problem betrifft eher Datenbank-Instances, die Verbindungen mit hoher Geschwindigkeit öffnen und schließen.



- Es wurde ein Problem behoben, das einen Datenbank-Neustart verursachen kann, wenn verbundene Binärprotokoll-Benutzer (Binlog) doppelte Binlog-Replikationsserver-IDs verwenden.

## Integration von MySQL-Fehlerbehebungen (Community Edition)

Diese Version enthält zusätzlich zu den unten aufgeführten Fehlerbehebungen alle Community-Fehlerbehebungen bis einschließlich 5.7.40. Weitere Informationen finden Sie unter [MySQL-Fehlerbehebungen durch Aurora-MySQL-2.x-Datenbank-Engine-Updates](#).

- Es wurde ein Problem behoben, das dazu führen kann, dass bestehende und neue Remote-Verbindungen blockiert werden, wenn sie gleichzeitig mit SHOW PROCESSLIST-Anweisung ausgeführt werden (Community-Fehler 34857411)
- Replikation: Einige binäre Protokollereignisse wurden nicht immer korrekt behandelt (Fehler 34617506)

## In Aurora-MySQL-Version 2 nicht unterstützte Funktionen

Die folgenden Features werden derzeit in Aurora MySQL Version 2 nicht unterstützt (kompatibel mit MySQL 5.7):

- Scan-Batching

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Die SQL-Anweisung CREATE TABLESPACE
- Plugin für die Gruppenreplikation
- Größere Seitengröße

- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- X Protocol

## Aurora-MySQL-Datenbank-Engine-Updates 25.10.2023 (Version 2.12.0.1, kompatibel mit MySQL 5.7.40) Beta

Version: 2.12.0.1

Aurora MySQL 2.12.0.1 ist allgemein in den folgenden Regionen verfügbar: USA Ost (Nord-Virginia), USA Ost (Ohio), USA West (Nordkalifornien), USA West (Oregon), AWS GovCloud (USA-Ost) und AWS GovCloud (USA-West). Dies ist eine frühe Version, die nur für Sicherheitsupdates verfügbar ist. Diese Korrekturen werden mit der nächsten Patch-Version, 2.12.1, in größerem Umfang in allen Regionen verfügbar sein. Aurora-MySQL-2.12-Versionen sind kompatibel mit MySQL 5.7.40.

Derzeit werden die Aurora-MySQL-Versionen 2.07.\*, 2.11.\*, 2.12.\*, 3.01.\*, 3.02.\*, 3.03.\* und 3.04\* unterstützt.

Sie können einen vorhandenen Aurora-MySQL-2.\*-Datenbank-Cluster zu Aurora MySQL 2.12.0.1 aktualisieren. Sie können auch einen Snapshot von jeder derzeit unterstützten Aurora-MySQL-Version zu Aurora MySQL 2.12.0.1 wiederherstellen.

Wenn Sie eine globale Aurora-MySQL-Datenbank zu Version 2.12.\* aktualisieren, müssen Sie Ihre primären und sekundären DB-Cluster auf exakt dieselbe Version aktualisieren, einschließlich des Patch-Levels. Weitere Informationen zum Upgrade der Nebenversion einer globalen Aurora-Datenbank finden Sie unter [Upgrades für Nebenversionen](#).

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Behobene Sicherheitsprobleme und CVEs sind unten aufgeführt:

Diese Version enthält alle Community-CVEs-Korrekturen bis einschließlich MySQL 5.7.40.

- [CVE-2023-38545](#)

Verbesserungen der Verfügbarkeit:

- Schnelles Einfügen ist in dieser Aurora MySQL-Version aufgrund eines Problems, das zu Inkonsistenzen bei der Ausführung von Abfragen wie INSERT INTO, und führen kannSELECT, nicht aktiviertFROM. Weitere Informationen zur Optimierung schneller Einfügungen finden Sie unter [Leistungsverbesserungen für Amazon Aurora MySQL](#).

## Aurora-MySQL-Datenbank-Engine-Updates 25.07.2023 (Version 2.12.0, kompatibel mit MySQL 5.7.40)

Version: 2.12.0

Aurora MySQL 2.12.0 ist allgemein verfügbar. Aurora-MySQL 2.12-Versionen sind bis zu MySQL 5.7.40 kompatibel. Weitere Informationen zu Community-Änderungen finden Sie unter [Änderungen in MySQL 5.7.40 \(11.10.22, Allgemeine Verfügbarkeit\)](#).

Derzeit werden die Aurora-MySQL-Versionen 2.07.\*, 2.11.\*, 2.12.\*, 3.01.\*, 3.02.\* und 3.03.\* unterstützt.

Sie können einen vorhandenen Aurora-MySQL-2.\*-Datenbank-Cluster zu Aurora MySQL 2.12.0 aktualisieren. Sie können auch einen Snapshot von jeder derzeit unterstützten Aurora-MySQL-Version in Aurora MySQL 2.12.0 wiederherstellen.

Wenn Sie eine globale Aurora-MySQL-Datenbank zu Version 2.12.\* aktualisieren und die Schreibweiterleitung aktiviert haben, müssen Sie Ihre primären und sekundären DB-Cluster auf exakt dieselbe Version aktualisieren, einschließlich des Patch-Levels, um die Schreibweiterleitung weiterhin verwenden zu können. Weitere Informationen zum Upgrade der Nebenversion einer globalen Aurora-Datenbank finden Sie unter [Upgrades für Nebenversionen](#).

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

### Note

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Behobene Sicherheitsprobleme und CVEs sind unten aufgeführt:

Diese Version enthält alle Community-CVEs-Korrekturen bis einschließlich MySQL 5.7.40.

- Die von Aurora MySQL verwendeten Standard-SSL-Verschlüsselungen wurden aktualisiert, um die weniger sicheren DES-CBC3-SHA-Werte aus dem [SSL\\_CIPHER](#)-Datenbankparameter auszuschließen. Wenn Sie aufgrund der Entfernung der DES-CBC3-SHA-Verschlüsselung auf SSL-Verbindungsprobleme stoßen, verwenden Sie bitte eine entsprechende sichere Verschlüsselung aus dieser Liste, [Konfigurieren von Verschlüsselungs-Suites für Verbindungen zu Aurora-MySQL-DB-Clustern](#). Weitere Informationen zur [Konfiguration der Verbindungsverschlüsselung](#) auf MySQL-Clients finden Sie in der MySQL-Dokumentation.
- [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](#)

Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem bei der Verschlüsselung von Datenbankaktivitätsströmen behoben, das zu Neustarts der Datenbank führen kann.
- Es wurden zwei Probleme behoben, die dazu führen können, dass ein Datenbankneustart fehlschlägt, wenn dieser während der Ausführung einer DDL-Abfrage (Data Definition Language) erfolgt.
- Es wurde ein Problem behoben, durch das Verbindungsspitzen zu einer erhöhten Abfragelatenz oder einem Neustart der Datenbank-Instance führen können.
- Es wurde ein Problem behoben, das in seltenen Fällen dazu führen kann, dass ein Aurora-Replikat neu gestartet wird, wenn große Aktualisierungsvorgänge oder Data Definition Language (DDL)-Workloads auf der Writer-Instance und Lesevorgänge für denselben Tabellensatz auf dem Aurora-Replikat gleichzeitig ausgeführt werden.
- Es wurde ein Problem behoben, bei dem Verbindungsspitzen dazu führen können, dass der Verbindungsaufbau länger dauert oder mit Timeout-Fehlern fehlschlägt.
- Es wurde ein Problem behoben, bei dem die Protokollrotation von Advanced Audit den freien Speicherplatz reduzieren kann, was dazu führen könnte, dass die Datenbank-Instance neu gestartet wird.
- Es wurde ein Problem behoben, das dazu führen kann, dass eine Aurora-MySQL-Reader-Instance neu gestartet wird, während eine Abfrage ausgeführt wird, die einen parallelen Aurora-Abfrageausführungsplan verwendet.
- Es wurde ein Problem behoben, das dazu führen kann, dass die Writer-Instance neu gestartet wird, während die OPTIMIZE TABLE-Abfrage in einer Tabelle mit einem Volltextsuche-Index (FTS) ausgeführt wird.
- Es wurde ein Problem behoben, das dazu führen kann, dass die Writer-Instance in einer primären Aurora- AWS Datenbankregion neu gestartet wird, wenn eine SELECT FOR UPDATE Abfrage mithilfe der globalen Schreibweiterleitung aus einer sekundären Aurora-Datenbankregion ausgeführt wird.
- Es wurde ein Problem behoben, das dazu führen kann, dass eine Reader-Instance der sekundären AWS Region der globalen Aurora-Datenbank, die die globale Schreibweiterleitung verwendet, neu gestartet wird, wenn eine weitergeleitete [implizite Commit-Anweisung](#) auf einen Fehler stößt.
- Schnelles Einfügen ist in dieser Aurora MySQL-Version aufgrund eines Problems, das zu Inkonsistenzen bei der Ausführung von Abfragen wie INSERT INTO, und führen kannSELECT, nicht aktiviertFROM. Weitere Informationen zur Optimierung schneller Einfügungen finden Sie unter [Leistungsverbesserungen von Amazon Aurora MySQL](#).

## Allgemeine Verbesserungen:

- Es wurden Leistungsoptimierungen bei der Dateiverwaltung für Binlog-Replikate eingeführt, um Konflikte beim Schreiben in Relay-Protokolldateien zu reduzieren.
- Es wurde ein Problem behoben, das dazu führen kann, dass der `buffer_pool_read_requests`-Zähler in den `information_schema`-Metriken falsch gemeldet wird.
- Es wurde ein Problem behoben, das dazu führen kann, dass der lokale Speicher bei der Ausführung von `LOAD FROM S3`- oder `SELECT INTO S3`-Vorgängen voll wird. Das Problem kann außerdem zu einer höheren CPU-Auslastung, Datenbankneustarts aufgrund von zu wenig Arbeitsspeicher und zu einer erhöhten Latenz bei diesen Abfragen führen.
- Es wurde ein Problem behoben, durch das es bei Datenbank-Instances, die die binäre Protokollreplikation verwenden, zu einer erhöhten CPU-Auslastung und zu Verbindungsausfällen kommen kann, wenn mehrere Benutzer für die binäre Protokollreplikation angeschlossen sind.
- Es wurde ein Problem behoben, bei dem die SSL-Serverstatusvariablen nicht gefüllt wurden.
- Es wurde ein Problem behoben, bei dem DML-Anweisungen (Data Manipulation Language), die doppelte Schreibvorgänge ausführen, zu übermäßiger Fehlerprotokollierung und erhöhten Abfragelatenzen führen können.
- Die Zeitzonendefinitionen wurden auf die Version IANA 2023c aktualisiert.
- Es wurde Unterstützung für die Aktivierung und Deaktivierung der binären Protokollierung auf Sitzungsebene hinzugefügt. Weitere Informationen finden Sie unter [Gespeicherte Prozeduren – Replizieren](#) im Amazon-Aurora-Benutzerhandbuch
- Es wurde Unterstützung für die Einstellung des binären Protokollformats auf Sitzungsebene hinzugefügt. Weitere Informationen finden Sie unter [Gespeicherte Prozeduren – Replizieren](#) im Amazon-Aurora-Benutzerhandbuch
- Es wurde ein Problem behoben, bei dem das Einstellen des `aurora_disable_hash_join`-Parameters auf 1 oder ON nicht verhindern kann, dass der Optimizer einen Hash-Join verwendet.
- Es wurde ein Problem mit Index-Scans behoben, bei dem möglicherweise ein falsches Ergebnis zurückgegeben wurde, wenn eine `SELECT`-Abfrage ausgeführt wurde, bei der die `GROUP BY`-Klausel verwendet und der `aurora_parallel_query`-Parameter zu ON geändert wurde.
- Es wurde ein Problem behoben, das in seltenen Fällen dazu führen kann, dass eine Amazon-Aurora-Reader-Instance neu gestartet wird, wenn auf eine Tabelle zugegriffen wird, in der umfangreiche Update- oder Data Definition Language (DDL)-Operationen gleichzeitig auf der Writer-Instance ausgeführt werden.

- Es wurde ein Problem behoben, das dazu führen kann, dass der `buffer_pool_read_requests`-Zähler in den `information_schema`-Metriken falsch gemeldet wird.
- Es wurde ein Problem behoben, das dazu führen kann, dass ein Binlog-Replikat neu gestartet wird, wenn die Systemvariable [server uuid](#) der Quelle fehlt oder einen ungültigen Wert hat.
- Es wurde ein Problem behoben, das verhindert hat, dass InnoDB-Statistiken veralten, was manchmal zu einem suboptimalen Abfrageausführungsplan und damit zu einer Verlängerung der Abfrageausführungszeit führen kann.
- Es wurde ein Problem behoben, bei dem die `AuroraGlobalDBRPOLag` CloudWatch Metriken unabhängig von der Benutzer-Workload immer Null anzeigten.

#### Upgrades und Migrationen:

- Zur Durchführung eines Nebenversions-Upgrades für eine globale Aurora-Datenbank von Aurora MySQL Version 2.07 oder 2.11 zu Aurora MySQL Version 2.12 oder höher siehe [Aktualisierung von Aurora MySQL durch Ändern der Engine-Version](#).

## Integration von MySQL-Fehlerbehebungen (Community Edition)

Diese Version enthält zusätzlich zu den unten aufgeführten Fehlerbehebungen alle Community-Fehlerbehebungen bis einschließlich 5.7.40. Weitere Informationen finden Sie unter [MySQL-Fehlerbehebungen durch Aurora-MySQL-2.x-Datenbank-Engine-Updates](#).

- Es wurde ein Problem behoben, das aufgrund der Rotation von TLS-Zertifikaten im Hintergrund zu einer höheren CPU-Auslastung führen kann (Community-Fehlerbehebung #34284186)

## In Aurora-MySQL-Version 2 nicht unterstützte Funktionen

Die folgenden Features werden derzeit in Aurora MySQL Version 2 nicht unterstützt (kompatibel mit MySQL 5.7):

- Scan-Batching.

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Die SQL-Anweisung CREATE TABLESPACE
- Plugin für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- X Protocol

## Aurora MySQL-Datenbank-Engine-Updates 2024-03-26 (Version 2.11.5, kompatibel mit MySQL 5.7.12) Standard

Version: 2.11.5

Aurora MySQL 2.11.5 ist allgemein verfügbar. Aurora-MySQL-2.11-Versionen sind kompatibel mit MySQL 5.7.12. Weitere Informationen zu Community-Änderungen finden Sie unter [Änderungen in MySQL 5.7.12 \(11.04.2016, Allgemeine Verfügbarkeit\)](#).

Die derzeit unterstützten Aurora MySQL-Versionen sind 2.07.9, 2.07.10, 2.11.\*, 2.12.\*, 3.01.\*, 3.02.\*, 3.03.\*, 3.04.\*, 3.05.\* und 3.06.\*.



Sie können einen vorhandenen Aurora MySQL 2.\*-Datenbankcluster auf Aurora MySQL 2.11.5 aktualisieren. Sie können auch einen Snapshot von jeder aktuell unterstützten niedrigeren Version von Aurora MySQL Version 2 in Aurora MySQL 2.11.5 wiederherstellen.

Wenn Sie eine globale Aurora-MySQL-Datenbank auf Version 2.11.\* aktualisieren, müssen Sie Ihre primären und sekundären DB-Cluster auf exakt dieselbe Version aktualisieren, einschließlich des Patch-Levels. Weitere Informationen zum Upgrade der Nebenversion einer globalen Aurora-Datenbank finden Sie unter [Upgrades für Nebenversionen](#).

Unmittelbar nach der Durchführung eines In-Place-Engine-Versions-Upgrades zu Aurora MySQL 2.11.\* wird automatisch ein Betriebssystem-Upgrade auf alle betroffenen Instances der DB-Instance-Klassen db.r4, db.r5, db.t2 und db.t3 angewendet, sofern auf den Instances eine alte Betriebssystemversion ausgeführt wird. In einem Multi-AZ-DB-Cluster wenden alle Reader-Instances zuerst das Betriebssystem-Upgrade an. Wenn das Betriebssystem-Upgrade in der ersten Reader-Instance abgeschlossen ist, erfolgt ein Failover und die vorherige Writer-Instance wird aktualisiert.

#### Note

Das Betriebssystem-Upgrade wird bei Hauptversions-Upgrades nicht automatisch auf die globalen Aurora-Datenbanken angewendet.

#### Note

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Sicherheitsprobleme und CVEs wurden behoben:

Die folgenden CVE-Fixes sind in dieser Version enthalten:

- [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](#)

#### Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem behoben, bei dem eine Aurora MySQL Writer-DB-Instance aufgrund eines Defekts in der Komponente, die mit dem Aurora-Speicher kommuniziert, ein Failover durchführen kann. Der Fehler ist auf einen Ausfall der Kommunikation zwischen der DB-Instance und dem zugrunde liegenden Speicher nach einem Softwareupdate zurückzuführen.
- Es wurde ein Problem behoben, das in seltenen Fällen dazu führen kann, dass die Reader-DB-Instances neu gestartet werden.
- Es wurde ein Problem mit Sperrkonflikten behoben, die durch einen Prüfprotokoll-Thread verursacht wurden und zu einer hohen CPU-Auslastung und Timeouts für Client-Anwendungen führen können.

#### Allgemeine Verbesserungen:

- Es wurde ein Problem behoben, das dazu führen kann, dass eine parallel Abfrage aufgrund vorübergehender Netzwerkprobleme beim Lesen von Daten aus dem Aurora-DB-Cluster-Volume fehlschlägt.
- Es wurde ein Problem im Zusammenhang mit der Verwaltung von Audit-Logdateien behoben, das dazu führen kann, dass auf Protokolldateien für den Download oder die Rotation nicht zugegriffen werden kann und das in einigen Fällen die CPU-Auslastung erhöht.
- Es wurde ein Problem behoben, das bei Verwendung der Schreibweiterleitung zu einem falschen Wert der `Threads_running` Variablen in den Tabellen `information_schema` und den `performance_schema` globalen Statustabellen führen kann.

#### Upgrades und Migrationen:

- Es wurde ein Problem behoben, das die Initiierung der binären Protokollreplikation auf Aurora MySQL-DB-Clustern verhinderte, die von RDS für MySQL 5.7 migriert wurden.
- Der Datenbank-Event-Scheduler wurde bei Hauptversions-Upgrades auf Aurora MySQL Version 3 deaktiviert. Auf diese Weise können Änderungen an der Datenbank durch die Ausführung eines Ereignisses während des Upgrades der Hauptversion vermieden werden.

## In Aurora-MySQL-Version 2 nicht unterstützte Funktionen

Die folgenden Features werden derzeit in Aurora MySQL Version 2 nicht unterstützt (kompatibel mit MySQL 5.7):

- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung CREATE TABLESPACE

# Aurora-MySQL-Datenbank-Engine-Updates 17.10.2023 (Version 2.11.4, kompatibel mit MySQL 5.7.12)

Version: 2.11.4

Aurora MySQL 2.11.4 ist allgemein verfügbar. Aurora-MySQL-2.11-Versionen sind kompatibel mit MySQL 5.7.12. Weitere Informationen zu Community-Änderungen finden Sie unter [Änderungen in MySQL 5.7.12 \(11.04.2016, Allgemeine Verfügbarkeit\)](#).

Derzeit werden die Aurora-MySQL-Versionen 2.07.9, 2.07.10, 2.11.\*, 2.12.\*, 3.01.\*, 3.02.\*, 3.03.\* und 3.04.\* unterstützt.

Sie können einen vorhandenen Aurora-MySQL-2.\*-Datenbank-Cluster zu Aurora MySQL 2.11.4 aktualisieren. Sie können auch einen Snapshot von jeder derzeit unterstützten Aurora-MySQL-Version in Aurora MySQL 2.11.4 wiederherstellen.

Wenn Sie eine globale Aurora-MySQL-Datenbank auf Version 2.11.\* aktualisieren, müssen Sie Ihre primären und sekundären DB-Cluster auf exakt dieselbe Version aktualisieren, einschließlich des Patch-Levels. Weitere Informationen zum Upgrade der Nebenversion einer globalen Aurora-Datenbank finden Sie unter [Upgrades für Nebenversionen](#).

Unmittelbar nach der Durchführung eines In-Place-Engine-Versions-Upgrades zu Aurora MySQL 2.11.\* wird automatisch ein Betriebssystem-Upgrade auf alle betroffenen Instances der DB-Instance-Klassen db.r4, db.r5, db.t2 und db.t3 angewendet, sofern auf den Instances eine alte Betriebssystemversion ausgeführt wird. In einem Multi-AZ-DB-Cluster wenden alle Reader-Instances zuerst das Betriebssystem-Upgrade an. Wenn das Betriebssystem-Upgrade in der ersten Reader-Instance abgeschlossen ist, erfolgt ein Failover und die vorherige Writer-Instance wird aktualisiert.

## Note

Das Betriebssystem-Upgrade wird bei Hauptversions-Upgrades nicht automatisch auf die globalen Aurora-Datenbanken angewendet.

**Note**

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Behobene Sicherheitsprobleme und CVEs sind unten aufgeführt:

- Es wurde ein Problem behoben, bei dem Ereignisse, die während der Verarbeitung von Audit-Protokoll-Rotationen gemeldet wurden, möglicherweise nicht in das Audit-Protokoll geschrieben wurden.
- [CVE-2022-24407](#)

Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem behoben, bei dem es bei Aurora-MySQL-Datenbank-Instances, die parallele Abfragen verwenden, zu einem Datenbankneustart kommen kann, wenn eine große Anzahl gleichzeitiger paralleler Abfragen ausgeführt wird.
- Es wurde ein Problem behoben, das dazu führen kann, dass eine Datenbank-Instance neu gestartet wird, während E/A-intensive Lese-Workloads ausgeführt werden.
- Es wurde ein Problem behoben, das dazu führen kann, dass eine Datenbank-Instance neu gestartet wird, wenn versucht wird, eine Datenbankseite zu lesen, die zu einer gelöschten Tabelle gehört.
- Es wurde ein Problem behoben, das dazu führen kann, dass die Reader-Instance neu gestartet wird, wenn die Writer-Instance das Datenbank-Volumen auf ein Vielfaches von 160 GB erhöht.
- Es wurde ein Problem behoben, das dazu führen kann, dass der Datenbank-Cluster nicht mehr verfügbar ist, wenn die Writer-Instance neu gestartet wird, während die Datenbank Auslöser für interne Systemtabellen erstellt oder löscht.

- Es wurde ein Problem behoben, das dazu führen kann, dass eine Reader-Instance neu gestartet wird, wenn DML-Abfragen (Data Manipulation Language) in einer Tabelle ausgeführt werden, die einen Volltextindex enthält.
- Es wurde ein Problem behoben, das dazu führen kann, dass eine Reader-Instance neu gestartet wird, während eine Abfrage ausgeführt wird, die einen parallelen Aurora-Abfrageausführungsplan verwendet.
- Es wurde ein Problem behoben, das dazu führen kann, dass die Writer-Instance neu gestartet wird, während die `OPTIMIZE TABLE`-Abfrage in einer Tabelle mit einem Volltextsuche-Index (FTS) ausgeführt wird.
- Das schnelle Einfügen ist in dieser Aurora MySQL-Version nicht aktiviert, da ein Problem zu Inkonsistenzen beim Ausführen von Abfragen wie `INSERT INTOSELECT`, und führen kann. Weitere Informationen zur Fast-Insert-Optimierung finden Sie unter [Amazon Aurora MySQL Performance Enhancements](#).

#### Allgemeine Verbesserungen:

- Es wurde ein Problem behoben, bei dem es bei kleinen Lesereplikant-Instances nach einem Upgrade von Versionen unter 2.11.\* zu einer erhöhten Replikationsverzögerung kommen kann.
- Es wurde ein Problem behoben, das zu einer übermäßigen Anzahl von Protokollmeldungen führen kann, wenn die [Grant-Tabelle `procs\_priv`](#) zur Überprüfung von Anfragen konsultiert wird, die gespeicherte Routinen beinhalten.
- Es wurde ein Speicherverwaltungsproblem behoben, das dazu führen kann, dass die Datenbank-Instance bei der Ausführung von Abfragen mithilfe der Hash-Join-Optimierung zu viel Speicher beansprucht.
- Es wurde ein Problem behoben, das zu einem Neustart der Datenbank führte, wenn `SELECT`-Anweisungen mit partitionierten Tabellen (erstellt in einer Version von MySQL, die den alten `ha_partition`-Partitions-Handler unterstützt) ausgeführt wurden und die parallele Abfrage vom Abfrageplaner ausgewählt wurde.
- Es wurde ein Problem behoben, das verhindern kann, dass neue Client-Verbindungen zur Datenbank hergestellt werden, wenn die Schreibweiterleitung aktiviert ist.
- Reduzierte Verzögerung bei der Replikation von Binärprotokollen (Binlog), wenn ein Aurora-MySQL-Binlog-Replikant `QUERY`-Ereignisse ausführt, die in die Binlog-Datei der Quelle geschrieben werden, ohne dass eine durch den Befehl `USE` definierte Standarddatenbank vorhanden ist.

- Es wurde ein Problem mit Index-Scans behoben, bei dem möglicherweise ein falsches Ergebnis zurückgegeben wurde, wenn eine SELECT-Abfrage ausgeführt wurde, bei der die GROUP BY-Klausel verwendet und der `aurora_parallel_query`-Parameter zu ON geändert wurde.
- Es wurde Unterstützung für die Aktivierung und Deaktivierung der binären Protokollierung auf Sitzungsebene hinzugefügt. Siehe [Gespeicherte Prozeduren – Replizieren](#) im Amazon-Aurora-Benutzerhandbuch.
- Es wurde ein Problem behoben, das dazu führen kann, dass ein Binlog-Replikat neu gestartet wird, wenn die Systemvariable `server_uuid` der Quelle fehlt oder einen ungültigen Wert hat.
- Es wurde Unterstützung für die Einstellung des binären Protokollformats auf Sitzungsebene hinzugefügt. Siehe [Gespeicherte Prozeduren – Replizieren](#) im Amazon-Aurora-Benutzerhandbuch.
- Es wurde ein Problem behoben, das dazu führen kann, dass die CommitLatency CloudWatch Metrik falsch gemeldet wird, wenn der `innodb_flush_log_at_trx_commit` Parameter nicht auf 1 gesetzt ist.
- Es wurde ein Problem behoben, das verhindert hat, dass InnoDB-Statistiken veralten, was manchmal zu einem suboptimalen Abfrageausführungsplan und damit zu einer Verlängerung der Abfrageausführungszeit führen kann.
- Es wurde ein Problem behoben, das einen Datenbank-Neustart verursachen kann, wenn verbundene Binärprotokoll-Benutzer (Binlog) doppelte Binlog-Replikationsserver-IDs verwenden.

## Integration von MySQL-Fehlerbehebungen (Community Edition)

Diese Version enthält alle Community-Fehlerbehebungen bis einschließlich 5.7.12, zusätzlich zu den unten aufgeführten. Weitere Informationen finden Sie unter [MySQL-Fehlerbehebungen durch Aurora-MySQL-2.x-Datenbank-Engine-Updates](#).

- Replikation: Einige binäre Protokollereignisse wurden nicht immer korrekt behandelt. (Fehler #34617506)
- Es wurde ein Problem behoben, das aufgrund einer TLS-Zertifikatsrotation im Hintergrund zu einer höheren CPU-Auslastung führen kann (Community-Fehler-Behebung #34284186).
- In vorbereiteten Anweisungen konnten einige Arten von Unterabfragen einen Serverabbruch verursachen. (Fehler #33100586)

## In Aurora-MySQL-Version 2 nicht unterstützte Funktionen

Die folgenden Features werden derzeit in Aurora MySQL Version 2 nicht unterstützt (kompatibel mit MySQL 5.7):

- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`

## Aurora-MySQL-Datenbank-Engine-Updates 09.06.2023 (Version 2.11.3, kompatibel mit MySQL 5.7.12)

Version: 2.11.3



Aurora MySQL 2.11.3 ist allgemein verfügbar. Aurora-MySQL-2.11-Versionen sind kompatibel mit MySQL 5.7.12. Weitere Informationen zu Community-Änderungen finden Sie unter [Änderungen in MySQL 5.7.12 \(11.04.2016, Allgemeine Verfügbarkeit\)](#).

Die derzeit unterstützten Aurora-MySQL-Versionen sind 2.07.\*, 2.11.\*, 3.01.\*, 3.02.\* und 3.03.\*.

Sie können einen vorhandenen Aurora-MySQL-2.\*-Datenbank-Cluster zu Aurora MySQL 2.11.3 aktualisieren. Sie können auch einen Snapshot von jeder derzeit unterstützten Aurora-MySQL-Version in Aurora MySQL 2.11.3 wiederherstellen.

Wenn Sie eine globale Aurora-MySQL-Datenbank auf Version 2.11.\* aktualisieren, müssen Sie Ihre primären und sekundären DB-Cluster auf exakt dieselbe Version aktualisieren, einschließlich des Patch-Levels. Weitere Informationen zum Upgrade der Nebenversion einer globalen Aurora-Datenbank finden Sie unter [Upgrades für Nebenversionen](#).

Unmittelbar nach der Durchführung eines In-Place-Engine-Versions-Upgrades zu Aurora MySQL 2.11.\* wird automatisch ein Betriebssystem-Upgrade auf alle betroffenen Instances der DB-Instance-Klassen db.r4, db.r5, db.t2 und db.t3 angewendet, sofern auf den Instances eine alte Betriebssystemversion ausgeführt wird. In einem Multi-AZ-DB-Cluster wenden alle Reader-Instances zuerst das Betriebssystem-Upgrade an. Wenn das Betriebssystem-Upgrade in der ersten Reader-Instance abgeschlossen ist, erfolgt ein Failover und die vorherige Writer-Instance wird aktualisiert.

#### Note

Das Betriebssystem-Upgrade wird bei Hauptversions-Upgrades nicht automatisch auf die globalen Aurora-Datenbanken angewendet.

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

#### Note

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Behobene Sicherheitsprobleme und CVEs sind unten aufgeführt:

- Die von Aurora MySQL verwendeten Standard-SSL-Verschlüsselungen wurden aktualisiert, um die weniger sicheren DES-CBC3-SHA-Werte aus dem Datenbankparameter [SSL\\_CIPHER](#) auszuschließen. Wenn aufgrund der Entfernung der DES-CBC3-SHA Verschlüsselung SSL-Verbindungsprobleme auftreten, verwenden Sie bitte eine entsprechende sichere Verschlüsselung aus dieser Liste, [ConfiguringCipherSuites](#). Weitere Informationen zur [Konfiguration der Verbindungsverschlüsselung](#) auf MySQL-Clients finden Sie in der MySQL-Dokumentation.
- [CVE-2023-21963](#)
- [CVE-2023-21912](#)
- [CVE-2023-0215](#)
- [CVE-2022-43551](#)
- [CVE-2022-37434](#)

Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem bei der Verschlüsselung von DAS-Ereignissen (Database Activity Streams) behoben, das zu Datenbankneustarts führen kann.
- Es wurden zwei Probleme behoben, die dazu führen können, dass ein Datenbankneustart fehlschlägt, wenn er während der Ausführung einer DDL-Abfrage (Data Definition Language) auftritt.
- Schnelles Einfügen ist in dieser Aurora MySQL-Version aufgrund eines Problems, das zu Inkonsistenzen bei der Ausführung von Abfragen wie `INSERT INTO`, und führen kann `SELECT`, nicht aktiviert `FROM`. Weitere Informationen zur Optimierung schneller Einfügungen finden Sie unter [Leistungsverbesserungen für Amazon Aurora MySQL](#).

Allgemeine Verbesserungen:

- Es wurden Leistungsoptimierungen bei der Dateiverwaltung für Binärprotokoll-Replikate eingeführt, um Konflikte beim Schreiben in Relay-Protokolldateien zu reduzieren.
- Es wurde ein Problem behoben, bei dem das Setzen des `aurora_disable_hash_join`-Parameters auf 1 oder ON den Optimizer möglicherweise nicht daran hindert, einen Hash-Join zu verwenden.

- Es wurde ein Problem behoben, das dazu führen kann, dass der `buffer_pool_read_requests`-Zähler in den `information_schema`-Metriken falsch gemeldet wird.
- Es wurde ein Problem behoben, das dazu führen kann, dass der lokale Speicher bei der Ausführung von `LOAD FROM S3-` oder `SELECT INTO S3-`Vorgängen voll wird. Das Problem kann außerdem zu einer höheren CPU-Auslastung, Datenbankneustarts aufgrund von zu wenig Arbeitsspeicher und zu einer erhöhten Latenz bei diesen Abfragen führen.

## In Aurora-MySQL-Version 2 nicht unterstützte Funktionen

Die folgenden Features werden derzeit in Aurora MySQL Version 2 nicht unterstützt (kompatibel mit MySQL 5.7):

- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen

- Replikationsfilter
- Die SQL-Anweisung CREATE TABLESPACE

## Aurora-MySQL-Datenbank-Engine-Updates 24.03.2023 (Version 2.11.2, kompatibel mit MySQL 5.7.12)

Version: 2.11.2

Aurora MySQL 2.11.2 ist allgemein verfügbar. Aurora-MySQL-2.11-Versionen sind kompatibel mit MySQL 5.7.12. Weitere Informationen zu Community-Änderungen finden Sie unter [Änderungen in MySQL 5.7.12 \(11.04.2016, Allgemeine Verfügbarkeit\)](#).

Die derzeit unterstützten Aurora-MySQL-Versionen sind 2.07.\*, 2.11.\*, 3.01.\*, 3.02.\* und 3.03.\*.

Sie können einen vorhandenen Aurora-MySQL-2.\*-Datenbank-Cluster zu Aurora MySQL 2.11.2 aktualisieren. Sie können auch einen Snapshot von jeder derzeit unterstützten Aurora-MySQL-Version in Aurora MySQL 2.11.2 wiederherstellen.

Wenn Sie eine globale Aurora-MySQL-Datenbank auf Version 2.11.\* aktualisieren, müssen Sie Ihre primären und sekundären DB-Cluster auf exakt dieselbe Version aktualisieren, einschließlich des Patch-Levels. Weitere Informationen zum Upgrade der Nebenversion einer globalen Aurora-Datenbank finden Sie unter [Upgrades für Nebenversionen](#).

Unmittelbar nach der Durchführung eines In-Place-Engine-Versions-Upgrades zu Aurora MySQL 2.11.\* wird automatisch ein Betriebssystem-Upgrade auf alle betroffenen Instances der DB-Instance-Klassen db.r4, db.r5, db.t2 und db.t3 angewendet, sofern auf den Instances eine alte Betriebssystemversion ausgeführt wird. In einem Multi-AZ-DB-Cluster wenden alle Reader-Instances zuerst das Betriebssystem-Upgrade an. Wenn das Betriebssystem-Upgrade in der ersten Reader-Instance abgeschlossen ist, erfolgt ein Failover und die vorherige Writer-Instance wird aktualisiert.

### Note

Das Betriebssystem-Upgrade wird bei Hauptversions-Upgrades nicht automatisch auf die globalen Aurora-Datenbanken angewendet.

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

#### Note

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Allgemeine Verbesserungen:

- Es wurde ein Problem behoben, bei dem es bei Datenbank-Instances, die die binäre Protokollreplikation verwenden, zu einer erhöhten CPU-Auslastung und zu Verbindungsausfällen kommen kann, wenn mehrere Benutzer für die binäre Protokollreplikation angeschlossen sind.
- Es wurde ein Problem behoben, das dazu führen kann, dass eine Reader-Instance in einer sekundären globalen Datenbankregion nach dem Upgrade auf Aurora MySQL Version 2.11 nicht mehr synchron ist, wenn der primäre Datenbank-Writer auf Aurora MySQL Version 2.10 läuft.

Verbesserungen der Verfügbarkeit:

- Schnelles Einfügen ist in dieser Aurora MySQL-Version aufgrund eines Problems, das zu Inkonsistenzen bei der Ausführung von Abfragen wie `INSERT INTO`, und führen kann `SELECT`, nicht aktiviert `FROM`. Weitere Informationen zur Optimierung schneller Einfügungen finden Sie unter [Leistungsverbesserungen für Amazon Aurora MySQL](#).

## In Aurora-MySQL-Version 2 nicht unterstützte Funktionen

Die folgenden Features werden derzeit in Aurora MySQL Version 2 nicht unterstützt (kompatibel mit MySQL 5.7):

- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`

## Aurora-MySQL-Datenbank-Engine-Updates 14.02.2023 (Version 2.11.1, kompatibel mit MySQL 5.7.12)

Version: 2.11.1

Aurora MySQL 2.11.1 ist allgemein verfügbar. Aurora-MySQL-2.11-Versionen sind kompatibel mit MySQL 5.7.12. Weitere Informationen zu Community-Änderungen finden Sie unter [Änderungen in MySQL 5.7.12 \(11.04.2016, Allgemeine Verfügbarkeit\)](#).

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.07.\*, 2.09.\*, 2.10.\*, 2.11.\*, 3.01.\* und 3.02.\* unterstützt.

Sie können einen vorhandenen Aurora-MySQL-2.\*-Datenbank-Cluster zu Aurora MySQL 2.11.1 aktualisieren. Bei Clustern mit Aurora MySQL Version 1 können Sie einen bestehenden Cluster mit

Aurora MySQL 1.23 oder höher direkt zu 2.11.1 aktualisieren. Sie können auch einen Snapshot von jeder derzeit unterstützten Aurora-MySQL-Version in Aurora MySQL 2.11.1 wiederherstellen.

Wenn Sie eine globale Aurora MySQL-Datenbank auf Version 2.11.\* aktualisieren und die Schreibweiterleitung aktiviert haben, müssen Sie Ihre primären und sekundären DB-Cluster auf exakt dieselbe Version aktualisieren, einschließlich des Patch-Levels, um die Schreibweiterleitung weiterhin verwenden zu können. Weitere Informationen zum Upgrade der Nebenversion einer globalen Aurora-Datenbank finden Sie unter [Upgrades für Nebenversionen](#).

Unmittelbar nach der Durchführung eines In-Place-Engine-Versions-Upgrades zu Aurora MySQL 2.11.\* wird automatisch ein Betriebssystem-Upgrade auf alle betroffenen Instances der DB-Instance-Klassen db.r4, db.r5, db.t2 und db.t3 angewendet, sofern auf den Instances eine alte Betriebssystemversion ausgeführt wird. In einem Multi-AZ-DB-Cluster wenden alle Reader-Instances zuerst das Betriebssystem-Upgrade an. Wenn das Betriebssystem-Upgrade in der ersten Reader-Instance abgeschlossen ist, erfolgt ein Failover und die vorherige Writer-Instance wird aktualisiert.

#### Note

Das Betriebssystem-Upgrade wird bei Hauptversions-Upgrades nicht automatisch auf die globalen Aurora-Datenbanken angewendet.

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

#### Note

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Behobene Sicherheitsprobleme und CVEs sind unten aufgeführt:

Korrekturen und andere Verbesserungen bei der Feinabstimmung der Handhabung in einer verwalteten Umgebung. Weitere CVE-Korrekturen unten:

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

#### Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem behoben, bei dem Verbindungsspitzen zu einer erhöhten Abfragelatenz oder einem Neustart der Datenbank-Instance führen können.
- Es wurde ein Problem behoben, das in seltenen Fällen dazu führen kann, dass ein Aurora-Replikat neu gestartet wird, wenn große Aktualisierungsvorgänge oder Data Definition Language (DDL)-Workloads auf der Writer-Instance und Lesevorgänge für denselben Tabellensatz auf dem Aurora-Replikat gleichzeitig ausgeführt werden.
- Es wurde ein Problem behoben, bei dem Verbindungsspitzen dazu führen können, dass der Verbindungsaufbau länger dauert oder mit Timeout-Fehlern fehlschlägt.
- Es wurde ein Problem behoben, bei dem die Protokollrotation von Advanced Audit den freien Speicherplatz reduzieren kann, was dazu führen kann, dass die Datenbank-Instance neu gestartet wird.
- Schnelles Einfügen ist in dieser Aurora MySQL-Version aufgrund eines Problems, das zu Inkonsistenzen bei der Ausführung von Abfragen wie `INSERT INTO`, und führen kann `SELECT`, nicht aktiviert `FROM`. Weitere Informationen zur Optimierung schneller Einfügungen finden Sie unter [Leistungsverbesserungen von Amazon Aurora MySQL](#).

#### Allgemeine Verbesserungen:

- Es wurde ein Problem behoben, bei dem die [SSL-Serverstatusvariablen](#) nicht gefüllt wurden.
- Es wurde ein Problem behoben, bei dem DML-Anweisungen (Data Manipulation Language), die doppelte Schreibvorgänge ausführen, zu übermäßiger Fehlerprotokollierung und erhöhten Abfragelatenzen führen könnten.

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon-Aurora-MySQL-Funktionen werden in Aurora-MySQL-Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora-MySQL-Version 2 (MySQL 5.7 kompatibel) unterstützt.



- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen zum synchronen Aufrufen von AWS Lambda Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung CREATE TABLESPACE

# Aurora-MySQL-Datenbank-Engine-Updates 25.10.2022 (Version 2.11.0, kompatibel mit MySQL 5.7.12). Diese Version ist für neue Erstellungen nicht verfügbar.

Version: 2.11.0

Aurora MySQL 2.11.0 ist allgemein verfügbar. Aurora-MySQL-2.x-Versionen sind kompatibel mit MySQL 5.7.12. Weitere Informationen zu Community-Änderungen finden Sie unter [Änderungen in MySQL 5.7.12 \(11.04.2016\)](#), Allgemeine Verfügbarkeit).

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 2.11\*, 3.01.\* und 3.02.\* unterstützt.

Sie können einen vorhandenen Aurora-MySQL-2.\*-Datenbank-Cluster zu Aurora MySQL 2.11.0 aktualisieren. Bei Clustern mit Aurora MySQL Version 1 können Sie einen bestehenden Cluster mit Aurora MySQL 1.23 oder höher direkt zu 2.11.0 aktualisieren. Sie können auch einen Snapshot von jeder derzeit unterstützten Aurora-MySQL-Version in Aurora MySQL 2.11.0 wiederherstellen.

Wenn Sie eine globale Aurora MySQL-Datenbank auf Version 2.11.\* aktualisieren und die Schreibweiterleitung aktiviert haben, müssen Sie Ihre primären und sekundären DB-Cluster auf exakt dieselbe Version aktualisieren, einschließlich des Patch-Levels, um die Schreibweiterleitung weiterhin verwenden zu können. Weitere Informationen zum Upgrade der Nebenversion einer globalen Aurora-Datenbank finden Sie unter [Upgrades für Nebenversionen](#).

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Note

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Behobene Sicherheitsprobleme und CVEs sind unten aufgeführt:

Korrekturen und andere Verbesserungen bei der Feinabstimmung der Handhabung in einer verwalteten Umgebung. Weitere CVE-Korrekturen unten:

- [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](#)

Neue Funktionen:

- Mit der Veröffentlichung von Aurora MySQL Version 2.11 ist ein neues Betriebssystem-Upgrade verfügbar. Wir empfehlen, dass Sie dieses ausstehende Betriebssystemupdate nach dem

Upgrade auf Version 2.11 auf alle Ihre Aurora-MySQL-Datenbank-Instances anwenden. Weitere Informationen finden Sie unter Arbeiten mit [Betriebssystemaktualisierungen](#).

- Eine neue dynamische Konfigurationsoption, `innodb_deadlock_detect`, kann verwendet werden, um die Deadlock-Erkennung zu deaktivieren. Auf Systemen mit hoher Parallelität kann die Deadlock-Erkennung zu einer Verlangsamung führen, wenn zahlreiche Threads auf dieselbe Sperre warten. Manchmal kann es effizienter sein, die Deadlock-Erkennung zu deaktivieren und sich auf die `innodb_lock_wait_timeout`-Einstellung für das Rollback von Transaktionen zu stützen, wenn ein Deadlock auftritt. (Fehler #23477773) Weitere Informationen zur Innodb-Deadlock-Erkennung finden Sie in der [MySQL-Dokumentation](#).
- Die Funktionen `UUID_TO_BIN`, `BIN_TO_UUID` und `IS_UUID` aus MySQL 8.0 wurden hinzugefügt. Weitere Informationen zur Verwendung dieser Funktionen finden Sie unter [MySQL – verschiedene Funktionen](#).
- Es wurde Unterstützung für Optimierer-Hinweise hinzugefügt, die es dem Benutzer ermöglichen, Aurora-MySQL-Parallelabfragen pro Tabelle oder pro Abfrage zu aktivieren oder zu deaktivieren.
  - [Arbeiten mit paralleler Abfrage Query für Amazon Aurora MySQL](#)
  - [Aurora-MySQL-Hinweise](#)
- Die Unterstützung für den R3-Instanztyp wurde entfernt.
- Es werden jetzt R6i-Instances unterstützt.

#### Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem behoben, das die regionsübergreifende logische Replikation in einem Datenbank-Cluster verhindern kann, weil eine falsche Binlog-Datei und Position in die Fehlerprotokolle geschrieben wurden. Dieses Problem kann auftreten, wenn die Engine nach der Ausführung einer DDL-Anweisung neu gestartet wird.
- Es wurde ein Problem behoben, das unter seltenen Bedingungen dazu führen kann, dass Aurora-Reader-Instances neu gestartet werden, wenn Access-Control List (ACL)-Anweisungen wie `GRANT` und `FLUSH` auf der Writer-Instance neu gestartet werden. Dieses Problem betrifft eher Reader-Instances mit einer großen Anzahl von Benutzern und ACL-Vorgängen (z. B. Berechtigungsänderungen).
- Es wurde ein Problem behoben, das unter seltenen Bedingungen dazu führen kann, dass die Writer-Instance neu gestartet oder ein Failover durchgeführt wird, wenn eine Transaktion auf eine Zeile zugreift, die durch eine andere Transaktion gelöscht wurde.

- Die Leistung der Volltext-Phrasensuche wurde verbessert, um den Zeitaufwand für die Suche nach Phrasen in einer Tabelle mit Volltextindizes erheblich zu reduzieren.
- Es wurde ein Problem behoben, bei dem eine Writer-Instance nach dem Neustart bei der langsamen Wiederherstellung hängen bleibt und anschließend erneut gestartet wird. Dieses Problem tritt auf, wenn sich zum Zeitpunkt des ersten Neustarts eine große Anzahl nicht festgeschriebener Zeilen in der Datenbank befindet.
- Es wurde ein Problem behoben, das in seltenen Fällen dazu führen kann, dass der Datenbank-Server aufgrund einer langen Semaphor-Wartezeit neu gestartet wird, wenn der [Deadlock-Detector-Thread](#) stecken bleibt.
- Es wurde ein Problem behoben, das in seltenen Fällen dazu führen kann, dass die Datenbank aufgrund einer langen Semaphor-Wartezeit neu gestartet wird, wenn E/A-Threads blockiert werden.
- Schnelles Einfügen ist in dieser Aurora MySQL-Version aufgrund eines Problems, das zu Inkonsistenzen bei der Ausführung von Abfragen wie INSERT INTO, und führen kannSELECT, nicht aktiviertFROM. Weitere Informationen zur Optimierung schneller Einfügungen finden Sie unter [Leistungsverbesserungen für Amazon Aurora MySQL](#).

#### Allgemeine Verbesserungen:

- Es wurde ein Problem behoben, das dazu führen kann, dass der Datenbankserver neu gestartet wird, wenn alle der folgenden Bedingungen erfüllt sind:
  - ALLOW\_INVALID\_DATES ist im SQL MODE deaktiviert.
  - Der Datenbankserver verarbeitet eine INSERT-, UPDATE-, DELETE- oder SELECT-Anweisung mit einem ungültigen Wert vom Typ DATETIME, so dass der Monat nicht zwischen 1 und 12 liegt.
- Es wurde ein Problem behoben, bei dem die Aufbewahrungsfrist für Binärprotokolle nicht eingehalten wurde, wenn Log-Bin auf OFF gesetzt war, was zu einer höheren Speicherauslastung als erwartet führte. Nach dieser Korrektur werden die Binärprotokolle entsprechend Ihrer Aufbewahrungsdauer gelöscht. Weitere Informationen zur Konfiguration des Aufbewahrungszeitraums für Binärprotokolle finden Sie im [Aurora-MySQL-Benutzerhandbuch](#).
- Es wurde ein Problem behoben, das dazu führen kann, dass der freie Speicherplatz auf der Datenbank-Instance reduziert wird, wenn bestimmte Data Control Language (DCL)-SQL-Anweisungen wie GRANT, FLUSH PRIVILEGES usw. auf dieser Instance ausgeführt werden. Die häufige Verwendung solcher Anweisungen kann dazu führen, dass der freisetzbare Speicher weiter reduziert wird, und kann dazu führen, dass die Datenbank-Instance aufgrund von out-of-

- memory Problemen neu gestartet wird. Die Verwendung solcher Anweisungen auf der Writer-Instance kann auch dazu führen, dass der freie Speicher auf den Reader-Instances reduziert wird.
- Es wurde ein größerer Lesebuffer für Lesevorgänge aus den Relay-Protokollen eingeführt, um die Anzahl der E/A-Lesevorgänge zu minimieren, wodurch Konflikte zwischen den E/A- und SQL-Threads reduziert werden.
  - Es wurde ein Problem behoben, das dazu führen kann, dass die gespeicherte Prozedur `mysql.rds_rotate_slow_log` mit der Fehlermeldung „Die Tabelle 'mysql.slow\_log\_backup' existiert nicht“ fehlschlägt.
  - Es wurde ein Problem behoben, bei dem eine übermäßige Invalidierung des Abfrage-Caches zu einer höheren CPU-Auslastung und zu Latenzen auf dem Lesereplikat als erwartet führt, da das Lesereplikat die Daten von der Festplatte statt aus dem Abfrage-Cache lesen muss.
  - Es wurde ein Problem behoben, das es Benutzern ermöglichte, die Befehle `INSTALL PLUGIN` und `UNINSTALL PLUGIN` auf einer Reader-Instance auszuführen, was zu einem Deadlock bei `Lock_Plugin`, `Lock_System_Variables_Hash` und `Lock_Global_System_Variables` führen kann. Diese Anweisungen können jetzt nur auf der Writer-Instance in einem Datenbank-Cluster ausgeführt werden.
  - Es wurde ein Problem behoben, durch das in Clustern beim Aktivieren der Binärprotokollierung gelegentlich eine höhere Commit-Latenz erreicht wurde. Dies betrifft alle Transaktionen, die große Binlog-Ereignisse (mit einer Größe von über 500 MB) generieren.
  - Es wurde ein Problem behoben, das dazu führen kann, dass die Metrik `trx_active_transactions` in der Tabelle `INFORMATION_SCHEMA.INNODB_METRICS` einen falschen Wert enthält.
  - Es wurde ein Problem behoben, durch das die logische Replikation gestoppt werden kann, wenn die Binlog-Datei beim Ausführen eines Rollbacks zum Savepoint für eine große Transaktion inkonsistent wird.
  - Maskierte Hashes für Anmeldeinformationen im allgemeinen Protokoll `slow-query-log` und Audit-Protokoll standardmäßig mit einem konsistenten Maskierungs-Secret. Dies ist über den Parameter `aurora_mask_password_hashes_type` konfigurierbar.
  - Es wurde ein Problem behoben, bei dem die Dauer des Zero-Downtime-Restart (ZDR) in den vom Kunden beobachteten Ereignissen falsch angegeben wird.
  - Es wurde ein Problem behoben, das dazu führen kann, dass Aufrufe von [mysql\\_rds\\_import\\_binlog\\_ssl\\_material](#) mit dem [MySQL-Serverfehler 1457](#) fehlschlagen.
  - Es wurde ein Problem behoben, durch das die Initialisierung von Dump-Threads mit dem Thread zum Löschen von Binärprotokollen blockiert werden kann. Dadurch kann verhindert werden, dass

die aktive Binlog-Datei rotiert und stattdessen weiter wächst oder Probleme mit neuen Binlog-Replikativverbindungen auftreten.

- Es wurde ein Problem behoben, durch das der Abfrage-Cache veraltete Ergebnisse für eine Reader-Instance zurückgeben kann.

## Integration von MySQL-Fehlerbehebungen (Community Edition)

Diese Version enthält alle Community-Fehlerbehebungen bis einschließlich 5.7, zusätzlich zu den unten aufgeführten. Weitere Informationen finden Sie unter [MySQL-Fehlerbehebungen durch Aurora-MySQL-2.x-Datenbank-Engine-Updates](#).

- Es wurde ein Problem behoben, bei dem der Code zum Lesen von Zeichensatzinformationen aus den Ereignistabellen von Performance-Schema-Anweisungen (z. B. `events_statements_current`) das gleichzeitige Schreiben in diese Zeichensatzinformationen nicht verhindert. Aufgrund dieses Problems kann der Zeichensatz für den SQL-Abfragetext möglicherweise ungültig sein, was zu einem Serverabbruch führen kann. Mit dieser Korrektur führt ein ungültiger Zeichensatz zu einer Kürzung der `SQL_TEXT`-Spalte und verhindert Serverabbrüche. (Fehler #23540008)
- InnoDB: Backport einer Korrektur für Community-Fehler #25189192, Fehler #84038. Es wurde ein Problem behoben, bei dem InnoDB nach einer `RENAME TABLE`-Operation, bei der eine Tabelle in ein anderes Schema verschoben wurde, die Datenwörterbuchtafel `INNODB_SYS_DATAFILES` nicht aktualisieren konnte. Dies führte zu einem Fehler beim Neustart, der darauf hinwies, dass die Tablespace-Datendatei nicht gefunden werden konnte.
- InnoDB: Es wurde ein Problem behoben, bei dem der Server beim Hinzufügen eines neuen Indexes einen intern definierten Fremdschlüsselindex löschte und versuchte, einen sekundären Index, der für eine virtuell generierte Spalte definiert ist, als Fremdschlüsselindex zu verwenden, was zu einem Serverabsturz führte. InnoDB erlaubt nun einer Fremdschlüsseleinschränkung, auf einen sekundären Index zu verweisen, der für eine virtuell generierte Spalte definiert ist. (Fehler #23533396)
- Es wurde ein Problem behoben, bei dem zwei Sitzungen, die gleichzeitig eine `INSERT... ON DUPLICATE KEY UPDATE`-Operation ausführten, ein Deadlock generierten. Während eines teilweisen Rollbacks eines Tupels konnte es in einer anderen Sitzung aktualisiert werden. Der Fix für diesen Fehler macht die Korrekturen für Fehler #11758237, Fehler #17604730 und Fehler #20040791 rückgängig. (Fehler #25966845)
- Backport einer Korrektur für den Community-Fehler #27407480: Es wurde ein Problem behoben, bei dem die Rechte `EXECUTE` und `ALTER ROUTINE` den Erstellern von Routinen nicht korrekt erteilt wurden, selbst wenn `automatic_sp_privileges` aktiviert war.

- Backport einer Korrektur für Community-Fehler #24671968: Es wurde ein Problem behoben, bei dem eine Abfrage falsche Ergebnisse liefern konnte, wenn die WHERE-Klausel eine abhängige Unterabfrage enthielt, die Tabelle einen sekundären Index für die Spalten in der Auswahlliste hatte, gefolgt von den Spalten in der Unterabfrage, und GROUP BY oder DISTINCT erlaubte, einen Loose Index Scan zu verwenden.
- Es wurde ein Problem behoben, bei dem die Replikation unterbrochen wurde, wenn eine Löschanweisung für mehrere Tabellen mit Fremdschlüsseln ausgegeben wurde. (Fehler #80821)
- Es wurde ein Problem behoben, bei dem in Sonderfällen bestimmte Slave-Fehler auch dann nicht ignoriert wurden, wenn [slave\\_skip\\_errors](#) aktiviert war. In Fällen, in denen das Öffnen und Sperren einer Tabelle fehlschlug oder Feldkonvertierungen auf einem Server fehlschlagen, auf dem die zeilenbasierte Replikation ausgeführt wird, wird der Fehler als kritisch angesehen und der Status von [slave\\_skip\\_errors](#) wird ignoriert. Die Korrektur stellt sicher, dass bei aktivierter Option [slave\\_skip\\_errors](#) alle Fehler, die beim Anwenden einer Transaktion gemeldet wurden, korrekt behandelt werden. (Fehler #70640, Fehler #17653275)
- Es wurde ein Problem behoben, bei dem eine [SET PASSWORD](#)-Anweisung von einem MySQL 5.6-Master auf einen MySQL 5.7-Slave oder von einem MySQL 5.7-Master, bei dem die Systemvariable [log\\_builtin\\_as\\_identified\\_by\\_password](#) auf ON gesetzt war, auf einen MySQL 5.7-Slave repliziert wurde. Der Passwort-Hash selbst wurde ebenfalls ghasht, bevor er auf dem Slave gespeichert wurde. Das Problem ist jetzt behoben und der replizierte Passwort-Hash wird so gespeichert, wie er ursprünglich an den Slave übergeben wurde. (Fehler #24687073)
- Es wurde ein Problem behoben, bei dem die Serialisierung eines JSON-Werts, der aus einem großen Unterdokument besteht, das in viele Ebenen von JSON-Arrays, Objekten oder beidem eingeschlossen ist, manchmal übermäßig viel Zeit in Anspruch nahm. (Fehler #23031146)
- Anweisungen, die nicht analysiert werden können (z. B. aufgrund von Syntaxfehlern), werden nicht mehr in das Protokoll für langsame Abfragen geschrieben. (Fehler #33732907)

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon-Aurora-MySQL-Funktionen werden in Aurora-MySQL-Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora-MySQL-Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.



- Native Funktionen zum synchronen Aufrufen von AWS Lambda Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`

## Aurora-MySQL-Datenbank-Engine-Updates 01.11.2022 (Version 2.10.3) (veraltet)

## Version: 2.10.3

Aurora MySQL 2.10.3 ist allgemein verfügbar. Aurora-MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora-MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 2.11\*, 3.01.\* und 3.02.\* unterstützt.

Sie können einen vorhandenen Aurora-MySQL-2.\*-Datenbank-Cluster zu Aurora MySQL 2.10.3 aktualisieren. Bei Clustern mit Aurora MySQL Version 1 können Sie einen bestehenden Cluster mit Aurora MySQL 1.23 oder höher direkt zu 2.10.3 aktualisieren. Sie können auch einen Snapshot von jeder derzeit unterstützten Aurora-MySQL-Version in Aurora MySQL 2.10.3 wiederherstellen.

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

### Note

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Behobene Sicherheitsprobleme und CVEs sind unten aufgeführt:

Korrekturen und andere Verbesserungen bei der Feinabstimmung der Handhabung in einer verwalteten Umgebung. Weitere CVE-Korrekturen unten:

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

Allgemeine Verbesserungen:

- Es wurde ein Problem behoben, das unter seltenen Bedingungen dazu führen kann, dass der Datenbank-Server aufgrund einer langen Semaphor-Wartezeit neu gestartet wird, wenn der [Deadlock-Detector-Thread](#) stecken bleibt.
- Es wurde ein Problem behoben, das dazu führen kann, dass der freie Speicherplatz auf der Datenbank-Instance reduziert wird, wenn bestimmte Data Control Language (DCL)-SQL-Anweisungen wie GRANT, FLUSH PRIVILEGES usw. auf dieser Instance ausgeführt werden. Die häufige Verwendung solcher Anweisungen kann dazu führen, dass der freie Speicher immer kleiner wird und die Datenbank-Instance aufgrund von Speicherproblemen neu gestartet wird. Die Verwendung solcher Anweisungen auf der Writer-Instance kann auch dazu führen, dass der freie Speicher auf den Reader-Instances reduziert wird.
- Es wurde ein Problem behoben, bei dem Abfragen der Tabelle „performance\_schema.events\_waits\_summary\_global\_by\_event\_name“ langsam werden können, wenn eine Datenbank-Instance stark ausgelastet und das performance\_schema-Warteereignis „wait/io/aurora\_respond\_to\_client“ aktiviert ist.
- Es wurde ein Problem behoben, das unter seltenen Bedingungen dazu führen kann, dass der Datenbank-Server angehalten und neu gestartet wird, wenn ein teilweises Rollback von Transaktionen aufgrund einer Beschränkungsverletzung bei den Sekundärindizes erreicht wird.
- Es wurde ein Problem behoben, das unter seltenen Bedingungen dazu führen kann, dass die Writer-Instance neu gestartet oder ein Failover durchgeführt wird, wenn eine Transaktion auf eine Zeile zugreift, die durch eine andere Transaktion gelöscht wurde.
- Es wurde ein Problem behoben, das unter seltenen Bedingungen dazu führen kann, dass die Datenbank aufgrund einer langen Semaphor-Wartezeit neu gestartet wird, wenn E/A-Threads blockiert werden.
- Es wurde ein Problem behoben, das in seltenen Fällen dazu führen kann, dass das Lesereplikat während eines Failovers neu gestartet wird, wenn die Unix-Socket-Lock-Datei verwendet wird.
- Es wurde ein Problem behoben, bei dem eine übermäßige Invalidierung des Abfrage-Caches zu einer höheren CPU-Auslastung und zu Latenzen auf dem Lesereplikat als erwartet führt, da das Lesereplikat die Daten von der Festplatte statt aus dem Abfrage-Cache lesen muss.

## Integration von MySQL-Fehlerbehebungen (Community Edition)

Diese Version enthält alle Community-Fehlerbehebungen bis einschließlich 5.7, zusätzlich zu den unten aufgeführten. Weitere Informationen finden Sie unter [MySQL-Fehlerbehebungen durch Aurora-MySQL-2.x-Datenbank-Engine-Updates](#).

- Es wurde ein Problem behoben, bei dem der Code zum Lesen von Zeichensatzinformationen aus den Ereignistabellen von Performance-Schema-Anweisungen (z. B. `events_statements_current`) das gleichzeitige Schreiben in diese Zeichensatzinformationen nicht verhindert. Aufgrund dieses Problems kann der Zeichensatz für den SQL-Abfragetext möglicherweise ungültig sein, was zu einem Serverabbruch führen kann. Mit dieser Korrektur führt ein ungültiger Zeichensatz zu einer Kürzung der `SQL_TEXT`-Spalte und verhindert Serverabbrüche. (Fehler #23540008)
- Es wurde ein Problem behoben, bei dem für ein `UPDATE` eine temporäre Tabelle mit einem Primärschlüssel von mehr als 1024 Byte erforderlich war und diese Tabelle mit InnoDB erstellt wurde. Der Server konnte dann beendet werden. (Fehler #25153670)
- Es wurde ein Problem behoben, bei dem zwei Sitzungen, die gleichzeitig eine `INSERT... ON DUPLICATE KEY UPDATE`-Operation ausführten, ein Deadlock generierten. Während eines teilweisen Rollbacks eines Tupels konnte es in einer anderen Sitzung aktualisiert werden. Der Fix für diesen Fehler macht die Korrekturen für Fehler #11758237, Fehler #17604730 und Fehler #20040791 rückgängig. (Fehler #25966845)

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon-Aurora-MySQL-Funktionen werden in Aurora-MySQL-Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora-MySQL-Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen für den synchronen Aufruf von AWS Lambda-Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native

Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`

## Aurora-MySQL-Datenbank-Engine-Updates 26.01.2022 (Version 2.10.2) (veraltet)

Version: 2.10.2

Aurora MySQL 2.10.2 ist allgemein verfügbar. Aurora-MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora-MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Sie können einen vorhandenen Aurora-MySQL-2.\*-Datenbank-Cluster zu Aurora MySQL 2.10.0 aktualisieren. Für Cluster, auf denen Aurora MySQL Version 1 ausgeführt wird, können Sie einen vorhandenen Cluster Aurora MySQL 1.23 oder höher direkt auf 2.10.0 aktualisieren. Sie können einen Snapshot aus einer derzeit unterstützten Aurora MySQL-Version zu Aurora MySQL 2.10.0 wiederherstellen.

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

### Note

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Behobene Sicherheitsprobleme und CVEs sind unten aufgeführt:

Korrekturen und andere Verbesserungen bei der Feinabstimmung der Handhabung in einer verwalteten Umgebung. Weitere CVE-Korrekturen unten:

- [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](#)

Allgemeine Verbesserungen:

- Die Leistung wurde optimiert, um die IO-Latenz der Datenbank in 24XL-Instance-Klassen zu reduzieren.
- Unterstützung für ECDHE SSL-Chiffren hinzugefügt. Weitere Informationen zur Konfiguration Ihrer Clients für die Verwendung dieser SSL-Chiffren finden Sie in der folgenden MySQL-Dokumentation: [Chiffren für verschlüsselte Verbindungsprotokolle](#)

- Sicherheitsprobleme im Zusammenhang mit der Aurora-MySQL-Integration mit anderen AWS-Services wie Amazon S3, Amazon ML und AWS Lambda wurden behoben.
- Es wurde ein Problem behoben, das dazu führen kann, dass ein Neustart der Datenbank-Instance fehlschlug, wenn die Datenbank über ca. 1 GB an Benutzer- und Berechtigungskombinationen verfügt.
- Es wurde ein Problem mit Parallel Query behoben, durch das die Datenbank falsche Gruppierungen oder Sortierreihenfolge zurückgab, wenn Abfragen mit einer GROUP BY-Klausel und einer WHERE-Klausel mit einem Bereichsprädikat ausgeführt wurden.
- Es wurde ein Problem behoben, das dazu führt, dass general\_log- und slow\_log-Tabellen nach dem In-Place-Upgrade von Aurora MySQL 1.x (kompatibel mit MySQL 5.6) auf Aurora MySQL 2.x (kompatibel mit MySQL 5.7) unzugänglich wurden.
- Es wurde ein Problem behoben, das in seltenen Fällen dazu führt, dass die Datenbank-Instance neu gestartet wird, wenn innodb\_trx-, innodb\_locks- oder innodb\_lockwaits-Tabellen abgefragt werden, während die Datenbank unter starkem Workload steht. Überwachungstools wie Performance Insights können solche Tabellen abfragen.
- Es wurde ein Problem behoben, durch das der Wert einer TIMESTAMP-Spalte einer vorhandenen Zeile auf den neuesten Zeitstempel aktualisiert wurde, wenn alle folgenden Bedingungen erfüllt sind:
  1. Für die Tabelle existiert ein Auslöser.
  2. Ein INSERT wird für die Tabelle ausgeführt, die eine ON DUPLICATE KEY UPDATE-Klausel enthält.
  3. Die eingefügte Zeile verursacht eine Verletzung doppelter Werte in einem UNIQUE-Index oder PRIMÄRSCHLÜSSEL.
  4. Eine oder mehrere Spalten sind vom Datentyp TIMESTAMP und haben einen Standardwert von CURRENT\_TIMESTAMP.
- Es wurde ein Problem behoben, das in seltenen Fällen verhindern konnte, dass ein Binlog-Replikat eine Verbindung zu einer Instance mit aktiviertem Binlog herstellt.
- Es wurde ein Problem behoben, durch das Transaktionen unter seltenen Bedingungen keinen Commit ausführen konnten, wenn sie auf einer Instance mit aktiviertem Binlog ausgeführt wurden.
- Es wurde ein Problem behoben, durch das keine neuen Verbindungen zu einer Instance mit aktiviertem Binlog hergestellt werden konnten.
- Es wurde ein Problem behoben, das zu einer übermäßigen internen Protokollierung führen kann, wenn das Patchen und Neustarten ohne Ausfallzeiten versucht wurde, wodurch der lokale Speicher aufgefüllt wurde.

- Es wurde ein Problem behoben, das dazu führen kann, dass ein Binlog-Replikat mit einem HA\_ERR\_FOUND\_DUPP\_KEY-Fehler stoppt, wenn bestimmte DDL- und DCL-Anweisungen repliziert wurden. Das Problem tritt auf, wenn die Quell-Instance mit dem gemischten Binärprotokollierungsformat und READ COMMITTED oder READ UNCOMMITTED Isolationsstufe konfiguriert.
- Es wurde ein Problem behoben, durch das der I/O-Thread der Binlog-Replikation nicht mit der primären Instance Schritt halten konnte, wenn die Multi-Thread-Replikation aktiviert war
- Es wurde ein Problem behoben, durch das unter seltenen Bedingungen eine hohe Anzahl aktiver Verbindungen zur Datenbank-Instance dazu führen konnte, dass die Metrik CloudWatch CommitLatency falsch gemeldet wurde.
- Es wurde ein Problem behoben, durch das sich der lokale Speicher auf Graviton-Instances füllte, wenn LOAD FROM S3 oder SELECT INTO S3 ausgeführt wurde.
- Es wurde ein Problem behoben, das zu falschen Abfrageergebnissen führen konnte, wenn eine Tabelle mit einem Fremdschlüssel abgefragt wurde und beide der folgenden Bedingungen erfüllt waren:
  1. Abfrage-Cache ist aktiviert
  2. Eine Transaktion mit einem kaskadierenden Löschen oder Update für diese Tabelle wird zurückgesetzt
- Es wurde ein Problem behoben, das unter seltenen Bedingungen dazu führen kann, dass Aurora-Reader-Instances neu gestartet wurden. Die Wahrscheinlichkeit, dass dieses Problem auftritt, steigt mit zunehmender Anzahl der Transaktions-Rollbacks.
- Es wurde ein Problem behoben, durch das die Anzahl der Mutex-Vorkommen von 'LOCK\_epoch\_id\_master' in Performance Schema zunimmt, wenn eine Sitzung geöffnet und geschlossen wird.
- Es wurde ein Problem behoben, das zu einer zunehmenden Anzahl von Deadlocks für Workloads führen kann, bei denen viele Transaktionen dieselbe Reihe von Zeilen gleichzeitig aktualisieren.
- Es wurde ein Problem behoben, das unter seltenen Bedingungen dazu führen kann, dass die Instances neu gestartet werden, wenn das Datenbank-Volume ein Vielfaches von 160 GB erreicht.
- Es wurde ein Problem mit Parallel Query behoben, durch das die Datenbank beim Ausführen von SQL-Anweisungen mit einer LIMIT-Klausel neu gestartet wurde.
- Es wurde ein Problem behoben, das unter seltenen Bedingungen dazu führen kann, dass die Datenbank-Instance bei Verwendung von XA-Transaktionen mit der Isolationsstufe READ COMMITTED neu gestartet wurde.



- Es wurde ein Problem behoben, durch das nach dem Neustart einer Aurora-Read-Instance möglicherweise erneut gestartet wurde, wenn während des Neustarts ein hoher DDL-Workload vorliegt.
- Es wurde ein Fehler mit falscher Meldung der Aurora-Reader-Replikationsverzögerung behoben.
- Es wurde ein Problem behoben, das unter seltenen Bedingungen dazu führen kann, dass eine Writer-Instance neu gestartet wurde, wenn eine Überprüfung der Datenintegrität im Speicher fehlschlägt.
- Es wurde ein Problem behoben, das unter seltenen Bedingungen für das Diagramm „Database Load“ in Performance Insights (PI)-Sitzungen fälschlicherweise eine aktive CPU-Nutzung anzeigte, obwohl die Sitzungen mit der Verarbeitung fertig und im Leerlauf waren.
- Es wurde ein Problem behoben, das unter seltenen Bedingungen dazu führen konnte, dass der Datenbank-Server neu gestartet wurde, wenn ein Abfrage-Cache verarbeitet wurde.
- Es wurde ein Problem behoben, das unter seltenen Bedingungen dazu führen konnte, dass die Writer-Instance in einem primären Global-Database-Cluster aufgrund einer Race-Bedingung während der Global-Database-Replikation neu gestartet wurde.
- Es wurde ein Problem, behoben, das während eines Neustarts einer Datenbank-Instance auftreten konnte, das mehr als einen Neustart verursachen kann.

## Integration von MySQL-Fehlerbehebungen (Community Edition)

- Es wurde ein Problem in InnoDB behoben, bei dem ein Codefehler im Zusammenhang mit Tabellenstatistiken eine Assertion in der Quelldatei dict0stats.cc auslöste. (Fehler #24585978)
- Es wurde ein Problem behoben, wobei ein sekundärer Index über eine virtuelle Spalte beschädigt wurde, wenn der Index online erstellt wurde. Bei [UPDATE](#)-Anweisungen beheben wir dies wie folgt: Wenn der virtuelle Spaltenwert des Indexdatensatzes auf NULL festgelegt ist, generieren wir diesen Wert aus dem Clusterindexdatensatz. (Fehler #30556595)
- Es wurde ein Problem in InnoDB behoben, bei dem das Löschen markierter Zeilen eine externe Lesesperre verursachen konnte, bevor ein teilweiser Rollback abgeschlossen wurde. Die externe Lesesperre verhinderte die Umwandlung einer impliziten Sperre in eine explizite Sperre während des teilweisen Rollbacks, was zu einem Assert-Fehler führte. (Fehler #29195848)
- Es wurde ein Problem behoben, durch das die leeren Hostnamen in Konten die Funktion des Servers beeinträchtigen konnten. (Fehler #28653104)
- Es wurde ein Problem in InnoDB behoben, durch das eine Abfrageunterbrechung während einer Sperrwartezeit einen Fehler verursachen kann. (Fehler #28068293)

- Es wurde Problem bei der Replikation behoben, wobei Doppelspieltransaktionen manchmal den Slave-Applier blockieren konnten, wenn die Transaktionsisolationsstufe auf [REPEATABLE READ](#) festgelegt wurde. (Bug #25040331)
- Es wurde ein Problem behoben, das dazu führen kann, dass Binlog-Replikate aufgrund eines Sperrwartzeitlimits zum Stillstand kamen. (Fehler #27189701)

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon-Aurora-MySQL-Funktionen werden in Aurora-MySQL-Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora-MySQL-Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen für den synchronen Aufruf von AWS Lambda-Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten

- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`

## Aurora-MySQL-Datenbank-Engine-Updates 21.10.2021 (Version 2.10.1) (veraltet)

Version: 2.10.1

Aurora MySQL 2.10.1 ist allgemein verfügbar. Aurora-MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora-MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Sie können einen vorhandenen Aurora-MySQL-2.\*-Datenbank-Cluster zu Aurora MySQL 2.10.0 aktualisieren. Für Cluster, auf denen Aurora MySQL Version 1 ausgeführt wird, können Sie einen vorhandenen Cluster Aurora MySQL 1.23 oder höher direkt auf 2.10.0 aktualisieren. Sie können einen Snapshot aus einer derzeit unterstützten Aurora MySQL-Version zu Aurora MySQL 2.10.0 wiederherstellen.

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

### Note

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Behobene Sicherheitsprobleme und CVEs sind unten aufgeführt:

Korrekturen und andere Verbesserungen bei der Feinabstimmung der Handhabung in einer verwalteten Umgebung. Weitere CVE-Korrekturen unten:

- [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](#)

Verbesserungen der Verfügbarkeit:

- Es wurde die Möglichkeit hinzugefügt, den Cluster für zukünftige größere Versions-Upgrades sauber herunterzufahren.

Allgemeine Verbesserungen:

- Es wurde ein Problem behoben, das aufgrund der übermäßigen Protokollierung von Informationsmeldungen in internen Diagnoseprotokolldateien zu einem hohen CPU-Verbrauch bei den Reader-Instances führen konnte.
- Es wurde ein Problem behoben, durch das der Wert einer TIMESTAMP-Spalte einer vorhandenen Zeile auf den neuesten Zeitstempel aktualisiert wurde, wenn alle folgenden Bedingungen erfüllt sind:
  1. Für die Tabelle existiert ein Auslöser.
  2. Ein INSERT wird für die Tabelle ausgeführt, die eine ON DUPLICATE KEY UPDATE-Klausel enthält.

3. Die eingefügte Zeile verursacht eine Verletzung doppelter Werte in einem UNIQUE-Index oder PRIMÄRSCHLÜSSEL.
  4. Eine oder mehrere Spalten sind vom Datentyp TIMESTAMP und haben einen Standardwert von CURRENT\_TIMESTAMP.
- Es wurde ein in Version 2.10.0 eingeführtes Problem behoben, das dazu führte, dass die Verwendung der json\_merge-Funktion in bestimmten Fällen einen Fehlercode auslöste. Insbesondere wenn die json\_merge-Funktion in einer DDL verwendet wird, die generierte Spalten enthält, kann sie den Fehlercode 1305 zurückgeben.
  - Es wurde ein Problem behoben, bei dem Lesereplikate unter seltenen Bedingungen neu gestartet wurden, wenn der Aktualisierungsverlauf eines großen Objekts für die Leseansicht einer Transaktion im Read-Replikat validiert wurde.
  - Es wurde ein Problem behoben, das unter seltenen Bedingungen dazu führte, dass eine Writer-Instance neu gestartet wurde, wenn eine Überprüfung der Datenintegrität im Speicher fehlschlägt.

## Integration von MySQL-Fehlerbehebungen (Community Edition):

- CURRENT\_TIMESTAMP ERZEUGT NULLEN IM AUSLÖSER. (Fehler #25209512)

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon-Aurora-MySQL-Funktionen werden in Aurora-MySQL-Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora-MySQL-Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen für den synchronen Aufruf von AWS Lambda-Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`

## Aurora-MySQL-Datenbank-Engine-Updates 25.05.2021 (Version 2.10.0) (veraltet)

Version: 2.10.0

Aurora MySQL 2.10.0 ist allgemein verfügbar. Aurora - 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora-MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Sie können einen vorhandenen Aurora-MySQL-2.\*-Datenbank-Cluster zu Aurora MySQL 2.10.0 aktualisieren. Für Cluster, auf denen Aurora MySQL Version 1 ausgeführt wird, können Sie einen vorhandenen Cluster Aurora MySQL 1.23 oder höher direkt auf 2.10.0 aktualisieren. Sie können

einen Snapshot aus einer derzeit unterstützten Aurora MySQL-Version zu Aurora MySQL 2.10.0 wiederherstellen.

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

#### Note

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Behobene Sicherheitsprobleme und CVEs sind unten aufgeführt:

Korrekturen und andere Verbesserungen bei der Feinabstimmung der Handhabung in einer verwalteten Umgebung. Weitere CVE-Korrekturen unten:

- [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](#)

## Neue Funktionen:

- Die `db.t3.large` Instance-Klasse wird jetzt unterstützt für Aurora MySQL.
- Binäre Protokoll-Replikation:
  - Einführung des Binlog-I/O-Cache zur Verbesserung der Binlog-Leistung durch Reduzierung der Konflikte zwischen Writer-Threads und Dump-Threads. Weitere Informationen finden Sie unter [Optimieren der binären Protokollreplikation](#) im Amazon-Aurora-Benutzerhandbuch.
  - In [Aurora MySQL-Version 2.08](#) haben wir eine verbesserte Binärprotokoll-Verarbeitung (binlog) eingeführt, um die Wiederherstellungszeit nach einem Absturz und die Latenzzeit für die Commit-Zeit bei sehr großen Transaktionen zu reduzieren. Diese Verbesserungen werden jetzt für Cluster unterstützt, bei denen GTID aktiviert ist.
- Verbesserte Verfügbarkeit von Reader-Instances:
  - Zuvor wurden, als eine Writer-Instance neu gestartet wurde, auch alle Reader-Instances in einem Aurora MySQL-Cluster neu gestartet. Mit dem heutigen Start stellen In-Region-Reader-Instances während eines Neustarts der Writer-Instance weiterhin Leseanfragen bereit, was die Leseverfügbarkeit im Cluster verbessert. Weitere Informationen finden Sie unter [Neustart eines Aurora-MySQL-Clusters \(Version 2.10 und höher\)](#) im Amazon-Aurora-Benutzerhandbuch.

### Important

Nach dem Upgrade auf Aurora MySQL 2.10 führt ein Neustart der Writer-Instance nicht zu einem Neustart des gesamten Clusters. Wenn Sie den gesamten Cluster neu starten möchten, starten Sie jetzt alle Reader-Instances im Cluster neu, nachdem Sie die Writer-Instance neu gestartet haben.

- Die Leistung der Read-Ahead-Seiten wurde verbessert, die durch die logische Read-Ahead-(LRA)-Technik angefordert wurden. Dies geschah durch Stapeln der mehrseitigen Lesevorgänge in einer einzigen Anfrage, die an den Aurora Speicher gesendet wurde. Infolgedessen werden die Abfragen, die die LRA-Optimierung verwenden, bis zu dreifach schneller ausgeführt.
- Neustart und Patches ohne Ausfallzeiten:
  - Verbesserter Neustart ohne Ausfallzeiten (ZDR) und Patchen ohne Ausfallzeiten (ZDP), um ZDR und ZDP in einem breiteren Spektrum von Szenarien zu ermöglichen, einschließlich der zusätzlichen Unterstützung für Fälle, in denen die binäre Protokollierung aktiviert ist. Außerdem wurde die Sichtbarkeit von ZDR- und ZDP-Ereignissen verbessert. Einzelheiten finden Sie in der Dokumentation: [Zero-Downtime Restart \(ZDR\) für Amazon Aurora MySQL](#) und [Verwenden von Zero-Downtime-Patching](#) im Amazon-Aurora-Benutzerhandbuch.



## Verbesserungen der Verfügbarkeit:

- Verbesserungen für einen schnelleren Start, wenn die Datenbank über eine große Anzahl temporärer Indizes und Tabellen verfügt, die während einer zuvor unterbrochenen DDL-Aktivität erstellt wurden.
- Es wurden mehrere Probleme im Zusammenhang mit wiederholten Neustarts während der Wiederherstellung nach einem Ausfall von unterbrochenen DDL-Vorgängen behoben, z. B. DROP TRIGGER, ALTER TABLE und speziell ALTER TABLE, das die Art der Partitionierung oder die Anzahl der Partitionen in einer Tabelle ändert.
- Es wurde ein Problem behoben, das zu einem Serverneustart während der Protokollverarbeitung von Database Activity Streams (DAS) führen konnte.
- Es wurde ein Problem beim Drucken einer Fehlermeldung beim Verarbeiten einer ALTER Abfrage für Systemtabellen behoben.

## Allgemeine Verbesserungen:

- Es wurde ein Problem behoben, durch das der Abfrage-Cache veraltete Ergebnisse für eine Reader-Instance zurückgeben konnte.
- Es wurde ein Problem behoben, durch das einige Aurora Commit-Metriken nicht aktualisiert wurden, als die Systemvariable `innodb_flush_log_at_trx_commit` auf 0 oder 2 festgelegt wurde.
- Es wurde ein Problem behoben, durch das ein im Abfrage-Cache gespeichertes Abfrageergebnis nicht durch Transaktionen mit mehreren Anweisungen aktualisiert wurde.
- Es wurde ein Problem behoben, das dazu führen konnte, dass der zuletzt geänderte Zeitstempel der binären Protokolldateien nicht korrekt aktualisiert wurde. Dies könnte dazu führen, dass binäre Protokolldateien vorzeitig gelöscht werden, bevor die vom Kunden konfigurierte Aufbewahrungsfrist erreicht wird.
- Falscher gemeldeter Binlog-Dateiname und Position von InnoDB nach der Wiederherstellung nach einem Ausfall behoben.
- Es wurde ein Problem behoben, das dazu führen konnte, dass große Transaktionen zu falschen Binärprotokollereignissen führen konnten, wenn `binlog_checksum`-Parameter auf NONE eingestellt wurde.
- Es wurde ein Problem behoben, durch das ein Binlog-Replikat mit einem Fehler gestoppt wird, wenn die replizierte Transaktion eine DDL-Anweisung und eine große Anzahl von Zeilenänderungen enthält.

- Es wurde ein Problem behoben, das zu einem Neustart in einer Reader-Instance beim Löschen einer Tabelle führte.
- Es wurde ein Problem behoben, das dazu führte, dass Open-Source-Konnektoren fehlschlugen, wenn versucht wurde, eine Binlog-Datei mit einer großen Transaktion zu verbrauchen.
- Es wurde ein Problem behoben, das zu falschen Abfrageergebnissen für die große Geometriespalte führen konnte, nachdem ein räumlicher Index für die Tabelle mit den großen Geometriewerten erstellt wurde.
- Die Datenbank erstellt jetzt den temporären Tablespace während des Neustarts neu, wodurch der zugehörige Speicherplatz freigegeben und zurückgewonnen werden kann.
- Es wurde ein Problem behoben, das verhinderte, dass `performance_schema` Tabellen auf Aurora Reader-Instances abgeschnitten wurden.
- Es wurde ein Problem behoben, das dazu führte, dass ein Binlog-Replikat mit einem `HA_ERR_KEY_NOT_FOUND`-Fehler stoppt.
- Es wurde ein Problem behoben, durch das die Datenbank beim Ausführen der `FLUSH TABLES WITH READ LOCK` Anweisung neu gestartet wurde.
- Es wurde ein Problem behoben, das die Verwendung von Sperrfunktionen auf Benutzerebene bei Aurora Reader-Instances verhinderte.

## Integration von MySQL-Fehlerbehebungen (Community Edition):

- Verschachtelte Transaktionen konnten manchmal den Replikat-Applier blockieren, wenn die Transaktionsisolationstufe auf [REPEATABLE READ](#) eingestellt wurde. (Bug #25040331)
- Wenn eine gespeicherte Prozedur eine Anweisung enthielt, die auf eine Ansicht verweist, die wiederum auf eine andere Ansicht verweist, konnte die Prozedur nicht mehr als einmal erfolgreich aufgerufen werden. (Fehler #87858, Fehler #26864199)
- Bei Abfragen mit vielen OR Bedingungen ist der Optimierer jetzt speichereffizienter und überschreitet weniger wahrscheinlich die Speichergrenze, die durch die Systemvariable [range\\_optimizer\\_max\\_mem\\_size](#) auferlegt wird. Darüber hinaus wurde der Standardwert für diese Variable von 1.536.000 auf 8.388.608 angehoben. (Fehler #79450, Fehler #22283790)
- Replikation: In der `next_event()`-Funktion, die vom SQL-Thread eines Replikats aufgerufen wird, um das nächste Ereignis aus dem Relay-Protokoll zu lesen, hat der SQL-Thread das erfasste `relaylog.log_lock` nicht freigegeben, als ein Fehler aufgetreten ist (z. B. aufgrund eines geschlossenen Relay-Protokolls), dazu führen, dass alle anderen Threads, die darauf warten, eine Sperre für das Relay-Protokoll zu erhalten, hängen bleiben. Mit dieser Fehlerbehebung wird

- die Sperre aufgehoben, bevor der SQL-Thread die Funktion unter der Situation verlässt. (Fehler #21697821)
- Beheben einer Speicherbeschädigung für ALTER TABLE mit virtueller Spalte. (Fehler #24961167, Fehler #24960450)
  - Replikation: Multithreaded-Replikate konnten nicht mit kleinen Warteschlangengrößen unter Verwendung von [slave\\_pending\\_jobs\\_size\\_max](#) konfiguriert werden, wenn sie jemals Transaktionen verarbeiten mussten, die größer als diese Größe sind. Jedes Paket, das größer als [slave\\_pending\\_jobs\\_size\\_max](#) ist, wurde mit dem Fehler ER\_MTS\_EVENT\_BIGGER\_PENDING\_JOBS\_SIZE\_MAX abgelehnt, auch wenn das Paket kleiner war als das von [slave\\_max\\_allowed\\_packet](#) festgelegte Limit. Mit dieser Fehlerbehebung wird [slave\\_pending\\_jobs\\_size\\_max](#) eher zu einem sanften Limit als zu einem harten Limit. Wenn die Größe eines Pakets [slave\\_pending\\_jobs\\_size\\_max](#) überschreitet, aber kleiner als [slave\\_max\\_allowed\\_packet](#) ist, wird die Transaktion solange gehalten, bis alle Replikat-Worker leere Warteschlangen haben und dann verarbeitet. Alle nachfolgenden Transaktionen werden solange gehalten, bis die große Transaktion abgeschlossen ist. Die Warteschlangengröße für Replikat-Worker kann daher begrenzt sein, während gelegentlich größere Transaktionen zugelassen werden. (Fehler #21280753, Fehler #77406)
  - Replikation: Bei Verwendung eines Multithread-Replikats zeigten Applier-Fehler Worker ID-Daten an, die mit Daten, die in den Replikationstabellen des Leistungsschemas externalisierten Daten nicht übereinstimmten. (Fehler #25231367)
  - Replikation: Auf einem GTID-basierten Replikationsreplikat, das mit [-gtid-mode=ON](#), [-log-bin=OFF](#) und mit [slave-skip-errors](#) ausgeführt wird, wenn ein Fehler aufgetreten ist, der ignoriert werden sollte, Exec\_Master\_Log\_Pos wurde die Synchronisierung mit nicht korrekt aktualisiert, wodurch Exec\_Master\_Log\_Pos die Synchronisierung mit verliertRead\_master\_log\_pos. Wenn ein GTID\_NEXT nicht angegeben wurde, würde das Replikat seinen GTID-Status niemals aktualisieren, wenn es von einer Transaktion mit einer einzelnen Anweisung zurückgesetzt wird. Der Exec\_Master\_Log\_Pos würde nicht aktualisiert werden, denn obwohl die Transaktion abgeschlossen war, zeigte ihr GTID-Status etwas anderes an. Die Fehlerbehebung hebt die Zurückhaltung der Aktualisierung des GTID-Status auf, wenn eine Transaktion nur zurückgesetzt wird, wenn in einer Transaktion GTID\_NEXT angegeben wird. (Fehler #22268777)
  - Replikation: Eine teilweise fehlgeschlagene Anweisung verbrauchte eine automatisch generierte oder angegebene GTID nicht korrekt, wenn die binäre Protokollierung deaktiviert wurde. Die Fehlerbehebung stellt sicher, dass eine teilweise fehlgeschlagene [DROP TABLE](#), ein teilweise fehlgeschlagener [DROP USER](#) oder ein teilweise fehlgeschlagener [DROP VIEW](#) die relevante

- GTID verbrauchen und sie in der @@GLOBAL.GTID\_EXECUTED- und mysql.gtid\_executed-Tabelle speichern, wenn die binäre Protokollierung deaktiviert wird. (Fehler #21686749)
- Replikation: Replikate, auf denen MySQL 5.7 ausgeführt wird, konnten aufgrund eines Fehlers beim Abrufen von [server\\_uuid](#), die nicht Teil von MySQL 5.5 ist, keine Verbindung zu einer MySQL 5.5-Quelle herstellen. Dies wurde durch Änderungen in der Methode zum Abrufen von verursacht [server\\_uuid](#). (Fehler #22748612)
  - Replikation: Der Mechanismus zum Überspringen von GTID-Transaktionen, der eine zuvor ausgeführte GTID-Transaktion stillschweigend überspringt, funktionierte für XA-Transaktionen nicht richtig. (Fehler #25041920)
  - [">XA ROLLBACK-Anweisungen](#), die fehlgeschlagen sind, weil eine falsche Transaktions-ID angegeben wurde, konnten im Binärprotokoll mit der richtigen Transaktions-ID aufgezeichnet werden und konnten daher durch Replikationsreplikate verwendet werden. Vor der binären Protokollierung wird nun die Fehlersituation geprüft und fehlgeschlagene XA ROLLBACK-Anweisungen nicht protokolliert. (Fehler #26618925)
  - Replikation: Wenn ein Replikat mit einer Anweisung [CHANGE MASTER TO](#) eingerichtet wurde, die den Namen der Quellprotokolldatei und die Position des Quellprotokolls nicht angegeben hat, fahren Sie herunter, bevor [START SLAVE](#) ausgegeben wurde, und starten Sie die Replikation dann nicht mit der Option [-relay-log-recovery](#) gesetzt. Dies geschah, weil der Empfänger-Thread vor dem Versuch der Wiederherstellung des Relay-Protokolls nicht gestartet wurde, sodass im Relay-Protokoll kein Protokollrotationsereignis verfügbar war, um den Namen der Quellprotokolldatei und die Quellprotokollposition bereitzustellen. In diesem Fall überspringt das Replikat nun die Wiederherstellung des Relay-Protokolls und protokolliert eine Warnung und beginnt dann mit der Replikation. (Fehler #28996606, Fehler #93397)
  - Replikation: Bei der zeilenbasierten Replikation wurde eine Meldung zurückgegeben, die Feldlängen falsch anzeigte, wenn von einer Tabelle mit einer utf8mb3-Spalte in eine Tabelle derselben Definition repliziert wurde, in der die Spalte mit einem utf8mb4-Zeichensatz definiert wurde. (Fehler #25135304, Fehler #83918)
  - Replikation: Als eine [RESET SLAVE-Anweisung](#) für ein Replikations-Replikat mit verwendeten GTIDs ausgegeben wurde, wurden die vorhandenen Relay-Protokolldateien bereinigt, aber die neue Relay-Protokolldatei für den Kanal wurde generiert, bevor der Satz der empfangenen GTIDs für den Kanal gelöscht wurde. Der frühere GTID-Satz wurde daher als PREVIOUS\_GTIDS Ereignis in die neue Relay-Protokolldatei geschrieben, was zu einem schwerwiegenden Fehler bei der Replikation führte, der besagt, dass das Replikat mehr GTIDs als die Quelle hatte, obwohl der für beide Server festgelegte Wert „gtid\_executed“ leer war. Wenn nun RESET SLAVE ausgegeben

- wird, wird der Satz der empfangenen GTIDs gelöscht, bevor die neue Relay-Protokolldatei generiert wird, damit diese Situation nicht auftritt. (Fehler #27411175)
- Replikation: Bei Verwendung von GTIDs, die für die Replikation verwendet wurden, konnten Transaktionen einschließlich Anweisungen, die einen Analysefehler verursachten ([ER\\_PARSE\\_ERROR](#)), nicht manuell durch die empfohlene Methode zum Einspritzen einer leeren oder Ersatztransaktion mit derselben GTID übersprungen werden. Diese Aktion sollte dazu führen, dass das Replikat die GTID als bereits verwendet identifiziert und daher die unerwünschte Transaktion überspringt, die seine GTID geteilt hat. Im Falle eines Parsing-Fehlers wurde der Replikationsanwendungs-Thread jedoch aufgrund des Parsing-Fehlers gestoppt, da die Anweisung geparkt wurde, bevor die GTID überprüft wurde, um zu sehen, ob sie übersprungen werden musste, obwohl die Transaktion trotzdem übersprungen werden sollte. Mit dieser Fehlerbehebung ignoriert der Replikations-Applier-Thread jetzt Parsingfehler, wenn die betreffende Transaktion übersprungen werden muss, da die GTID bereits verwendet wurde. Beachten Sie, dass diese Verhaltensänderung nicht bei Workloads gilt, die aus einer von erzeugten Binärprotokollausgabe bestehe `mysqlbinlog`. In dieser Situation bestünde die Gefahr, dass eine Transaktion mit einem Parsing-Fehler, der unmittelbar auf eine übersprungene Transaktion folgt, auch stillschweigend übersprungen wird, wenn sie einen Fehler auslösen sollte. (Fehler #27638268)
  - Replikation: Aktivieren Sie den SQL-Thread, um eine Teiltransaktion mit GTID zu überspringen. (Fehler #25800025)
  - Replikation: Wenn ein negativer oder fraktionierter Zeitüberschreitungs-Parameter an `WAIT_UNTIL_SQL_THREAD_AFTER_GTIDS()` angegeben wurde, verhielt sich der Server auf unerwartete Weise. Mit dieser Fehlerbehebung.
    - Wird ein fraktionierter Zeitüberschreitungs-Wert unverändert gelesen, ohne Abrunden.
    - Wird ein negativer Zeitüberschreitungs-Wert mit einem Fehler zurückgewiesen, wenn sich der Server in einem strikten SQL-Modus befindet. Wenn sich der Server nicht in einem strikten SQL-Modus befindet, bewirkt der Wert, dass die Funktion `NULL` sofort ohne Wartezeit zurückgibt und dann eine Warnung ausgibt. (Fehler #24976304, Fehler #83537)
  - Replikation: Wenn die `WAIT_FOR_EXECUTED_GTID_SET()`-Funktion mit einem Zeitüberschreitungs-Wert verwendet wurde, der einen Bruchteil enthält (z. B. 1,5), führte ein Fehler in der Umwandlungslogik dazu, dass die Zeitüberschreitung auf die nächste ganze Sekunde abgerundet wurde und bei Werten unter 1 Sekunde auf null (zum Beispiel 0,1). Die Casting-Logik wurde nun korrigiert, sodass der Zeitüberschreitungs-Wert wie ursprünglich angegeben ohne Rundung angewendet wird. Danke an Dirkjan Bussink für den Beitrag. (Fehler #29324564, Fehler #94247)

- Wenn GTIDs aktiviert sind, löste [XA COMMIT](#) bei einer nicht verbundenen XA-Transaktion innerhalb einer Transaktion mit mehreren Anweisungen eine Zusicherung aus. (Fehler #22173903)
- Replikation: In Debug-Builds wurde eine Zusicherung ausgelöst, wenn eine [XA ROLLBACK-Anweisung](#) für eine unbekannte Transaktions-ID ausgegeben wurde, als der `gtid_next`-Wert manuell festgelegt wurde. Der Server versucht jetzt nicht, den GTID-Status zu aktualisieren, wenn eine ROLLBACK XA-Anweisung mit einem Fehler fehlschlägt. (Fehler #27928837, Fehler #90640)
- Behebt das Problem mit der falschen Sortierreihenfolge, wenn mehrere CASE Funktionen in der ORDER BY Klausel verwendet werden (Fehler #22810883).
- Einige Abfragen, bei denen die Sortierung verwendet wurde, konnten während der Optimierung auf eine nicht initialisierte Spalte zugreifen und verursachten ein Server-Exit. (Fehler #27389294)

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon-Aurora-MySQL-Funktionen werden in Aurora-MySQL-Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora-MySQL-Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen zum synchronen Aufrufen von AWS Lambda Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`

## Aurora-MySQL-Datenbank-Engine-Updates 12.11.2021 (Version 2.09.3) (veraltet)

Version: 2.09.3


Aurora MySQL 2.09.3 ist allgemein verfügbar. Aurora-MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora-MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Sie können einen vorhandenen Aurora-MySQL-2.\*-Datenbank-Cluster zu Aurora MySQL 2.10.0 aktualisieren. Für Cluster, auf denen Aurora MySQL Version 1 ausgeführt wird, können Sie einen vorhandenen Cluster Aurora MySQL 1.23 oder höher direkt auf 2.10.0 aktualisieren. Sie können einen Snapshot aus einer derzeit unterstützten Aurora MySQL-Version zu Aurora MySQL 2.10.0 wiederherstellen.

Geben Sie zum Erstellen eines Clusters mit einer älteren Version von Aurora MySQL die Engine-Version über AWS Management Console, AWS CLI oder die Amazon-RDS-API an.

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

 Note

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Fehlerbehebungen bei der Sicherheit:

Korrekturen und andere Verbesserungen bei der Feinabstimmung der Handhabung in einer verwalteten Umgebung. Weitere CVE-Korrekturen unten:

- [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](#)

#### Verbesserungen der Verfügbarkeit:

- Es wurde eine Optimierung eingeführt, die die Konflikte für Abfragen reduzieren kann, die für Tabellen in `information_schema` ausgeführt werden.
- Unterstützung für ECDHE SSL-Chiffren hinzugefügt.

#### Allgemeine Verbesserungen:

- Es wurde ein Problem behoben, das unter seltenen Bedingungen dazu führen kann, dass eine Writer-Instance neu gestartet wurde, wenn eine Überprüfung der Datenintegrität im Speicher fehlschlägt.
- Es wurde ein Problem behoben, das unter seltenen Bedingungen dazu führen kann, dass die Datenbank-Instance neu gestartet wurde, wenn das Cluster-Volumen erweitert wird, während die binäre Protokollierung aktiviert ist.
- Es wurde eine seltene Race-Bedingung während eines Neustarts einer Datenbank-Instance behoben, die mehr als einen Neustart verursachen kann.
- Es wurde ein Problem behoben, das dazu führen kann, dass ein Neustart der Datenbank-Instance fehlschlug, wenn die Datenbank über eine große Anzahl von Benutzer- und Berechtigungskombinationen verfügt.
- Es wurde ein Problem mit einer parallelen Abfrage behoben, durch das die Datenbank beim Ausführen von SQL-Anweisungen mit LIMIT-Klausel neu gestartet wurde.
- Es wurde ein Fehler mit falscher Meldung der Aurora-Replikationsverzögerung behoben.
- Es wurde ein Problem behoben, das dazu führen kann, dass `general_log`- und `slow_log`-Tabellen nach dem In-Place-Upgrade von Aurora MySQL 1.x (basierend auf MySQL 5.6) auf Aurora MySQL 2.x (basierend auf MySQL 5.7) unzugänglich wurden.
- Es wurde ein Problem behoben, das in seltenen Fällen dazu führen kann, dass die Datenbank-Instance neu gestartet wurde, wenn `innodb_trx`-, `innodb_locks`- oder `innodb_lockwaits`-Tabellen abgefragt wurden, während die Datenbank unter starkem Workload steht. Überwachungstools und Funktionen wie Leistungserkenntnisse können solche Tabellen abfragen.
- Es wurde ein Problem behoben, durch das eine Datenbank-Instance neu gestartet wurde, wenn die SQL-Anweisung „`FLUSH TABLES WITH READ LOCK`“ ausgeführt wurde.

- Es wurde ein Problem behoben, durch das der InnoDB-Bereinigungsprozess während des Löschens einer Reader-Instance unterbrochen wurde, was zu einer vorübergehenden Erhöhung der Länge der Verlaufsliste führte.
- Es wurde ein Problem mit einer parallelen Abfrage behoben, das dazu führen kann, dass die Datenbank neu gestartet wurde, wenn eine SQL-Anweisung für eine Tabelle mit einer virtuellen Spalte ausgeführt wurde.
- Es wurde ein Problem mit paralleler Abfrage behoben, durch das die Datenbank falsche Gruppierungen oder Sortierreihenfolge zurückgab, wenn Abfragen mit GROUP BY-Klausel und einer WHERE-Klausel mit einem Bereichsprädikat ausgeführt wurden.
- Es wurde ein Problem bei der parallelen Abfrage behoben, das unter seltenen Bedingungen dazu führen kann, dass die Datenbank beim Ausführen von SQL-Anweisungen mit JSON-Funktionen neu gestartet wurde.
- Es wurde ein Problem behoben, das unter seltenen Bedingungen dazu führen kann, dass die Writer-Instance im primären Global Database-Cluster aufgrund einer Race-Bedingung während der globalen Datenbankreplikation neu gestartet wurde.
- Es wurde ein Problem behoben, das dazu führen kann, dass ein Binlog-Replikat mit einem HA\_ERR\_FOUND\_DUPP\_KEY-Fehler stoppt, wenn bestimmte DDL- und DCL-Anweisungen repliziert wurden. Das Problem tritt auf, wenn die Quell-Instance mit dem gemischten Binärprotokollierungsformat und READ COMMITTED oder READ UNCOMMITTED Isolationsstufe konfiguriert.
- Es wurde ein Problem behoben, das unter seltenen Bedingungen dazu führen kann, dass die Datenbank-Instance bei Verwendung von XA-Transaktionen in READ COMMITTED Isolationsstufe neu gestartet wurde.
- Es wurde ein Problem behoben, durch das der Wert einer TIMESTAMP-Spalte einer vorhandenen Zeile auf den neuesten Zeitstempel aktualisiert wurde, wenn alle folgenden Bedingungen erfüllt sind: 1. Ein Auslöser für die Tabelle existiert; 2. ein INSERT wird für die Tabelle ausgeführt, die eine ON DUPLICATE KEY UPDATE-Klausel hat; 3. die eingefügte Zeile kann eine verursachen doppelte Wertverletzung in einem UNIQUE-Index oder PRIMARY KEY; und 4. eine oder mehrere Spalten sind vom Datentyp TIMESTAMP und haben einen Standardwert von CURRENT\_TIMESTAMP.
- Es wurde ein Problem behoben, das unter seltenen Bedingungen dazu führen kann, dass eine Reader-Instance aufgrund einer falschen Prüfverarbeitung neu gestartet wurde.
- Es wurde ein Problem behoben, durch das die Reader-Instance neu gestartet wurde, wenn die Writer-Instance das Datenbankvolumen um bestimmte Volumengrößengrenzen vergrößert.

- Es wurde ein Problem behoben, das längere Neustartzeiten für Datenbank-Instances mit geklonten Cluster-Volumes verursachen konnte.
- Es wurde ein Problem behoben, durch das ein Neustart der Datenbank-Instance ein- oder mehrmals fehlschlug, nachdem ein TRUNCATE TABLE-Vorgang für die Writer-Instance ausgeführt wurde.
- Es wurde ein Problem behoben, das unter seltenen Bedingungen dazu führen kann, dass die Datenbank-Instance neu gestartet wurde.
- Es wurde ein Problem behoben, das unter seltenen Bedingungen dazu führen kann, dass die Writer-Instance neu gestartet wird, wenn das Datenbank-Volume ein Vielfaches von 160 GB erreicht.

## Integration von MySQL-Fehlerbehebungen (Community Edition):

- Fehler #23533396 - Beim Hinzufügen eines neuen Index ließ der Server einen intern definierten Fremdschlüsselindex fallen und versuchte, einen sekundären Index, der für eine virtuell generierte Spalte definiert ist, als Fremdschlüsselindex zu verwenden, was zu einem Serveraustritt führte. InnoDB erlaubt nun einer Fremdschlüsseleinschränkung, auf einen sekundären Index zu verweisen, der für eine virtuell generierte Spalte definiert ist.
- Fehler #29550513 - Replikation: Ein Sperrproblem in der Funktion WAIT\_FOR\_EXECUTED\_GTID\_SET () kann dazu führen, dass der Server unter bestimmten Umständen hängen bleibt. Das Problem wurde jetzt behoben.

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon-Aurora-MySQL-Funktionen werden in Aurora-MySQL-Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora-MySQL-Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen für den synchronen Aufruf von AWS Lambda-Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.

- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`

## Aurora-MySQL-Datenbank-Engine-Updates 26.02.2021 (Version 2.09.2) (veraltet)

Version: 2.09.2

Aurora MySQL 2.09.2 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Sie können einen bestehenden Aurora MySQL 2.\* Datenbank-Cluster auf Aurora MySQL 2.09.2 aktualisieren. Bei Clustern mit Aurora MySQL Version 1 können Sie ein bestehendes Cluster mit Aurora MySQL 1.23 oder höher direkt auf 2.09.2 aktualisieren. Sie können auch einen Snapshot von jeder derzeit unterstützten Aurora MySQL-Version in Aurora MySQL 2.09.2 wiederherstellen.

Geben Sie zum Erstellen eines Clusters mit einer älteren Version von Aurora MySQL die Engine-Version über AWS Management Console, AWS CLI oder die Amazon-RDS-API an.

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

#### Note

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Neue Funktionen:

- Aurora-MySQL-Cluster unterstützen jetzt die folgenden EC2-R6g-Instances, die von ARM-basierten AWS-Graviton2-Prozessoren unterstützt werden: r6g.large, r6g.xlarge, r6g.2xlarge, r6g.4xlarge, r6g.8xlarge, r6g.12xlarge, r6g.16xlarge. Weitere Informationen finden Sie unter [Aurora-DB-Instance-Klassen](#) im Amazon-Aurora-Benutzerhandbuch.

Fehlerbehebungen bei der Sicherheit:

Korrekturen und andere Verbesserungen bei der Feinabstimmung der Handhabung in einer verwalteten Umgebung. Weitere CVE Fixes unten:

- [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](#)

#### Verbesserungen der Verfügbarkeit:

- Es wurde ein in 2.09.0 eingeführtes Problem behoben, das während der Skalierung des Clusterspeichervolumens zu einer erhöhten Schreiblatenz führen konnte.
- Es wurde ein Problem mit der Funktion zur dynamischen Größenänderung behoben, das zum Neustart von Aurora Read Replicas führen konnte.
- Es wurde ein Problem behoben, das während des Upgrades von 1.23.\* auf 2.09.\* zu längeren Ausfallzeiten führen konnte.
- Es wurde ein Problem behoben, bei dem eine DDL oder DML während einer Prefetch-Anfrage zu einem Neustart einer Engine des Motors führen konnte.
- Es wurde ein Problem behoben, durch das ein Binlog-Replik mit einem Fehler gestoppt wird, wenn die replizierte Transaktion eine DDL-Anweisung und eine große Anzahl von Zeilenänderungen enthält.
- Es wurde ein Problem behoben, bei dem eine als Binlog-Replik fungierende Datenbank neu gestartet werden kann, während ein DDL-Ereignis auf der `mysql.time_zone`-Tabelle repliziert wird.
- Es wurde ein Problem behoben, das dazu führen konnte, dass große Transaktionen zu falschen Binärprotokollereignissen führen konnten, wenn `binlog_checksum`-Parameter auf `NONE` eingestellt wurde.
- Es wurde ein Problem behoben, das dazu führte, dass ein Binlog-Replik mit einem `HA_ERR_KEY_NOT_FOUND`-Fehler stoppt.
- Verbesserte Gesamtstabilität.

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora MySQL Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen für den synchronen Aufruf von AWS Lambda-Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter

- Die SQL-Anweisung CREATE TABLESPACE

## Aurora-MySQL-Datenbank-Engine-Updates 11.12.2020 (Version 2.09.1) (veraltet)

Version: 2.09.1

Aurora MySQL 2.09.1 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Sie können einen bestehenden Aurora MySQL 2.\* Datenbank-Cluster auf Aurora MySQL 2.09.1 aktualisieren. Bei Clustern mit Aurora MySQL Version 1 können Sie ein bestehendes Cluster mit Aurora MySQL 1.23 oder höher direkt auf 2.09.1 aktualisieren. Sie können auch einen Snapshot von jeder derzeit unterstützten Aurora MySQL-Version in Aurora MySQL 2.09.1 wiederherstellen.

Geben Sie zum Erstellen eines Clusters mit einer älteren Version von Aurora MySQL die Engine-Version über AWS Management Console, AWS CLI oder die RDS-API an.

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

### Note

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Fehlerbehebungen bei der Sicherheit:

Korrekturen und andere Verbesserungen bei der Feinabstimmung der Handhabung in einer verwalteten Umgebung. Weitere CVE Fixes unten:



- [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](#)

#### Inkompatible Änderungen:

Diese Version führt eine Berechtigungsänderung ein, die sich auf das Verhalten des `mysqldump` Befehls auswirkt. Benutzer müssen das `PROCESS` Recht haben, auf die `INFORMATION_SCHEMA.FILES` Tabelle zuzugreifen. Um den `mysqldump` Befehl ohne Änderungen auszuführen, erteilen Sie dem Datenbankbenutzer, mit dem sich der `PROCESS` Befehl verbindet, die `mysqldump` Berechtigung. Sie können den `mysqldump` Befehl auch mit der `--no-tablespaces` Option ausführen. Mit dieser Option enthält die `mysqldump` Ausgabe keine `CREATE LOGFILE GROUP` oder `CREATE TABLESPACE` Anweisungen. In diesem Fall greift der `mysqldump` Befehl nicht auf die `INFORMATION_SCHEMA.FILES` Tabelle zu, und Sie müssen die `PROCESS` Berechtigung nicht erteilen.

#### Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem behoben, das dazu führen konnte, dass sich eine Clientsitzung aufhängte, wenn die Datenbankengine beim Lesen oder Schreiben in das Netzwerk auf einen Fehler stieß.
- Es wurde ein Speicherleck in der Funktion zur dynamischen Größenänderung behoben, das in 2.09.0 eingeführt wurde.

#### Globale Datenbanken:

- Es wurden mehrere Probleme behoben, bei dem die Replikate einer sekundären Datenbankregion beim Upgrade auf Version 2.09.0 neu gestartet werden konnten, während der primäre Region-Writer eine ältere Release-Version hatte.

## Integration von MySQL-Fehlerbehebungen (Community Edition):

- Replikation: Doppelspieltransaktionen konnten manchmal den Slave-Applier blockieren, wenn die Transaktionsisolationsstufe auf [REPEATABLE READ](#) festgelegt wurde. (Bug #25040331)
- Bei einer Tabelle mit einer [TIMESTAMP](#)- oder [DATETIME](#)-Spalte mit dem Standardwert [CURRENT\\_TIMESTAMP](#) könnte die Spalte auf 0000-00-00 00:00:00 initialisiert werden, wenn die Tabelle einen BEFORE INSERT-Auslöser hat. (Bug #25209512, Bug #84077)
- Bei einer [INSERT](#)-Anweisung, für die die VALUES-Liste mit einer Unterabfrage, die eine Verknüpfung enthält, Werte für die zweite oder spätere Zeile erzeugte, konnte es zu einer Beendigung des Servers kommen, nachdem die erforderlichen Berechtigungen nicht aufgelöst wurden. (Bug #23762382)

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora MySQL Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen für den synchronen Aufruf von AWS Lambda-Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`

## Aurora-MySQL-Datenbank-Engine-Updates 17.09.2020 (Version 2.09.0) (veraltet)

Version: 2.09.0

Aurora MySQL 2.09.0 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Sie können einen Snapshot von Aurora MySQL 1.23.\* in Aurora MySQL 2.09.0 wiederherstellen. Sie haben auch die Möglichkeit bestehende Aurora MySQL 2.\*-Datenbank-Cluster auf Aurora

MySQL 2.09.0 aktualisieren. Sie können einen vorhandenen Aurora MySQL 1.23.\*-Cluster nicht direkt auf Aurora MySQL 2.09.0 aktualisieren. Sie können jedoch seinen Snapshot zu Aurora MySQL 2.09.0 wiederherstellen.

### Important

Die Verbesserungen des Aurora Speichers in dieser Version beschränken die verfügbaren Upgrade-Pfade von Aurora MySQL 1 auf Aurora MySQL 2.09.\*. Wenn Sie einen Aurora MySQL 1.\* Cluster auf 2.09.\* aktualisieren, müssen Sie ein Upgrade von Aurora MySQL 1.23 durchführen.

Geben Sie zum Erstellen eines Clusters mit einer älteren Version von Aurora MySQL die Engine-Version über AWS Management Console, AWS CLI oder die RDS-API an.

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

### Note

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Neue Funktionen:

- Mit dieser Version können Sie Amazon Aurora MySQL-Datenbank-Instances mit bis zu 128 Tebibyte (TiB) Speicher erstellen. Das neue Speicherlimit stellt eine Erhöhung gegenüber den vorherigen 64 TiB dar. Die Speichergröße von 128 TiB unterstützt größere Datenbanken. Diese Funktion wird für kleine Instance-Größen (db.t2 oder db.t3) nicht unterstützt. Ein einzelner Tabellenraum kann aufgrund von [InnoDB-Einschränkungen mit 16 KB Seitengröße](#) nicht über 64 TiB hinausgehen.

Aurora benachrichtigt Sie, wenn die Cluster-Volume-Größe nahe 128 TiB liegt, sodass Sie Maßnahmen ergreifen können, bevor Sie die Größenbeschränkung erreichen. Die

Benachrichtigungen werden im mysql-Protokoll und die RDS-Ereignisse in der angezeigt AWS Management Console.

- Sie können jetzt durch Ändern des Werts für den DB-Cluster-Parameter parallele Abfragen für einen vorhandenen Cluster aktivieren oder deaktivieren `aurora_parallel_query`. Sie müssen beim Erstellen des Clusters nicht die Einstellung `parallelquery` für den Parameter `--engine-mode` verwenden.

Parallele Abfragen sind jetzt in allen Regionen verfügbar, in denen Aurora MySQL verfügbar ist.

Es gibt eine Reihe weiterer Funktionsverbesserungen und Änderungen für die Verfahren zum Aktualisieren und Aktivieren paralleler Abfragen in Aurora-Clustern. Weitere Informationen finden Sie im Abschnitt zum [Arbeiten mit parallelen Abfragen für Amazon Aurora MySQL](#) im Amazon-Aurora-Benutzerhandbuch.

- Aurora passt die Größe Ihres Cluster-Speicherplatzes dynamisch an. Bei dynamischer Größenanpassung verringert sich der Speicherplatz für den Aurora-DB-Cluster automatisch, wenn Sie Daten aus dem DB-Cluster entfernen. Weitere Informationen finden Sie unter [Speicherskalierung](#) im Amazon-Aurora-Benutzerhandbuch.

#### Note

Die Funktion zur dynamischen Größenanpassung wird in Phasen in den AWS-Regionen bereitgestellt, in denen Aurora verfügbar ist. Abhängig von der Region, in der sich Ihr Cluster befindet, ist diese Funktion möglicherweise noch nicht verfügbar. Weitere Informationen finden Sie in der Ankündigung [„Was ist neu“](#).

#### Wichtige Fehlerbehebungen:

- Backport von Community-Bug #27659490: SELECT USING DYNAMIC RANGE AND INDEX MERGE USE TOO MUCH MEMORY (OOM) (SELECT MIT DYNAMISCHEM BEREICH UND INDEXZUSAMMENFÜHRUNG VERBRAUCHT ZUVIEL SPEICHER)
- Bug #26881508: MYSQL #1: DISABLE\_ABORT\_ON\_ERROR IN AUTH\_COMMON.H (DISABLE\_ABORT\_ON\_ERROR IN AUTH\_COMMON.H)
- Backport von Community-Bug #24437124: POSSIBLE BUFFER OVERFLOW ON CREATE TABLE (MÖGLICHER PUFFERÜBERLAUF BEI CREATE TABLE)
- Backport von Bug #27158030: INNODB ONLINE ALTER CRASHES WITH CONCURRENT DML (INNODB ONLINE ALTER STÜRZT BEI GLEICHZEITIGER DML AB)

- Fehler #29770705: DER SERVER STÜRZTE BEIM AUSFÜHREN EINER AUSWAHL MIT SPEZIFISCHER WHERE-KLAUSEL AB
- Backport von Bug #26502135: MYSQLD SEGFAULTS IN MDL\_CONTEXT:: TRY\_ACQUIRE\_LOCK\_IMPL (MYSQLD SEGFAULTS IN MDL\_CONTEXT::TRY\_ACQUIRE\_LOCK\_IMPL)
- Backport von Bug #26935001: ALTER TABLE AUTO\_INCREMENT TRIES TO READ INDEX FROM DISCARDED TABLESPACE (ALTER TABLE AUTO\_INCREMENT VERSUCHT, INDEX AUS VERWORFENEM TABELLENRAUM ZU LESEN)
- Bug #28491099: [FATAL] MEMORY BLOCK IS INVALID | INNODB: ASSERTION FAILURE: UT0UT.CC:670 (SPEICHERBLOCK IST UNGÜLTIG | INNODB: ZUSICHERUNGSFEHLER: UT0UT.CC:670)
- Bug #30499288: GCC 9.2.1 REPORTS A NEW WARNING FOR OS\_FILE\_GET\_PARENT\_DIR (GCC 9.2.1 MELDET NEUE WARNUNG FÜR OS\_FILE\_GET\_PARENT\_DIR)
- Bug #29952565: MYSQLD GOT SIGNAL 11 WHILE EXECUTING A QUERY(UNION + ORDER BY + SUB-QUERY) (MYSQLD ERHÄLT SIGNAL 11 WÄHREND AUSFÜHRUNG EINER ABFRAGE (UNION + SORTIEREN NACH + UNTERABFRAGE))
- Bug #30628268: ABSTURZ WEGEN FEHLENDEM SPEICHER
- Fehler #30441969: FEHLER #29723340: ABSTURZ DES MYSQL-SERVERS NACH SQL-ABFRAGE MIT DATA ?AST
- Bug #30569003: 5.7 REPLICATION BREAKAGE WITH SYNTAX ERROR WITH GRANT MANAGEMENT (5.7-REPLIKATIONSBRUCH MIT SYNTAXFEHLER MIT ERTEILUNGSVERWALTUNG)
- Fehler #29915479: DAS AUSFÜHREN VON COM\_REGISTER\_SLAVE OHNE COM\_BINLOG\_DUMP KANN ZU EINEM BEENDEN DES SERVERS FÜHREN
- Bug #30569003: 5.7 REPLICATION BREAKAGE WITH SYNTAX ERROR WITH GRANT MANAGEMENT (5.7-REPLIKATIONSBRUCH MIT SYNTAXFEHLER MIT ERTEILUNGSVERWALTUNG)
- Fehler #29915479: DAS AUSFÜHREN VON COM\_REGISTER\_SLAVE OHNE COM\_BINLOG\_DUMP KANN ZU EINEM BEENDEN DES SERVERS FÜHREN
- Bug #20712046: SHOW PROCESSLIST AND PERFORMANCE\_SCHEMA TABLES DO NOT MASK PASSWORD FROM QUERY (PROZESSLISTE ANZEIGEN UND PERFORMANCE\_SCHEMA-TABELLEN MASKIEREN PASSWORT AUS ABFRAGE NICHT)
- Backport-Bug #18898433: EXTREMELY SLOW PERFORMANCE WITH OUTER JOINS AND JOIN BUFFER (fixed in 5.7.21) (EXTREM LANGSAME LEISTUNG BEI EXTERNEN JOINS)

- UND JOIN-PUFFER (behooben in 5.7.21)) Abfragen mit zahlreichen Links-Joins waren langsam, wenn die Join-Pufferung verwendet wurde (z. B. bei Verwendung mit dem blockverschachtelten Schleifenalgorithmus). (Bug #18898433, Bug #72854)
- Backport-Bug #26402045: MYSQLD CRASHES ON QUERY) (MYSQLD STÜRZT BEI ABFRAGE AB (behooben in MySQL 5.7.23)) Bestimmte Unterabfragen-Materialisierungen konnten einen Server-Exit verursachen. Diese Abfragen erzeugen nun einen Fehler, der darauf hindeutet, dass die Materialisierung deaktiviert ist. (Bug #26402045)
  - [Backport von MySQL] Andere Benutzer als rdsadmin dürfen die PFS-Tabelle im Reader-Replikat nicht aktualisieren.
  - Der Fehler, bei dem der Kunde das perfschema im Reader-Replikat nicht aktualisieren kann, wurde behoben.
  - Bug #26666274: INFINITE LOOP IN PERFORMANCE SCHEMA BUFFER CONTAINER (UNENDLICHE SCHLEIFE IM LEISTUNGSSCHEMA-PUFFERCONTAINER)
  - [Bug #26997096](#): Der relay\_log\_space-Wert wird nicht synchronisiert aktualisiert, sodass sein Wert manchmal viel höher ist als der tatsächlich von Relay-Protokollen belegte Festplattenspeicher.
  - Bug #25082593: FOREIGN KEY VALIDATION DOESN'T NEED TO ACQUIRE GAP LOCK IN READ COMMITTED (FREMDSCHLÜSSELVALIDIERUNG MUSS KEIN GAP-LOCK IN READ COMMITTED ERWERBEN)
  - [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](#)

#### Verbesserungen der Verfügbarkeit:

- Aktivieren Sie den Sperrverwalter ABA fix standardmäßig.
- Es wurde ein Problem im Sperrverwalter behoben, bei dem eine Race-Bedingung dazu führen konnte, dass eine Sperre von zwei Transaktionen gemeinsam genutzt wurde, was zu einem Neustart der Datenbank führte.
- Es wurde ein Problem beim Erstellen einer temporären Tabelle mit komprimiertem Zeilenformat behoben, das zu einem Neustart führen konnte.
- Korrigierter Standardwert von `table_open_cache` auf 16XL- und 24XL-Instances, der wiederholte Failover und hohe CPU-Auslastung auf großen Instance-Klassen (R4/R5-16XL, R5-12XL, R5-24XL) verursachen konnte. Dies wirkt sich auf 2.07.x aus.
- Es wurde ein Problem behoben, durch das die Wiederherstellung eines Clusters aus Amazon S3 zu Aurora MySQL Version 2.08.0 länger dauert als erwartet, wenn das S3-Backup nicht die Tabelle `mysql.host` enthält.
- Es wurde ein Problem behoben, das aufgrund von Aktualisierungen virtueller Spalten mit sekundären Indizes wiederholte Failover verursachen konnte.
- Es wurde ein Problem im Zusammenhang mit der Transaktionssperr-Speicherverwaltung bei lang ausgeführten Schreibtransaktionen behoben, was zu einem Neustart der Datenbank führte.
- Mehrere Probleme wurden behoben, bei denen die Engine während des Null-Downtime-Patchings abstürzen konnte, während sie den sicheren Punkt für das Patching prüfte.
- Es wurde ein Problem behoben, bei dem die Redo-Protokollierung für temporäre Tabellen übersprungen wurde, was zuvor zu einem Absturz führte.
- Es wurde eine Race-Bedingung im Sperrverwalter zwischen dem Beenden der Verbindung/Abfrage und dem Beenden der Sitzung behoben.
- Es wurde ein Problem behoben, bei dem die Datenbank abstürzen konnte, wenn es sich bei ihr um ein Binlog-Replikat handelte und sie ein DDL-Ereignis über die MySQL-Tabelle `time_zone` empfing.

#### Globale Datenbanken:



- Die Ansicht MySQL INFORMATION\_SCHEMA.REPLICA\_HOST\_STATUS in einer sekundären Region zeigt jetzt die Einträge für die Replikate an, die zu dieser Region gehören.
- Es wurden unerwartete Abfragefehler behoben, die in einer sekundären Global DB-Region im Anschluss an temporären Problemen mit der Netzwerkkonnektivität zwischen primären und sekundären Regionen auftreten konnten.

- 

#### Parallele Abfrage:

- Es wurde ein Problem mit dem EXPLAIN-Plan für eine Parallel Query-Abfrage behoben, der für eine einfache Einzeltabellenabfrage nicht korrekt ist.
- Es wurde ein Problem mit der Selbstsperrung behoben, das auftreten kann, wenn Parallel Query aktiviert ist.

#### Allgemeine Verbesserungen:

- Der Export zu S3 unterstützt jetzt das Schlüsselwort ENCRYPTION.
- Der maximal akzeptierte Wert des Parameters `aurora_binlog_replication_max_yield_seconds` beträgt jetzt 36.000. Der vorherige maximal akzeptierte Wert betrug 45. Dieser Parameter funktioniert nur, wenn der Parameter `aurora_binlog_use_large_read_buffer` auf 1 festgelegt wird.
- Das Verhalten wurde geändert, indem MIXED `binlog_format` jetzt ROW und nicht STATEMENT zugeordnet wird, wenn `LOAD DATA FROM INFILE | S3` ausgeführt wird.
- Es wurde ein Problem behoben, bei dem ein mit einem Aurora MySQL Binlog-Primary verbundenes Binlog-Replikat unvollständige Daten anzeigen konnte, wenn der Primary `LOAD DATA FROM S3` ausführte und `binlog_format` auf STATEMENT festgelegt war.
- Die maximal zulässige Länge für die Audit-Systemvariablen `server_audit_incl_users` und `server_audit_excl_users` wurde von 1.024 Bytes auf 2.000 Bytes erhöht.
- Es wurde ein Problem behoben, bei dem Benutzer möglicherweise den Zugriff auf die Datenbank verlieren, wenn der Parameter `max_connections` in der Parametergruppe gesenkt wird, wenn die Zahl der aktuellen Verbindungen den festgelegten Wert übersteigt.
- Es wurde ein Problem in Datenaktivitätsströmen behoben, bei dem ein einzelnes Anführungszeichen und ein umgekehrter Schrägstrich nicht richtig maskiert wurden.

## Integration von MySQL-Fehlerbehebungen (Community Edition):

- Fehler #27659490: AUSWAHL MIT DYNAMISCHEN BEREICH UND INDEX-ZUSAMMENFÜHRUNG VERBRAUCHT ZU VIEL SPEICHER (NICHT GENÜGENDE SPEICHER)
- Bug #26881508: MYSQL #1: DISABLE\_ABORT\_ON\_ERROR IN AUTH\_COMMON.H (DISABLE\_ABORT\_ON\_ERROR IN AUTH\_COMMON.H)
- Bug #24437124: POSSIBLE BUFFER OVERFLOW ON CREATE TABLE (MÖGLICHER PUFFERÜBERLAUF BEI CREATE TABLE)
- Bug #27158030: INNODB ONLINE ALTER CRASHES WITH CONCURRENT DML (INNODB ONLINE ALTER STÜRZT BEI GLEICHZEITIGER DML AB)
- Fehler #29770705: DER SERVER STÜRZTE BEIM AUSFÜHREN EINER AUSWAHL MIT SPEZIFISCHER WHERE-KLAUSEL AB
- Bug #26502135: MYSQLD SEGFAULTS IN MDL\_CONTEXT::TRY\_ACQUIRE\_LOCK\_IMPL (MYSQLD SEGFAULTS IN MDL\_CONTEXT::TRY\_ACQUIRE\_LOCK\_IMPL)
- Fehler #26935001: „ALTER TABLE AUTO\_INCREMENT“ VERSUCHT, DEN INDEX AUS VERWORFENEM TABELLENRAUM ZU LESEN
- Bug #28491099: [FATAL] MEMORY BLOCK IS INVALID | INNODB: ASSERTION FAILURE: UT0UT.CC:670 (SPEICHERBLOCK IST UNGÜLTIG | INNODB: ZUSICHERUNGSFEHLER: UT0UT.CC:670)
- Bug #30499288: GCC 9.2.1 REPORTS A NEW WARNING FOR OS\_FILE\_GET\_PARENT\_DIR (GCC 9.2.1 MELDET NEUE WARNUNG FÜR OS\_FILE\_GET\_PARENT\_DIR)
- Bug #29952565: MYSQLD GOT SIGNAL 11 WHILE EXECUTING A QUERY(UNION + ORDER BY + SUB-QUERY) (MYSQLD ERHÄLT SIGNAL 11 WÄHREND AUSFÜHRUNG EINER ABFRAGE (UNION + SORTIEREN NACH + UNTERABFRAGE))
- Bug #30628268: ABSTURZ WEGEN FEHLENDEM SPEICHER
- Fehler #30441969: FEHLER #29723340: ABSTURZ DES MYSQL-SERVERS NACH SQL-ABFRAGE MIT DATA ?AST
- Bug #30569003: 5.7 REPLICATION BREAKAGE WITH SYNTAX ERROR WITH GRANT MANAGEMENT (5.7-REPLIKATIONSBRUCH MIT SYNTAXFEHLER MIT ERTEILUNGSVERWALTUNG)
- Fehler #29915479: DAS AUSFÜHREN VON COM\_REGISTER\_SLAVE OHNE COM\_BINLOG\_DUMP KANN ZU EINEM BEENDEN DES SERVERS FÜHREN

- Bug #30569003: 5.7 REPLICATION BREAKAGE WITH SYNTAX ERROR WITH GRANT MANAGEMENT (5.7-REPLIKATIONSBRUCH MIT SYNTAXFEHLER MIT ERTEILUNGSVERWALTUNG)
- Fehler #29915479: DAS AUSFÜHREN VON COM\_REGISTER\_SLAVE OHNE COM\_BINLOG\_DUMP KANN ZU EINEM BEENDEN DES SERVERS FÜHREN
- Bug #20712046: SHOW PROCESSLIST AND PERFORMANCE\_SCHEMA TABLES DO NOT MASK PASSWORD FROM QUERY (PROZESSLISTE ANZEIGEN UND PERFORMANCE\_SCHEMA-TABELLEN MASKIEREN PASSWORT AUS ABFRAGE NICHT)
- Bug #18898433: EXTREMELY SLOW PERFORMANCE WITH OUTER JOINS AND JOIN BUFFER (EXTREM LANGSAME LEISTUNG BEI EXTERNEN JOINS UND JOIN-PUFFER (behoben in 5.7.21))
- Bug #26402045: MYSQLD CRASHES ON QUERY) (MYSQLD STÜRZT BEI ABFRAGE AB (behoben in MySQL 5.7.23))
- Fehler #23103937: PS\_TRUNCATE\_ALL\_TABLES() DOES NOT WORK IN SUPER\_READ\_ONLY MODE
- Bug #26666274: INFINITE LOOP IN PERFORMANCE\_SCHEMA BUFFER CONTAINER (UNENDLICHE SCHLEIFE IM LEISTUNGSSCHEMA-PUFFERCONTAINER)
- Fehler #26997096: Der relay\_log\_space-Wert wird nicht synchronisiert aktualisiert, sodass sein Wert manchmal viel höher ist als der tatsächlich von den Relay-Protokollen belegte Festplattenspeicher. (<https://github.com/mysql/mysql-server/commit/78f25d2809ad457e81f90342239c9bc32a36cdfa>)
- Bug #25082593: FOREIGN KEY VALIDATION DOESN'T NEED TO ACQUIRE GAP LOCK IN READ COMMITTED (FREMDSCHLÜSSELVALIDIERUNG MUSS KEIN GAP-LOCK IN READ COMMITTED ERWERBEN)
- Bug #24764800: REPLICATION FAILING ON SLAVE WITH XAER\_RMFAIL ERROR (REPLIKATIONSFEHLER AUF SLAVE MIT FEHLER XAER\_RMFAIL)
- Bug #81441: WARNING ABOUT LOCALHOST WHEN USING SKIP-NAME-RESOLVE (WARNUNG ZU LOCALHOST BEI VERWENDUNG VON SKIP-NAME-RESOLVE)

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora MySQL Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen für den synchronen Aufruf von AWS Lambda-Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter

- Die SQL-Anweisung CREATE TABLESPACE

## Aurora-MySQL-Datenbank-Engine-Updates 06.01.2022 (Version 2.08.4) (veraltet)

Version: 2.08.4

Aurora MySQL 2.08.4 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Sie können einen vorhandenen Aurora-MySQL-2.\*-Datenbank-Cluster zu Aurora MySQL 2.10.0 aktualisieren. Für Cluster, auf denen Aurora MySQL Version 1 ausgeführt wird, können Sie einen vorhandenen Cluster Aurora MySQL 1.23 oder höher direkt auf 2.10.0 aktualisieren. Sie können einen Snapshot aus einer derzeit unterstützten Aurora MySQL-Version zu Aurora MySQL 2.10.0 wiederherstellen.

Geben Sie zum Erstellen eines Clusters mit einer älteren Version von Aurora MySQL die Engine-Version über AWS Management Console, AWS CLI oder die Amazon-RDS-API an.

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

### Note

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Sicherheitsupdates und allgemeine Verbesserungen:

- Sicherheitsprobleme im Zusammenhang mit der Aurora-MySQL-Integration mit anderen AWS-Services wie Amazon S3, Amazon ML und AWS Lambda wurden behoben.

- Es wurde ein Problem behoben, durch das der Wert einer TIMESTAMP-Spalte einer vorhandenen Zeile auf den neuesten Zeitstempel aktualisiert wurde, wenn alle folgenden Bedingungen erfüllt sind: 1. Ein Auslöser für die Tabelle existiert; 2. ein INSERT wird für die Tabelle ausgeführt, die eine ON DUPLICATE KEY UPDATE-Klausel hat; 3. die eingefügte Zeile kann eine verursachen doppelte Wertverletzung in einem UNIQUE-Index oder PRIMARY KEY; und 4. eine oder mehrere Spalten sind vom Datentyp TIMESTAMP und haben einen Standardwert von CURRENT\_TIMESTAMP.
- Es wurde ein Problem behoben, das unter seltenen Bedingungen dazu führte, dass eine Writer-Instance neu gestartet wurde, wenn eine Überprüfung der Datenintegrität im Speicher fehlschlägt.
- Es wurde ein Problem mit Parallel Query behoben, durch das die Datenbank beim Ausführen von SQL-Anweisungen mit einer LIMIT-Klausel neu gestartet wurde.

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora MySQL Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen für den synchronen Aufruf von AWS Lambda-Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native

Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`

## Aurora-MySQL-Datenbank-Engine-Updates 12.11.2020 (Version 2.08.3) (veraltet)

Version: 2.08.3

Aurora MySQL 2.08.3 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Sie können bestehende Aurora MySQL 2.\*-Datenbank-Cluster direkt auf Aurora MySQL 2.08.3 aktualisieren. Sie können einen vorhandenen Aurora MySQL 1.\*-Cluster direkt auf 2.07.3 oder höher upgraden und dann direkt auf 2.08.3 aktualisieren.

Geben Sie zum Erstellen eines Clusters mit einer älteren Version von Aurora MySQL die Engine-Version über AWS Management Console, AWS CLI oder die RDS-API an.

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

### Note

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Fehlerbehebungen bei der Sicherheit:

Korrekturen und andere Verbesserungen bei der Feinabstimmung der Handhabung in einer verwalteten Umgebung. Weitere CVE Fixes unten:

- [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](#)

Inkompatible Änderungen:

Diese Version führt eine Berechtigungsänderung ein, die sich auf das Verhalten des `mysqldump` Befehls auswirkt. Benutzer müssen das `PROCESS` Recht haben, auf die `INFORMATION_SCHEMA.FILES` Tabelle zuzugreifen. Um den `mysqldump` Befehl ohne Änderungen



auszuführen, erteilen Sie dem Datenbankbenutzer, mit dem sich der `PROCESS` Befehl verbindet, die `mysqldump` Berechtigung. Sie können den `mysqldump` Befehl auch mit der `--no-tablespaces` Option ausführen. Mit dieser Option enthält die `mysqldump` Ausgabe keine `CREATE LOGFILE GROUP` oder `CREATE TABLESPACE` Anweisungen. In diesem Fall greift der `mysqldump` Befehl nicht auf die `INFORMATION_SCHEMA.FILES` Tabelle zu, und Sie müssen die `PROCESS` Berechtigung nicht erteilen.

## Integration von MySQL-Fehlerbehebungen (Community Edition):

- Bug #23762382 - WERTE EINFÜGEN ABFRAGE MIT JOIN IN EINEM SELECT VERURSACHT FALSCHES VERHALTEN.
- Bug #25209512 - CURRENT\_TIMESTAMP ERZEUGT NULLEN IM TRIGGER.

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora MySQL Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen für den synchronen Aufruf von AWS Lambda-Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native

Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`

## Aurora-MySQL-Datenbank-Engine-Updates 28.08.2020 (Version 2.08.2) (veraltet)

Version: 2.08.2

Aurora MySQL 2.08.2 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Sie können einen Snapshot aus einer aktuell unterstützten Aurora MySQL-Version in Aurora MySQL 2.08.2 wiederherstellen. Sie haben auch die Möglichkeit, bestehende Aurora MySQL 2.\* Datenbank-Cluster auf Aurora MySQL 2.08.2 upzugraden. Sie können einen vorhandenen Aurora MySQL 1.\*-Cluster nicht direkt auf Aurora MySQL 2.08.2 aktualisieren. Sie können jedoch seinen Snapshot zu Aurora MySQL 2.08.2 wiederherstellen. Weitere Informationen zur Wiederherstellung von Snapshots finden Sie unter [Wiederherstellen aus einem DB-Cluster-Snapshot](#) im Amazon-Aurora-Benutzerhandbuch.

Geben Sie zum Erstellen eines Clusters mit einer älteren Version von Aurora MySQL die Engine-Version über AWS Management Console, AWS CLI oder die RDS-API an.

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

#### Note

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Kritische Fehlerbehebungen:

- Es wurde ein Problem behoben, das zu einem ungeplanten Ausfall und einer Beeinträchtigung der Datenbankverfügbarkeit führen könnte.

Verfügbarkeitskorrekturen:

- Es wurde ein Problem behoben, durch das die Aurora MySQL-Datenbank neu gestartet werden konnte, wenn es sich um ein Binlog-Replikat handelte und ein DDL-Ereignis über die mysql-Tabelle `time_zone` repliziert wurde.

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora MySQL Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.

- Native Funktionen für den synchronen Aufruf von AWS Lambda-Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung CREATE TABLESPACE

# Aurora-MySQL-Datenbank-Engine-Updates 18.06.2020 (Version 2.08.1) (veraltet)

Version: 2.08.1

Aurora MySQL 2.08.1 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Sie können einen Snapshot aus einer aktuell unterstützten Aurora MySQL-Version zu Aurora MySQL 2.08.1 wiederherstellen. Sie haben die Möglichkeit, bestehende Aurora MySQL 2.\* Datenbank-Cluster auf Aurora MySQL 2.08.1 upzugraden. Sie können einen vorhandenen Aurora MySQL 1.\*-Cluster nicht direkt auf Aurora MySQL 2.08.1 aktualisieren. Sie können jedoch seinen Snapshot zu Aurora MySQL 2.08.1 wiederherstellen.

Geben Sie zum Erstellen eines Clusters mit einer älteren Version von Aurora MySQL die Engine-Version über AWS Management Console, AWS CLI oder die RDS-API an.

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Note

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Neue Funktionen:

- Globale Datenbank-Schreibweiterleitung. In einer Aurora Global Database können Sie jetzt bestimmte Schreiboperationen ausführen, z. B. DML-Anweisungen, während Sie mit einem sekundären Cluster verbunden sind. Die Schreiboperationen werden an den primären Cluster weitergeleitet, und alle Änderungen werden auf die sekundären Cluster repliziert. Weitere

Informationen finden Sie unter [Verwenden der Schreibweiterleitung in einer globalen Amazon-Aurora-Datenbank](#) im Amazon-Aurora-Benutzerhandbuch.

Allgemeine Fehlerbehebungen zur Erhöhung der Stabilität:

- Es wurde ein Problem behoben, bei dem das Wiederherstellen eines Clusters von Amazon S3 zu Aurora MySQL-Version 2.08.0 länger dauerte als erwartet, wenn das S3-Backup die `mysql.host-` Tabelle nicht enthielt.

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora MySQL Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen für den synchronen Aufruf von AWS Lambda-Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`

## Aurora-MySQL-Datenbank-Engine-Updates 02.06.2020 (Version 2.08.0) (veraltet)

Version: 2.08.0

Aurora MySQL 2.08.0 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Sie können einen Snapshot aus einer derzeit unterstützten Aurora MySQL-Version auf Aurora MySQL 2.08.0 wiederherstellen. Sie haben die Möglichkeit, bestehende Aurora MySQL 2.\* Datenbank-Cluster auf Aurora MySQL 2.08.0 upzugraden. Sie können einen vorhandenen Aurora MySQL 1.\*-Cluster nicht direkt auf Aurora MySQL 2.08.0 aktualisieren. Sie können jedoch seinen Snapshot zu Aurora MySQL 2.08.0 wiederherstellen.

Geben Sie zum Erstellen eines Clusters mit einer älteren Version von Aurora MySQL die Engine-Version über AWS Management Console, AWS CLI oder die RDS-API an.

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

### Note

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

### Neue Funktionen:

- Verbesserte Binärprotokoll-Verarbeitung zur Reduzierung der Wiederherstellungszeit nach einem Absturz und der Commit-Zeitlatenz bei sehr großen Transaktionen.
- Starten der Funktion Datenbankaktivitäts-Streams (DAS) für Aurora MySQL. Diese Funktion bietet einen nahezu Echtzeit-Datenstrom der Datenbankaktivitäten in Ihrer relationalen Datenbank, um Sie bei der Überwachung der Aktivitäten zu unterstützen. Weitere Informationen finden Sie unter [Überwachung von Amazon Aurora mit Datenbankaktivitäts-Streams](#) im Amazon-Aurora-Benutzerhandbuch.
- Aktualisierte Zeitzonendateien zur Unterstützung der neusten Zeitzoneänderung in Brasilien.
- Neue Schlüsselwörter in SQL eingeführt, um die Hash-Join-Funktion für eine bestimmte Tabelle und/oder innere Tabelle auszuführen: HASH\_JOIN, HASH\_JOIN\_PROBING und HASH\_JOIN\_BUILDING. Weitere Informationen finden Sie in den [Aurora-MySQL-Hinweisen](#) im Amazon-Aurora-Benutzerhandbuch.
- In Aurora MySQL 5.7 wurde die Unterstützung für Join-Reihenfolge-Hinweise durch Rückportierung einer [MySQL 8.0-Funktion](#) eingeführt. Die neuen Hinweise sind JOIN\_FIXED\_ORDER, JOIN\_ORDER, JOIN\_PREFIX und JOIN\_SUFFIX. Eine ausführliche Dokumentation zur Unterstützung für Join-Reihenfolge-Hinweise finden Sie unter [WL#9158: Join Order Hints](#).
- Das Aurora Machine Learning unterstützt jetzt benutzerdefinierte Funktionen mit MEDIUMINT als Rückgabebetyp.
- Die lambda\_async() -gespeicherte Prozedur unterstützt nun alle MySQL-utf8-Zeichen.

### Wichtige Fehlerbehebungen:



- Es wurde ein Problem behoben, das dazu führen konnte, dass eine DB-Reader-Instance unvollständige Ergebnisse für eine FTS-Abfrage zurückgab, nachdem die Tabelle `INFORMATION_SCHEMA.INNODB_SYS_TABLES` auf der DB-Writer-Instance abgefragt wurde.
- [CVE-2019-5443](#)
- [CVE-2019-3822](#)

#### Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem behoben, das zu einem Neustart der Datenbank führte, nachdem eine Anweisung mit mehreren Abfragen, die auf mehrere Tabellen oder Datenbanken zugreift, bei aktiviertem Abfrage-Cache ausgeführt wurde.
- Es wurde eine race-Bedingung im Lock-Manager behoben, die zu einem Neustart oder Failover der Datenbank beim Transaktionsrollback führte.
- Es wurde ein Problem behoben, das einen Neustart oder Failover der Datenbank auslöste, wenn mehrere Verbindungen versuchen, dieselbe Tabelle mit einem Volltextsuchindex zu aktualisieren.
- Es wurde ein Problem behoben, das einen Neustart oder Failover der Datenbank während eines `kill session`-Befehls auslösen konnte. Wenn dieses Problem auftritt, wenden Sie sich an den AWS Support, um diesen Fix auf Ihrer Instance zu aktivieren.
- Es wurde ein Problem behoben, das zum Neustart der DB-Reader-Instance während einer mehrere Anweisungen umfassende Transaktion mit mehreren `SELECT`-Anweisungen und einer hohen Schreib-Workload auf der DB-Writer-Instance bei aktiviertem `AUTOCOMMIT` führte.
- Es wurde ein Problem behoben, das einen Neustart der DB-Reader-Instance nach der Ausführung lang andauernder Abfragen verursachte, während die DB-Writer-Instance unter einer hohen OLTP-Schreib-Workload stand.

#### Allgemeine Verbesserungen:

- Verbesserte Datenbank-Wiederherstellungszeit und Commit-Latenz für lang andauernde Transaktionen bei aktiviertem Binärprotokoll.
- Der Algorithmus wurde verbessert, um bessere Statistiken für die Schätzung unterschiedlicher Wertzählungen auf indizierten Spalten, einschließlich Spalten mit verzerrten Datenverteilungen, zu erzeugen.
- Es wurde die Antwortzeit und CPU-Auslastung von Join-Abfragen reduziert, die auf temporäre MyISAM-Tabellen zugreifen, und die Ergebnisse werden in den lokalen Speicher überführt.

- Es wurde ein Problem behoben, das verhinderte, dass Aurora MySQL 5.6-Snapshots mit Datenbank- oder Tabellennamen, die Leerzeichen enthalten, in einem neuen Aurora MySQL 5.7-Cluster wiederhergestellt werden konnten.
- Es wurden Informationen über die Transaktion von Opfern aufgenommen, wenn das Deadlock in gelöst ist `show engine innodb status`.
- Es wurde ein Problem behoben, das dazu führte, dass Verbindungen hängen blieben, wenn Clients mit mehreren verschiedenen Versionen mit derselben Datenbank verbunden sind und auf den Abfrage-Cache zugreifen.
- Es wurde ein Speicherleck behoben, das durch mehrfache Aufrufe des ZDP-(Zero-Downtime Patch)- oder ZDR-(Zero-Downtime Restart)-Workflows während der Lebensdauer einer Datenbank-Instance entstand.
- Es wurde eine Fehlermeldung bei Zero-Downtime Patch (ZDP)- oder Zero-Downtime Restart (ZDR)-Operationen behoben, die fälschlicherweise angab, dass die letzte Transaktion abgebrochen wurde, wenn das Auto-Commit-Flag deaktiviert ist.
- Es wurde ein Problem bei ZDP-Operationen (Zero-Downtime Patch) behoben, das beim Wiederherstellen von Benutzersitzungsvariablen im neuen Datenbankprozess zu einer Serverausfall-Fehlermeldung führen konnte.
- Es wurde ein Problem bei ZDP-Operationen (Zero Downtime Patch) behoben, das zu sporadischen Datenbankausfällen führen konnte, wenn während des Patchens lange andauernde Abfragen ausgeführt wurden.
- Es wurde ein Problem behoben, bei dem Abfragen, die eine Aurora-Machine-Learning-Funktion enthalten, leere Fehlermeldungen aufgrund einer falsch behandelten Fehlerantwort von Machine-Learning-Services wie Amazon Sagemaker und Amazon Comprehend zurückgaben.
- Es wurde ein Problem in der Funktion zur Überwachung auf fehlenden Speicherplatz behoben, bei dem ein benutzerdefinierter Wert des `table_definition_cache`-Parameters nicht berücksichtigt wurde.
- Die Fehlermeldung „Query execution was interrupted“ (Abfrageausführung wurde unterbrochen) wird zurückgegeben, wenn eine Aurora-Machine-Learning-Abfrage unterbrochen wird. Zuvor wurde stattdessen die generische Meldung „Internal error in processing ML request“ (Interner Fehler bei Verarbeitung der ML-Anforderung) zurückgegeben.
- Es wurde ein Problem behoben, das bei einem Binärprotokoll-Worker zu einem Verbindungstimeout führen konnte, wenn der Parameter `slave_net_timeout` kleiner als der Parameter `aurora_binlog_replication_max_yield_seconds` ist und die Workload auf dem Binärprotokoll-Master-Cluster gering ist.

- Verbesserte Überwachung des Binärprotokoll-Wiederherstellungsfortschritts durch Ausgabe von Informationsmeldungen im Fehlerprotokoll mit einer Frequenz von einer Meldung pro Minute.
- Es wurde ein Problem behoben, das dazu führen konnte, dass aktive Transaktionen von der `SHOW ENGINE INNODB STATUS`-Abfrage nicht gemeldet wurden.

## Integration von MySQL-Fehlerbehebungen (Community Edition):

- [Fehler #25289359](#): Eine bei der Datensynchronisierung vorgenommene Volltext-Cache-Sperre wurde nicht freigegeben, wenn die Volltext-Cache-Größe das Volltext-Cache-Größenlimit überschritt.
- [Fehler #29138644](#): Das manuelle Ändern der Systemzeit bei ausgeführtem MySQL-Server verursachte Verzögerungen in den Page-Cleaner-Threads.
- [Fehler #25222337](#): Ein Feldname einer virtuellen NULL-Spalte in einem virtuellen Index führte zu einem Beenden des Servers während eines Feldnamensabgleichs, der beim Befüllen von virtuellen Spalten auftritt, die von einer Fremdschlüsselbeschränkung betroffen sind.
- [Fehler #25053286](#): Das Ausführen einer gespeicherten Prozedur, die eine Abfrage enthält, die auf eine Ansicht zugreift, konnte Speicher zuweisen, der bis zum Ende der Sitzung nicht freigegeben wurde.
- [Fehler #25586773](#): Das Ausführen einer gespeicherten Prozedur, die eine Anweisung enthält, die aus dem Inhalt bestimmter `SELECT`-Anweisungen eine Tabelle erzeugt hat, konnte zu einem Speicherleck führen.
- [Fehler #28834208](#): Während der Protokollanwendung, nach einer `OPTIMIZE TABLE`-Operation, füllte InnoDB die virtuellen Spalten nicht auf, bevor nach Aktualisierungen des virtuellen Spaltenindexes gesucht wurde.
- [Fehler #26666274](#): Endlosschleife im Leistungsschema-Puffercontainer aufgrund eines Überlaufs einer 32-Bit-Ganzzahl ohne Vorzeichen.

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora MySQL Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.

- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen für den synchronen Aufruf von AWS Lambda-Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung CREATE TABLESPACE

# Aurora-MySQL-Datenbank-Engine-Updates 15.08.2023 (Version 2.07.10, kompatibel mit MySQL 5.7.12)

Version: 2.07.10

Aurora MySQL 2.07.10 ist allgemein verfügbar. Aurora-MySQL-2.07-Versionen sind kompatibel mit MySQL 5.7.12. Weitere Informationen zu Community-Änderungen finden Sie unter [Änderungen in MySQL 5.7.12 \(11.04.2016, Allgemeine Verfügbarkeit\)](#).

## Note

Diese Version ist als Long-Term Support- (LTS, Langzeit-Support)-Version ausgewiesen. Weitere Informationen finden Sie unter [Aurora-MySQL-Long-Term-Support \(LTS, Langzeit-Support\)-Versionen](#) im Amazon-Aurora-Benutzerhandbuch.

Derzeit werden die Aurora-MySQL-Versionen 2.07.\*, 2.11.\*, 3.01.\*, 3.02.\*, 3.03.\* und 3.04\* unterstützt.

Sie können einen Snapshot von einer aktuell unterstützten Aurora-MySQL-Version in Aurora MySQL 2.07.10 wiederherstellen. Sie haben auch die Möglichkeit, bestehende Aurora-MySQL-2.\*-Datenbank-Cluster zu Aurora MySQL 2.07.10 zu aktualisieren. Ein direktes Upgrade ist für Aurora-MySQL-1.\*-Cluster zu Aurora MySQL 2.\* verfügbar (siehe [Upgrade von Aurora MySQL 1.x zu 2.x](#)). Ein direktes Upgrade ist für Aurora-MySQL-2.\*-Cluster zu Aurora MySQL 3.\* verfügbar (siehe [Upgrade von Aurora MySQL 2.x zu 3.x](#)).

Unmittelbar nach der Durchführung eines In-Place-Engine-Versions-Upgrades zu Aurora MySQL 2.07.10\* wird automatisch ein Betriebssystem-Upgrade auf alle betroffenen Instances der DB-Instance-Klassen db.r4, db.r5, db.t2 und db.t3 angewendet, sofern auf den Instances eine alte Betriebssystemversion ausgeführt wird. In einem Multi-AZ-DB-Cluster wenden alle Reader-Instances zuerst das Betriebssystem-Upgrade an. Wenn das Betriebssystem-Upgrade in der ersten Reader-Instance abgeschlossen ist, erfolgt ein Failover und die vorherige Writer-Instance wird aktualisiert.

## Note

Das Betriebssystem-Upgrade wird bei Hauptversions-Upgrades nicht automatisch auf die globalen Aurora-Datenbanken angewendet.

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Behobene Sicherheitsprobleme und CVEs sind unten aufgeführt:

Korrekturen und andere Verbesserungen bei der Feinabstimmung der Handhabung in einer verwalteten Umgebung. Weitere CVE-Korrekturen unten:

- [CVE-2023-21963](#)
- [CVE-2023-21912](#)
- [CVE-2023-0215](#)
- [CVE-2022-43551](#)
- [CVE-2022-37434](#)
- Es wurde ein Problem behoben, bei dem Ereignisse, die während der Verarbeitung von Prüfprotokoll-Rotationen gemeldet wurden, möglicherweise nicht in das Prüfprotokoll geschrieben wurden.
- Die von Aurora MySQL verwendeten Standard-SSL-Verschlüsselungen wurden aktualisiert, um die weniger sicheren DES-CBC3-SHA-Werte aus dem [SSL\\_CIPHER](#)-Datenbankparameter auszuschließen. Wenn Sie aufgrund der Entfernung der DES-CBC3-SHA-Verschlüsselung auf SSL-Verbindungsprobleme stoßen, verwenden Sie bitte eine entsprechende sichere Verschlüsselung aus dieser Liste, [Konfigurieren von Verschlüsselungs-Suites für Verbindungen zu Aurora-MySQL-DB-Clustern](#).
- OpenSSL wurde zu Version 1.0.2zh aktualisiert.

Allgemeine Verbesserungen:

- Es wurde Unterstützung für ECDHE-RSA-SSL-Verschlüsselungen hinzugefügt, die kleinere Schlüsselgrößen für die Verschlüsselung verwenden.
- Ein Speicherverwaltungsproblem bei der Ausführung von Abfragen mit Hash-Joins wurde behoben.

## In Aurora-MySQL-Version 2 nicht unterstützte Funktionen

Die folgenden Features werden derzeit in Aurora MySQL Version 2 nicht unterstützt (kompatibel mit MySQL 5.7):

- Asynchrones Key Prefetch (AKP).
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Die SQL-Anweisung CREATE TABLESPACE
- Plugin für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- X Protocol

# Aurora-MySQL-Datenbank-Engine-Updates 04.05.2023 (Version 2.07.9, kompatibel mit MySQL 5.7.12)

Version: 2.07.9

Aurora MySQL 2.07.9 ist allgemein verfügbar. Aurora-MySQL-2.07-Versionen sind kompatibel mit MySQL 5.7.12. Weitere Informationen zu Community-Änderungen finden Sie unter [Änderungen in MySQL 5.7.12 \(11.04.2016, Allgemeine Verfügbarkeit\)](#).

## Note

Diese Version ist als Long-Term Support- (LTS, Langzeit-Support)-Version ausgewiesen. Weitere Informationen finden Sie unter [Aurora-MySQL-Long-Term-Support \(LTS, Langzeit-Support\)-Versionen](#) im Amazon-Aurora-Benutzerhandbuch.

Derzeit werden die Aurora-MySQL-Versionen 2.07.\*, 2.11.\*, 3.01.\*, 3.02.\* und 3.03.\* unterstützt.

Sie können einen Snapshot von einer aktuell unterstützten Aurora-MySQL-Version in Aurora MySQL 2.07.9 wiederherstellen. Sie haben auch die Möglichkeit, bestehende Aurora-MySQL-2.\*-Datenbank-Cluster zu Aurora MySQL 2.07.9 zu aktualisieren. Ein direktes Upgrade ist für Aurora-MySQL-1.\*-Cluster zu Aurora MySQL 2.\* verfügbar (siehe [Upgrade von Aurora MySQL 1.x zu 2.x](#)). Ein direktes Upgrade ist für Aurora-MySQL-2.\*-Cluster zu Aurora MySQL 3.\* verfügbar (siehe [Upgrade von Aurora MySQL 2.x zu 3.x](#)).

Unmittelbar nach der Durchführung eines In-Place-Engine-Versions-Upgrades zu Aurora MySQL 2.07.9\* wird automatisch ein Betriebssystem-Upgrade auf alle betroffenen Instances der DB-Instance-Klassen db.r4, db.r5, db.t2 und db.t3 angewendet, sofern auf den Instances eine alte Betriebssystemversion ausgeführt wird. In einem Multi-AZ-DB-Cluster wenden alle Reader-Instances zuerst das Betriebssystem-Upgrade an. Wenn das Betriebssystem-Upgrade in der ersten Reader-Instance abgeschlossen ist, erfolgt ein Failover und die vorherige Writer-Instance wird aktualisiert.

## Note

Das Betriebssystem-Upgrade wird bei Hauptversions-Upgrades nicht automatisch auf die globalen Aurora-Datenbanken angewendet.



Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Behobene Sicherheitsprobleme und CVEs sind unten aufgeführt:

Korrekturen und andere Verbesserungen bei der Feinabstimmung der Handhabung in einer verwalteten Umgebung. Weitere CVE-Korrekturen unten:

- [CVE-2022-32221](#)

Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem behoben, bei dem die Protokollrotation von Advanced Audit den freien Speicherplatz reduzieren kann, was dazu führen könnte, dass die Datenbank-Instance neu gestartet wird.
- Es wurde ein Problem behoben, das bei Datenbankneustarts auftreten kann und dazu führt, dass die Datenbank über einen längeren Zeitraum nicht erfolgreich gestartet wird.

Allgemeine Verbesserungen:

- Es wurde ein Problem behoben, das unter seltenen Bedingungen dazu führen kann, dass die Instances neu gestartet werden, wenn das Datenbank-Volume ein Vielfaches von 160 GB erreicht.

## In Aurora-MySQL-Version 2 nicht unterstützte Funktionen

Die folgenden Features werden derzeit in Aurora MySQL Version 2 nicht unterstützt (kompatibel mit MySQL 5.7):

- Asynchrones Key Prefetch (AKP).
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`

## Aurora-MySQL-Datenbank-Engine-Updates 16.06.2022 (Version 2.07.8) (veraltet)

Version: 2.07.8

Aurora MySQL 2.07.8 ist allgemein verfügbar. Aurora MySQL 2.\*-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.\*-Versionen sind mit MySQL 5.6 kompatibel.

### Note

Diese Version ist als Long-Term Support- (LTS, Langzeit-Support)-Version ausgewiesen. Weitere Informationen finden Sie unter [Versionen von Aurora MySQL Long-Term Support \(LTS\)](#) im Amazon-Aurora-Benutzerhandbuch.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Sie können einen Snapshot von einer aktuell unterstützten Aurora-MySQL-Version in Aurora MySQL 2.07.8 wiederherstellen. Sie haben auch die Möglichkeit, bestehende Aurora-MySQL-2.\*-Datenbank-Cluster zu Aurora MySQL 2.07.8 zu aktualisieren. Ein direktes Upgrade ist für Aurora-MySQL-1.\*-Cluster zu Aurora MySQL 2.\* verfügbar (siehe [Upgrade von Aurora MySQL 1.x zu 2.x](#)). Ein direktes Upgrade ist für Aurora-MySQL-2.\*-Cluster zu Aurora MySQL 3.\* verfügbar (siehe [Upgrade von Aurora MySQL 2.x zu 3.x](#)).

Geben Sie zum Erstellen eines Clusters mit einer älteren Version von Aurora MySQL die Engine-Version über AWS Management Console, AWS CLI oder die RDS-API an.

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

#### Note

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Fehlerbehebungen bei der Sicherheit:

Korrekturen und andere Verbesserungen bei der Feinabstimmung der Handhabung in einer verwalteten Umgebung. Weitere CVE-Korrekturen unten:

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

Allgemeine Verbesserungen:

- Es wurde ein Problem behoben, das unter seltenen Bedingungen dazu führen kann, dass der Datenbank-Server neu gestartet wird, wenn der Deadlock-Detector-Thread aufgrund einer Race-Bedingung stecken bleibt.

## Integration von MySQL-Fehlerbehebungen (Community Edition):

- Wenn für ein UPDATE eine temporäre Tabelle mit einem Primärschlüssel von mehr als 1024 Byte erforderlich war und diese Tabelle mit InnoDB erstellt wurde, konnte der Server beendet werden. (Fehler #25153670)

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora MySQL Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen für den synchronen Aufruf von AWS Lambda-Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine

> 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`

## Aurora-MySQL-Datenbank-Engine-Updates 24.11.2021 (Version 2.07.7) (veraltet)

Version: 2.07.7

Aurora MySQL 2.07.7 ist allgemein verfügbar. Aurora MySQL 2.\*-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.\*-Versionen sind mit MySQL 5.6 kompatibel.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Sie können einen Snapshot von einer aktuell unterstützten Aurora-MySQL-Version in Aurora MySQL 2.07.7 wiederherstellen. Sie haben auch die Möglichkeit, bestehende Aurora-MySQL-2.\*-Datenbank-Cluster zu Aurora MySQL 2.07.7 zu aktualisieren. Sie können einen vorhandenen Aurora-MySQL-1.\*-Cluster nicht direkt zu 2.07.7 aktualisieren. Sie können jedoch seinen Snapshot zu Aurora MySQL 2.07.7 wiederherstellen.

Geben Sie zum Erstellen eines Clusters mit einer älteren Version von Aurora MySQL die Engine-Version über AWS Management Console, AWS CLI oder die RDS-API an.

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

### Note

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Fehlerbehebungen bei der Sicherheit:

Korrekturen und andere Verbesserungen bei der Feinabstimmung der Handhabung in einer verwalteten Umgebung. Weitere CVE-Korrekturen unten:

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

Allgemeine Verbesserungen:

- Sicherheitsprobleme im Zusammenhang mit der Aurora-MySQL-Integration mit anderen AWS-Services wie Amazon S3, Amazon ML, Lambda behoben.
- Es wurde ein Fehler mit falscher Meldung einer Aurora-Replikationsverzögerung behoben.
- Es wurde ein Problem behoben, das dazu führen kann, dass ein Neustart der Datenbank-Instance fehlschlug, wenn die Datenbank über eine große Anzahl von Benutzer- und Berechtigungskombinationen verfügt.
- Es wurde ein Problem behoben, das dazu führen kann, dass general\_log- und slow\_log-Tabellen nach dem In-Place-Upgrade von Aurora MySQL 1.x (basierend auf MySQL 5.6) auf Aurora MySQL 2.x (basierend auf MySQL 5.7) unzugänglich wurden.
- Es wurde ein Problem behoben, das unter seltenen Bedingungen dazu führen kann, dass eine Reader-Instance aufgrund einer falschen Prüfverarbeitung neu gestartet wurde.
- Es wurde ein Problem behoben, das unter seltenen Bedingungen das Diagramm „Database Load“ in Performance Insights (PI) -Sitzungen so aktiv anzeigt, dass die CPU aktiv verwendet wurde, obwohl die Sitzungen mit der Verarbeitung abgeschlossen waren und im Leerlauf sind.

- Es wurde ein Problem mit einer parallelen Abfrage behoben, durch das die Datenbank beim Ausführen von SQL-Anweisungen mit einer LIMIT-Klausel neu gestartet wurde.
- Es wurde ein Problem behoben, durch das der Wert einer TIMESTAMP-Spalte einer vorhandenen Zeile auf den neuesten Zeitstempel aktualisiert wurde, wenn alle folgenden Bedingungen erfüllt sind: 1. Für die Tabelle existiert ein Auslöser; 2. Ein INSERT wird für die Tabelle ausgeführt, die eine ON DUPLICATE KEY UPDATE-Klausel hat; 3. Die eingefügte Zeile kann eine Verletzung doppelter Werte in einem UNIQUE-Index oder PRIMARY KEY verursachen; und 4. eine oder mehrere Spalten sind vom TIMESTAMP-Datentyp und haben einen Standardwert von CURRENT\_TIMESTAMP.
- Es wurde ein Problem behoben, das unter seltenen Bedingungen dazu führen kann, dass die Datenbank-Instance bei Verwendung von XA-Transaktionen in READ COMMITTED Isolationsstufe neu gestartet wurde.

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora MySQL Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen für den synchronen Aufruf von AWS Lambda-Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`

## Aurora-MySQL-Datenbank-Engine-Updates 02.09.2021 (Version 2.07.6) (veraltet)

### Version 2.07.6

Aurora MySQL 2.07.6 ist allgemein verfügbar. Aurora MySQL 2.\*-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.\*-Versionen sind mit MySQL 5.6 kompatibel.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Sie können einen Snapshot von einer aktuell unterstützten Aurora MySQL-Version in Aurora MySQL 2.07.6 wiederherstellen. Sie haben auch die Möglichkeit, bestehende Aurora MySQL 2.\* Datenbank-



Cluster auf Aurora MySQL 2.07.6 zu aktualisieren. Sie können einen bestehenden Aurora MySQL 1.\*-Cluster nicht direkt auf 2.07.6 aktualisieren; Sie können jedoch seinen Snapshot auf Aurora MySQL 2.07.6 wiederherstellen.

Um einen Cluster mit einer älteren Version von Aurora MySQL zu erstellen, geben Sie bitte die Engine-Version über die AWS Management Console, die AWS CLI oder die RDS-API an.

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Integration von MySQL-Fehlerbehebungen (Community Edition):

- DAS EINFÜGEN VON DATENSÄTZEN DER GRÖSSE 64K NIMMT ZU VIEL ZEIT IN ANSPRUCH. ([Fehler #23031146](#))

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora MySQL Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen für den synchronen Aufruf von AWS Lambda-Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`

## Aurora-MySQL-Datenbank-Engine-Updates 06.07.2021 (Version 2.07.5) (veraltet)

Version: 2.07.5

Aurora MySQL 2.07.5 ist allgemein verfügbar. Aurora MySQL 2.\*-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.\*-Versionen sind mit MySQL 5.6 kompatibel.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Sie können einen Snapshot aus einer derzeit unterstützten Aurora MySQL-Version auf Aurora MySQL 2.07.5 wiederherstellen. Sie haben die Möglichkeit, bestehende Aurora-MySQL 2.\*-

Datenbank-Cluster auf Aurora MySQL 2.07.5 upzugraden. Sie können einen vorhandenen Aurora MySQL 1.\*-Cluster nicht direkt auf Aurora MySQL 2.07.5 aktualisieren. Sie können jedoch seinen Snapshot zu Aurora MySQL 2.07.5 wiederherstellen.

Geben Sie zum Erstellen eines Clusters mit einer älteren Version von Aurora MySQL die Engine-Version über AWS Management Console, AWS CLI oder die RDS-API an.

#### Note

Diese Version ist als Long-Term Support- (LTS, Langzeit-Support)-Version ausgewiesen. Weitere Informationen finden Sie unter [Aurora-MySQL-Long-Term-Support \(LTS, Langzeit-Support\)-Versionen](#) im Amazon-Aurora-Benutzerhandbuch.

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem behoben, bei dem Sperren auf Benutzerebene auf einem Aurora-Replikat nicht zulässig sind.
- Es wurde ein Problem behoben, das einen Neustart einer Datenbank verursachen konnte, wenn XA-Transaktionen in Isolationsstufe READ COMMITTED verwendet wurden.
- Die maximal zulässige Länge wurde auf 2000 für die globalen Parameter `server_audit_incl_users` und `server_audit_excl_users` erweitert.

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora MySQL Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.

- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen für den synchronen Aufruf von AWS Lambda-Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung CREATE TABLESPACE

# Aurora-MySQL-Datenbank-Engine-Updates 04.03.2021 (Version 2.07.4) (veraltet)

Version: 2.07.4

Aurora MySQL 2.07.4 ist allgemein verfügbar. Aurora MySQL 2.\*-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.\*-Versionen sind mit MySQL 5.6 kompatibel.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Sie können einen Snapshot aus einer derzeit unterstützten Aurora MySQL-Version zu Aurora MySQL 2.07.4 wiederherstellen. Sie haben auch die Möglichkeit bestehende Aurora MySQL 2.\*-Datenbank-Cluster auf Aurora MySQL 2.07.4 aktualisieren. Sie können einen vorhandenen Aurora MySQL 1.\*-Cluster nicht direkt auf Aurora MySQL 2.07.4 aktualisieren. Sie können jedoch seinen Snapshot zu Aurora MySQL 2.07.4 wiederherstellen.

Um einen Cluster mit einer älteren Version von Aurora MySQL zu erstellen, geben Sie bitte die Engine-Version über die AWS Management Console, die AWS CLI oder die RDS-API an.

## Note

Diese Version ist als Long-Term Support- (LTS, Langzeit-Support)-Version ausgewiesen. Weitere Informationen finden Sie unter [Aurora-MySQL-Long-Term-Support \(LTS, Langzeit-Support\)-Versionen](#) im Amazon-Aurora-Benutzerhandbuch.

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Fehlerbehebungen bei der Sicherheit:

- [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](#)

#### Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem behoben, das dazu führen könnte, dass ein Client im Falle eines Netzwerkfehlers beim Lesen oder Schreiben eines Netzwerkpakets hängen bleibt.
- In einigen Fällen wurden die Wiederanlaufzeiten der Engine nach unterbrochener DDL verbessert.
- Es wurde ein Problem behoben, bei dem eine DDL oder DML während einer Prefetch-Anfrage zu einem Neustart einer Engine des Motors führen konnte.
- Es wurde ein Problem behoben, bei dem eine Replica neu gestartet werden konnte, während ein Reverse-Scan einer Tabelle/eines Index für eine Aurora-Read Replica ausgeführt wurde.
- Es wurde ein Problem beim Clone-Cluster-Betrieb behoben, das dazu führen konnte, dass der Klon länger dauert.
- Es wurde ein Problem behoben, das einen Neustart einer Datenbank verursachen könnte, wenn die parallele Abfrageoptimierung für geospatiale Spalten verwendet wird.
- Es wurde ein Problem behoben, das dazu führte, dass ein Binlog-Replikat mit einem HA\_ERR\_KEY\_NOT\_FOUND-Fehler stoppt.

#### Integration von MySQL-Fehlerbehebungen (Community Edition):

- Es wurde ein Problem im Volltext-Ngram-Parser behoben, das bei Token mit „ „ (Leerzeichen), „%“ oder „,“ bestand. Kunden sollten ihre FTS-Indizes neu erstellen, wenn sie einen Ngram-Parser verwenden. (Bug #25873310)
- Ein Problem wurde behoben, das bei der Ausführung von Abfragen mit verschachtelten SQL-Ansichten zu einem Neustart der Engine führen konnte. (Bug #27214153, Bug #26864199)

#### Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora MySQL Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen für den synchronen Aufruf von AWS Lambda-Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter

- Die SQL-Anweisung CREATE TABLESPACE

## Aurora-MySQL-Datenbank-Engine-Updates 10.11.2020 (Version 2.07.3) (veraltet)

Version: 2.07.3

Aurora MySQL 2.07.3 ist allgemein verfügbar. Aurora MySQL 2.\*-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.\*-Versionen sind mit MySQL 5.6 kompatibel.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Sie können einen Snapshot aus einer derzeit unterstützten Aurora MySQL-Version zu Aurora MySQL 2.07.3 wiederherstellen. Sie haben auch die Möglichkeit bestehende Aurora MySQL 2.\*-Datenbank-Cluster auf Aurora MySQL 2.07.3 zu aktualisieren. Sie können einen vorhandenen Aurora MySQL 1.\*-Cluster nicht direkt auf Aurora MySQL 2.07.3 aktualisieren. Sie können jedoch seinen Snapshot zu Aurora MySQL 2.07.3 wiederherstellen.

Um einen Cluster mit einer älteren Version von Aurora MySQL zu erstellen, geben Sie bitte die Engine-Version über die AWS Management Console AWS CLI, die oder die RDS-API an.

### Note

Diese Version ist als Long-Term Support- (LTS, Langzeit-Support)-Version ausgewiesen. Weitere Informationen finden Sie unter [Aurora-MySQL-Long-Term-Support \(LTS, Langzeit-Support\)-Versionen](#) im Amazon-Aurora-Benutzerhandbuch.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Fehlerbehebungen bei der Sicherheit:

Korrekturen und andere Verbesserungen bei der Feinabstimmung der Handhabung in einer verwalteten Umgebung.



- [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](#)

#### Inkompatible Änderungen:

Diese Version führt eine Berechtigungsänderung ein, die sich auf das Verhalten des `mysqldump` Befehls auswirkt. Benutzer müssen das `PROCESS` Recht haben, auf die `INFORMATION_SCHEMA.FILES` Tabelle zuzugreifen. Um den `mysqldump` Befehl ohne Änderungen auszuführen, erteilen Sie dem Datenbankbenutzer, mit dem sich der `PROCESS` Befehl verbindet, die `mysqldump` Berechtigung. Sie können den `mysqldump` Befehl auch mit der `--no-tablespaces` Option ausführen. Mit dieser Option enthält die `mysqldump` Ausgabe keine `CREATE LOGFILE GROUP` oder `CREATE TABLESPACE` Anweisungen. In diesem Fall greift der `mysqldump` Befehl nicht auf die `INFORMATION_SCHEMA.FILES` Tabelle zu, und Sie müssen die `PROCESS` Berechtigung nicht erteilen.

#### Verbesserungen der Verfügbarkeit:

- Es wurde eine Race-Condition im Sperrmanager zwischen dem Beenden einer Verbindung/ Abfrage und dem Beenden der Sitzung behoben, die zu einem Neustart der Datenbank geführt hat.
- Es wurde ein Problem behoben, das zu einem Neustart der Datenbank führt, nachdem eine Anweisung mit mehreren Abfragen, die auf mehrere Tabellen oder Datenbanken zugreift, bei aktiviertem Abfrage-Cache ausgeführt wird.

- Es wurde ein Problem behoben, das aufgrund von Aktualisierungen virtueller Spalten mit sekundären Indizes wiederholte Neustarts verursachen konnte.

## Integration von MySQL-Fehlerbehebungen (Community Edition):

- InnoDB: Gleichzeitige XA-Transaktionen, die erfolgreich zur XA-Prüfungsphase auf dem Master liefen, führten zu einem Konflikt bei der Wiederholung auf dem Slave, was zu einem Timeout für die Sperrwartezeit im Applier-Thread führte. Der Konflikt war auf den GAP-Sperrbereich zurückzuführen, der sich unterschied, als die Transaktionen auf dem Slave seriell wiedergegeben wurden. Um diese Art von Konflikt zu verhindern, werden GAP-Sperren, die von XA-Transaktionen in der [READ COMMITTED](#)-Isolationsstufe übernommen werden, jetzt veröffentlicht (und nicht mehr geerbt), wenn XA-Transaktionen die Vorbereitungsphase erreichen. (Bug #27189701, Bug #25866046)
- InnoDB: Eine GAP-Sperre wurde während der Überprüfung von Fremdschlüsseln unnötig durchgeführt, während die [READ COMMITTED](#)-Isolationsstufe verwendet wurde. (Bug #25082593)
- Replikation: Wenn bei Verwendung von XA-Transaktionen ein Timeout für ein Zeitlimit einer Sperrung oder ein Deadlock für den Applier (SQL) -Thread auf einem Replikations-Slave aufgetreten ist, funktionierte die automatische Wiederholung nicht. Die Ursache war, dass der SQL-Thread zwar einen Rollback durchführen würde, die XA-Transaktion jedoch nicht zurücksetzen würde. Dies bedeutete, dass bei der Wiederholung der Transaktion das erste Ereignis XA START war, was ungültig war, da die XA-Transaktion bereits im Gange war, was zu einem XAER\_RMFAIL-Fehler führte. (Bug #24764800)
- Replikation: Doppelspieltransaktionen konnten manchmal den Slave-Applier blockieren, wenn die Transaktionsisolationsstufe auf [REPEATABLE READ](#) festgelegt wurde. (Bug #25040331)
- Replikation: Der von einer [SHOW SLAVE STATUS](#)-Anweisung zurückgegebene Wert für die gesamte kombinierte Größe aller vorhandenen Relay-Protokolldateien (Relay\_Log\_Space) könnte viel größer werden als der tatsächlich von den Relay-Protokolldateien belegte Speicherplatz. Der I/O-Thread hat die Variable nicht gesperrt, während sie den Wert aktualisiert hat, so dass der SQL-Thread automatisch eine Relay-Protokolldatei löschen und einen reduzierten Wert schreiben konnte, bevor der I/O-Thread mit der Aktualisierung des Werts fertig war. Der I/O-Thread schrieb dann seine ursprüngliche Größenberechnung, ignorierte das Update des SQL-Threads und fügte so den Platz für die gelöschte Datei hinzu. Der Wert Relay\_Log\_Space ist jetzt während Updates gesperrt, um gleichzeitige Aktualisierungen zu verhindern und eine genaue Berechnung sicherzustellen. (Bug #26997096, Bug #87832)

- Bei einer [INSERT](#)-Anweisung, für die die VALUES-Liste mit einer Unterabfrage, die eine Verknüpfung enthält, Werte für die zweite oder spätere Zeile erzeugte, konnte es zu einer Beendigung des Servers kommen, nachdem die erforderlichen Berechtigungen nicht aufgelöst wurden. (Bug #23762382)
- Bei einer Tabelle mit einer [TIMESTAMP](#)- oder [DATETIME](#)-Spalte mit dem Standardwert [CURRENT\\_TIMESTAMP](#) könnte die Spalte auf 0000-00-00 00:00:00 initialisiert werden, wenn die Tabelle einen BEFORE INSERT-Auslöser hat. (Bug #25209512, Bug #84077)
- Ein Server-Exit könnte aus gleichzeitigen Versuchen mehrerer Threads resultieren, Metadaten Performance Schema-Objekte zu registrieren und abzumelden. (Bug #26502135)
- Das Ausführen einer gespeicherten Prozedur, die eine Anweisung enthält, die aus dem Inhalt bestimmter [SELECT](#)-Anweisungen eine Tabelle erzeugt hat, konnte zu einem Speicherleck führen. (Bug #25586773)
- Das Ausführen einer gespeicherten Prozedur mit einer auf eine Ansicht zugreifenden Abfrage konnte Speicher zuweisen, der bis zum Ende der Sitzung nicht freigegeben wurde. (Bug #25053286)
- Bestimmte Unterabfragen-Materialisierungen konnten einen Server-Exit verursachen. Diese Abfragen erzeugen nun einen Fehler, der darauf hindeutet, dass die Materialisierung deaktiviert ist. (Bug #26402045)
- Abfragen mit zahlreichen Links-Joins waren langsam, wenn die Join-Pufferung verwendet wurde (z. B. bei Verwendung mit dem blockverschachtelten Schleifenalgorithmus). (Bug #18898433, Bug #72854)
- Der Optimierer übersprang die zweite Spalte in einem zusammengesetzten Index, als er eine Verknüpfung mit einer LIKE Klausel gegen die zweite Spalte ausführte. (Bug #28086754)

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora MySQL Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.

- Native Funktionen zum synchronen Aufrufen AWS Lambda von Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung CREATE TABLESPACE

# Aurora-MySQL-Datenbank-Engine-Updates 17.04.2020 (Version 2.07.2) (veraltet)

Version: 2.07.2

Aurora MySQL 2.07.2 ist allgemein verfügbar. Aurora MySQL 2.\*-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.\*-Versionen sind mit MySQL 5.6 kompatibel.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Sie können einen Snapshot aus einer derzeit unterstützten Aurora MySQL-Version auf Aurora MySQL 2.07.2 wiederherstellen. Sie haben die Möglichkeit, bestehende Aurora MySQL 2.\* Datenbank-Cluster auf Aurora MySQL 2.07.2 upzugraden. Sie können einen vorhandenen Aurora MySQL 1.\*-Cluster nicht direkt auf Aurora MySQL 2.07.2 aktualisieren. Sie können jedoch seinen Snapshot zu Aurora MySQL 2.07.2 wiederherstellen.

Um einen Cluster mit einer älteren Version von Aurora MySQL zu erstellen, geben Sie bitte die Engine-Version über die AWS Management Console AWS CLI, die oder die RDS-API an.

## Note

Diese Version ist als Long-Term Support- (LTS, Langzeit-Support)-Version ausgewiesen. Weitere Informationen finden Sie unter [Aurora-MySQL-Long-Term-Support \(LTS, Langzeit-Support\)-Versionen](#) im Amazon-Aurora-Benutzerhandbuch.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Fehlerbehebungen bei der Sicherheit:

- [CVE-2016-8287](#)
- [CVE-2016-5634](#)

Wichtige Fehlerbehebungen:

- Es wurde ein Problem behoben, durch das das Klonen bei einigen Datenbank-Clustern mit hohen Schreiblasten länger dauert.
- Es wurde ein Problem behoben, durch das Abfragen auf einer Reader-DB-Instance mit Ausführungsplänen mit sekundären Indizes nicht festgeschriebene Daten zurückgeben konnten. Das Problem beschränkt sich auf Daten, die von DML-Operationen (Data Manipulation Language) betroffen sind, die Primär- oder Sekundärindexschlüsselspalten ändern.

#### Allgemeine Verbesserungen:

- Es wurde ein Problem behoben, das zu einer langsamen Wiederherstellung eines Aurora 1.x-DB-Clusters mit FTS-Indizes (Full Text Search) in einem Aurora 2.x-DB-Cluster führte.
- Es wurde ein Problem behoben, das zu langsameren Wiederherstellungen eines Aurora 1.x-Datenbank-Snapshots mit partitionierten Tabellen mit Sonderzeichen in Tabellennamen in einem Aurora 2.x-DB-Cluster führte.
- Es wurde ein Problem behoben, das Fehler beim Abfragen langsamer Abfrageprotokolle und allgemeiner Protokolle in Reader-DB-Instances verursachte.

#### Integration von MySQL-Fehlerbehebungen (Community Edition):

- Fehler #23104498: Es wurde ein Problem in Performance Schema behoben, bei dem der gesamte verwendete Speicher gemeldet wurde. (<https://github.com/mysql/mysql-server/commit/20b6840df5452f47313c6f9a6ca075bfb00a96b>)
- Fehler #22551677: Es wurde ein Problem in Performance Schema behoben, das dazu führen konnte, dass die Datenbank-Engine abstürzte, wenn sie offline geschaltet wurde. (<https://github.com/mysql/mysql-server/commit/05e2386eccd32b6b444b900c9f8a87a1d8d531e9>)
- Fehler #23550835, Fehler #23298025, Fehler #81464: Es wurde ein Problem in Performance Schema behoben, bei dem eine Datenbank-Engine aufgrund der Überschreitung der Kapazität eines internen Puffers zum Absturz gebracht wurde. (<https://github.com/mysql/mysql-server/commit/b4287f93857bf2f99b18fd06f555bbe5b12debfc>)

#### Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora MySQL Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen zum synchronen Aufrufen AWS Lambda von Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter

- Die SQL-Anweisung CREATE TABLESPACE

## Aurora-MySQL-Datenbank-Engine-Updates 23.12.2019 (Version 2.07.1) (veraltet)

Version: 2.07.1

Aurora MySQL 2.07.1 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Sie können einen Snapshot aus einer derzeit unterstützten Aurora MySQL-Version auf Aurora MySQL 2.07.1 wiederherstellen. Sie haben die Möglichkeit, bestehende Aurora MySQL 2.\* Datenbank-Cluster auf Aurora MySQL 2.07.1 upzugraden. Sie können einen vorhandenen Aurora MySQL 1.\*-Cluster nicht direkt auf Aurora MySQL 2.07.1 aktualisieren. Sie können jedoch seinen Snapshot zu Aurora MySQL 2.07.1 wiederherstellen.

Um einen Cluster mit einer älteren Version von Aurora MySQL zu erstellen, geben Sie bitte die Engine-Version über die AWS Management Console AWS CLI, die oder die RDS-API an.

### Note

Diese Version ist derzeit in den folgenden AWS Regionen nicht verfügbar: AWS GovCloud (US-Ost) [us-gov-east-1], AWS GovCloud (US-West) [us-gov-west-1], China (Ningxia) [cn-northwest-1], Asien-Pazifik (Hongkong) [ap-east-1] und Naher Osten (Bahrain) [me-south-1]. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.



**Note**

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Wichtige Fehlerbehebungen:

- Es wurde ein langsames Speicherleck in bestimmten Aurora-Datenbankprotokollierungs- und Protokollierungs-Subsystem behoben, das den freizugebenden Speicher verringert.

Allgemeine Fehlerbehebungen zur Erhöhung der Stabilität:

- Ein Absturz während der Ausführung einer komplexen Abfrage mit Joins und Aggregation mit mehreren Tabellen, die intern Zwischentabellen verwenden, wurde behoben.

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora MySQL Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native

Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`

## Aurora-MySQL-Datenbank-Engine-Updates 25.11.2019 (Version 2.07.0) (veraltet)

Version: 2.07.0

Aurora MySQL 2.07.0 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Sie können einen Snapshot aus einer derzeit unterstützten Aurora MySQL-Version auf Aurora MySQL 2.07.0 wiederherstellen. Sie haben die Möglichkeit, bestehende Aurora MySQL 2.\* Datenbank-Cluster auf Aurora MySQL 2.07.0 upzugraden. Sie können einen vorhandenen Aurora MySQL 1.\*-Cluster nicht direkt auf Aurora MySQL 2.07.0 aktualisieren. Sie können jedoch seinen Snapshot zu Aurora MySQL 2.07.0 wiederherstellen.

Um einen Cluster mit einer älteren Version von Aurora MySQL zu erstellen, geben Sie bitte die Engine-Version über die AWS Management Console AWS CLI, die oder die RDS-API an.

#### Note

Diese Version ist derzeit in den folgenden AWS Regionen nicht verfügbar: AWS GovCloud (US-Ost) [us-gov-east-1], AWS GovCloud (US-West) [us-gov-west-1], China (Ningxia) [cn-northwest-1], Asien-Pazifik (Hongkong) [ap-east-1], Naher Osten (Bahrain) [me-south-1] und Südamerika (São Paulo) [sa-east-1]. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

#### Note

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Neue Funktionen:

- Globale Datenbanken ermöglichen jetzt das Hinzufügen sekundärer schreibgeschützter Replikatregionen für Datenbankcluster, die in diesen AWS Regionen bereitgestellt werden: Regionen: USA Ost (Nord-Virginia) [us-east-1], USA Ost (Ohio) [us-east-2], USA West (Nordkalifornien) [us-west-1], USA West (Oregon) [us-west-2], Europa (Irland) [eu-west-1], Europa (London) [eu-west-2], Europa (Paris) [eu-west-2] -3], Asien-Pazifik (Tokio) [ap-northeast-1], Asien-Pazifik (Seoul) [ap-northeast-2], Asien-Pazifik (Singapur) [ap-southeast-1], Asien-Pazifik (Sydney) [ap-southeast-2], Kanada (Zentral) [ca-central-1], Europa (Frankfurt) [ eu-central-1] und Asien-Pazifik (Mumbai) [ap-Süd-1].
- Amazon Aurora Machine Learning ist eine hochoptimierte Integration zwischen der Aurora MySQL-Datenbank und den Services für AWS maschinelles Lernen (ML). Aurora Machine Learning

erlaubt es Entwicklern, ihren Datenbankanwendungen eine Vielzahl an ML-basierten Prognosen hinzuzufügen, indem sie ML-Modelle aufrufen mithilfe der vertrauten SQL-Programmiersprache, die sie bereits für Datenbankentwicklung nutzen, und ohne eigene Integrationen erstellen oder separate Tools lernen zu müssen. Weitere Informationen finden Sie unter [Verwendung von Machine Learning \(ML\)-Funktionen mit Amazon Aurora](#).

- Unterstützung für ANSI-READ COMMITTED-Isolationsstufe auf den Read Replicas hinzugefügt. Diese Isolationsstufe ermöglicht das Ausführen von langandauernden Abfragen auf der Read Replica ohne Auswirkungen auf den hohen Durchsatz von Schreibvorgängen auf dem Schreiber-Knoten. Weitere Informationen finden Sie unter [Aurora MySQL-Isolationsstufen](#).

Kritische Fehlerbehebungen:

- [CVE-2019-2922](#)
- [CVE-2019-2923](#)
- [CVE-2019-2924](#)
- [CVE-2019-2910](#)

Wichtige Fehlerbehebungen:

- Behebung eines Problems mit der DDL-Wiederherstellung, das zu einer verlängerten Datenbank-Ausfallzeit führte. Cluster, die nach dem Ausführen einer Multi-Tabellen-DROP-Anweisung, wie z. B. `DROP TABLE t1, t2, t3`, nicht mehr verfügbar werden, sollten auf diese Version aktualisiert werden.
- Behebung eines Problems mit der DDL-Wiederherstellung, das zu einer verlängerten Datenbank-Ausfallzeit führte. Cluster, die nach dem Ausführen von `INPLACE ALTER TABLE-DDL`-Anweisungen nicht mehr verfügbar werden, sollten auf diese Version aktualisiert werden.

Allgemeine Fehlerbehebungen zur Erhöhung der Stabilität:

- Behebung eines Problems, das inkonsistente Daten in der `information_schema.replica_host_status`-Tabelle generiert hat.

Integration von MySQL-Fehlerbehebungen (Community Edition):

- Bug #26251621: INCORRECT BEHAVIOR WITH TRIGGER AND GCOL

- Bug #22574695: ASSERTION `!TABLE || (!TABLE->READ\_SET || BITMAP\_IS\_SET(TABLE->READ\_SET, FIEL
- Bug #25966845: INSERT ON DUPLICATE KEY GENERATE A DEADLOCK
- Bug #23070734: CONCURRENT TRUNCATE TABLES CAUSE STALL
- Bug #26191879: FOREIGN KEY CASCADES USE EXCESSIVE MEMORY
- Bug #20989615: INNODB AUTO\_INCREMENT PRODUCES SAME VALUE TWICE

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora MySQL Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Aurora MySQL 2.07.0 ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Aurora MySQL 2.07.0 bietet aktuell keine Unterstützung für die folgenden MySQL 5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser

- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`

## Aurora-MySQL-Datenbank-Engine-Updates 22.11.2019 (Version 2.06.0) (veraltet)

Version: 2.06.0

Aurora MySQL 2.06.0 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Zurzeit werden die Aurora MySQL-Versionen 1.14.\*, 1.15.\*, 1.16.\*, 1.17.\*, 1.18.\*, 1.19.\*, 2.01.\*, 2.02.\*, 2.03.\*, 2.04.\*, 2.05.\*, 2.06.\*, 2.04.\*, 2.05.\* und 2.06.\* unterstützt.

Sie können einen Snapshot aus einer derzeit unterstützten Aurora MySQL-Version auf Aurora MySQL 2.06.0 wiederherstellen. Sie haben die Möglichkeit, bestehende Aurora MySQL 2.\* Datenbank-Cluster auf Aurora MySQL 2.06.0 upzugraden. Sie können einen vorhandenen Aurora MySQL 1.\*-Cluster nicht direkt auf Aurora MySQL 2.06.0 aktualisieren. Sie können jedoch seinen Snapshot zu Aurora MySQL 2.06.0 wiederherstellen.

Um einen Cluster mit einer älteren Version von Aurora MySQL zu erstellen, geben Sie bitte die Engine-Version über die AWS Management Console AWS CLI, die oder die RDS-API an.

### Note

Diese Version ist derzeit in den folgenden AWS Regionen nicht verfügbar: AWS GovCloud (US-Ost) [us-gov-east-1], AWS GovCloud (US-West) [us-gov-west-1], China (Ningxia) [cn-northwest-1], Asien-Pazifik (Hongkong) [ap-east-1] und Naher Osten (Bahrain) [me-south-1]. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

### Note

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Neue Funktionen:

- Aurora MySQL-Cluster unterstützen jetzt die Instance-Typen db.r5.8xlarge, db.r5.16xlarge und db.r5.24xlarge. Weitere Informationen zu Instance-Typen für Aurora-MySQL-Cluster finden Sie unter [Aurora-DB-Instance-Klassen](#) im Amazon-Aurora-Benutzerhandbuch.
- Die Hash Join-Funktion ist jetzt allgemein verfügbar und erfordert nicht, dass die Aurora-Lab-Modus-Einstellung auf ON gesetzt wird. Diese Funktion kann die Abfrageleistung verbessern, wenn Sie eine große Datenmenge mithilfe eines Equi-Joins verbinden müssen. Weitere Informationen zur Verwendung dieses Features finden Sie unter [Verwenden der Daten-API für Aurora Serverless](#) im Amazon-Aurora-Benutzerhandbuch.
- Die Hot Row Contention-Funktion ist jetzt allgemein verfügbar und erfordert nicht, dass die Aurora-Lab-Modus-Einstellung auf ON gesetzt wird. Diese Funktion sorgt für eine deutliche Verbesserung des Durchsatzes bei Workloads mit zahlreichen Transaktionen, die um Zeilen auf derselben Seite konkurrieren.
- Aurora MySQL 2.06 und höher unterstützt jetzt das „Zurückspulen“ eines DB-Clusters auf einen bestimmten Zeitpunkt, ohne dass die Daten aus einem Backup wiederhergestellt werden müssen. Diese Funktion („Backtrack“) bietet eine schnelle Möglichkeit zur Wiederherstellung bei Benutzerfehlern, wie dem Löschen der falschen Tabelle oder dem Löschen der falschen Zeile. Backtrack funktioniert in nur wenigen Sekunden, selbst bei großen Datenbanken. Lesen Sie [den AWS Blog](#) für einen Überblick. Weitere Informationen finden Sie unter [Backtracking an Aurora DB-Cluster](#) im Amazon Aurora Benutzerhandbuch.
- Aurora 2.06 und höher unterstützen synchrone AWS Lambda Aufrufe über die native Funktion. `lambda_sync()` Außerdem ist die native Funktion `lambda_async()` vorhanden, die als Alternative zu der bestehenden gespeicherten Prozedur für asynchrone Lambda-Aufrufe

verwendet werden kann. Informationen zum Aufrufen von Lambda-Funktionen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.

Kritische Fehlerbehebungen:

Keine.

Wichtige Fehlerbehebungen:

Fehlerbehebungen bei der Sicherheit

- [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](#)

Umgang mit Verbindungen

- Die Datenbankverfügbarkeit wurde verbessert, um besser mit einer Zunahme der Clientverbindungen während der Ausführung einer oder mehrerer DDLs umgehen zu können. Dazu werden bei Bedarf zusätzliche Threads eingerichtet. Sie sollten ein Upgrade durchführen, wenn die Datenbank nach einem Anstieg der Verbindungen während der DDL-Verarbeitung nicht mehr gut reagiert.



## Engine-Neustart

- Behebung eines Problems mit längerer Nichtverfügbarkeit beim Neustart der Engine. Dies behebt ein Problem bei der Buffer-Pool-Initialisierung. Dieses Problem tritt selten auf, kann sich aber potenziell auf jede unterstützte Version auswirken.
- Behebung eines Problems, durch das eine als Binlog-Master konfigurierte Datenbank neu gestartet wurde, während ein umfangreicher Schreib-Workload stattfand.

## Allgemeine Fehlerbehebungen zur Erhöhung der Stabilität:

- Verbesserungen bei Anfragen mit Zugriff auf nicht zwischengespeicherte Daten, die langsamer als gewöhnlich sein konnten. Kunden, die unerklärliche Leselatenzen beim Zugriff auf nicht zwischengespeicherte Daten erfahren, sollten ein Upgrade durchführen, da es sich möglicherweise um genau dieses Problem handelt.
- Behebung eines Problems, bei dem partitionierte Tabellen aus einem Datenbank-Snapshot nicht wiederhergestellt werden konnten. Kunden, bei denen Fehler auftreten, wenn sie auf partitionierte Tabellen in einer Datenbank zugreifen, die aus dem Snapshot einer Aurora MySQL 1.\*-Datenbank wiederhergestellt wurde, sollten diese Version verwenden.
- Verbesserung der Stabilität der Aurora-Replicas durch die Korrektur der Sperrenkonkurrenz zwischen Threads für Lese-Anfragen und solchen, die Schemaänderungen anwenden, während eine DDL-Anfrage auf der Writer DB-Instance läuft.
- Behebung eines Stabilitätsproblems im Zusammenhang mit dem von DDL-Operationen ausgelösten `mysql.innodb_table_stats`-Tabellenupdate.
- Behebung eines Problems, bei dem inkorrektweise ERROR 1836 gemeldet wurde, wenn eine eingebettete Anfrage gegen eine temporäre Tabelle auf der Aurora-Replica ausgeführt wurde.

## Leistungsverbesserungen

- Verbesserte Leistung der Binlog-Replikation durch Verhindern unnötiger API-Aufrufe an den Zwischenspeicher, wenn der Abfragezwischenspeicher auf dem Binlog-Worker deaktiviert wurde.

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora MySQL Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Aurora MySQL 2.06.0 ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Aurora MySQL 2.06.0 bietet aktuell keine Unterstützung für die folgenden MySQL 5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung CREATE TABLESPACE

## Aurora-MySQL-Datenbank-Engine-Updates 11.11.2019 (Version 2.05.0) (veraltet)

Version: 2.05.0

Aurora MySQL 2.05.0 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Derzeit unterstützte Aurora MySQL-Versionen: 1.14.\*, 1.15.\*, 1.16.\*, 1.17.\*, 1.18.\*, 1.19.\*, 2.01.\*, 2.02.\*, 2.03.\* und 2.04.\*.

Sie können einen Snapshot aus einer derzeit unterstützten Aurora MySQL-Version auf Aurora MySQL 2.05.0 wiederherstellen. Sie haben die Möglichkeit, bestehende Aurora MySQL 2.\* Datenbank-Cluster auf 2.04.6 bis Aurora MySQL 2.05.0. upzugraden. Sie können einen vorhandenen Aurora MySQL 1.\*-Cluster nicht direkt auf Aurora MySQL 2.05.0 aktualisieren. Sie können jedoch seinen Snapshot zu Aurora MySQL 2.05.0 wiederherstellen.

Um einen Cluster mit einer älteren Version von Aurora MySQL zu erstellen, geben Sie bitte die Engine-Version über die AWS Management Console AWS CLI, die oder die RDS-API an.

#### Note

Diese Version ist derzeit in den folgenden AWS Regionen nicht verfügbar: AWS GovCloud (US-Ost) [us-gov-east-1], AWS GovCloud (US-West) [us-gov-west-1], China (Ningxia) [cn-northwest-1], Asien-Pazifik (Hongkong) [ap-east-1], Europa (Stockholm) [eu-north-1] und Naher Osten (Bahrain) [me-south-1]. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

#### Note

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Kritische Fehlerbehebungen:

- [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](#)
- Es wurde ein Problem behoben, bei dem die Ereignisse in der aktuellen Binärprotokolldatei auf dem Master nicht auf dem Worker repliziert wurden, wenn der Wert des Parameters `sync_binlog` nicht auf 1 eingestellt war.

#### Wichtige Fehlerbehebungen:

- Kunden mit einer Datenbankgröße von fast 64 Tebibytes (TiB) wird dringend empfohlen, auf diese Version zu aktualisieren, um Ausfallzeiten aufgrund von Stabilitätsfehlern zu vermeiden, die Volumes in der Nähe des Aurora Speicherlimits betreffen.
- Der Standardwert des Parameters `aurora_binlog_replication_max_yield_seconds` wurde auf Null geändert, um eine Erhöhung der Replikationsverzögerung zugunsten der Vordergrundabfrageleistung auf dem Binärprotokollmaster zu verhindern.

## Integration von MySQL-Fehlerbehebungen

- Fehler #23054591: `PURGE BINARY LOGS TO` liest die gesamte Binlog-Datei und führt dazu MySQL, dass es zum Stillstand kommt

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora MySQL Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.

- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen zum synchronen Aufrufen von Funktionen. AWS Lambda Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Aurora MySQL 2.05.0 ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Aurora MySQL 2.05.0 bietet aktuell keine Unterstützung für die folgenden MySQL 5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung CREATE TABLESPACE

# Aurora-MySQL-Datenbank-Engine-Updates 14.08.2018 (Version 2.04.9) (veraltet)

Version: 2.04.9

Aurora MySQL 2.04.9 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Um einen Cluster mit einer älteren Version von Aurora MySQL zu erstellen, geben Sie bitte die Engine-Version über die AWS Management Console AWS CLI, die oder die RDS-API an.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Note

Diese Version ist derzeit in den folgenden AWS Regionen nicht verfügbar: AWS GovCloud (US-Ost) [us-gov-east-1], AWS GovCloud (US-West) [us-gov-west-1], Asien-Pazifik (Hongkong) [ap-east-1] und Naher Osten (Bahrain) [me-south-1]. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

## Note

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Wichtige Fehlerbehebungen:

Kritische Fehlerbehebungen:

- [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](#)

Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem behoben, das zu einem Neustart der Datenbank oder einem Failover aufgrund der Ausführung eines `kill session`-Befehls führen konnte. Wenn dieses Problem auftritt, wenden Sie sich an den AWS Support, um diesen Fix auf Ihrer Instance zu aktivieren.
- Es wurde ein Problem behoben, das einen Neustart der Datenbank während der Ausführung einer komplexen Abfrage mit Multi-Table-Joins und Aggregation verursacht, die intern Zwischentabellen verwenden.
- Es wurde ein Problem behoben, durch das die Datenbank neu gestartet wurde, weil `DROP TABLE` auf mehreren Tabellen unterbrochen wurde.
- Es wurde ein Problem behoben, durch das während der Datenbankwiederherstellung ein Datenbank-Failover verursacht wurde.
- Ein Datenbankneustart wurde behoben, der durch eine falsche Meldung von `threads_running` verursacht wurde, wenn Prüfungs- und langsame Abfrageprotokolle aktiviert sind.
- Es wurde ein Problem behoben, durch das ein `kill query`-Befehl während der Ausführung hängen bleiben könnte.
- Es wurde eine race-Bedingung im Lock-Manager behoben, die zu einem Neustart oder Failover der Datenbank beim Transaktionsrollback führte.
- Es wurde ein Problem behoben, das einen Neustart oder Failover der Datenbank auslöste, wenn mehrere Verbindungen versuchen, dieselbe Tabelle mit einem Volltextsuchindex zu aktualisieren.
- Es wurde ein Problem behoben, das zu einem Deadlatch führen konnte, wenn ein Index gelöscht wurde, was zu einem Failover oder einem Neustart führte.

#### Allgemeine Verbesserungen:

- Es wurden Probleme behoben, die dazu führen konnten, dass Abfragen auf Read Replicas Daten aus einer nicht festgeschriebenen Transaktion verwenden konnten. Dieses Problem beschränkt sich auf Transaktionen, die unmittelbar nach einem Neustart der Datenbank gestartet werden.
- Es wurde ein Problem behoben, das während `INPLACE ALTER TABLE` für eine Tabelle mit definierten Auslösern auftrat, sowie dann, wenn die DDL keine `RENAME`-Klausel enthielt.
- Es wurde ein Problem behoben, durch das das Klonen bei einigen Datenbank-Clustern mit hohen Schreiblasten länger dauerte.
- Es wurde ein Problem behoben, das während eines Upgrades aufgetreten ist, wenn eine partitionierte Tabelle eingebettete Leerzeichen im Namen enthielt.
- Es wurde ein Problem behoben, durch das Read Replicas vorübergehend Teilergebnisse einer kürzlich festgeschriebenen Transaktion auf dem Writer sehen konnten.



- Es wurde ein Problem behoben, durch das Abfragen auf Read Replicas gegen eine FTS-Tabelle veraltete Ergebnisse liefern konnten. Dies tritt nur auf, wenn die FTS-Abfrage auf der Read Replica unmittelbar auf eine Abfrage auf `INFORMATION_SCHEMA.INNODB_SYS_TABLES` für dieselbe FTS-Tabelle auf dem Writer folgt.
- Es wurde ein Problem behoben, das zu einer langsamen Wiederherstellung eines Aurora 1.x-Datenbank-Clusters mit FTS-Indizes (Full Text Search) in einem Aurora 2.x-Datenbank-Cluster führte.
- Die maximal zulässige Länge wurde auf 2000 für die globalen Parameter `server_audit_incl_users` und `server_audit_excl_users` erweitert.
- Es wurde ein Fehler behoben, durch den die Wiederherstellung von Aurora 1.x zu Aurora 2.x längere Zeit in Anspruch nehmen konnte.
- Es wurde ein Problem behoben, durch das ein `lambda_async`-Aufruf über gespeicherte Prozedur nicht mit Unicode funktioniert.
- Es wurde ein Problem behoben, das auftrat, wenn ein räumlicher Index eine Off-Record-Geometrie-Spalte nicht ordnungsgemäß verarbeitete.
- Es wurde ein Problem behoben, durch das eine Abfrage auf einer Reader-DB-Instance mit dem Fehler `InternalFailureException` und der Meldung „Operation beendet (interner Fehler)“ fehlschlagen konnte.

## Integration von MySQL-Fehlerbehebungen

- Fehler #23070734, Bug #80060: Concurrent TRUNCATE TABLEs cause stalls
- Fehler #23103937: `PS_TRUNCATE_ALL_TABLES()` DOES NOT WORK IN SUPER\_READ\_ONLY MODE
- Fehler #22551677: Wenn der Server offline genommen wurde, konnte eine seltene Bedingung im Performance Schema zu einem Server-Exit führen.
- Fehler #27082268: Ungültige FTS-Synchronisierung.
- FEHLER #12589870: Es wurde ein Fehler behoben, der einen Neustart mit mehrfacher Abfrage auslöste, wenn der Abfrage-Cache aktiviert ist.
- Fehler #26402045: Bestimmte Fälle der Materialisierung von Unterabfragen konnten einen Server-Exit verursachen. Diese Abfragen erzeugen nun einen Fehler, der darauf hindeutet, dass die Materialisierung deaktiviert ist.
- Fehler #18898433: Abfragen mit vielen Links-Joins waren langsam, wenn die Join-Pufferung verwendet wurde (zum Beispiel mit dem block-verschachtelten Schleifenalgorithmus).

- Fehler #25222337: Ein Feldname einer virtuellen NULL-Spalte in einem virtuellen Index führte zu einem Beenden des Servers während eines Feldnamensabgleichs, der beim Befüllen von virtuellen Spalten auftritt, die von einer Fremdschlüsselbeschränkung betroffen sind. (<https://github.com/mysql/mysql-server/commit/273d5c9d7072c63b6c47dbef6963d7dc491d5131>)
- Fehler #25053286: Das Ausführen einer gespeicherten Prozedur mit einer auf eine Ansicht zugreifenden Abfrage konnte Speicher zuweisen, der bis zum Ende der Sitzung nicht freigegeben wurde. (<https://github.com/mysql/mysql-server/commit/d7b37d4d141a95f577916448650c429f0d6e193d>)
- Fehler #25586773: Das Ausführen einer gespeicherten Prozedur mit einer Anweisung, die eine Tabelle aus dem Inhalt bestimmter SELECT (https://dev.mysql.com/doc/refman/5.7/en/select.html) -Anweisungen erstellt, konnte zu einem Speicherverlust führen. (<https://github.com/mysql/mysql-server/commit/88301e5adab65f6750f66af284be410c4369d0c1>)
- Fehler #26666274: INFINITE LOOP IN PERFORMANCE SCHEMA BUFFER CONTAINER.
- Fehler #23550835, Fehler #23298025, Fehler #81464: SELECT Performance Schema-Tabellen, konnten bei vollem internen Puffer einen Server-Exit verursachen.

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora MySQL Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen zum synchronen Aufrufen AWS Lambda von Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Aurora MySQL 2.04.9 ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Aurora MySQL 2.04.9 bietet aktuell keine Unterstützung für die folgenden MySQL 5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`

## Aurora-MySQL-Datenbank-Engine-Updates 20.11.2019 (Version 2.04.8) (veraltet)

Version: 2.04.8

Aurora MySQL 2.04.8 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Sie können einen Snapshot jeder 2.\* Aurora MySQL-Version in Aurora MySQL 2.04.8 wiederherstellen. Sie haben die Möglichkeit, bestehende Aurora MySQL 2.\* Datenbank-Cluster auf Aurora MySQL 2.04.8 upzugraden.

Um einen Cluster mit einer älteren Version von Aurora MySQL zu erstellen, geben Sie bitte die Engine-Version über die AWS Management Console, die AWS CLI, die oder die RDS-API an.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

#### Note

Diese Version ist derzeit in den folgenden AWS Regionen nicht verfügbar: AWS GovCloud (US-Ost) [us-gov-east-1], AWS GovCloud (US-West) [us-gov-west-1], Asien-Pazifik (Hongkong) [ap-east-1] und Naher Osten (Bahrain) [me-south-1]. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

#### Note

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Neue Funktionen:

- Lesereplikation-Verbesserungen:
  - Geringerer Netzwerkdatenverkehr von der Writer-Instance durch die effiziente Übertragung von Daten zu Reader-Instances innerhalb des Aurora-DB-Clusters. Diese Verbesserung ist standardmäßig aktiviert, da sie dabei hilft, zu vermeiden, dass Replicas zurückfallen und neu gestartet werden. Der Parameter für diese Funktion ist `aurora_enable_repl_bin_log_filtering`.
  - Geringerer Netzwerkdatenverkehr von Writer- zu Reader-Instances innerhalb des Aurora-DB-Clusters mit Komprimierung. Diese Verbesserung ist standardmäßig nur für die Instance-Klassen `8xlarge` und `16xlarge` aktiviert, da diese Instances mit zusätzlichem CPU-Overhead für die Komprimierung umgehen können. Der Parameter für diese Funktion ist `aurora_enable_replica_log_compression`.

## Wichtige Fehlerbehebungen:

- Verbesserung des Speichermanagements in der Aurora-Writer-Instance, durch die der Neustart des Writers durch Speicherüberlastung bei intensiven Workloads beim gleichzeitigen Vorhandensein von Reader-Instances innerhalb des Aurora-DB-Clusters verhindert wird.
- Fehlerbehebung für eine nicht-deterministische Bedingung im Scheduler, die zu einem Neustart der Engine führt, während gleichzeitig auf das Performance-Schemaobjekt zugegriffen wird.

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora MySQL Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen zum synchronen Aufrufen von Funktionen. AWS Lambda Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Aurora MySQL 2.04.8 ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Aurora MySQL 2.04.8 bietet aktuell keine Unterstützung für die folgenden MySQL 5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung CREATE TABLESPACE

## Aurora-MySQL-Datenbank-Engine-Updates 14.11.2019 (Version 2.04.7) (veraltet)

Version: 2.04.7

Aurora MySQL 2.04.7 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Sie können einen Snapshot aus einer derzeit unterstützten Aurora MySQL-Version auf Aurora MySQL 2.04.7 wiederherstellen. Sie haben die Möglichkeit, bestehende Aurora MySQL 2.\* Datenbank-Cluster auf Aurora MySQL 2.04.7 upzugraden. Sie können einen vorhandenen Aurora MySQL 1.\*-Cluster nicht direkt auf Aurora MySQL 2.04.7 aktualisieren. Sie können jedoch seinen Snapshot zu Aurora MySQL 2.04.7 wiederherstellen.

Um einen Cluster mit einer älteren Version von Aurora MySQL zu erstellen, geben Sie bitte die Engine-Version über die AWS Management Console AWS CLI, die oder die RDS-API an.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

**Note**

Diese Version ist derzeit in den folgenden AWS Regionen nicht verfügbar: AWS GovCloud (US-Ost) [us-gov-east-1], AWS GovCloud (US-West) [us-gov-west-1], Asien-Pazifik (Hongkong) [ap-east-1] und Naher Osten (Bahrain) [me-south-1]. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

**Note**

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Wichtige Fehlerbehebungen:

Umgang mit Verbindungen

- Die Datenbankverfügbarkeit wurde verbessert, um besser mit einer Zunahme der Clientverbindungen während der Ausführung einer oder mehrerer DDLs umgehen zu können. Dazu werden bei Bedarf zusätzliche Threads eingerichtet. Sie sollten ein Upgrade durchführen, wenn die Datenbank nach einem Anstieg der Verbindungen während der DDL-Verarbeitung nicht mehr gut reagiert.
- Es wurde ein Problem behoben, das zu einem falschen Wert für die globale Statusvariable `Threads_running` führte.

Engine-Neustart

- Behebung eines Problems mit längerer Nichtverfügbarkeit beim Neustart der Engine. Dies behebt ein Problem bei der Buffer-Pool-Initialisierung. Dieses Problem tritt selten auf, kann sich aber potenziell auf jede unterstützte Version auswirken.

Allgemeine Fehlerbehebungen zur Erhöhung der Stabilität:

- Verbesserungen bei Anfragen mit Zugriff auf nicht zwischengespeicherte Daten, die langsamer als gewöhnlich sein konnten. Kunden, die unerklärliche Leselatenzen beim Zugriff auf nicht zwischengespeicherte Daten erfahren, sollten ein Upgrade durchführen, da es sich möglicherweise um genau dieses Problem handelt.

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora MySQL Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen zum synchronen Aufrufen von Funktionen. AWS Lambda Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Aurora MySQL 2.04.7 ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Aurora MySQL 2.04.7 bietet aktuell keine Unterstützung für die folgenden MySQL 5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße



- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung CREATE TABLESPACE

## Aurora-MySQL-Datenbank-Engine-Updates 19.09.2019 (Version 2.04.6) (veraltet)

Version: 2.04.6

Aurora MySQL 2.04.6 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Sie haben die Möglichkeit, bestehende Aurora MySQL 2.\* Datenbank-Cluster auf Aurora MySQL 2.04.6 upzugraden. Wir gestatten kein In-Place-Upgrade von Aurora MySQL 1.\* Clustern. Diese Einschränkung wird in einer späteren Aurora MySQL 2.\* Version aufgehoben. Sie können Snapshots von Aurora MySQL 1.14.\*, 1.15.\*, 1.16.\*, 1.17.\*, 1.18.\*, 1.19.\*, 2.01.\*, 2.02.\*, 2.03.\* und 2.04.\* in Aurora MySQL 2.04.6 wiederherstellen.

Um eine ältere Version von Aurora MySQL zu verwenden, können Sie neue Datenbank-Cluster erstellen, indem Sie die Engine-Version über die AWS Management Console AWS CLI, die oder die Amazon RDS-API angeben.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

**Note**

Diese Version ist derzeit in den folgenden AWS Regionen nicht verfügbar: Europa (London) [eu-west-2], AWS GovCloud (US-Ost) [us-gov-east-1], (US-West) [-1], China AWS GovCloud (Ningxia) [us-gov-westcn-northwest-1] und Asien-Pazifik (Hongkong) [ap-east-1]. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

**Note**

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

- Es wurde ein Problem behoben, bei dem die Ereignisse in der aktuellen Binärprotokolldatei auf dem Master nicht auf dem Worker repliziert wurden, wenn der Wert des Parameters `sync_binlog` nicht auf 1 eingestellt war.
- Der Standardwert des Parameters `aurora_binlog_replication_max_yield_seconds` wurde auf Null geändert, um eine Erhöhung der Replikationsverzögerung zugunsten der Vordergrundabfrageleistung auf dem Binärprotokollmaster zu verhindern.

## Integration von MySQL-Fehlerbehebungen

- Fehler #23054591: `PURGE BINARY LOGS TO` liest die gesamte Binlog-Datei und verursacht einen Stall MySQL

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora MySQL Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen zum synchronen Aufrufen von Funktionen. AWS Lambda Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Aurora MySQL 2.04.6 ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Aurora MySQL 2.04.6 bietet aktuell keine Unterstützung für die folgenden MySQL 5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung CREATE TABLESPACE

# Aurora-MySQL-Datenbank-Engine-Updates 08.07.2019 (Version 2.04.5) (veraltet)

Version: 2.04.5

Aurora MySQL 2.04.5 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Sie haben die Möglichkeit, bestehende Aurora MySQL 2.\* Datenbank-Cluster auf Aurora MySQL 2.04.5 aufzugraden. Wir gestatten kein In-Place-Upgrade von Aurora MySQL 1.\* Clustern. Diese Einschränkung wird in einer späteren Aurora MySQL 2.\* Version aufgehoben. Sie können Snapshots von Aurora MySQL 1.14.\*, 1.15.\*, 1.16.\*, 1.17.\*, 1.18.\*, 1.19.\*, 2.01.\*, 2.02.\*, 2.03.\* und 2.04 in Aurora MySQL und 2.04.5 wiederherstellen.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Note

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Fehlerbehebungen bei der Sicherheit:

- [CVE-2016-3518](#)

Allgemeine Korrekturen:

- Es wurde eine Race-Condition während des Speichervolumewachstums behoben, die einen Neustart der Datenbank zur Folge hatte.

- Es wurde ein interner Kommunikationsfehler beim Öffnen des Volumes behoben, der zum Neustart der Datenbank führte.
- Unterstützung für die DDL-Wiederherstellung für `ALTER TABLE ALGORITHM=INPLACE` bei partitionierten Tabellen hinzugefügt.
- Es wurde ein Problem mit der DDL-Wiederherstellung von `ALTER TABLE ALGORITHM=COPY` behoben, das zum Neustart der Datenbank führte.
- Aurora-Replica-Stabilität unter hohem Lösch-Workload über den Writer verbessert.
- Es wurde ein Datenbankneustart korrigiert, der durch eine Deadlatch zwischen dem Thread, der die Volltextsuchindex-Synchronisierung durchführt, und dem Thread, der die Entleerung der Volltextsuche-Tabelle aus dem Dictionary-Cache durchführt.
- Es wurde ein Stabilitätsproblem auf dem Binlog-Worker während der DDL-Replikation behoben, während die Verbindung mit dem Binlog-Master instabil ist.
- Es wurde ein out-of-memory Problem im Volltext-Suchcode behoben, das zu einem Neustart der Datenbank führte.
- Es wurde ein Problem mit dem Aurora-Writer behoben, das dazu führte, dass er neu gestartet wurde, wenn das gesamte 64-Tebibyte(TiB)-Volume verwendet wurde.
- Es wurde eine Race-Condition in der Leistungsschema-Funktion behoben, die einen Neustart der Datenbank zur Folge hatte.
- Beseitigung eines Fehlers, der beim Behandeln eines Fehlers in der Netzwerkprotokollverwaltung zum Abbruch von Verbindungen führte.

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora MySQL Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Systemeigene Funktionen zum synchronen Aufrufen AWS Lambda von Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.

- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Aurora MySQL 2.04.5 ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Aurora MySQL 2.04.5 bietet aktuell keine Unterstützung für die folgenden MySQL 5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`

## Aurora-MySQL-Datenbank-Engine-Updates 29.05.2019 (Version 2.04.4) (veraltet)

Version: 2.04.4

Aurora MySQL 2.04.4 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Beim Erstellen eines neuen Aurora MySQL-DB-Clusters (einschließlich Wiederherstellen eines Snapshots) können Sie Kompatibilität mit entweder MySQL 5.7 oder MySQL 5.6 auswählen. Wir lassen kein In-Place-Upgrade von Aurora MySQL 1.\* Clustern oder Wiederherstellung von Aurora MySQL 1.\* Clustern aus einem Amazon S3-Backup in Aurora MySQL 2.04.4 zu. Wir planen, diese Einschränkungen in einer späteren Veröffentlichung von Aurora MySQL 2.\* aufzuheben.

Sie können Snapshots von Aurora MySQL 1.14.\*, 1.15.\*, 1.16.\*, 1.17.\*, 1.18.\*, 1.19.\*, 2.01.\*, 2.02.\*, 2.03.\* und 2.04 in Aurora MySQL 2.04.4 wiederherstellen.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

#### Note

Diese Version ist derzeit in den Regionen AWS GovCloud (US-West) [us-gov-west-1], Europa (Stockholm) [eu-north-1], China (Ningxia) [cn-northwest-1] und Asien-Pazifik (Hongkong) [ap-east-1] nicht verfügbar. AWS Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

#### Note

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

- Beseitigung eines Problems, das beim Laden von Daten aus S3 in Aurora zu Fehlern führen könnte.
- Beseitigung eines Problems, das beim Hochladen von Daten aus Aurora zu S3 zu Fehlern führen könnte.

- Beseitigung eines Fehlers, der beim Behandeln eines Fehlers in der Netzwerkprotokollverwaltung zum Abbruch von Verbindungen führte.
- Beseitigung eines Problems, das beim Arbeiten mit partitionierten Tabellen zu einem Absturz führen konnte.
- Beseitigung eines Problems, bei dem die Funktion Performance Insights in einigen Regionen nicht verfügbar war.

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora MySQL Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen zum synchronen Aufrufen von Funktionen. AWS Lambda Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Aurora MySQL 2.04.4 ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Aurora MySQL 2.04.4 unterstützt zurzeit die folgenden MySQL 5.7-Funktionen nicht:



- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung CREATE TABLESPACE

## Aurora-MySQL-Datenbank-Engine-Updates 09.05.2019 (Version 2.04.3) (veraltet)

Version: 2.04.3

Aurora MySQL 2.04.3 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Beim Erstellen eines neuen Aurora MySQL-DB-Clusters (einschließlich Wiederherstellen eines Snapshots) können Sie Kompatibilität mit entweder MySQL 5.7 oder MySQL 5.6 auswählen. Wir lassen kein In-Place-Upgrade von Aurora MySQL 1.\* Clustern oder Wiederherstellung von Aurora MySQL 1.\* Clustern aus einem Amazon S3-Backup in Aurora MySQL 2.04.3 zu. Wir planen, diese Einschränkungen in einer späteren Veröffentlichung von Aurora MySQL 2.\* aufzuheben.

Sie können Snapshots von Aurora MySQL 1.14.\*, 1.15.\*, 1.16.\*, 1.17.\*, 1.18.\*, 1.19.\*, 2.01.\*, 2.02.\*, 2.03.\* und 2.04.\* in Aurora MySQL-MySQL 2.04.3 wiederherstellen.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

**Note**

Diese Version ist derzeit in den Regionen AWS GovCloud (US-West) [us-gov-west-1] und China (Ningxia) [cn-northwest-1] nicht verfügbar. AWS Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

**Note**

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

- Es wurde ein Fehler in der Binärprotokoll-Replikation behoben, der ein Problem auf Aurora-Instances, die als Binärprotokoll-Worker konfiguriert sind, verursachen kann.
- Ein Problem beim Umgang mit großen gespeicherten Routinen wurde behoben out-of-memory .
- Ein Fehler bei der Verarbeitung bestimmter ALTER TABLE-Befehle wurde behoben.
- Ein Fehler in Form von Verbindungsabbrüchen aufgrund eines Problems in der Netzwerkprotokollverwaltung wurde behoben.

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora MySQL Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen zum synchronen Aufrufen AWS Lambda von Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.

- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Aurora MySQL 2.04.3 ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Aurora MySQL 2.04.3 bietet aktuell keine Unterstützung für die folgenden MySQL 5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`

## Aurora-MySQL-Datenbank-Engine-Updates 02.05.2019 (Version 2.04.2) (veraltet)

Version: 2.04.2

Aurora MySQL 2.04.2 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Beim Erstellen eines neuen Aurora MySQL-DB-Clusters (einschließlich Wiederherstellen eines Snapshots) können Sie Kompatibilität mit entweder MySQL 5.7 oder MySQL 5.6 auswählen. Wir lassen kein In-Place-Upgrade von Aurora MySQL 1.\* Clustern oder Wiederherstellung von Aurora MySQL 1.\* Clustern aus einem Amazon S3-Backup in Aurora MySQL 2.04.2 zu. Wir planen, diese Einschränkungen in einer späteren Veröffentlichung von Aurora MySQL 2.\* aufzuheben.

Sie können Snapshots von Aurora MySQL 1.14.\*, 1.15.\*, 1.16.\*, 1.17.\*, 1.18.\*, 1.19.\*, 2.01.\*, 2.02.\*, 2.03.\*, 2.04.0 und 2.04.1 in Aurora MySQL 2.04.2 wiederherstellen.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

#### Note

Diese Version ist derzeit in den Regionen AWS GovCloud (US-West) [us-gov-west-1] und China (Ningxia) [cn-northwest-1] nicht verfügbar. AWS Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

#### Note

Informationen zum Upgrade Ihres Aurora-MySQL-Datenbank-Clusters finden Sie unter [Upgrade der Nebenversion oder des Patch-Levels eines Aurora-MySQL-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

- Die SSL-Binärprotokoll-Replikation mithilfe von benutzerdefinierten Zertifikaten wird jetzt unterstützt. Weitere Informationen zur Verwendung der SSL-Binärprotokoll-Replikation in Aurora MySQL finden Sie unter [mysql\\_rds\\_import\\_binlog\\_ssl\\_material](#).
- Es wurde ein Fehler behoben, der auf der primären Aurora-Instance beim Optimieren einer Tabelle mit einem Volltextsuchindex auftritt.

- Es wurde ein Fehler auf den Aurora-Replicas behoben, der die Leistung bestimmter Abfragen mithilfe von `SELECT (*)` auf Tabellen mit sekundären Indizes beeinträchtigen konnte.
- Es wurde ein Zustand behoben, bei dem Fehler 1032 angezeigt wurde.
- Die Stabilität von Aurora-Replicas wurde durch Korrigieren mehrerer Fehler verbessert.

## Integration von MySQL-Fehlerbehebungen

- Fehler #24829050 – INDEX\_MERGE\_INTERSECTION-OPTIMIERUNG FÜHRT ZU FALSCHEN ABFRAGEERGEBNISSEN

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora MySQL Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen zum synchronen Aufrufen von Funktionen. AWS Lambda Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Aurora MySQL 2.04.2 ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine

> 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Aurora MySQL 2.04.2 bietet aktuell keine Unterstützung für die folgenden MySQL 5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung CREATE TABLESPACE

## Aurora-MySQL-Datenbank-Engine-Updates 25.03.2019 (Version 2.04.1) (veraltet)

Version: 2.04.1

Aurora MySQL 2.04.1 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Beim Erstellen eines neuen Aurora MySQL-DB-Clusters (einschließlich Wiederherstellen eines Snapshots) können Sie Kompatibilität mit entweder MySQL 5.7 oder MySQL 5.6 auswählen. Wir lassen kein In-Place-Upgrade von Aurora MySQL 1.\* Clustern oder Wiederherstellung von Aurora MySQL 1.\* Clustern aus einem Amazon S3-Backup in Aurora MySQL 2.04.1 zu. Wir planen, diese Einschränkungen in einer späteren Veröffentlichung von Aurora MySQL 2.\* aufzuheben.

Sie können Snapshots von Aurora MySQL 1.14.\*, 1.15.\*, 1.16.\*, 1.17.\*, 1.18.\*, 1.19.\*, 2.01.\*, 2.02.\*, 2.03.\*, 2.04.0 in Aurora MySQL 2.04.1 wiederherstellen.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

#### Note

Diese Version ist derzeit in der Region AWS GovCloud (US-West) [us-gov-west-1] nicht verfügbar. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

#### Note

Die Vorgehensweise zum Upgraden eines DB-Clusters hat sich geändert. Weitere Informationen finden Sie unter [Upgrade von Nebenversionen oder der Patch-Ebene eines Aurora-MYSQL-DB-Clusters](#) in der Amazon-Aurora-Benutzeranleitung.

## Verbesserungen

- Es wurde ein Problem behoben, bei dem ein Aurora MySQL 5.6-Snapshot für Versionen unter 1.16 nicht in den neuesten Aurora MySQL 5.7-Cluster wiederhergestellt werden konnte.

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora MySQL Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen zum synchronen Aufrufen AWS Lambda von Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).

- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`

## Aurora-MySQL-Datenbank-Engine-Updates 25.03.2019 (Version 2.04.0) (veraltet)

Version: 2.04

Aurora MySQL 2.04 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.



Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Beim Erstellen eines neuen Aurora MySQL-DB-Clusters (einschließlich Wiederherstellen eines Snapshots) können Sie Kompatibilität mit entweder MySQL 5.7 oder MySQL 5.6 auswählen. Wir lassen kein In-Place-Upgrade von Aurora MySQL 1.\* Clustern oder Wiederherstellung von Aurora MySQL 1.\* Clustern aus einem Amazon S3-Backup in Aurora MySQL 2.04.0 zu. Wir planen, diese Einschränkungen in einer späteren Veröffentlichung von Aurora MySQL 2.\* aufzuheben.

Sie können Snapshots von Aurora MySQL 1.19.\*, 2.01.\*, 2.02.\* und 2.03.\* in Aurora MySQL 2.04.0 wiederherstellen. Sie können keine Snapshots von Aurora MySQL 1.14.\* oder niedriger, 1.15.\*, 1.16.\*, 1.17.\* und 1.18.\* in Aurora MySQL 2.04.0 wiederherstellen. Diese Einschränkung wird in Aurora MySQL 2.04.1 entfernt.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

#### Note

Diese Version ist derzeit in der Region AWS GovCloud (US-West) [us-gov-west-1] nicht verfügbar. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

#### Note

Die Vorgehensweise zum Upgraden eines DB-Clusters hat sich geändert. Weitere Informationen finden Sie unter [Upgrade von Nebenversionen oder der Patch-Ebene eines Aurora-MYSQL-DB-Clusters](#) in der Amazon-Aurora-Benutzeranleitung.

## Verbesserungen

- Unterstützt GTID-basierte Replikation. Informationen über die Verwendung der GTID-basierten Replikation mit Aurora MySQL finden Sie unter [Verwenden der GTID-basierten Replikation für Aurora MySQL](#) im Amazon-Aurora-Benutzerhandbuch.

- Es wurde ein Problem behoben, bei dem eine Aurora Replica fälschlicherweise den Fehler `Running in read-only mode` ausgab, wenn eine Anweisung zum Löschen oder Aktualisieren von Zeilen in einer temporären Tabelle eine InnoDB-Unterabfrage enthielt.

## Integration von MySQL-Fehlerbehebungen

- Bug #26225783: MYSQL CRASH ON CREATE TABLE (REPRODUCEABLE) -> INNODB: ALONG SEMAPHORE WAIT.

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora MySQL Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen zum synchronen Aufrufen AWS Lambda von Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Diese Aurora-MySQL-Version ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Diese Aurora-MySQL-Version bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`

## Aurora-MySQL-Datenbank-Engine-Updates 07.02.2019 (Version 2.03.4) (veraltet)

Version: 2.03.4

Aurora MySQL 2.03.4 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Beim Erstellen eines neuen Aurora MySQL-DB-Clusters (einschließlich der Wiederherstellung aus einem Snapshot) können Sie die Kompatibilität mit entweder MySQL 5.7 oder MySQL 5.6 auswählen.

Wir erlauben kein direktes Upgrade von Aurora MySQL 1.\*-Clustern in Aurora MySQL 2.03.4 oder die Wiederherstellung in Aurora MySQL 2.03.4 aus einer Amazon S3-Backup. Wir planen, diese Einschränkungen in einer späteren Veröffentlichung von Aurora MySQL 2.\* aufzuheben.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

**Note**

Diese Version ist derzeit in den Regionen AWS GovCloud (US-West) [us-gov-west-1] und China (Peking) [cn-north-1] nicht verfügbar. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

**Note**

Die Vorgehensweise zum Upgraden eines DB-Clusters hat sich geändert. Weitere Informationen finden Sie unter [Upgrade von Nebenversionen oder der Patch-Ebene eines Aurora-MYSQL-DB-Clusters](#) in der Amazon-Aurora-Benutzeranleitung.

## Verbesserungen

- Unterstützung für UTF8MB4 Unicode 9.0-Sortierung unter Beachtung von Akzenten und Groß-/ Kleinschreibung, `utf8mb4_0900_as_ci`.

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon-Aurora-MySQL-Funktionen werden in Aurora-MySQL-Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora-MySQL-Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen zum synchronen Aufrufen von Funktionen. AWS Lambda Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).

- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Aurora MySQL 2.03.4 ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Aurora MySQL 2.03.4 unterstützt zurzeit die folgenden MySQL-5.7-Funktionen nicht:

- Globale Transaktionskennungen (GTIDs) Aurora MySQL unterstützt GTIDs in Version 2.04 und höher.
- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`
- X Protocol

## Aurora-MySQL-Datenbank-Engine-Updates 18.01.2019 (Version 2.03.3) (veraltet)

Version: 2.03.3

Aurora MySQL 2.03.3 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Beim Erstellen eines neuen Aurora MySQL-DB-Clusters (einschließlich der Wiederherstellung aus einem Snapshot) können Sie die Kompatibilität mit entweder MySQL 5.7 oder MySQL 5.6 auswählen.

Wir erlauben kein direktes Upgrade von Aurora MySQL 1.\*-Clustern in Aurora MySQL 2.03.3 oder die Wiederherstellung in Aurora MySQL 2.03.3 aus einer Amazon S3-Backup. Wir planen, diese Einschränkungen in einer späteren Veröffentlichung von Aurora MySQL 2.\* aufzuheben.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

#### Note

Diese Version ist derzeit in den Regionen AWS GovCloud (US-West) [us-gov-west-1] und China (Peking) [cn-north-1] nicht verfügbar. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

#### Note

Die Vorgehensweise zum Upgraden eines DB-Clusters hat sich geändert. Weitere Informationen finden Sie unter [Upgrade von Nebenversionen oder der Patch-Ebene eines Aurora-MYSQL-DB-Clusters](#) in der Amazon-Aurora-Benutzeranleitung.

## Verbesserungen

### CVE-Fehlerbehebungen

- [CVE-2016-5436](#)

### Kritische Fehlerbehebungen:

- Es wurde ein Problem behoben, bei dem eine Aurora Replica abstürzen konnte, wenn ein Index rückwärts gescannt wurde.

- Es wurde ein Problem behoben, bei dem eine Aurora Replica möglicherweise neu gestartet wurde, wenn die primäre Aurora-Instance direkte DDL-Operationen für partitionierte Tabellen ausführte.
- Es wurde ein Problem behoben, bei dem eine Aurora Replica während der Invalidierung des Abfrage-Caches nach einer DDL-Operation für eine primäre Aurora-Instance möglicherweise neu gestartet wurde.
- Es wurde ein Problem behoben, bei dem eine Aurora Replica einer SELECT-Abfrage für eine Tabelle möglicherweise neu gestartet wurde, während die primäre Aurora-Instance eine Kürzung der Tabelle durchführte.
- Es wurde ein Problem mit einem falschen Ergebnis bei temporären MyISAM-Tabellen behoben, bei dem nur auf indizierte Spalten zugegriffen wurde.
- Es wurde ein Problem in langsamen Protokollen behoben, bei dem regelmäßig nach ca. 40.000 Abfragen fälschlicherweise zu große Werte für `query_time` und `lock_time` erzeugt wurden.
- Es wurde ein Problem behoben, bei dem ein Schema mit dem Namen „tmp“ dazu führen konnte, dass die Migration von RDS for MySQL zu Aurora MySQL hängen blieb.
- Es wurde ein Problem behoben, bei dem im Auditprotokoll möglicherweise Ereignisse während der Protokollrotation fehlten.
- Es wurde ein Problem behoben, bei dem die aus einem Aurora 5.6-Snapshot wiederhergestellte primäre Aurora-Instance möglicherweise neu gestartet wurde, wenn die Fast DDL-Funktion im Labor-Modus aktiviert ist.
- Es wurde ein Problem behoben, bei dem die CPU-Nutzung bedingt durch den Wörterbuchstatistik-Thread bei 100 % lag.
- Es wurde ein Problem behoben, bei dem eine Aurora Replica beim Ausführen einer CHECK TABLE-Anweisung möglicherweise neu gestartet wurde.

## Integration von MySQL-Fehlerbehebungen

- Bug #25361251: INCORRECT BEHAVIOR WITH INSERT ON DUPLICATE KEY IN SP
- Bug #26734162: INCORRECT BEHAVIOR WITH INSERT OF BLOB + ON DUPLICATE KEY UPDATE
- Bug #27460607: INCORRECT BEHAVIOR OF IODKU WHEN INSERT SELECT's SOURCE TABLE IS EMPTY
- Eine Abfrage mit der Klausel DISTINCT oder GROUP BY könnte falsche Ergebnisse zurückgeben. (MySQL 5.7 Bug #79591, Bug #22343910)

- Ein DELETE aus verbundenen Tabellen, die eine abgeleitete Tabelle in der WHERE Klausel verwendet, schlägt mit dem Fehler 1093 (Bug #23074801) fehl.
- GCOLS: INCORRECT BEHAVIOR WITH CHARSET CHANGES (Bug #25287633).

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon-Aurora-MySQL-Funktionen werden in Aurora-MySQL-Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora-MySQL-Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen zum synchronen Aufrufen von Funktionen. AWS Lambda Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Aurora MySQL 2.03.3 ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Aurora MySQL 2.03.3 bietet aktuell keine Unterstützung für die folgenden MySQL 5.7-Funktionen:

- Globale Transaktionskennungen (GTIDs) Aurora MySQL unterstützt GTIDs in Version 2.04 und höher.
- Plug-In für die Gruppenreplikation



- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`
- X Protocol

## Aurora-MySQL-Datenbank-Engine-Updates 09.01.2019 (Version 2.03.2) (veraltet)

Version: 2.03.2

Aurora MySQL 2.03.2 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Beim Erstellen eines neuen Aurora MySQL-DB-Clusters (einschließlich der Wiederherstellung aus einem Snapshot) können Sie die Kompatibilität mit entweder MySQL 5.7 oder MySQL 5.6 auswählen.

Wir erlauben kein direktes Upgrade von Aurora MySQL 1.\*-Clustern in Aurora MySQL 2.03.2 oder die Wiederherstellung in Aurora MySQL 2.03.2 aus einer Amazon S3-Backup. Wir planen, diese Einschränkungen in einer späteren Veröffentlichung von Aurora MySQL 2.\* aufzuheben.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

### Note

Diese Version ist derzeit in den Regionen AWS GovCloud (US-West) [us-gov-west-1] und China (Peking) [cn-north-1] nicht verfügbar. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

**Note**

Die Vorgehensweise zum Upgraden eines DB-Clusters hat sich geändert. Weitere Informationen finden Sie unter [Upgrade von Nebenversionen oder der Patch-Ebene eines Aurora-MYSQL-DB-Clusters](#) in der Amazon-Aurora-Benutzeranleitung.

## Verbesserungen

- Aurora-Versionsauswahl – Seit Aurora MySQL 2.03.2 können Sie in der AWS Management Console zwischen mehreren MySQL 5.7-kompatiblen Aurora-Versionen auswählen. Weitere Informationen finden Sie unter [Aurora-MySQL-Engine-Versionen überprüfen oder angeben AWS](#) im Amazon-Aurora-Benutzerhandbuch.

Kritische Fehlerbehebungen:

- [CVE-2016-3495](#)

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon-Aurora-MySQL-Funktionen werden in Aurora-MySQL-Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora-MySQL-Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen zum synchronen Aufrufen von Funktionen. AWS Lambda Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Aurora MySQL 2.03.2 ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Aurora MySQL 2.03.2 bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Globale Transaktionskennungen (GTIDs) Aurora MySQL unterstützt GTIDs in Version 2.04 und höher.
- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`
- X Protocol

## Aurora-MySQL-Datenbank-Engine-Updates 24.10.2018 (Version 2.03.1) (veraltet)

Version: 2.03.1

Aurora MySQL 2.03.1 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Beim Erstellen eines neuen Aurora MySQL-DB-Clusters können Sie die Kompatibilität mit entweder MySQL 5.7 oder MySQL 5.6 auswählen. Beim Wiederherstellen eines mit MySQL 5.6 kompatiblen Snapshots können Sie die Kompatibilität mit entweder MySQL 5.7 oder MySQL 5.6 auswählen.

Sie können Snapshots von Aurora MySQL 1.14.\*, 1.15.\*, 1.16.\*, 1.17.\*, 1.18.\*, 2.01.\*, 2.02.\* und 2.03 in Aurora-MySQL 2.03.1 wiederherstellen.

Wir erlauben kein direktes Upgrade von Aurora MySQL 1.\*-Clustern in Aurora MySQL 2.03.1 oder die Wiederherstellung in Aurora MySQL 2.03.1 aus einer Amazon S3-Backup. Wir planen, diese Einschränkungen in einer späteren Veröffentlichung von Aurora MySQL 2.\* aufzuheben.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

### Note

Diese Version ist derzeit in den Regionen AWS GovCloud (US-West) [us-gov-west-1] und China (Peking) [cn-north-1] nicht verfügbar. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

## Verbesserungen

- Ein Problem wurde behoben, das bei der Erkennung von Transaktionsdeadlocks möglicherweise zum Neustart des Aurora Writers führt.

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon-Aurora-MySQL-Funktionen werden in Aurora-MySQL-Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora-MySQL-Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen zum synchronen Aufrufen von Funktionen. AWS Lambda Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).

- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Aurora MySQL 2.03.1 ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Aurora MySQL 2.03.1 bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Globale Transaktionskennungen (GTIDs) Aurora MySQL unterstützt GTIDs in Version 2.04 und höher.
- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`
- X Protocol

## Aurora-MySQL-Datenbank-Engine-Updates 11.10.2018 (Version 2.03) (veraltet)

Version: 2.03

Aurora MySQL 2.03 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Beim Erstellen eines neuen Aurora MySQL-DB-Clusters können Sie die Kompatibilität mit entweder MySQL 5.7 oder MySQL 5.6 auswählen. Beim Wiederherstellen eines mit MySQL 5.6 kompatiblen Snapshots können Sie die Kompatibilität mit entweder MySQL 5.7 oder MySQL 5.6 auswählen.

Sie können Snapshots von Aurora MySQL 1.14.\*, 1.15.\*, 1.16.\*, 1.17.\*, 1.18.\*, 2.01.\* und 2.02.\* in Aurora-MySQL 2.03 wiederherstellen.

Wir erlauben kein direktes Upgrade von Aurora MySQL 1.\*-Clustern in Aurora MySQL 2.03 oder die Wiederherstellung in Aurora MySQL 2.03 aus einer Amazon S3-Backup. Wir planen, diese Einschränkungen in einer späteren Veröffentlichung von Aurora MySQL 2.\* aufzuheben.

#### Note

Diese Version ist derzeit in den Regionen AWS GovCloud (US-West) [us-gov-west-1] und China (Peking) [cn-north-1] nicht verfügbar. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

- Performance-Schema ist verfügbar.
- Ein Problem wurde behoben, bei dem Zombiesitzungen mit entferntem Status möglicherweise mehr CPU verbrauchen.
- Ein Problem mit einem Dead-Latch wurde behoben. Es tritt auf, wenn eine schreibgeschützte Transaktion eine Sperre für einen Datensatz auf dem Aurora-Writer erhält.
- Ein Problem wurde behoben, bei dem das Aurora-Replica ohne Kundenworkload möglicherweise eine hohe CPU-Auslastung aufweist.
- Mehrere Probleme wurden behoben, die einen Neustart des Aurora-Replicas oder des Aurora-Writers zufolge haben können.
- Die Diagnoseprotokollierung lässt sich nun überspringen, wenn die Datenträgerdurchsatzgrenze erreicht ist.

- Ein Speicherleck wurde behoben, bei dem die Binärprotokollierung auf dem Aurora-Writer aktiviert ist.

## Integration von MySQL-Fehlerbehebungen (Community Edition):

- REVERSE-SCAN AUF EINER PARTITIONIERTEN TABELLE FÜHRT ICP AUS – ORDER BY DESC (Fehler #24929748).
- JSON\_OBJECT ERSTELLT UNGÜLTIGEN JSON-CODE (Fehler #26867509).
- DAS EINFÜGEN GROßER JSON-DATEN DAUERT ÜBERMÄßIG LANGE. (Fehler #22843444).
- DIE PARTITIONIERTEN TABELLEN NUTZEN UNTER VERSION 5.7 MEHR SPEICHER ALS UNTER VERSION 5.6 (Fehler #25080442).

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon-Aurora-MySQL-Funktionen werden in Aurora-MySQL-Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora-MySQL-Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen zum synchronen Aufrufen AWS Lambda von Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Aurora MySQL 2.03 ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native

Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Aurora MySQL 2.03 bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Globale Transaktionskennungen (GTIDs) Aurora MySQL unterstützt GTIDs in Version 2.04 und höher.
- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`
- X Protocol

## Aurora-MySQL-Datenbank-Engine-Updates 08.10.2018 (Version 2.02.5) (veraltet)

Version: 2.02.5

Aurora MySQL 2.02.5 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Beim Erstellen eines neuen Aurora MySQL-DB-Clusters können Sie die Kompatibilität mit entweder MySQL 5.7 oder MySQL 5.6 auswählen. Beim Wiederherstellen eines mit MySQL 5.6 kompatiblen Snapshots können Sie die Kompatibilität mit entweder MySQL 5.7 oder MySQL 5.6 auswählen.

Sie können Snapshots von Aurora MySQL 1.14.\*, 1.15.\*, 1.16.\*, 1.17.\*, 1.18.\*, 2.01.\* und 2.02.\* in Aurora-MySQL 2.02.5 wiederherstellen. Sie können ebenfalls ein direktes Upgrade von Aurora MySQL 2.01.\* oder 2.02.\* auf Aurora MySQL 2.02.5 durchführen.



Wir erlauben kein direktes Upgrade von Aurora MySQL 1.\*-Clustern in Aurora MySQL 2.02.5 oder die Wiederherstellung in Aurora MySQL 2.02.5 aus einer Amazon S3-Backup. Wir planen, diese Einschränkungen in einer späteren Veröffentlichung von Aurora MySQL 2.\* aufzuheben.

Das Performance-Schema ist für diese Version von Aurora MySQL 5.7 deaktiviert. Aktualisierung auf Aurora 2.03 für Support des Performance-Schemas.

#### Note

Diese Version ist derzeit in den Regionen AWS GovCloud (US-West) [us-gov-west-1] und China (Peking) [cn-north-1] nicht verfügbar. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

- Es wurde ein Problem behoben, bei dem ein Aurora-Replica möglicherweise neu gestartet wird, wenn auf einer Tabelle ein Reverse-Scan ausgeführt wird.

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon-Aurora-MySQL-Funktionen werden in Aurora-MySQL-Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora-MySQL-Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen zum synchronen Aufrufen AWS Lambda von Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).

- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Aurora MySQL 2.02.5 ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Aurora MySQL 2.02.5 bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Globale Transaktionskennungen (GTIDs) Aurora MySQL unterstützt GTIDs in Version 2.04 und höher.
- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`
- X Protocol

## Aurora-MySQL-Datenbank-Engine-Updates 21.09.2018 (Version 2.02.4) (veraltet)

Version: 2.02.4

Aurora MySQL 2.02.4 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Beim Erstellen eines neuen Aurora MySQL-DB-Clusters können Sie die Kompatibilität mit entweder MySQL 5.7 oder MySQL 5.6 auswählen. Beim Wiederherstellen eines mit MySQL 5.6 kompatiblen Snapshots können Sie die Kompatibilität mit entweder MySQL 5.7 oder MySQL 5.6 auswählen.

Sie können Snapshots von Aurora MySQL 1.14.\*, 1.15.\*, 1.16.\*, 1.17.\*, 1.18.\*, 2.01.\* und 2.02.\* in Aurora-MySQL 2.02.4 wiederherstellen. Sie können ebenfalls ein direktes Upgrade von Aurora MySQL 2.01.\* oder 2.02.\* auf Aurora MySQL 2.02.4 durchführen.

Wir erlauben kein direktes Upgrade von Aurora MySQL 1.\*-Clustern in Aurora MySQL 2.02.4 oder die Wiederherstellung in Aurora MySQL 2.02.4 aus einer Amazon S3-Backup. Wir planen, diese Einschränkungen in einer späteren Veröffentlichung von Aurora MySQL 2.\* aufzuheben.

Das Performance-Schema ist für diese Version von Aurora MySQL 5.7 deaktiviert. Aktualisierung auf Aurora 2.03 für Support des Performance-Schemas.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

- Ein Stabilitätsproblem im Zusammenhang mit Volltextsuchindizes in Tabellen wurde behoben, die aus einem Aurora MySQL 5.6-Snapshot wiederhergestellt wurden.

## Integration von MySQL-Fehlerbehebungen (Community Edition):

- 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

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon-Aurora-MySQL-Funktionen werden in Aurora-MySQL-Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora-MySQL-Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen zum synchronen Aufrufen AWS Lambda von Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten aus MySQL mithilfe eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

## Kompatibilität mit MySQL 5.7

Aurora MySQL 2.02.4 ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Aurora MySQL 2.02.4 bietet aktuell keine Unterstützung für die folgenden MySQL-5.7-Funktionen:

- Globale Transaktionskennungen (GTIDs) Aurora MySQL unterstützt GTIDs in Version 2.04 und höher.
- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`
- X Protocol

## Aurora-MySQL-Datenbank-Engine-Updates 23.08.2018 (Version 2.02.3) (veraltet)

Version: 2.02.3

Aurora MySQL 2.02.3 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Beim Erstellen eines neuen Aurora MySQL-DB-Clusters können Sie die Kompatibilität mit entweder MySQL 5.7 oder MySQL 5.6 auswählen. Beim Wiederherstellen eines mit MySQL 5.6 kompatiblen Snapshots können Sie die Kompatibilität mit entweder MySQL 5.7 oder MySQL 5.6 auswählen.

Sie können Snapshots von Aurora MySQL 1.14.\*, 1.15.\*, 1.16.\*, 1.17.\*, 2.01.\* und 2.02.\* in Aurora-MySQL 2.02.3 wiederherstellen. Sie können ebenfalls ein direktes Upgrade von Aurora MySQL 2.01.\* oder 2.02.\* auf Aurora MySQL 2.02.3 durchführen.

Wir erlauben kein direktes Upgrade von Aurora MySQL 1.\*-Clustern in Aurora MySQL 2.02.3 oder die Wiederherstellung in Aurora MySQL 2.02.3 aus einer Amazon S3-Backup. Wir planen, diese Einschränkungen in einer späteren Veröffentlichung von Aurora MySQL 2.\* aufzuheben.

Das Performance-Schema ist für diese Version von Aurora MySQL 5.7 deaktiviert. Aktualisierung auf Aurora 2.03 für Support des Performance-Schemas.

### Note

Diese Version ist derzeit in den Regionen AWS GovCloud (US-West) [us-gov-west-1] und China (Peking) [cn-north-1] nicht verfügbar. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Vergleich mit Aurora MySQL Version 1

Die folgenden Amazon-Aurora-MySQL-Funktionen werden in Aurora-MySQL-Version 1 (mit MySQL 5.6 kompatibel), jedoch derzeit nicht in Aurora-MySQL-Version 2 (MySQL 5.7 kompatibel) unterstützt.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen zum synchronen Aufrufen AWS Lambda von Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).

- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten von MySQL unter Verwendung eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

Derzeit unterstützt Aurora MySQL 2.01 keine der in Aurora MySQL Version 1.16 und höher hinzugefügten Funktionen. Weitere Informationen zu Aurora MySQL Version 1.16 finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).

## Kompatibilität mit MySQL 5.7

Aurora MySQL 2.02.3 ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Aurora MySQL 2.02.3 bietet aktuell keine Unterstützung für die folgenden MySQL 5.7-Funktionen:

- Globale Transaktionskennungen (GTIDs) Aurora MySQL unterstützt GTIDs in Version 2.04 und höher.
- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung CREATE TABLESPACE
- X Protocol

## CLI-Unterschiede zwischen Aurora MySQL 2.x und Aurora MySQL 1.x

- Der Engine-Name für Aurora MySQL 2.x lautet `aurora-mysql`. Der Engine-Name für Aurora MySQL 1.x ist weiterhin `aurora`.
- Die Standardparametergruppe für Aurora MySQL 2.x ist `default.aurora-mysql5.7`. Die Standardparametergruppe für Aurora MySQL 1.x ist weiterhin `default.aurora5.6`.
- Der DB-Cluster-Parametergruppenfamiliename für Aurora MySQL 2.x ist `aurora-mysql5.7`. Der DB-Cluster-Parametergruppenfamiliename für Aurora MySQL 1.x ist weiterhin `aurora5.6`.

Den vollständigen CLI-Befehlssatz sowie eine Auflistung der Unterschiede zwischen Aurora MySQL 2.x und Aurora MySQL 1.x finden Sie in der Aurora-Dokumentation.

## Verbesserungen

- Es wurde ein Problem behoben, bei dem eine Aurora-Replica neu gestartet werden konnte, wenn optimistische Cursor-Wiederherstellungen beim Lesen von Datensätzen verwendet wurden.
- Der Standardwert des Parameters `innodb_stats_persistent_sample_pages` wurde zur Verbesserung der Indexstatistiken auf 128 aktualisiert.
- Ein Problem wurde behoben, bei dem ein Aurora-Replica beim Zugriff auf eine Tabelle, die derzeit in der primären Aurora-Instance geändert wird, möglicherweise neu gestartet wird.
- `ANALYZE TABLE` wurde behoben, damit der Tabellendefinitions-cache nicht bereinigt wird.
- Ein Problem wurde behoben, bei dem die primäre Aurora-Instance oder ein Aurora-Replica beim Umwandeln einer Punkt-Abfrage für räumliche Daten zu einem Suchbereich möglicherweise neu gestartet wird.

## Aurora-MySQL-Datenbank-Engine-Updates 04.06.2018 (Version 2.02.2) (veraltet)

Version: 2.02.2

Aurora MySQL 2.02.2 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.



Beim Erstellen eines neuen Aurora MySQL-DB-Clusters können Sie die Kompatibilität mit entweder MySQL 5.7 oder MySQL 5.6 auswählen. Beim Wiederherstellen eines mit MySQL 5.6 kompatiblen Snapshots können Sie die Kompatibilität mit entweder MySQL 5.7 oder MySQL 5.6 auswählen.

Sie können Snapshots von Aurora MySQL 1.14.\*, 1.15.\*, 1.16.\*, 1.17.\*, 2.01.\* und 2.02 in Aurora MySQL 2.02.2 wiederherstellen. Sie können ebenfalls ein direktes Upgrade von Aurora MySQL 2.01.\* oder 2.02 auf Aurora MySQL 2.02.2 durchführen.

Wir erlauben kein direktes Upgrade von Aurora MySQL 1.\*-Clustern in Aurora MySQL 2.02.2 oder die Wiederherstellung in Aurora MySQL 2.02.2 aus einer Amazon S3-Backup. Wir planen, diese Einschränkungen in einer späteren Veröffentlichung von Aurora MySQL 2.\* aufzuheben.

Das Performance-Schema ist für diese Version von Aurora MySQL 5.7 deaktiviert. Aktualisierung auf Aurora 2.03 für Support des Performance-Schemas.

#### Note

Diese Version ist derzeit in den Regionen AWS GovCloud (US-West) [us-gov-west-1] und China (Peking) [cn-north-1] nicht verfügbar. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Kritische Fehlerbehebungen:

- [CVE-2016-3486](#)

## Vergleich mit Aurora MySQL 5.6

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL 5.6 unterstützt, in Aurora MySQL 5.7 derzeit jedoch nicht.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.

- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen zum synchronen Aufrufen AWS Lambda von Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten von MySQL unter Verwendung eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

Derzeit unterstützt Aurora MySQL 2.01 keine der in Aurora MySQL Version 1.16 und höher hinzugefügten Funktionen. Weitere Informationen zu Aurora MySQL Version 1.16 finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).

## Kompatibilität mit MySQL 5.7

Aurora MySQL 2.02.2 ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Aurora MySQL 2.02.2 bietet aktuell keine Unterstützung für die folgenden MySQL 5.7-Funktionen:

- Globale Transaktionskennungen (GTIDs) Aurora MySQL unterstützt GTIDs in Version 2.04 und höher.
- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen

- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`
- X Protocol

## CLI-Unterschiede zwischen Aurora MySQL 2.x und Aurora MySQL 1.x

- Der Engine-Name für Aurora MySQL 2.x lautet `aurora-mysql`. Der Engine-Name für Aurora MySQL 1.x ist weiterhin `aurora`.
- Die Standardparametergruppe für Aurora MySQL 2.x ist `default.aurora-mysql5.7`. Die Standardparametergruppe für Aurora MySQL 1.x ist weiterhin `default.aurora5.6`.
- Der DB-Cluster-Parametergruppenfamilienname für Aurora MySQL 2.x ist `aurora-mysql5.7`. Der DB-Cluster-Parametergruppenfamilienname für Aurora MySQL 1.x ist weiterhin `aurora5.6`.

Den vollständigen CLI-Befehlssatz sowie eine Auflistung der Unterschiede zwischen Aurora MySQL 2.x und Aurora MySQL 1.x finden Sie in der Aurora-Dokumentation.

## Verbesserungen

- Es wurde ein Problem behoben, bei dem ein Aurora Writer gelegentlich neu gestartet werden konnte, wenn er den Aurora Replica-Fortschritt verfolgte.
- Es wurde ein Problem behoben, bei dem eine Aurora-Replica neu startet oder einen Fehler aufwirft, wenn auf eine partitionierte Tabelle zugegriffen wird, nachdem Index Create- oder Drop-Anweisungen für die Tabelle auf dem Aurora-Writer ausgeführt wurden.
- Es wurde ein Problem behoben, bei dem der Zugriff auf eine Tabelle auf einer Aurora-Replica nicht möglich war, während die Änderungen, die durch die Ausführung von `ALTER table ADD/DROP column`-Anweisungen auf dem Aurora-Writer verursacht wurden, angewendet wurden.

## Aurora-MySQL-Datenbank-Engine-Updates 03.05.2018 (Version 2.02) (veraltet)

Version: 2.02

Aurora MySQL 2.02 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Beim Erstellen eines neuen Aurora MySQL-DB-Clusters können Sie die Kompatibilität mit entweder MySQL 5.7 oder MySQL 5.6 auswählen. Beim Wiederherstellen eines mit MySQL 5.6 kompatiblen Snapshots können Sie die Kompatibilität mit entweder MySQL 5.7 oder MySQL 5.6 auswählen.

Sie können Snapshots von Aurora MySQL 1.14.\*, 1.15.\*, 1.16.\*, 1.17.\* und 2.01.\* in Aurora-MySQL 2.02 wiederherstellen. Sie können ebenfalls ein direktes Upgrade von Aurora MySQL 2.01.\* auf Aurora MySQL 2.02 durchführen.

Wir erlauben kein direktes Upgrade von Aurora MySQL 1.x-Clustern in Aurora MySQL 2.02 oder die Wiederherstellung in Aurora MySQL 2.02 aus einer Amazon S3-Backup. Wir haben vor, diese Einschränkungen in einer späteren Veröffentlichung von Aurora MySQL 2.x aufzuheben.

Das Performance-Schema ist für diese Version von Aurora MySQL 5.7 deaktiviert. Aktualisierung auf Aurora 2.03 für Support des Performance-Schemas.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Vergleich mit Aurora MySQL 5.6

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL 5.6 unterstützt, in Aurora MySQL 5.7 derzeit jedoch nicht.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen zum synchronen Aufrufen AWS Lambda von Funktionen. Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten von MySQL unter Verwendung eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

Derzeit unterstützt Aurora MySQL 2.01 keine der in Aurora MySQL Version 1.16 und höher hinzugefügten Funktionen. Weitere Informationen zu Aurora MySQL Version 1.16 finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).

## Kompatibilität mit MySQL 5.7

Aurora MySQL 2.02 ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Aurora MySQL 2.02 bietet aktuell keine Unterstützung für die folgenden MySQL 5.7-Funktionen:

- Globale Transaktionskennungen (GTIDs) Aurora MySQL unterstützt GTIDs in Version 2.04 und höher.
- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`
- X Protocol

## CLI-Unterschiede zwischen Aurora MySQL 2.x und Aurora MySQL 1.x

- Der Engine-Name für Aurora MySQL 2.x lautet `aurora-mysql`. Der Engine-Name für Aurora MySQL 1.x ist weiterhin `aurora`.
- Die Standardparametergruppe für Aurora MySQL 2.x ist `default.aurora-mysql5.7`. Die Standardparametergruppe für Aurora MySQL 1.x ist weiterhin `default.aurora5.6`.

- Der DB-Cluster-Parametergruppenfamilienname für Aurora MySQL 2.x ist `aurora-mysql15.7`. Der DB-Cluster-Parametergruppenfamilienname für Aurora MySQL 1.x ist weiterhin `aurora5.6`.

Den vollständigen CLI-Befehlssatz sowie eine Auflistung der Unterschiede zwischen Aurora MySQL 2.x und Aurora MySQL 1.x finden Sie in der Aurora-Dokumentation.

## Verbesserungen

- Ein Problem wurde behoben, durch das ein Aurora Writer neu gestartet wird, wenn INSERT-Anweisungen mit Fast-Insert-Optimierung ausgeführt werden.
- Ein Problem wurde behoben, durch das ein Aurora-Replica neu gestartet wird, wenn mit diesem Aurora-Replica ALTER DATABASE-Anweisungen ausgeführt werden.
- Ein Problem wurde behoben, durch das ein Aurora Replica neu gestartet wird, wenn Abfragen mit Tabellen ausgeführt wurden, die gerade auf dem Aurora Writer entfernt wurden.
- Ein Problem wurde behoben, durch das ein Aurora-Replica neu gestartet wird, wenn die Einstellung `innodb_adaptive_hash_index` im Aurora Replica auf OFF gesetzt wird.
- Ein Problem wurde behoben, durch das ein Aurora-Replica neu gestartet wird, wenn TRUNCATE TABLE-Abfragen auf dem Aurora-Writer ausgeführt werden.
- Ein Problem wurde behoben, durch das der Aurora Writer unter bestimmten Umständen beim Ausführen von INSERT-Anweisungen einfriert In einem Cluster mit mehreren Knoten kann dies zu einem Failover führen.
- Ein Speicherleck im Zusammenhang mit dem Zuweisen von Sitzungsvariablen wurde beseitigt.
- Ein Problem wurde behoben, durch das der Aurora Writer unter bestimmten Umständen beim Bereinigen der rückgängig-Schritte für Tabellen mit generierten Spalten einfriert.
- Ein Problem wurde behoben, durch das der Aurora Writer beim Aktivieren der Binärprotokollierung gelegentlich neu gestartet wird.

## Integration von MySQL-Fehlerbehebungen

- Left Join gibt falsche Ergebnisse auf der äußeren Seite zurück (Fehlernr. 22833364).

# Aurora-MySQL-Datenbank-Engine-Updates 13.03.2018 (Version 2.01.1) (veraltet)

Version: 2.01.1

Aurora MySQL 2.01.1 ist allgemein verfügbar. Aurora MySQL 2.x-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.x-Versionen sind mit MySQL 5.6 kompatibel.

Beim Erstellen eines neuen Aurora MySQL-DB-Clusters können Sie die Kompatibilität mit entweder MySQL 5.7 oder MySQL 5.6 auswählen. Beim Wiederherstellen eines mit MySQL 5.6 kompatiblen Snapshots können Sie die Kompatibilität mit entweder MySQL 5.7 oder MySQL 5.6 auswählen.

Sie können Snapshots von Aurora MySQL 1.14.\*, 1.15.\*, 1.16.\* und 1.17.\* in Aurora MySQL 2.01.1 wiederherstellen.

Wir erlauben kein direktes Upgrade von Aurora MySQL 1.x-Clustern in Aurora MySQL 2.01.1 oder die Wiederherstellung in Aurora MySQL 2.01.1 aus einer Amazon S3-Backup. Wir haben vor, diese Einschränkungen in einer späteren Veröffentlichung von Aurora MySQL 2.x aufzuheben.

Das Performance-Schema ist für diese Version von Aurora MySQL 5.7 deaktiviert. Aktualisierung auf Aurora 2.03 für Support des Performance-Schemas.

## Vergleich mit Aurora MySQL 5.6

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL 5.6 unterstützt, in Aurora MySQL 5.7 derzeit jedoch nicht.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen zum synchronen Aufrufen von Funktionen. AWS Lambda Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).

- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten von MySQL unter Verwendung eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

Derzeit unterstützt Aurora MySQL 2.01.1 keine der in Aurora MySQL Version 1.16 und höher hinzugefügten Funktionen. Weitere Informationen zu Aurora MySQL Version 1.16 finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).

## Kompatibilität mit MySQL 5.7

Aurora MySQL 2.01.1 ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Aurora MySQL 2.01.1 bietet aktuell keine Unterstützung für die folgenden MySQL 5.7-Funktionen:

- Globale Transaktionskennungen (GTIDs) Aurora MySQL unterstützt GTIDs in Version 2.04 und höher.
- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`
- X Protocol



## CLI-Unterschiede zwischen Aurora MySQL 2.x und Aurora MySQL 1.x

- Der Engine-Name für Aurora MySQL 2.x lautet `aurora-mysql`. Der Engine-Name für Aurora MySQL 1.x ist weiterhin `aurora`.
- Die Standardparametergruppe für Aurora MySQL 2.x ist `default.aurora-mysql5.7`. Die Standardparametergruppe für Aurora MySQL 1.x ist weiterhin `default.aurora5.6`.
- Der DB-Cluster-Parametergruppenfamiliename für Aurora MySQL 2.x ist `aurora-mysql5.7`. Der DB-Cluster-Parametergruppenfamiliename für Aurora MySQL 1.x ist weiterhin `aurora5.6`.

Den vollständigen CLI-Befehlssatz sowie eine Auflistung der Unterschiede zwischen Aurora MySQL 2.x und Aurora MySQL 1.x finden Sie in der Aurora-Dokumentation.

## Verbesserungen

- Behebung des Problems einer Snapshot-Wiederherstellung, bei der Aurora-spezifische Datenbankberechtigungen fehlerhaft erstellt wurden, wenn ein MySQL 5.6-kompatibler Snapshot mit MySQL 5.7-Kompatibilität wiederhergestellt wurde.
- Unterstützung für 1.17-Snapshot-Wiederherstellungen hinzugefügt.

## Aurora-MySQL-Datenbank-Engine-Updates 06.02.2018 (Version 2.01) (veraltet)

Version: 2.01

Aurora MySQL 2.01 ist allgemein verfügbar. In Zukunft werden Aurora MySQL 2.x-Versionen kompatibel mit MySQL 5.7 und Aurora MySQL 1.x-Versionen kompatibel mit MySQL 5.6 sein.

Beim Erstellen eines neuen Aurora MySQL-DB-Clusters (einschließlich der aus Snapshots wiederhergestellten) können Sie die Kompatibilität mit entweder MySQL 5.7 oder MySQL 5.6 auswählen.

Sie können Snapshots von Aurora MySQL 1.14.\*, 1.15.\* und 1.16.\* in Aurora MySQL 2.01 wiederherstellen.

Wir erlauben kein direktes Upgrade von Aurora MySQL 1.x-Clustern in Aurora MySQL 2.01 oder die Wiederherstellung in Aurora MySQL 2.01 aus einer Amazon S3-Backup. Wir haben vor, diese Einschränkungen in einer späteren Veröffentlichung von Aurora MySQL 2.x aufzuheben.

Das Performance-Schema ist für diese Version von Aurora MySQL 5.7 deaktiviert. Aktualisierung auf Aurora 2.03 für Support des Performance-Schemas.

## Vergleich mit Aurora MySQL 5.6

Die folgenden Amazon Aurora MySQL-Funktionen werden in Aurora MySQL 5.6 unterstützt, in Aurora MySQL 5.7 derzeit jedoch nicht.

- Asynchrones Key Prefetch (AKP). Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.
- Hash-Verknüpfungen. Weitere Informationen finden Sie unter [Optimieren großer Aurora-MySQL-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.
- Native Funktionen zum synchronen Aufrufen von Funktionen. AWS Lambda Weitere Informationen finden Sie unter [Aufrufen einer Lambda-Funktion mit einer nativen Aurora-MySQL-Funktion](#) im Amazon-Aurora-Benutzerhandbuch.
- Scan-Batching. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).
- Migrieren von Daten aus MySQL mithilfe eines Amazon S3-Buckets. Weitere Informationen finden Sie unter [Migrieren von Daten von MySQL unter Verwendung eines Amazon-S3-Buckets](#) im Amazon-Aurora-Benutzerhandbuch.

Derzeit unterstützt Aurora MySQL 2.01 keine der in Aurora MySQL Version 1.16 und höher hinzugefügten Funktionen. Weitere Informationen zu Aurora MySQL Version 1.16 finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#).

## Kompatibilität mit MySQL 5.7

Aurora MySQL 2.01 ist drahtkompatibel mit MySQL 5.7 und enthält Funktionen wie JSON-Unterstützung, räumliche Indizes und generierte Spalten. Aurora MySQL verwendet eine native Implementierung der räumlichen Indexierung unter Verwendung von Kurven der Z-Ordnung, um eine > 20 x bessere Schreibleistung und eine > 10 x bessere Leseleistung als MySQL 5.7 für räumliche Datensätze zu liefern.

Aurora MySQL 2.01 bietet aktuell keine Unterstützung für die folgenden MySQL 5.7-Funktionen:

- Globale Transaktionskennungen (GTIDs) Aurora MySQL unterstützt GTIDs in Version 2.04 und höher.

- Plug-In für die Gruppenreplikation
- Größere Seitengröße
- Laden des InnoDB-Pufferpools beim Starten
- Plugin für den InnoDB-Volltext-Parser
- Replikation aus mehreren Quellen
- Größenanpassung des Online-Pufferpools
- Plugin für die Passwortvalidierung
- Plugins für die Umformulierung von Abfragen
- Replikationsfilter
- Die SQL-Anweisung `CREATE TABLESPACE`
- X Protocol

## CLI-Unterschiede zwischen Aurora MySQL 2.x und Aurora MySQL 1.x

- Der Engine-Name für Aurora MySQL 2.x lautet `aurora-mysql`. Der Engine-Name für Aurora MySQL 1.x ist weiterhin `aurora`.
- Die Standardparametergruppe für Aurora MySQL 2.x ist `default.aurora-mysql5.7`. Die Standardparametergruppe für Aurora MySQL 1.x ist weiterhin `default.aurora5.6`.
- Der DB-Cluster-Parametergruppenfamilienname für Aurora MySQL 2.x ist `aurora-mysql5.7`. Der DB-Cluster-Parametergruppenfamilienname für Aurora MySQL 1.x ist weiterhin `aurora5.6`.

Den vollständigen CLI-Befehlssatz sowie eine Auflistung der Unterschiede zwischen Aurora MySQL 2.x und Aurora MySQL 1.x finden Sie in der Aurora-Dokumentation.

# Aktualisierungen der Datenbank-Engine für Amazon Aurora MySQL Version 1 (veraltet)

Im Folgenden finden Sie die Aktualisierungen der Amazon Aurora Version 1 Datenbank-Engine:

- [Aurora-MySQL-Datenbank-Engine-Updates 30.09.2021 \(Version 1.23.4\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 28.06.2021 \(Version 1.23.3\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 18.03.2021 \(Version 1.23.2\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 24.11.2020 \(Version 1.23.1\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 02.09.2020 \(Version 1.23.0\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 03.06.2021 \(Version 1.22.5\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 04.03.2021 \(Version 1.22.4\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 09.11.2020 \(Version 1.22.3\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 05.03.2020 \(Version 1.22.2\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 23.12.2019 \(Version 1.22.1\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 25.11.2019 \(Version 1.22.0\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 25.11.2019 \(Version 1.21.0\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 05.03.2020 \(Version 1.20.1\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 11.11.2019 \(Version 1.20.0\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 05.03.2020 \(Version 1.19.6\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 19.09.2019 \(Version 1.19.5\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 05.06.2019 \(Version 1.19.2\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 09.05.2019 \(Version 1.19.1\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 07.02.2019 \(Version 1.19.0\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 20.09.2018 \(Version 1.18.0\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 05.03.2020 \(Version 1.17.9\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 17.01.2019 \(Version 1.17.8\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 08.10.2018 \(Version 1.17.7\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 06.09.2018 \(Version 1.17.6\) \(veraltet\)](#)

- [Aurora-MySQL-Datenbank-Engine-Updates 14.08.2018 \(Version 1.17.5\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 07.08.2018 \(Version 1.17.4\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 05.06.2018 \(Version 1.17.3\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 27.04.2018 \(Version 1.17.2\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 23.03.2018 \(Version 1.17.1\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 13.03.2018 \(Version 1.17\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 \(Version 1.16\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 20.11.2017 \(Version 1.15.1\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 24.10.2017 \(Version 1.15\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 13.03.2018 \(Version 1.14.4\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 22.09.2017 \(Version 1.14.1\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 07.08.2017 \(Version 1.14\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 15.05.2017 \(Version 1.13\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 05.04.2017 \(Version 1.12\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 23.02.2017 \(Version 1.11\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 12.01.2017 \(Version 1.10.1\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 14.12.2016 \(Version 1.10\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 10.11.2016 \(Versionen 1.9.0, 1.9.1\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 26.10.2016 \(Version 1.8.1\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 18.10.2016 \(Version 1.8\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 20.09.2016 \(Version 1.7.1\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 30.08.2016 \(Version 1.7.0\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 01.06.2016 \(Version 1.6.5\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 06.04.2016 \(Version 1.6\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 11.01.2016 \(Version 1.5\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 03.12.2015 \(Version 1.4\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 16.10.2015 \(Versionen 1.2, 1.3\) \(veraltet\)](#)
- [Aurora-MySQL-Datenbank-Engine-Updates 24.08.2015 \(Version 1.1\) \(veraltet\)](#)

# Aurora-MySQL-Datenbank-Engine-Updates 30.09.2021 (Version 1.23.4) (veraltet)

Version: 1.23.4

Aurora MySQL 1.23.4 ist allgemein verfügbar. Aurora MySQL 2.\*-Versionen sind mit MySQL 5.7 kompatibel. Aurora MySQL 1.\*-Versionen sind mit MySQL 5.6 kompatibel.

Diese Engine-Version wird planmäßig am 28. Februar 2023 veralten. Weitere Informationen finden Sie unter [Vorbereitung auf das Lebenszyklusende der mit Amazon Aurora MySQL kompatiblen Edition Version 1](#).

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Geben Sie zum Erstellen eines Clusters mit einer älteren Version von Aurora MySQL die Engine-Version über die RDS-Konsole, die AWS-CLI oder die Amazon-RDS-API an.

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Allgemeine Verbesserungen:

- Es wurde ein Problem behoben, das aufgrund der übermäßigen Protokollierung von Informationsmeldungen in internen Diagnoseprotokolldateien zu einem hohen CPU-Verbrauch bei den Reader-Instances führen konnte.

Wichtige Fehlerbehebungen:

- [CVE-2021-2307](#)
- [CVE-2021-2226](#)
- [CVE-2021-2160](#)
- [CVE-2021-2154](#)
- [CVE-2021-2060](#)
- [CVE-2021-2032](#)

- [CVE-2021-2001](#)

## Aurora-MySQL-Datenbank-Engine-Updates 28.06.2021 (Version 1.23.3) (veraltet)

Version: 1.23.3

Aurora MySQL 1.23.3 ist allgemein verfügbar. Aurora MySQL 1.\*-Versionen sind mit MySQL 5.6 kompatibel. Aurora MySQL 2.\*-Versionen sind mit MySQL 5.7 kompatibel.

Diese Engine-Version wird planmäßig am 28. Februar 2023 veralten. Weitere Informationen finden Sie unter [Vorbereitung auf das Lebenszyklusende der mit Amazon Aurora MySQL kompatiblen Edition Version 1](#).

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Geben Sie zum Erstellen eines Clusters mit einer älteren Version von Aurora MySQL die Engine-Version über die RDS-Konsole, die AWS-CLI oder die Amazon-RDS-API an.

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Allgemeine Stabilitäts- und Verfügbarkeitsverbesserungen

Fehlerbehebungen bei der Sicherheit:

- [CVE-2021-23841](#)
- [CVE-2021-3449](#)
- [CVE-2020-28196](#)

## Aurora-MySQL-Datenbank-Engine-Updates 18.03.2021 (Version 1.23.2) (veraltet)

Version: 1.23.2

Aurora MySQL 1.23.2 ist allgemein verfügbar. Aurora MySQL 1.\*-Versionen sind mit MySQL 5.6 kompatibel. Aurora MySQL 2.\*-Versionen sind mit MySQL 5.7 kompatibel.

Diese Engine-Version wird planmäßig am 28. Februar 2023 veralten. Weitere Informationen finden Sie unter [Vorbereitung auf das Lebenszyklusende der mit Amazon Aurora MySQL kompatiblen Edition Version 1](#).

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Geben Sie zum Erstellen eines Clusters mit einer älteren Version von Aurora MySQL die Engine-Version über die RDS-Konsole, die AWS-CLI oder die Amazon-RDS-API an.

#### Note

Diese Version ist derzeit in den folgenden Regionen nicht verfügbar: AWS GovCloud (USA-Ost) [us-gov-east-1], AWS GovCloud (USA-West) [us-gov-west-1]. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Wichtige Fehlerbehebungen:

- [CVE-2020-14867](#)
- [CVE-2020-14812](#)
- [CVE-2020-14769](#)
- [CVE-2020-14765](#)
- [CVE-2020-14793](#)
- [CVE-2020-14672](#)
- [CVE-2020-1971](#)
- [CVE-2018-3143](#)

Verbesserungen der Verfügbarkeit:



- Es wurde ein Problem in der Funktion zur Größenänderung des dynamischen Clusterspeichers behoben, das zum Neustart von Reader-DB-Instances führen kann.
- Ein Failover-Problem aufgrund einer Race-Bedingung in der `RESET QUERY CACHE`-Anweisung wurde behoben.
- Ein Absturz in einem verschachtelten Prozeduraufruf mit Abfrage-Cache wurde behoben
- Ein Problem wurde behoben, um einen wiederholten Neustart von `mysqld` bei der Wiederherstellung von einer unvollständigen Kürzung von partitionierten oder unterpartitionierten Tabellen zu verhindern.
- Es wurde ein Problem behoben, das zur Verhinderung der Migration von On-Prem oder RDS for MySQL auf Aurora MySQL führen konnte.
- Es wurde eine seltene Race-Bedingung behoben, bei der die Datenbank während der Skalierung des Speichervolumens hätte neu gestartet werden können.
- Es wurde ein Problem im Sperrverwalter behoben, bei dem eine Race-Bedingung dazu führen konnte, dass eine Sperre von zwei Transaktionen gemeinsam genutzt wurde, was zu einem Neustart der Datenbank führte.
- Es wurde ein Problem im Zusammenhang mit der Transaktionssperr-Speicherverwaltung bei lang ausgeführten Schreibtransaktionen behoben, was zu einem Neustart der Datenbank führte.
- Es wurde eine race-Bedingung im Lock-Manager behoben, die zu einem Neustart oder Failover der Datenbank beim Transaktionsrollback führte.
- Es wurde ein Problem während des Upgrades von 5.6 auf 5.7 behoben, wenn in der Tabelle Fast Online DDL im Labor-Modus in 5.6 aktiviert war.
- Es wurden mehrere Probleme behoben, bei denen die Engine während des Zero-Downtime-Patchings neu gestartet werden konnte, während sie für das Patching nach einem ruhigen Punkt in der Datenbankaktivität suchte.
- Es wurden mehrere Probleme im Zusammenhang mit wiederholten Neustarts aufgrund von unterbrochenen DDL-Vorgängen behoben, z. B. `DROP TRIGGER`, `ALTER TABLE`, und speziell `ALTER TABLE`, das die Art der Partitionierung oder die Anzahl der Partitionen in einer Tabelle ändert.
- Der Standardwert von `table_open_cache` auf 16XL- und 24XL-Instances wurde aktualisiert, um wiederholte Neustarts und hohe CPU-Auslastung bei großen Instance-Klassen (R4/R5-16XL, R5-12XL, R5-24XL) zu vermeiden. Dies wirkte sich auf Versionen 1.21.x und 1.22.x aus.
- Es wurde ein Problem behoben, das dazu führte, dass ein Binlog-Replikat mit einem `HA_ERR_KEY_NOT_FOUND`-Fehler stoppt.

## Integration von MySQL-Fehlerbehebungen (Community Edition):

- Replikation: Während eine `SHOW BINLOG EVENTS` Anweisung ausgeführt wurde, wurde jede parallele Transaktion blockiert. Der Fix stellt sicher, dass der `SHOW BINLOG EVENTS` Prozess jetzt nur noch eine Sperre für die Dauer der Berechnung der Endposition der Datei erwirbt, daher werden parallele Transaktionen nicht für lange Zeiträume blockiert. (Fehler #76618, Bug #20928790)

## Aurora-MySQL-Datenbank-Engine-Updates 24.11.2020 (Version 1.23.1) (veraltet)

Version: 1.23.1

Aurora MySQL 1.23.1 ist allgemein verfügbar. Aurora MySQL 1.\*-Versionen sind mit MySQL 5.6 kompatibel. Aurora MySQL 2.\*-Versionen sind mit MySQL 5.7 kompatibel.

Diese Engine-Version wird planmäßig am 28. Februar 2023 veralten. Weitere Informationen finden Sie unter [Vorbereitung auf das Lebenszyklusende der mit Amazon Aurora MySQL kompatiblen Edition Version 1](#).

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Geben Sie zum Erstellen eines Clusters mit einer älteren Version von Aurora MySQL die Engine-Version über die RDS-Konsole, die AWS-CLI oder die Amazon-RDS-API an.

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Fehlerbehebungen bei der Sicherheit:

Korrekturen und andere Verbesserungen bei der Feinabstimmung der Handhabung in einer verwalteten Umgebung. Weitere CVE Fixes unten:

- [CVE-2020-14559](#)
- [CVE-2020-14539](#)

## Inkompatible Änderungen:

Diese Version führt eine Berechtigungsänderung ein, die sich auf das Verhalten des `mysqldump` Befehls auswirkt. Benutzer müssen das `PROCESS` Recht haben, auf die `INFORMATION_SCHEMA.FILES` Tabelle zuzugreifen. Um den `mysqldump` Befehl ohne Änderungen auszuführen, erteilen Sie dem Datenbankbenutzer, mit dem sich der `PROCESS` Befehl verbindet, die `mysqldump` Berechtigung. Sie können den `mysqldump` Befehl auch mit der `--no-tablespaces` Option ausführen. Mit dieser Option enthält die `mysqldump` Ausgabe keine `CREATE LOGFILE GROUP` oder `CREATE TABLESPACE` Anweisungen. In diesem Fall greift der `mysqldump` Befehl nicht auf die `INFORMATION_SCHEMA.FILES` Tabelle zu, und Sie müssen die `PROCESS` Berechtigung nicht erteilen.

## Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem behoben, durch das eine Aurora Reader-Instance in einem sekundären globalen Datenbankcluster mit 1.23.0 wiederholt neu gestartet wurde.
- Es wurde ein Problem behoben, bei dem die Replikate einer sekundären Datenbankregion beim Upgrade auf Version 1.23.0 neu gestartet werden konnten, während der primäre Region-Writer eine ältere Release-Version hatte.
- Es wurde ein Speicherleck in der Funktion zur dynamischen Größenänderung behoben, das in Aurora MySQL 1.23.0 eingeführt wurde.
- Es wurde ein Problem behoben, das während der Ausführung einer Abfrage mithilfe der Funktion für parallele Abfragen zu einem Serverneustart führen konnte.
- Es wurde ein Problem behoben, das dazu führen konnte, dass sich eine Clientsitzung aufhängte, wenn die Datenbankengine beim Lesen oder Schreiben in das Netzwerk auf einen Fehler stieß.

## Aurora-MySQL-Datenbank-Engine-Updates 02.09.2020 (Version 1.23.0) (veraltet)

Version: 1.23.0

Aurora MySQL 1.23.0 ist allgemein verfügbar. Aurora MySQL 1.\*-Versionen sind mit MySQL 5.6 kompatibel. Aurora MySQL 2.\*-Versionen sind mit MySQL 5.7 kompatibel.

Diese Engine-Version wird planmäßig am 28. Februar 2023 veralten. Weitere Informationen finden Sie unter [Vorbereitung auf das Lebenszyklusende der mit Amazon Aurora MySQL kompatiblen Edition Version 1.](#)

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Sie können den Snapshot einer Aurora MySQL 1.\*-Datenbank in Aurora MySQL 1.23.0 wiederherstellen.

#### Important

Die Verbesserungen des Aurora Speichers in dieser Version beschränken die verfügbaren Upgrade-Pfade von Aurora MySQL 1.23 auf Aurora MySQL 2.\*. Wenn Sie einen Aurora MySQL 1.23-Cluster auf 2.\* aktualisieren, müssen Sie ein Upgrade auf Aurora MySQL 2.09.0 oder höher durchführen.

Geben Sie zum Erstellen eines Clusters mit einer älteren Version von Aurora MySQL bitte die Engine-Version über die RDS-Konsole, die AWS-CLI oder die Amazon-RDS-API an.

#### Note

Diese Version ist derzeit in den folgenden Regionen nicht verfügbar: AWS GovCloud (USA-Ost) [us-gov-east-1], AWS GovCloud (USA-West) [us-gov-west-1]. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Neue Funktionen:

- Sie können jetzt durch Ändern des Werts für den DB-Cluster-Parameter parallele Abfragen für einen vorhandenen Cluster aktivieren oder deaktivieren `aurora_parallel_query`. Sie müssen beim Erstellen des Clusters nicht die Einstellung `parallelquery` für den Parameter `--engine-mode` verwenden.

Parallele Abfragen sind jetzt in allen Regionen verfügbar, in denen Aurora MySQL verfügbar ist.

Es gibt eine Reihe weiterer Funktionsverbesserungen und Änderungen für die Verfahren zum Aktualisieren und Aktivieren paralleler Abfragen in Aurora-Clustern. Weitere Informationen finden Sie im Abschnitt zum [Arbeiten mit parallelen Abfragen für Amazon Aurora MySQL](#) im Amazon-Aurora-Benutzerhandbuch.

- Mit dieser Version können Sie Amazon Aurora MySQL-Datenbank-Instances mit bis zu 128 Tebibyte (TiB) Speicher erstellen. Das neue Speicherlimit stellt eine Erhöhung gegenüber den vorherigen 64 TiB dar. Die Speichergröße von 128 TiB unterstützt größere Datenbanken. Diese Funktion wird für kleine Instance-Größen (db.t2 oder db.t3) nicht unterstützt. Ein einzelner Tabellenraum kann aufgrund von [InnoDB-Einschränkungen mit 16 KB Seitengröße](#) nicht über 64 TiB hinausgehen.

Aurora benachrichtigt Sie, wenn die Cluster-Volume-Größe nahe 128 TiB liegt, sodass Sie Maßnahmen ergreifen können, bevor Sie die Größenbeschränkung erreichen. Die Benachrichtigungen werden im mysql-Protokoll und die RDS-Ereignisse in der angezeigten AWS Management Console.

- Verbesserte Binärprotokoll-Verarbeitung zur Reduzierung der Wiederherstellungszeit nach einem Absturz und der Commit-Zeitlatenz bei sehr großen Transaktionen.
- Aurora passt die Größe Ihres Cluster-Speicherplatzes dynamisch an. Bei dynamischer Größenanpassung verringert sich der Speicherplatz für den Aurora-DB-Cluster automatisch, wenn Sie Daten aus dem DB-Cluster entfernen. Weitere Informationen finden Sie unter [Speicherskalierung](#) im Amazon-Aurora-Benutzerhandbuch.

#### Note

Die Funktion zur dynamischen Größenanpassung wird in Phasen in den AWS-Regionen bereitgestellt, in denen Aurora verfügbar ist. Abhängig von der Region, in der sich Ihr Cluster befindet, ist diese Funktion möglicherweise noch nicht verfügbar. Weitere Informationen finden Sie in der Ankündigung [„Was ist neu“](#).

Wichtige Fehlerbehebungen:

- [CVE-2019-2911](#)
- [CVE-2019-2537](#)
- [CVE-2018-2787](#)
- [CVE-2018-2784](#)

- [CVE-2018-2645](#)
- [CVE-2018-2640](#)

#### Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem im Sperrverwalter behoben, bei dem eine Race-Bedingung dazu führen konnte, dass eine Sperre von zwei Transaktionen gemeinsam genutzt wurde, was zu einem Neustart der Datenbank führte.
- Es wurde ein Problem im Zusammenhang mit der Transaktionssperr-Speicherverwaltung bei lang ausgeführten Schreibtransaktionen behoben, was zu einem Neustart der Datenbank führte.
- Es wurde eine race-Bedingung im Lock-Manager behoben, die zu einem Neustart oder Failover der Datenbank beim Transaktionsrollback führte.
- Es wurde ein Problem beim Upgrade von 5.6 auf 5.7 behoben, das auftrat, wenn sich das `innodb_file_format` in einer Tabelle änderte, bei der schnelle DDL aktiviert war.
- Es wurden mehrere Probleme behoben, bei denen die Engine während des Zero-Downtime-Patchings neu gestartet werden konnte, während sie für das Patching nach einem ruhigen Punkt in der Datenbankaktivität suchte.
- Es wurde ein Problem im Zusammenhang mit der DDL-Wiederherstellung behoben, das den Neustart der DB-Instance bei der Wiederherstellung einer unterbrochenen `DROP TRIGGER`-Operation beeinträchtigte.
- Es wurde ein Fehler behoben, der zur Nichtverfügbarkeit der Datenbank führen konnte, wenn während der Ausführung bestimmter Partitionierungsoperationen ein Absturz auftrat. Genauer gesagt, eine unterbrochene `ALTER TABLE`-Operation, bei der die Art der Partitionierung oder die Anzahl der Partitionen in einer Tabelle geändert wird.
- Korrigierter Standardwert von `table_open_cache` auf 16XL- und 24XL-Instances, der wiederholte Failover und hohe CPU-Auslastung auf großen Instance-Klassen (R4/R5-16XL, R5-12XL, R5-24XL) verursachen konnte. Dies wirkte sich auf 1.21.x und 1.22.x aus.

#### Globale Datenbanken:

- Füllen fehlender Daten in der MySQL-Ansicht `INFORMATION_SCHEMA.REPLICA_HOST_STATUS` zu primären und sekundären AWS-Regionen in einer globalen Aurora-Datenbank.
- Es wurden unerwartete Abfragefehler behoben, die in einer sekundären Global DB-Region aufgrund der Garbage Collection von rückgängig-Datensätzen in der primären Region nach

temporären Netzwerkkonnektivitätsproblemen zwischen der primären und der sekundären Region auftreten konnten.

#### Parallele Abfrage:

- Es wurde ein Problem behoben, bei dem die parallele Abfrage dazu führen konnte, dass eine lang laufende Abfrage ein leeres Ergebnis zurückgab.
- Es wurde ein Problem behoben, bei dem eine Abfrage in einer kleinen Tabelle auf der Aurora-Read-Replica länger als eine Sekunde dauern konnte.
- Es wurde ein Problem behoben, das zu einem Neustart führen konnte, wenn eine parallele Abfrage und eine DML-Anweisung unter einer hohen Workload gleichzeitig ausgeführt wurden.

#### Allgemeine Verbesserungen:

- Es wurde ein Problem behoben, bei dem Abfragen mit dem räumlichen Index Teilergebnisse zurückgeben konnten, wenn der räumliche Index auf Tabellen mit bereits vorhandenen großen räumlichen Werten erstellt wurde.
- Die maximal zulässige Länge für die Audit-Systemvariablen `server_audit_incl_users` und `server_audit_excl_users` wurde von 1.024 Bytes auf 2.000 Bytes erhöht.
- Es wurde ein Problem behoben, bei dem eine Binärprotokoll-Replica, die mit einem Aurora-MySQL-Binärprotokoll-Primary verbunden ist, unvollständige Daten anzeigen konnte, wenn das Aurora-MySQL-Binärprotokoll-Primary Daten von S3 unter `statement binlog_format` lud.
- Erfüllen des Community-Verhaltens, `mixed binlog_format row` statt `statement` zuzuordnen, um Daten zu laden.
- Es wurde ein Problem behoben, das dazu führte, dass die Binärprotokoll-Replikation nicht mehr funktionierte, wenn der Benutzer die Verbindung schloss und bei der Sitzung temporäre Tabellen verwendet wurden.
- Verbesserte Antwortzeit einer Abfrage mit temporären MyISAM-Tabellen.
- Es wurde ein Berechtigungsproblem behoben, wenn ein Binärprotokoll-Worker eine native Lambda-Funktion ausführte.
- Es wurde ein Problem bei Aurora-Read-Replicas behoben, wenn versucht wurde, das langsame Protokoll oder das allgemeine Protokoll abzufragen oder zu rotieren.

- Es wurde ein Problem behoben, durch das die logische Replikation unterbrochen wurde, wenn der Parameter `binlog_checksum` auf unterschiedliche Werte auf dem Master und dem Replikat festgelegt wurde.
- Es wurde ein Problem behoben, durch das Read Replicas vorübergehend Teilergebnisse einer kürzlich festgeschriebenen Transaktion auf dem Writer sehen konnten.
- Aufnahme von Transaktionsinformationen der zurückgerollten Transaktion in `show engine innodb status` bei Auflösen eines Deadlocks.

## Integration von MySQL-Fehlerbehebungen (Community Edition):

- Binärprotokoll-Ereignisse mit `ALTER TABLE ADD COLUMN ALGORITHM=QUICK` werden zu `ALGORITHM=DEFAULT` umgeschrieben, um mit der Community-Edition kompatibel zu sein.
- Fehler #22350047: WENN DER CLIENT NACH DEM ROLLBACK AUF DEN SAVEPOINT BEENDET WURDE, WURDEN VORHERIGE STMTS ÜBERTRAGEN
- Fehler #29915479: DAS AUSFÜHREN VON `COM_REGISTER_SLAVE` OHNE `COM_BINLOG_DUMP` KANN ZU EINEM BEENDEN DES SERVERS FÜHREN
- Fehler #30441969: FEHLER #29723340: ABSTURZ DES MYSQL-SERVERS NACH SQL-ABFRAGE MIT `DATA ?AST`
- Bug #30628268: ABSTURZ WEGEN FEHLENDEM SPEICHER
- Fehler #27081349: UNERWARTETES VERHALTEN BEIM LÖSCHEN MIT RÄUMLICHER FUNKTION
- Fehler #27230859: UNERWARTETES VERHALTEN BEIM HANDLING VON UNGÜLTIGEM POLYGON"
- Fehler #27081349: UNERWARTETES VERHALTEN BEIM LÖSCHEN MIT SPATIAL"
- Fehler #26935001: „ALTER TABLE AUTO\_INCREMENT“ VERSUCHT, DEN INDEX AUS VERWORFENEM TABELLENRAUM ZU LESEN
- Fehler #29770705: DER SERVER STÜRZTE BEIM AUSFÜHREN EINER AUSWAHL MIT SPEZIFISCHER WHERE-KLAUSEL AB
- Fehler #27659490: AUSWAHL MIT DYNAMISCHEN BEREICH UND INDEX-ZUSAMMENFÜHRUNG VERBRAUCHT ZU VIEL SPEICHER (NICHT GENÜGENDE SPEICHER)
- Fehler #24786290: REPLIKATION BRICHT NACH DEM AUFTRETEN DES FEHLERS #74145 IM MASTER AB
- Fehler #27703912: ÜBERMÄSSIGE SPEICHERNUTZUNG MIT VIEL VORBEREITUNG



- Fehler #20527363: ABSTURZ TEMPORÄRER TRUNCATE-TABELLE: !  
DICT\_TF2\_FLAG\_IS\_SET(TABELLE, DICT\_TF2\_TEMPORARY)
- Fehler #23103937 PS\_TRUNCATE\_ALL\_TABLES() FUNKTIONIERT NICHT IM MODUS SUPER\_READ\_ONLY
- Fehler #25053286: DIE VERWENDUNG DER ANSICHT MIT BEDINGUNG VERURSACHT FEHLERHAFTES VERHALTEN (behooben in 5.6.36)
- Fehler #25586773: FEHLERHAFTES VERHALTEN FÜR DAS ERSTELLEN EINER TABELLENAUSWAHL IN EINER SCHLEIFE IN SP (behooben in 5.6.39)
- Fehler #27407480: AUTOMATIC\_SP\_PRIVILEGES ERFORDERT EINFÜGEBERECHTIGUNGEN FÜR MYSQL.USER-TABELLE
- Fehler #26997096: Der relay\_log\_space-Wert wird nicht synchronisiert aktualisiert, sodass sein Wert manchmal viel höher ist als der tatsächlich von den Relay-Protokollen belegte Festplattenspeicher.
- Fehler #15831300 SLAVE\_TYPE\_CONVERSIONS=ALL\_NON\_LOSSY FUNKTIONIERT NICHT WIE ERWARTET
- SSL-Fehler Backport Fehler #17087862, Fehler #20551271
- Fehler #16894092: PERFORMANCE REGRESSION IN 5.6.6+ FOR INSERT INTO ... SELECT ... FROM (behooben in 5.6.15).
- Portierung einer Fehlerbehebung im Zusammenhang mit SLAVE\_TYPE\_CONVERSIONS.

## Aurora-MySQL-Datenbank-Engine-Updates 03.06.2021 (Version 1.22.5) (veraltet)

Version: 1.22.5

Aurora MySQL 1.22.5 ist allgemein verfügbar. Aurora MySQL 1.\*-Versionen sind mit MySQL 5.6 kompatibel. Aurora MySQL 2.\*-Versionen sind mit MySQL 5.7 kompatibel.

Diese Engine-Version wird planmäßig am 28. Februar 2023 veralten. Weitere Informationen finden Sie unter [Vorbereitung auf das Lebenszyklusende der mit Amazon Aurora MySQL kompatiblen Edition Version 1](#).

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Geben Sie zum Erstellen eines Clusters mit einer älteren Version von Aurora MySQL die Engine-Version über die RDS-Konsole, die AWS-CLI oder die Amazon-RDS-API an.

### Note

Diese Version ist als Long-Term Support- (LTS, Langzeit-Support)-Version ausgewiesen. Weitere Informationen finden Sie unter [Aurora-MySQL-Long-Term-Support \(LTS, Langzeit-Support\)-Versionen](#) im Amazon-Aurora-Benutzerhandbuch.

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem behoben, das dazu führen könnte, dass die Datenbank zum Stillstand gebracht und anschließend aufgrund eines Parallelitätskonflikts zwischen internen Bereinigungs-Threads neu gestartet oder fehlgeschlagen wurde.
- Es wurde ein Problem behoben, das dazu führen könnte, dass der Cluster nicht verfügbar wurde, wenn die Datenbank neu gestartet wurde, während XA-Transaktionen im vorbereiteten Zustand gehalten wurden und dann erneut gestartet wurden, bevor diese Transaktionen festgeschrieben oder zurückgesetzt wurden. Vor diesem Fix können Sie das Problem beheben, indem Sie den Cluster bis zu einem bestimmten Zeitpunkt vor dem ersten Neustart wiederherstellen.
- Es wurde ein Problem behoben, das dazu führen könnte, dass die InnoDB-Bereinigung blockiert wurde, wenn die Datenbank während der Verarbeitung einer DDL-Anweisung neu gestartet wird. Infolgedessen würde die Länge der InnoDB-Verlaufsliste zunehmen und das Cluster-Speichervolume würde weiter wachsen, bis es sich füllt, wodurch die Datenbank nicht verfügbar ist. Vor diesem Fix können Sie das Problem beheben, indem Sie die Datenbank erneut starten, um die Bereinigung zu entsperren.

## Aurora-MySQL-Datenbank-Engine-Updates 04.03.2021 (Version 1.22.4) (veraltet)

Version: 1.22.4

Aurora MySQL 1.22.4 ist allgemein verfügbar. Aurora MySQL 1.\*-Versionen sind mit MySQL 5.6 kompatibel. Aurora MySQL 2.\*-Versionen sind mit MySQL 5.7 kompatibel.

Diese Engine-Version wird planmäßig am 28. Februar 2023 veralten. Weitere Informationen finden Sie unter [Vorbereitung auf das Lebenszyklusende der mit Amazon Aurora MySQL kompatiblen Edition Version 1](#).

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Geben Sie zum Erstellen eines Clusters mit einer älteren Version von Aurora MySQL die Engine-Version über die RDS-Konsole, die AWS-CLI oder die Amazon-RDS-API an.

#### Note

Diese Version ist als Long-Term Support- (LTS, Langzeit-Support)-Version ausgewiesen. Weitere Informationen finden Sie unter [Aurora-MySQL-Long-Term-Support \(LTS, Langzeit-Support\)-Versionen](#) im Amazon-Aurora-Benutzerhandbuch.

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Fehlerbehebungen bei der Sicherheit:

Korrekturen und andere Verbesserungen bei der Feinabstimmung der Handhabung in einer verwalteten Umgebung. Weitere CVE Fixes unten:

- [CVE-2020-14867](#)
- [CVE-2020-14812](#)
- [CVE-2020-14793](#)
- [CVE-2020-14769](#)
- [CVE-2020-14765](#)
- [CVE-2020-14672](#)
- [CVE-2020-1971](#)

## Verbesserungen der Verfügbarkeit:

- Es wurde ein Problem behoben, das einen Neustart oder Failover der Datenbank während eines `kill session`-Befehls auslösen konnte. Wenn dieses Problem auftritt, wenden Sie sich an den AWS Support, um diesen Fix auf Ihrer Instance zu aktivieren.
- Verbesserte Binärprotokollierung zur Reduzierung der Wiederherstellungszeit nach einem Absturz und der Commit-Latenz bei großen Transaktionen.
- Es wurde ein Problem behoben, das dazu führte, dass ein Binlog-Replikat mit einem `HA_ERR_KEY_NOT_FOUND`-Fehler stoppt.

## Aurora-MySQL-Datenbank-Engine-Updates 09.11.2020 (Version 1.22.3) (veraltet)

Version: 1.22.3

Aurora MySQL 1.22.3 ist allgemein verfügbar. Aurora MySQL 1.\*-Versionen sind mit MySQL 5.6 kompatibel. Aurora MySQL 2.\*-Versionen sind mit MySQL 5.7 kompatibel.

Diese Engine-Version wird planmäßig am 28. Februar 2023 veralten. Weitere Informationen finden Sie unter [Vorbereitung auf das Lebenszyklusende der mit Amazon Aurora MySQL kompatiblen Edition Version 1](#).

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Geben Sie zum Erstellen eines Clusters mit einer älteren Version von Aurora MySQL die Engine-Version über die RDS-Konsole, die AWS-CLI oder die Amazon-RDS-API an.

### Note

Diese Version ist als Long-Term Support- (LTS, Langzeit-Support)-Version ausgewiesen. Weitere Informationen finden Sie unter [Versionen von Aurora MySQL Long-Term Support \(LTS\)](#) im Amazon-Aurora-Benutzerhandbuch.

Bei Fragen oder Bedenken steht Ihnen der AWS Support in den Community-Foren und über [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Fehlerbehebungen bei der Sicherheit:

Korrekturen und andere Verbesserungen bei der Feinabstimmung der Handhabung in einer verwalteten Umgebung. Weitere CVE Fixes unten:

- [CVE-2020-14559](#)
- [CVE-2020-14539](#)
- [CVE-2020-2579](#)
- [CVE-2020-2812](#)
- [CVE-2020-2780](#)
- [CVE-2020-2763](#)

Inkompatible Änderungen:

Diese Version führt eine Berechtigungsänderung ein, die sich auf das Verhalten des `mysqldump` Befehls auswirkt. Benutzer müssen das `PROCESS` Recht haben, auf die `INFORMATION_SCHEMA.FILES` Tabelle zuzugreifen. Um den `mysqldump` Befehl ohne Änderungen auszuführen, erteilen Sie dem Datenbankbenutzer, mit dem sich der `PROCESS` Befehl verbindet, die `mysqldump` Berechtigung. Sie können den `mysqldump` Befehl auch mit der `--no-tablespaces` Option ausführen. Mit dieser Option enthält die `mysqldump` Ausgabe keine `CREATE LOGFILE GROUP` oder `CREATE TABLESPACE` Anweisungen. In diesem Fall greift der `mysqldump` Befehl nicht auf die `INFORMATION_SCHEMA.FILES` Tabelle zu, und Sie müssen die `PROCESS` Berechtigung nicht erteilen.

Verbesserungen der Verfügbarkeit:

- Es wurden Probleme behoben, die zu Serverneustarts während der Wiederherstellung einer nicht festgeschriebenen DDL-Anweisung führen können.
- Die Race-Bedingungen im Sperrmanager, die einen Neustart des Servers verursachen können, wurden behoben.
- Es wurde ein Problem behoben, das dazu führen konnte, dass der Monitoring-Agent den Server während der Wiederherstellung einer großen Transaktion neu startet

Allgemeine Verbesserungen:

- Das Verhalten wurde geändert, indem `MIXED binlog_format` jetzt ROW und nicht STATEMENT zugeordnet wird, wenn `LOAD DATA FROM INFILE | S3` ausgeführt wird.
- Es wurde ein Problem behoben, bei dem ein mit einem Aurora MySQL Binlog-Primary verbundenes Binlog-Replikat unvollständige Daten anzeigen konnte, wenn der Primary `LOAD DATA FROM S3` ausführte und `binlog_format` auf STATEMENT festgelegt war.

## Integration von MySQL-Fehlerbehebungen (Community Edition):

- Fehler #26654685: Eine beschädigte Index-ID, die während der Überprüfung eines Fremdschlüssels aufgetreten ist, hat eine Behauptung ausgelöst
- Fehler #15831300: Beim Hochsetzen von Ganzzahlen von einem kleineren Typ auf dem Master auf einen größeren Typ auf dem Slave (z. B. von einer [SMALLINT-Spalte](#) auf dem Master zu einer [BIGINT-Spalte](#) auf dem Slave) werden die hochgesetzten Werte standardmäßig so behandelt, als wären sie signiert. In solchen Fällen ist es nun möglich, dieses Verhalten mit einem oder beiden von `ALL_SIGNED` und `ALL_UNSIGNED` aus einer Reihe von Werten, die in der für die Systemvariable [slave\\_type\\_conversions](#) angegebenen werden, zu ändern oder zu überschreiben. Weitere Informationen finden Sie unter [Zeilen-basierte Replikation: Attributförderung und -herabstufung](#) sowie in der Beschreibung der Variable.
- Fehler #17449901: Mit `foreign_key_checks=0` ermöglicht InnoDB das Löschen eines Index, der für eine Fremdschlüsseleinschränkung erforderlich ist, wodurch die Tabelle inkonsistent wurde und die beim Laden der Tabelle aufgetretene Fremdschlüsselprüfung fehlschlug. InnoDB verhindert jetzt, dass ein Index gelöscht wird, der für eine Fremdschlüsseleinschränkung erforderlich ist, selbst bei `foreign_key_checks=0`. Die Fremdschlüsseleinschränkung muss entfernt werden, bevor der Fremdschlüsselindex gelöscht wird.
- FEHLER #20768847: [ALTER TABLE ... Eine DROP INDEX](#)-Operation für eine Tabelle mit Fremdschlüsselabhängigkeiten hat eine Behauptung ausgelöst.

## Aurora-MySQL-Datenbank-Engine-Updates 05.03.2020 (Version 1.22.2) (veraltet)

Version: 1.22.2

Aurora MySQL 1.22.2 ist allgemein verfügbar. Aurora MySQL 1.\*-Versionen sind mit MySQL 5.6 kompatibel. Aurora MySQL 2.\*-Versionen sind mit MySQL 5.7 kompatibel.

Diese Engine-Version wird planmäßig am 28. Februar 2023 veralten. Weitere Informationen finden Sie unter [Vorbereitung auf das Lebenszyklusende der mit Amazon Aurora MySQL kompatiblen Edition Version 1](#).

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Um einen Cluster mit einer älteren Version von Aurora MySQL zu erstellen, geben Sie bitte die Engine-Version über die RDS-Konsole, die AWS CLI oder die Amazon RDS-API an.

#### Note

Diese Version ist derzeit in den folgenden Regionen nicht verfügbar: AWS GovCloud (US-Ost) [us-gov-east-1], AWS GovCloud (US-West) [us-gov-west-1]. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

Diese Version ist als Long-Term Support- (LTS, Langzeit-Support)-Version ausgewiesen. Weitere Informationen finden Sie unter [Aurora-MySQL-Long-Term-Support \(LTS, Langzeit-Support\)-Versionen](#) im Amazon-Aurora-Benutzerhandbuch.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Wichtige Fehlerbehebungen:

- Es wurde ein Problem von zeitweiligen Verbindungsfehlern nach der Zertifikatsrotation behoben.
- Es wurde ein Problem behoben, durch das das Klonen bei einigen Datenbank-Clustern mit hohen Schreiblasten länger dauert.
- Es wurde ein Problem behoben, durch das die logische Replikation unterbrochen wurde, wenn der Parameter `binlog_checksum` auf unterschiedliche Werte auf dem Master und dem Replikat festgelegt wurde.
- Es wurde ein Problem behoben, durch das langsames Protokoll und allgemeines Protokoll bei Lesereplikaten möglicherweise nicht ordnungsgemäß rotiert wurden.
- Es wurde ein Problem mit dem Verhalten von ANSI Read Committed Isolation Level behoben.

# Aurora-MySQL-Datenbank-Engine-Updates 23.12.2019 (Version 1.22.1) (veraltet)

Version: 1.22.1

Aurora MySQL 1.22.1 ist allgemein verfügbar. Aurora MySQL 1.\*-Versionen sind mit MySQL 5.6 kompatibel. Aurora MySQL 2.\*-Versionen sind mit MySQL 5.7 kompatibel.

Diese Engine-Version wird planmäßig am 28. Februar 2023 veralten. Weitere Informationen finden Sie unter [Vorbereitung auf das Lebenszyklusende der mit Amazon Aurora MySQL kompatiblen Edition Version 1](#).

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Um einen Cluster mit einer älteren Version von Aurora MySQL zu erstellen, geben Sie bitte die Engine-Version über die AWS Management Console, die AWS CLI oder die RDS-API an. Sie haben die Möglichkeit, bestehende Aurora MySQL 1.\* Datenbank-Cluster auf Aurora MySQL 1.22.1 upzugraden.

## Note

Diese Version ist derzeit in den folgenden AWS Regionen nicht verfügbar: AWS GovCloud (US-Ost) [us-gov-east-1], (US-West) [-1], China AWS GovCloud (Ningxia) [us-gov-westcn-northwest-1], Asien-Pazifik (Hongkong) [ap-east-1] und Naher Osten (Bahrain) [me-south-1]. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Note

Die Vorgehensweise zum Upgraden eines DB-Clusters hat sich geändert. Weitere Informationen finden Sie unter [Upgrade von Nebenversionen oder der Patch-Ebene eines Aurora-MYSQL-DB-Clusters](#) in der Amazon-Aurora-Benutzeranleitung.



## Verbesserungen

### Kritische Fehlerbehebungen:

- Es wurden Probleme behoben, durch die Modulwiederherstellung mit Tabellensperren und temporären Tabellen verhindert wurden.
- Die Stabilität des Binärprotokolls bei Verwendung temporärer Tabellen wurde verbessert.

### Wichtige Fehlerbehebungen:

- Es wurde ein langsames Speicherleck in bestimmten Aurora-Datenbankprotokollierungs- und Protokollierungs-Subsystem behoben, das den freizugebenden Speicher verringert.

## Aurora-MySQL-Datenbank-Engine-Updates 25.11.2019 (Version 1.22.0) (veraltet)

Version: 1.22.0

Aurora MySQL 1.22.0 ist allgemein verfügbar. Aurora MySQL 1.\*-Versionen sind mit MySQL 5.6 kompatibel. Aurora MySQL 2.\*-Versionen sind mit MySQL 5.7 kompatibel.

Diese Engine-Version wird planmäßig am 28. Februar 2023 veralten. Weitere Informationen finden Sie unter [Vorbereitung auf das Lebenszyklusende der mit Amazon Aurora MySQL kompatiblen Edition Version 1.](#)

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Um einen Cluster mit einer älteren Version von Aurora MySQL zu erstellen, geben Sie bitte die Engine-Version über die AWS Management Console, die AWS CLI oder die RDS-API an. Sie haben die Möglichkeit, bestehende Aurora MySQL 1.\* Datenbank-Cluster auf Aurora MySQL 1.22.0 aufzugraden.

### Note

Diese Version ist derzeit in den folgenden AWS Regionen nicht verfügbar: AWS GovCloud (US-Ost) [us-gov-east-1], AWS GovCloud (US-West) [us-gov-west-1], China (Ningxia) [cn-

northwest-1], Asien-Pazifik (Hongkong) [ap-east-1], Naher Osten (Bahrain) [me-south-1] und Südamerika (São Paulo) [sa-east-1]. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

### Note

Die Vorgehensweise zum Upgraden eines DB-Clusters hat sich geändert. Weitere Informationen finden Sie unter [Upgrade von Nebenversionen oder der Patch-Ebene eines Aurora-MYSQL-DB-Clusters](#) in der Amazon-Aurora-Benutzeranleitung.

## Verbesserungen

Neue Funktionen:

- Aurora MySQL-Cluster unterstützen jetzt die Instance-Typen r5.8xlarge, r5.16xlarge und r5.24xlarge.
- Binlog hat neue Erweiterung für bessere Commit Time-Latenz bei sehr großen Transaktionen.
- Aurora MySQL verfügt jetzt über einen Mechanismus für die Minimierung des Zeitfensters zum Schreiben der Events einer großen Transaktion beim Commitment zu Binlog. Dies verhindert effektiv längere Offline-Wiederherstellungen bei Datenbank abstürzen innerhalb diese Zeitfensters. Diese Funktion behebt auch ein Problem, bei dem eine große Transaktion kleine Transaktionen beim Binlog-Commitment blockiert. Diese Funktion ist standardmäßig deaktiviert und kann vom Serviceteam aktiviert werden, wenn dies für Ihr Workload erforderlich ist. Nach der Aktivierung wird sie bei Workloads > 500 MB ausgelöst.
- Unterstützung für ANSI-READ COMMITTED-Isolationsstufe auf den Read Replicas hinzugefügt. Diese Isolationsstufe ermöglicht das Ausführen von langandauernden Abfragen auf der Read Replica ohne Auswirkungen auf den hohen Durchsatz von Schreibvorgängen auf dem Schreiber-Knoten. Weitere Informationen finden Sie unter [Aurora MySQL-Isolationsstufen](#).
- Globale Datenbanken ermöglichen jetzt das Hinzufügen sekundärer schreibgeschützter Replikatregionen für Datenbankcluster, die in diesen AWS Regionen bereitgestellt werden:

Regionen: USA Ost (Nord-Virginia) [us-east-1], USA Ost (Ohio) [us-east-2], USA West (Nordkalifornien) [us-west-1], USA West (Oregon) [us-west-2], Europa (Irland) [eu-west-1], Europa (London) [eu-west-2], Europa (Paris) [eu-west-2] -3], Asien-Pazifik (Tokio) [ap-northeast-1], Asien-Pazifik (Seoul) [ap-northeast-2], Asien-Pazifik (Singapur) [ap-southeast-1], Asien-Pazifik (Sydney) [ap-southeast-2], Kanada (Zentral) [ca-central-1], Europa (Frankfurt) [ eu-central-1] und Asien-Pazifik (Mumbai) [ap-south-1].

- Die Hot Row Contention-Funktion ist jetzt allgemein verfügbar und erfordert nicht, dass die Aurora-Lab-Modus-Einstellung auf ON gesetzt wird. Diese Funktion sorgt für eine deutliche Verbesserung des Durchsatzes bei Workloads mit zahlreichen Transaktionen, die um Zeilen auf derselben Seite konkurrieren.
- Diese Version hat aktualisierte Zeitzonendateien zur Unterstützung der jüngsten brasilianischen Zeitzoneaktualisierung für neue Cluster.

#### Kritische Fehlerbehebungen:

- [CVE-2019-2922](#)
- [CVE-2019-2923](#)
- [CVE-2019-2924](#)
- [CVE-2019-2910](#)

#### Wichtige Fehlerbehebungen:

- [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](#)
- Behebung eines Problems in der DDL-Wiederherstellungskomponente, das zu einer verlängerten Datenbank-Ausfallzeit führte. Cluster, die nach dem Ausführen von TRUNCATE TABLE-DDL-

Anweisungen auf einer Tabelle mit einer AUTO\_INCREMENT-Spalte nicht mehr verfügbar werden, sollten auf diese Version aktualisiert werden.

- Behebung eines Problems in der DDL-Wiederherstellungskomponente, das zu einer verlängerten Datenbank-Ausfallzeit führte. Cluster, die nach dem Ausführen von DROP TABLE-DDL-Anweisungen auf mehreren Tabellen nicht mehr verfügbar werden, sollten aktualisiert werden.

Allgemeine Fehlerbehebungen zur Erhöhung der Stabilität:

- Behebung eines Problems, durch das Read Replicas während einer lang andauernden Transaktion neu gestartet wurden. Kunden, bei denen Replica-Neustarts auftreten, die mit einem beschleunigten Rückgang des freistellbaren Speichers einhergehen, sollten ein Upgrade zu dieser Version durchführen.
- Behebung eines Problems, bei dem inkorrektweise ERROR 1836 gemeldet wurde, wenn eine eingebettete Anfrage gegen eine temporäre Tabelle auf der Read Replica ausgeführt wurde.
- Behebung eines parallelen Abfrageabbruchfehlers auf einer Aurora-Reader-Instance, während ein hoher Workload mit Schreibvorgängen auf der Aurora-Writer-Instance ausgeführt wird.
- Behebung eines Problems, durch das eine als Binlog-Master konfigurierte Datenbank neu gestartet wurde, während ein umfangreicher Schreib-Workload stattfand.
- Behebung eines Problems mit längerer Nichtverfügbarkeit beim Neustart der Engine. Dies behebt ein Problem bei der Buffer-Pool-Initialisierung. Dieses Problem tritt selten auf, kann sich aber potenziell auf jede unterstützte Version auswirken.
- Behebung eines Problems, das inkonsistente Daten in der `information_schema.replica_host_status`-Tabelle generiert hat.
- Behebung einer Race-Condition zwischen der parallelen Abfrage und den Standard-Ausführungspfaden, das dazu führte, dass die Reader-Knoten abwechselnd neu gestartet wurden.
- Verbesserung der Stabilität der Datenbank, wenn die Anzahl der Clientverbindungen den `max_connections`-Parameterwert überschreitet.
- Verbesserung der Stabilität der Reader-Instances durch die Blockierung nicht unterstützter DDL- und LOAD FROM S3-Abfragen.

Integration von MySQL-Fehlerbehebungen (Community Edition):

- Bug#16346241 - SERVER CRASH IN ITEM\_PARAM::QUERY\_VAL\_STR
- Bug#17733850 - NAME\_CONST() CRASH IN ITEM\_NAME\_CONST::ITEM\_NAME\_CONST()

- Bug #20989615 - INNODB AUTO\_INCREMENT PRODUCES SAME VALUE TWICE
- Bug #20181776 - ACCESS CONTROL DOESN'T MATCH MOST SPECIFIC HOST WHEN IT CONTAINS WILDCARD
- Bug #27326796 - MYSQL CRASH WITH INNODB ASSERTION FAILURE IN FILE PARSOPARS.CC
- Bug #20590013 - IF YOU HAVE A FULLTEXT INDEX AND DROP IT YOU CAN NO LONGER PERFORM ONLINE DDL

## Aurora-MySQL-Datenbank-Engine-Updates 25.11.2019 (Version 1.21.0) (veraltet)

Version: 1.21.0

Aurora MySQL 1.21.0 ist allgemein verfügbar. Aurora MySQL 1.\*-Versionen sind mit MySQL 5.6 kompatibel. Aurora MySQL 2.\*-Versionen sind mit MySQL 5.7 kompatibel.

Zurzeit werden die Aurora MySQL-Versionen 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.\* und 2.07.\* unterstützt. Um einen Cluster mit einer älteren Version von Aurora MySQL zu erstellen, geben Sie bitte die Engine-Version über die AWS Management Console, die AWS CLI oder die RDS-API an. Sie haben die Möglichkeit, bestehende Aurora MySQL 1.\* Datenbank-Cluster auf Aurora MySQL 1.21.0 upzugraden.

### Note

Diese Version ist derzeit in den folgenden AWS Regionen nicht verfügbar: AWS GovCloud (US-Ost) [us-gov-east-1], AWS GovCloud (US-West) [us-gov-west-1], China (Ningxia) [cn-northwest-1], Asien-Pazifik (Hongkong) [ap-east-1], Europa (Stockholm) [eu-north-1] und Naher Osten (Bahrain) [me-south-1]. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

**Note**

Die Vorgehensweise zum Upgraden eines DB-Clusters hat sich geändert. Weitere Informationen finden Sie unter [Upgrade von Nebenversionen oder der Patch-Ebene eines Aurora-MYSQL-DB-Clusters](#) in der Amazon-Aurora-Benutzeranleitung.

## Verbesserungen

### Kritische Fehlerbehebungen:

- [CVE-2018-0734](#)
- [CVE-2019-2534](#)
- [CVE-2018-2612](#)
- [CVE-2017-3599](#)
- [CVE-2018-2562](#)
- [CVE-2017-3329](#)
- [CVE-2018-2696](#)
- [CVE-2015-4737](#)

### Wichtige Fehlerbehebungen:

- Kunden mit einer Datenbankgröße von fast 64 Tebibytes (TiB) wird dringend empfohlen, auf diese Version zu aktualisieren, um Ausfallzeiten aufgrund von Stabilitätsfehlern zu vermeiden, die Volumes in der Nähe des Aurora Speicherlimits betreffen.

### Allgemeine Fehlerbehebungen zur Erhöhung der Stabilität:

- Behebung eines parallelen Abfrageabbruchfehlers auf Aurora-Reader-Instances, während ein hoher Workload mit Schreibvorgängen auf der Aurora-Writer-Instance ausgeführt wird.
- Behebung eines Fehlers auf Aurora-Reader-Instances, der den freien Arbeitsspeicher bei lang andauernden Transaktionen reduzierte, während ein hoher Transaktions-Commit-Datenverkehr auf der Writer-Instance stattfindet.
- Der Wert des Parameters `aurora_disable_hash_join` ist jetzt nach einem Neustart der Datenbank oder einem Host-Ersatz persistent.

- Behebung eines Fehlers in Verbindung mit dem Cache für die Volltextsuche, aufgrund dessen der Aurora-Instance nicht mehr genügend Arbeitsspeicher zur Verfügung stand. Kunden, die die Volltextsuche nutzen, sollten ein Upgrade durchführen.
- Verbesserte Stabilität der Datenbanken, wenn die Hash-Join-Funktion aktiviert ist und der Instance wenig Speicherplatz zur Verfügung steht. Kunden, die Hash Join nutzen, sollten ein Upgrade durchführen.
- Behebung eines Problems im Abfrage-Cache, in dem der Fehler "Zu viele Verbindungen" zu einem Neustart führen konnte.
- Korrektur der Berechnung des freien Speichers auf T2-Instances mit Einbeziehung des Auslagerungsspeicherplatzes, um unnötige Neustarts zu vermeiden.

## Integration von MySQL-Fehlerbehebungen (Community Edition):

- Fehler #19929406: `HANDLE_FATAL_SIGNAL (SIG=11) IN __MEMMOVE_SSSE3_BACK FROM STRING::COPY`
- Fehler #17059925: Für [UNION](#)-Anweisungen wurde der zeilengeprüfte Wert falsch berechnet. Dies zeigte sich in Gestalt zu großer Werte für die Spalte `ROWS_EXAMINED` von Leistungsschema-Anweisungstabellen (wie z. B. [events\\_statements\\_current](#)).
- Fehler #11827369: Einige Abfragen mit verschachtelten `SELECT . . . FROM DUAL`-Unterabfragen erhöhten eine Assertion.
- Fehler #16311231: Falsche Ergebnisse wurden zurückgegeben, wenn eine Abfrage eine Unterabfrage in einer `IN`-Klausel enthielt, die eine [XOR](#)-Operation in der `WHERE`-Klausel enthielt.

## Aurora-MySQL-Datenbank-Engine-Updates 05.03.2020 (Version 1.20.1) (veraltet)

Version: 1.20.1

Aurora MySQL 1.20.1 ist allgemein verfügbar. Aurora MySQL 1.\*-Versionen sind mit MySQL 5.6 kompatibel. Aurora MySQL 2.\*-Versionen sind mit MySQL 5.7 kompatibel.

Zurzeit werden die Aurora MySQL-Versionen 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.\* und 2.07.\* unterstützt. Sie können den Snapshot einer Aurora MySQL 1.\*-Datenbank in Aurora MySQL 1.20.1 wiederherstellen.



Um einen Cluster mit einer älteren Version von Aurora MySQL zu erstellen, geben Sie bitte die Engine-Version über die RDS-Konsole, die AWS CLI oder die Amazon RDS-API an.

### Note

Diese Version ist derzeit in den folgenden Regionen nicht verfügbar: AWS GovCloud (US-Ost) [us-gov-east-1], AWS GovCloud (US-West) [us-gov-west-1]. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Wichtige Fehlerbehebungen:

- Es wurde ein Problem von zeitweiligen Verbindungsfehlern nach der Zertifikatsrotation behoben.
- Es wurde ein Problem im Zusammenhang mit der Parallelität beim Schließen von Verbindungen behoben, das zu einem Failover bei hohem Workload führen würde.

Allgemeine Fehlerbehebungen zur Erhöhung der Stabilität:

- Ein Absturz während der Ausführung einer komplexen Abfrage mit Joins und Aggregation mit mehreren Tabellen, die intern Zwischentabellen verwendet, wurde behoben.

## Aurora-MySQL-Datenbank-Engine-Updates 11.11.2019 (Version 1.20.0) (veraltet)

Version: 1.20.0

Aurora MySQL 1.20.0 ist allgemein verfügbar. Aurora MySQL 1.\*-Versionen sind mit MySQL 5.6 kompatibel. Aurora MySQL 2.\*-Versionen sind mit MySQL 5.7 kompatibel.

Derzeit unterstützte Aurora MySQL-Versionen: 1.14.\*, 1.15.\*, 1.16.\*, 1.17.\*, 1.18.\*, 1.19.\*, 1.20.\*, 2.01.\*, 2.02.\*, 2.03.\* und 2.04.\*. Um einen Cluster mit einer älteren Version von Aurora MySQL zu

erstellen, geben Sie bitte die Engine-Version über die AWS Management Console, die AWS CLI oder die RDS-API an. Sie haben auch die Möglichkeit, bestehende Aurora MySQL 1.\* Datenbank-Cluster auf 1.19.5 bis Aurora MySQL 1.20.0 upzugraden.

#### Note

Diese Version ist derzeit in den folgenden AWS Regionen nicht verfügbar: AWS GovCloud (US-Ost) [us-gov-east-1], AWS GovCloud (US-West) [us-gov-west-1], China (Ningxia) [cn-northwest-1], Asien-Pazifik (Hongkong) [ap-east-1], Europa (Stockholm) [eu-north-1] und Naher Osten (Bahrain) [me-south-1]. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

#### Note

Die Vorgehensweise zum Upgraden eines DB-Clusters hat sich geändert. Weitere Informationen finden Sie unter [Upgrade von Nebenversionen oder der Patch-Ebene eines Aurora-MYSQL-DB-Clusters](#) in der Amazon-Aurora-Benutzeranleitung.

## Verbesserungen

Kritische Fehlerbehebungen:

- [CVE-2018-0734](#)
- [CVE-2019-2534](#)
- [CVE-2018-2612](#)
- [CVE-2017-3599](#)
- [CVE-2018-2562](#)
- [CVE-2017-3329](#)
- [CVE-2018-2696](#)
- [CVE-2015-4737](#)

## Wichtige Fehlerbehebungen:

- Kunden mit einer Datenbankgröße von fast 64 Tebibytes (TiB) wird dringend empfohlen, auf diese Version zu aktualisieren, um Ausfallzeiten aufgrund von Stabilitätsfehlern zu vermeiden, die Volumes in der Nähe des Aurora Speicherlimits betreffen.

## Allgemeine Fehlerbehebungen zur Erhöhung der Stabilität:

- Behebung eines parallelen Abfrageabbruchfehlers auf Aurora-Reader-Instances, während ein hoher Workload mit Schreibvorgängen auf der Aurora-Writer-Instance ausgeführt wird.
- Behebung eines Fehlers auf Aurora-Reader-Instances, der den freien Arbeitsspeicher bei lang andauernden Transaktionen reduzierte, während ein hoher Transaktions-Commit-Datenverkehr auf der Writer-Instance stattfindet.
- Der Wert des Parameters `aurora_disable_hash_join` ist jetzt nach einem Neustart der Datenbank oder einem Host-Ersatz persistent.
- Behebung eines Fehlers in Verbindung mit dem Cache für die Volltextsuche, aufgrund dessen der Aurora-Instance nicht mehr genügend Arbeitsspeicher zur Verfügung stand. Kunden, die die Volltextsuche nutzen, sollten ein Upgrade durchführen.
- Verbesserte Stabilität der Datenbanken, wenn die Hash-Join-Funktion aktiviert ist und der Instance wenig Speicherplatz zur Verfügung steht. Kunden, die Hash Join nutzen, sollten ein Upgrade durchführen.
- Behebung eines Problems im Abfrage-Cache, in dem der Fehler "Zu viele Verbindungen" zu einem Neustart führen konnte.
- Korrektur der Berechnung des freien Speichers auf T2-Instances mit Einbeziehung des Auslagerungsspeicherplatzes, um unnötige Neustarts zu vermeiden.

## Integration von MySQL-Fehlerbehebungen (Community Edition):

- Fehler #19929406: `HANDLE_FATAL_SIGNAL (SIG=11) IN __MEMMOVE_SSSE3_BACK FROM STRING::COPY`
- Fehler #17059925: Für [UNION](#)-Anweisungen wurde der zeilengeprüfte Wert falsch berechnet. Dies zeigte sich in Gestalt zu großer Werte für die Spalte `ROWS_EXAMINED` von Leistungsschema-Anweisungstabellen (wie z. B. [events\\_statements\\_current](#)).
- Fehler #11827369: Einige Abfragen mit verschachtelten `SELECT ... FROM DUAL`-Unterabfragen erhöhten eine Assertion.

- Fehler #16311231: Falsche Ergebnisse wurden zurückgegeben, wenn eine Abfrage eine Unterabfrage in einer IN-Klausel enthielt, die eine [XOR](#)-Operation in der WHERE-Klausel enthielt.

## Aurora-MySQL-Datenbank-Engine-Updates 05.03.2020 (Version 1.19.6) (veraltet)

Version: 1.19.6

Aurora MySQL 1.19.6 ist allgemein verfügbar. Aurora MySQL 1.\*-Versionen sind mit MySQL 5.6 kompatibel. Aurora MySQL 2.\*-Versionen sind mit MySQL 5.7 kompatibel.

Diese Engine-Version wird planmäßig am 28. Februar 2023 veralten. Weitere Informationen finden Sie unter [Vorbereitung auf das Lebenszyklusende der mit Amazon Aurora MySQL kompatiblen Edition Version 1](#).

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Sie können den Snapshot einer Aurora MySQL 1.\*-Datenbank in Aurora MySQL 1.19.6 wiederherstellen.

Um einen Cluster mit einer älteren Version von Aurora MySQL zu erstellen, geben Sie bitte die Engine-Version über die RDS-Konsole, die AWS CLI oder die Amazon RDS-API an.

### Note

Diese Version ist derzeit in den folgenden Regionen nicht verfügbar: AWS GovCloud (US-Ost) [us-gov-east-1], AWS GovCloud (US-West) [us-gov-west-1]. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Wichtige Fehlerbehebungen:

- Es wurde ein Problem von zeitweiligen Verbindungsfehlern nach der Zertifikatsrotation behoben.

## Aurora-MySQL-Datenbank-Engine-Updates 19.09.2019 (Version 1.19.5) (veraltet)

Version: 1.19.5

Aurora MySQL 1.19.5 ist allgemein verfügbar. Aurora MySQL 1.\*-Versionen sind mit MySQL 5.6 kompatibel. Aurora MySQL 2.\*-Versionen sind mit MySQL 5.7 kompatibel.

Diese Engine-Version wird planmäßig am 28. Februar 2023 veralten. Weitere Informationen finden Sie unter [Vorbereitung auf das Lebenszyklusende der mit Amazon Aurora MySQL kompatiblen Edition Version 1](#).

Derzeit werden die Aurora-MySQL-Versionen 1.19.5, 1.19.6, 1.22.\*, 1.23.\*, 2.04.\*, 2.07.\*, 2.08.\*, 2.09.\*, 2.10.\*, 3.01.\* und 3.02.\* unterstützt.

Sie haben die Möglichkeit, bestehende Datenbank-Cluster auf Aurora MySQL 1.19.5 upzugraden. Sie können Snapshots von Aurora MySQL 1.14.\*, 1.15.\*, 1.16.\*, 1.17.\*, 1.18.\*, 1.19.1. und 1.19.2 in Aurora MySQL 1.19.5 wiederherstellen.

Um eine ältere Version von Aurora MySQL zu verwenden, können Sie neue Datenbankcluster erstellen, indem Sie die Engine-Version über die AWS Management Console AWS CLI, die oder die RDS-API angeben.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

### Note

Diese Version ist derzeit in den folgenden AWS Regionen nicht verfügbar: Europa (London) [eu-west-2], AWS GovCloud (US-Ost) [us-gov-east-1], (US-West) [-1], China AWS GovCloud (Ningxia) [us-gov-westcn-northwest-1] und Asien-Pazifik (Hongkong) [ap-east-1]. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

**Note**

Die Vorgehensweise zum Upgraden eines DB-Clusters hat sich geändert. Weitere Informationen finden Sie unter [Upgrade von Nebenversionen oder der Patch-Ebene eines Aurora-MYSQL-DB-Clusters](#) in der Amazon-Aurora-Benutzeranleitung.

## Verbesserungen

- Behebung eines Fehlers auf Aurora-Reader-Instances, der den freien Arbeitsspeicher bei lang andauernden Transaktionen reduzierte, während ein hoher Transaktions-Commit-Datenverkehr auf der Writer-Instance stattfindet.
- Behebung eines parallelen Abfrageabbruchfehlers auf Aurora-Reader-Instances, während ein hoher Workload mit Schreibvorgängen auf der Aurora-Writer-Instance ausgeführt wird.
- Der Wert des Parameters `aurora_disable_hash_join` ist jetzt nach einem Neustart der Datenbank oder einem Host-Ersatz persistent.
- Behebung eines Fehlers in Verbindung mit dem Cache für die Volltextsuche, aufgrund dessen der Aurora-Instance nicht mehr genügend Arbeitsspeicher zur Verfügung stand.
- Verbesserte Stabilität der Datenbanken, wenn die Volume-Größe nahe am 64-Tebibyte(TiB)-Volume-Limit liegt, indem 160 GB an Speicher für den Wiederherstellungs-Workflow reserviert wird, damit er ohne Failover abgeschlossen wird.
- Verbesserte Stabilität der Datenbanken, wenn die Hash-Join-Funktion aktiviert ist und der Instance wenig Speicherplatz zur Verfügung steht.
- Korrektur der freien Speicherberechnung zum Einbeziehen des Auslagerungsspeicherplatzes auf T2-Instances, wodurch sie vorzeitig neu gestartet wurden.
- Behebung eines Problems im Abfrage-Cache, in dem der Fehler "Zu viele Verbindungen" zu einem Neustart führen konnte.

## Integration von MySQL-Fehlerbehebungen (Community Edition):

- [CVE-2018-2696](#)
- [CVE-2015-4737](#)
- Fehler #19929406: `HANDLE_FATAL_SIGNAL (SIG=11) IN __MEMMOVE_SSE3_BACK FROM STRING::COPY`

- Fehler #17059925: Für [UNION](#)-Anweisungen wurde der zeilengeprüfte Wert falsch berechnet. Dies zeigte sich in Gestalt zu großer Werte für die Spalte ROWS\_EXAMINED von Leistungsschema-Anweisungstabellen (wie z. B. [events\\_statements\\_current](#)).
- Fehler #11827369: Einige Abfragen mit verschachtelten SELECT ... FROM DUAL-Unterabfragen erhöhten eine Assertion.
- Fehler #16311231: Falsche Ergebnisse wurden zurückgegeben, wenn eine Abfrage eine Unterabfrage in einer IN-Klausel enthielt, die eine [XOR](#)-Operation in der WHERE-Klausel enthielt.

## Aurora-MySQL-Datenbank-Engine-Updates 05.06.2019 (Version 1.19.2) (veraltet)

Version: 1.19.2

Aurora MySQL 1.19.2 ist allgemein verfügbar. Alle neuen mit MySQL 5.6 kompatiblen Aurora MySQL-Datenbank-Cluster einschließlich aus Snapshots wiederhergestellter Datenbank-Cluster können mit 1.17.8, 1.19.0, 1.19.1 oder 1.19.2 erstellt werden. Sie können optional für vorhandene Datenbank-Cluster ein Upgrade auf Aurora MySQL 1.19.2 ausführen, müssen dies jedoch nicht. Um eine ältere Version zu verwenden, erstellen Sie neue Datenbank-Cluster in Aurora MySQL 1.14.4, Aurora MySQL 1.15.1, Aurora MySQL 1.16, Aurora MySQL 1.17.8 oder Aurora MySQL 1.18. Sie können dazu die AWS CLI oder die Amazon RDS-API verwenden und die Engine-Version angeben.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

### Note

Diese Version ist derzeit in den Regionen AWS GovCloud (US-West) [us-gov-west-1], Europa (Stockholm) [eu-north-1], China (Ningxia) [cn-northwest-1] und Asien-Pazifik (Hongkong) [ap-east-1] nicht verfügbar. AWS Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

**Note**

Die Vorgehensweise zum Upgraden eines DB-Clusters hat sich geändert. Weitere Informationen finden Sie unter [Upgrade von Nebenversionen oder der Patch-Ebene eines Aurora-MYSQL-DB-Clusters](#) in der Amazon-Aurora-Benutzeranleitung.

## Verbesserungen

- Beseitigung eines Problems, das beim Laden von Daten aus Amazon S3 in Aurora zu Fehlern führen könnte.
- Beseitigung eines Problems, das beim Hochladen von Daten aus Aurora zu Amazon S3 zu Fehlern führen könnte.
- Beseitigung eines Fehlers, das zur Erstellung von Zombie-Sitzungen in einem beendeten Zustand führte.
- Beseitigung eines Fehlers, der beim Behandeln eines Fehlers in der Netzwerkprotokollverwaltung zum Abbruch von Verbindungen führte.
- Beseitigung eines Problems, das beim Arbeiten mit partitionierten Tabellen zu einem Absturz führen konnte.
- Beseitigung eines Problems im Zusammenhang mit der binlong-Replikation der Auslösererstellung.

## Aurora-MySQL-Datenbank-Engine-Updates 09.05.2019 (Version 1.19.1) (veraltet)

Version: 1.19.1

Aurora MySQL 1.19.1 ist allgemein verfügbar. Alle neuen mit MySQL 5.6 kompatiblen Aurora MySQL-Datenbank-Clustern, einschließlich der aus Snapshots wiederhergestellten, können mit 1.17.8, 1.19.0 oder 1.19.1 wiederhergestellt werden. Für bestehende Datenbank-Cluster können Sie ein Upgrade auf Aurora MySQL 1.19.1 ausführen (nicht verpflichtend). Um eine ältere Version zu verwenden, erstellen Sie neue Datenbank-Cluster in Aurora MySQL 1.14.4, Aurora MySQL 1.15.1, Aurora MySQL 1.16, Aurora MySQL 1.17.8 oder Aurora MySQL 1.18. Sie können dazu die AWS CLI oder die Amazon RDS-API verwenden und die Engine-Version angeben.



Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

#### Note

Diese Version ist derzeit in den Regionen AWS GovCloud (US-West) [us-gov-west-1] und China (Peking) [cn-north-1] nicht verfügbar. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

#### Note

Die Vorgehensweise zum Upgraden eines DB-Clusters hat sich geändert. Weitere Informationen finden Sie unter [Upgrade von Nebenversionen oder der Patch-Ebene eines Aurora-MYSQL-DB-Clusters](#) in der Amazon-Aurora-Benutzeranleitung.

## Verbesserungen

- Es wurde ein Fehler in der Binärprotokoll-Replikation behoben, der ein Problem auf Aurora-Instances, die als Binärprotokoll-Worker konfiguriert sind, verursachen kann.
- Ein Fehler bei der Verarbeitung bestimmter ALTER TABLE-Befehle wurde behoben.
- Ein Fehler in Form von Verbindungsabbrüchen aufgrund eines Problems in der Netzwerkprotokollverwaltung wurde behoben.

## Aurora-MySQL-Datenbank-Engine-Updates 07.02.2019 (Version 1.19.0) (veraltet)

Version: 1.19.0

Aurora MySQL 1.19.0 ist allgemein verfügbar. Alle neuen mit MySQL 5.6 kompatiblen Aurora MySQL-Datenbank-Cluster, einschließlich der aus Snapshots wiederhergestellten, können mit 1.17.8 oder 1.19.0 wiederhergestellt werden. Für bestehende Datenbank-Cluster können Sie ein Upgrade auf Aurora MySQL 1.19.0 ausführen (nicht verpflichtend). Zur Verwendung einer älteren Version erstellen Sie neue Datenbank-Cluster in Aurora MySQL 1.14.4, Aurora MySQL 1.15.1, Aurora MySQL

1.16, Aurora MySQL 1.17.8 oder Aurora MySQL 1.18.0. Sie können dazu die AWS CLI oder die Amazon RDS-API verwenden und die Engine-Version angeben.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

#### Note

Diese Version ist derzeit in den Regionen AWS GovCloud (US-West) [us-gov-west-1] und China (Peking) [cn-north-1] nicht verfügbar. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

#### Note

Die Vorgehensweise zum Upgraden eines DB-Clusters hat sich geändert. Weitere Informationen finden Sie unter [Upgrade von Nebenversionen oder der Patch-Ebene eines Aurora-MYSQL-DB-Clusters](#) in der Amazon-Aurora-Benutzeranleitung.

## Features

- Aurora-Versionsauswahl – Ab Aurora MySQL 1.19.0 können Sie in der Amazon RDS-Konsole zwischen mehreren MySQL 5.6-kompatiblen Aurora-Versionen auswählen. Weitere Informationen finden Sie unter [Aurora-MYSQL-Engine-Versionen überprüfen oder angeben AWS](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

- Es wurde ein Stabilitätsproblem im Zusammenhang mit der CHECK TABLE-Abfrage auf einer Aurora Replica behoben.
- Es wurde eine neue globale `aurora_disable_hash_join`-Benutzervariable zum Deaktivieren von Hash-Join eingeführt.
- Es wurde ein Stabilitätsproblem beim Erzeugen der Ausgabezeile beim Hash-Join mehrerer Tabellen erzeugt.

- Es wurde ein Problem behoben, bei dem aufgrund einer Planänderung während der Hash-Join-Anwendbarkeitsprüfung ein falsches Ergebnis zurückgegeben wurde.
- Bei lang andauernden Transaktionen wird Zero-Downtime-Patching (Patches ohne Ausfallzeiten) unterstützt. Diese Verbesserung tritt beim Aktualisieren von Version 1.19 auf eine höhere Version in Kraft.
- Zero-Downtime-Patching (Patches ohne Ausfallzeiten) wird nun unterstützt, wenn binlog aktiviert ist. Diese Verbesserung tritt beim Aktualisieren von Version 1.19 auf eine höhere Version in Kraft.
- Es wurde ein Problem behoben, durch das die CPU-Auslastung auf einer Aurora-Replica unabhängig von der Workload erhöht wurde.
- Es wurde eine race-Bedingung im Lock-Manager behoben, die zu einem Neustart der Datenbank führte.
- Es wurde eine race-Bedingung in der Lock-Manager-Komponente behoben, um die Stabilität von Aurora-Instances zu verbessern.
- Die Stabilität des Deadlock Detectors innerhalb der Lock Manager-Komponente wurde verbessert.
- INSERTDie Operation ist für eine Tabelle verboten, wenn InnoDB erkennt, dass der Index beschädigt wurde.
- Es wurde ein Stabilitätsproblem in Fast DDL behoben.
- Die Stabilität von Aurora wurde durch Reduzieren des Speicherverbrauchs beim Scan-Batching für einzeilige Unterabfragen verbessert.
- Es wurde ein Stabilitätsproblem behoben, das auftrat, nachdem ein Fremdschlüssel entfernt wurde, während die Systemvariable `foreign_key_checks` auf "0" eingestellt war.
- Es wurde ein Problem bei der Out-Of-Memory-Vermeidungsfunktion behoben, bei dem die vom Benutzer am `table_definition_cache`-Wert vorgenommenen Änderungen fälschlicherweise überschrieben wurden.
- Es wurden Stabilitätsprobleme bei der Out-Of-Memory-Vermeidungsfunktion behoben.
- Es wurde ein Problem behoben, bei dem `query_time` und `lock_time` in `slow_query_log` auf unzulässige Werte eingestellt wurden.
- Es wurde ein Stabilitätsproblem mit Parallelabfragen behoben, das durch die interne unsachgemäße Bearbeitung der Zeichenfolgensortierung ausgelöst wurde.
- Es wurde ein Stabilitätsproblem mit Parallelabfragen behoben, das durch eine Sekundärindexsuche ausgelöst wurde.
- Es wurde ein Stabilitätsproblem mit Parallelabfragen behoben, das durch die Aktualisierung mehrerer Tabellen ausgelöst wurde.

## Integration von MySQL-Fehlerbehebungen (Community Edition):

- BUG #32917: DETECT ORPHAN TEMP-POOL FILES, AND HANDLE GRACEFULLY
- BUG #63144 CREATE TABLE IF NOT EXISTS METADATA LOCK IS TOO RESTRICTIVE

## Aurora-MySQL-Datenbank-Engine-Updates 20.09.2018 (Version 1.18.0) (veraltet)

Version: 1.18.0

Aurora MySQL 1.18.0 ist allgemein verfügbar. Alle neuen mit MySQL 5.6 kompatiblen parallelen Aurora MySQL-Abfrage-Cluster, einschließlich der aus Snapshots wiederhergestellten, werden in Aurora MySQL 1.18.0 erstellt. Für bestehende parallele Abfrage-Cluster können Sie ein Upgrade auf Aurora MySQL 1.18.0 ausführen (nicht verpflichtend). Sie können neue DB-Cluster in Aurora MySQL 1.14.4, Aurora MySQL 1.15.1, Aurora MySQL 1.16 oder Aurora MySQL 1.17.6 erstellen. Sie können dazu die AWS CLI oder die Amazon RDS-API verwenden und die Engine-Version angeben.

Mit Aurora MySQL-Version 1.18.0 verwenden wir ein Cluster-Patching-Modell, bei dem alle Knoten in einem Aurora-DB-Cluster gleichzeitig gepatcht werden.

### Important

Aurora MySQL 1.18.0 gilt nur für parallele Aurora-Abfrage-Cluster. Wenn Sie einen bereitgestelltes 5.6.10a-Cluster aktualisieren, ergibt dies Version 1.17.8. Wenn Sie einen parallelen 5.6.10a-Abfrage-Cluster aktualisieren, ergibt dies Version 1.18.0.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Features

- Parallel Query ist mit dieser Version für neue Cluster und wiederhergestellte Schnappschüsse verfügbar. Aurora MySQL Parallel Query ist eine Optimierung, die bei der Verarbeitung datenintensiver Abfragen einen Teil der I/O-Operationen und Berechnungen parallelisiert. Die parallelisierten Vorgänge beinhalten das Abrufen von Zeilen aus dem Speicher, das Extrahieren

von Spaltenwerten und das Bestimmen der Zeilen, die den Bedingungen der WHERE-Klausel und der JOIN-Klauseln entsprechen. Diese datenintensive Arbeit wird (datenbankoptimiert, herabgestuft) an mehrere Knoten in der verteilten Speicherebene von Aurora delegiert. Ohne eine Parallelabfrage leitet jede Abfrage die gescannten Daten zu einem einzigen Knoten innerhalb des Aurora MySQL-Clusters (Hauptknoten) und die Verarbeitung jeglicher Abfragen erfolgt dort.

- Ist Parallel Query aktiviert, entscheidet die Aurora MySQL-Engine automatisch, wann Abfragen profitieren können. An SQL müssen dafür keine Änderungen (z. B. Hinweise oder Tabellenattribute) vorgenommen werden.

Weitere Informationen finden Sie im Abschnitt zum [Arbeiten mit parallelen Abfragen für Amazon Aurora MySQL](#) im Amazon-Aurora-Benutzerhandbuch.

- OOM-Vermeidung: Diese Funktion überwacht den Systemspeicher und verfolgt den von verschiedenen Datenbankkomponenten genutzten Speicher. Sobald das System nicht über ausreichend Arbeitsspeicher verfügt, wird zum Vermeiden von Out of Memory (OOM) und eines Datenbankneustarts eine Reihe an Aktionen ausgeführt, um Arbeitsspeicher von verschiedenen verfolgten Komponenten freizugeben. Diese Best-Effort-Funktion ist für t2-Instances standardmäßig aktiviert und kann mithilfe des neuen Instance-Parameters auch für andere Instance-Klassen aktiviert werden `aurora_oom_response`. Der Instance-Parameter führt eine Zeichenfolge mit durch Kommata getrennten Aktionen aus, die eine Instance durchführen muss, wenn nur wenig Arbeitsspeicher zur Verfügung steht. Gültige Aktionen sind „print“, „tune“, „decline“, „kill\_query“ und jegliche Kombinationen dieser. Jede leere Zeichenfolge bedeutet, dass keine Aktionen durchgeführt werden dürfen und die Funktion wird deaktiviert. Die Standardaktion für die Funktion lautet „print, tune“. Verwendungsbeispiele:
  - „print“ – druckt nur die Abfragen mit hohem Speicherbedarf.
  - „tune“ – stellt die Caches der internen Tabellen so ein, dass etwas Arbeitsspeicher für das System freigegeben wird.
  - „decline“ – lehnt neue Abfragen ab, sobald der Instance wenig Speicherplatz zur Verfügung steht.
  - „kill\_query“ – beendet die Abfragen in absteigender Reihenfolge des Speicherverbrauchs, bis der Instance-Speicher oberhalb des unteren Schwellenwerts liegt. Data Definition Language (DDL)-Anweisungen werden nicht beendet.
  - „print, tune“ – führt die beschriebenen Aktionen sowohl für „print“ als auch für „tune“ aus.
  - „tune, decline, kill\_query“ – führt die beschriebenen Aktionen für „tune“, „decline“ und „kill\_query“ aus.

Informationen zu den out-of-memory Bearbeitungsbedingungen und weitere Hinweise zur Fehlerbehebung finden Sie unter [Probleme mit Amazon Aurora MySQL out of memory](#) im Amazon Aurora Benutzerhandbuch.

## Aurora-MySQL-Datenbank-Engine-Updates 05.03.2020 (Version 1.17.9) (veraltet)

Version: 1.17.9

Aurora MySQL 1.17.9 ist allgemein verfügbar. Aurora MySQL 1.\*-Versionen sind mit MySQL 5.6 kompatibel. Aurora MySQL 2.\*-Versionen sind mit MySQL 5.7 kompatibel.

Zurzeit werden die Aurora MySQL-Versionen 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.\* und 2.07.\* unterstützt. Sie können den Snapshot einer Aurora MySQL 1.\*-Datenbank in Aurora MySQL 1.17.9 wiederherstellen.

Um einen Cluster mit einer älteren Version von Aurora MySQL zu erstellen, geben Sie bitte die Engine-Version über die RDS-Konsole, die AWS CLI oder die Amazon RDS-API an.

### Note

Diese Version ist derzeit in den folgenden Regionen nicht verfügbar: AWS GovCloud (US-Ost) [us-gov-east-1], AWS GovCloud (US-West) [us-gov-west-1]. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

Wichtige Fehlerbehebungen:

- Es wurde ein Problem von zeitweiligen Verbindungsfehlern nach der Zertifikatsrotation behoben.

# Aurora-MySQL-Datenbank-Engine-Updates 17.01.2019 (Version 1.17.8) (veraltet)

Version: 1.17.8

Aurora MySQL 1.17.8 ist allgemein verfügbar. Alle neuen mit MySQL 5.6 kompatiblen Aurora MySQL-Datenbank-Cluster, einschließlich der aus Snapshots wiederhergestellten, werden in Aurora MySQL 1.17.8 erstellt. Für bestehende Datenbank-Cluster können Sie ein Upgrade auf Aurora MySQL 1.17.8 ausführen (nicht verpflichtend). Zur Verwendung einer älteren Version erstellen Sie einen neuen Datenbank-Cluster in Aurora MySQL 1.14.4, 1.15.1, 1.16 oder 1.17.7. Sie können dazu die AWS CLI oder die Amazon RDS-API verwenden und die Engine-Version angeben.

Mit Aurora MySQL-Version 1.17.8 verwenden wir ein Cluster-Patching-Modell, bei dem alle Knoten in einem Aurora-DB-Cluster gleichzeitig gepatcht werden.

## Note

Diese Version ist derzeit in den Regionen AWS GovCloud (US-West) [us-gov-west-1] und China (Peking) [cn-north-1] nicht verfügbar. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

- Ein Leistungsproblem wurde behoben, durch das die CPU-Auslastung auf einer Aurora-Replica nach einem Neustart erhöht wurde.
- Ein Stabilitätsproblem für SELECT-Abfragen mit Hash-Join wurde behoben.

## Integration von MySQL-Fehlerbehebungen (Community Edition):

- Fehler 13418638: CREATE TABLE IF NOT EXISTS METADATA LOCK IS TOO RESTRICTIVE

# Aurora-MySQL-Datenbank-Engine-Updates 08.10.2018 (Version 1.17.7) (veraltet)

Version: 1.17.7

Aurora MySQL 1.17.7 ist allgemein verfügbar. Alle neuen mit MySQL 5.6 kompatiblen Aurora MySQL-Datenbank-Cluster, einschließlich der aus Snapshots wiederhergestellten, werden in Aurora MySQL 1.17.7 erstellt. Für bestehende Datenbank-Cluster können Sie ein Upgrade auf Aurora MySQL 1.17.7 ausführen (nicht verpflichtend). Zur Verwendung einer älteren Version erstellen Sie einen neuen Datenbank-Cluster in Aurora MySQL 1.14.4, 1.15.1, 1.16 oder 1.17.6. Sie können dazu die AWS CLI oder die Amazon RDS-API verwenden und die Engine-Version angeben.

Mit Aurora MySQL-Version 1.17.7 verwenden wir ein Cluster-Patching-Modell, bei dem alle Knoten in einem Aurora-DB-Cluster gleichzeitig gepatcht werden.

## Note

Diese Version ist derzeit in den Regionen AWS GovCloud (US-West) [us-gov-west-1] und China (Peking) [cn-north-1] nicht verfügbar. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

- Die Variable des InnoDB-Status `innodb_buffer_pool_size` wurde öffentlich zugänglich gemacht, damit die Kunden Änderungen vornehmen können.
- Ein Stabilitätsproblem auf dem Aurora-Cluster wurde behoben, das bei Failovers auftrat.
- Clusterverfügbarkeit wurde durch Behebung eines DDL-Wiederherstellungsproblems verbessert, das nach einer fehlgeschlagenen TRUNCATE-Operation auftrat.
- Ein durch DDL-Operationen ausgelöstes Stabilitätsproblem im Zusammenhang mit dem `mysql.innodb_table_stats`-Tabellenupdate wurde behoben.
- Stabilitätsprobleme des Aurora Replicas wurden behoben, die während der Invalidierung des Abfragecaches nach einer DDL-Operation ausgelöst wurden.



- Ein Stabilitätsproblem wurde behoben, das durch ungültigen Speicherzugriff während der periodischen Bereinigung des Datenverzeichniscaches im Hintergrund ausgelöst wurde.

## Integration von MySQL-Fehlerbehebungen (Community Edition):

- Fehler #16208542: Das Verwerfen eines Index auf einer Fremdschlüsselspalte führt zu einer fehlenden Tabelle.
- Fehler #76349: Speicherleck in `add_derived_key()`.
- Fehler #16862316: Für partitionierte Tabellen liefern Abfragen unterschiedliche Ergebnisse, je nachdem, ob Index Merge verwendet wurde.
- Fehler #17588348: Abfragen mit der `index_merge`-Optimierung (siehe [Index Merge Optimization](#)) können ungültige Ergebnisse liefern, wenn sie an Tabellen ausgeführt werden, die mithilfe von HASH partitioniert wurden.

## Aurora-MySQL-Datenbank-Engine-Updates 06.09.2018 (Version 1.17.6) (veraltet)

Version: 1.17.6

Aurora MySQL 1.17.6 ist allgemein verfügbar. Alle neuen mit MySQL 5.6 kompatiblen Aurora MySQL-Datenbank-Cluster, einschließlich der aus Snapshots wiederhergestellten, werden in Aurora MySQL 1.17.6 erstellt. Für bestehende Datenbank-Cluster können Sie ein Upgrade auf Aurora MySQL 1.17.6 ausführen (nicht verpflichtend). Zur Verwendung einer älteren Version erstellen Sie einen neuen Datenbank-Cluster in Aurora MySQL 1.14.4, 1.15.1, 1.16 oder 1.17.5. Sie können dazu die AWS CLI oder die Amazon RDS-API verwenden und die Engine-Version angeben.

Mit Aurora MySQL-Version 1.17.6 verwenden wir ein Cluster-Patching-Modell, bei dem alle Knoten in einem Aurora-DB-Cluster gleichzeitig gepatcht werden.

### Note

Diese Version ist derzeit in den Regionen AWS GovCloud (US-West) [us-gov-west-1] und China (Peking) [cn-north-1] nicht verfügbar. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

- Ein Stabilitätsproblem beim Aurora Reader für SELECT-Abfragen wurde behoben, während der Aurora Writer DDL-Operationen auf derselben Tabelle durchgeführt hat.
- Ein Stabilitätsproblem wurde behoben, das durch das Erstellen und Löschen von DDL-Protokollen für temporäre Tabellen mit einer Heap/Speicher-Engine verursacht wurde.
- Ein Stabilitätsproblem auf dem Binärprotokoll-Worker wurde behoben, bei dem DDL-Anweisungen repliziert werden, während die Verbindung zum Binärprotokoll-Master instabil ist.
- Ein Stabilitätsproblem wurde behoben, das beim Schreiben in das Slow-Query-Protokoll aufgetreten ist.
- Ein Problem mit der Replikationsstatustabelle wurde behoben, das falsche Aurora Reader-Verzögerungsinformationen bereitgestellt hat.

## Integration von MySQL-Fehlerbehebungen (Community Edition):

- Für eine [ALTER TABLE](#)-Anweisung, durch die der Standardwert einer [BINARY](#)-Spalte umbenannt oder geändert wurde, ist die Änderung mit einer Tabellenkopie und nicht direkt durchgeführt worden. (Fehler #67141, Fehler #14735373, Fehler #69580, Fehler #17024290)
- Eine äußere Join-Operation zwischen einer regulären Tabelle und einer abgeleiteten Tabelle, die implizit aus Gruppen besteht, könnte zum Beenden des Servers führen. (Fehler #16177639)

## Aurora-MySQL-Datenbank-Engine-Updates 14.08.2018 (Version 1.17.5) (veraltet)

Version: 1.17.5

Aurora MySQL 1.17.5 ist allgemein verfügbar. Alle neuen mit MySQL 5.6 kompatiblen Aurora MySQL-Datenbank-Cluster, einschließlich der aus Snapshots wiederhergestellten, werden in Aurora MySQL 1.17.5 erstellt. Für bestehende Datenbank-Cluster können Sie ein Upgrade auf Aurora MySQL 1.17.5 ausführen (nicht verpflichtend). Zur Verwendung einer älteren Version erstellen Sie

einen neuen Datenbank-Cluster in Aurora MySQL 1.14.4, 1.15.1, 1.16 oder 1.17.4. Sie können dazu die AWS CLI oder die Amazon RDS-API verwenden und die Engine-Version angeben.

Mit Aurora MySQL-Version 1.17.5 verwenden wir ein Cluster-Patching-Modell, bei dem alle Knoten in einem Aurora-DB-Cluster gleichzeitig gepatcht werden.

#### Note

Diese Version ist derzeit in den Regionen AWS GovCloud (US-West) [us-gov-west-1] und China (Peking) [cn-north-1] nicht verfügbar. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

- Es wurde ein Problem behoben, bei dem ein Aurora Writer möglicherweise neu startet, nachdem ein Aurora-Cluster mit der Funktion Zero-Downtime-Patching (Patchen ohne Ausfallzeiten) gepatcht wurde.

## Aurora-MySQL-Datenbank-Engine-Updates 07.08.2018 (Version 1.17.4) (veraltet)

Version: 1.17.4

Aurora MySQL 1.17.4 ist allgemein verfügbar. Alle neuen mit MySQL 5.6 kompatiblen Aurora MySQL-Datenbank-Cluster, einschließlich der aus Snapshots wiederhergestellten, werden in Aurora MySQL 1.17.4 erstellt. Für bestehende Datenbank-Cluster können Sie ein Upgrade auf Aurora MySQL 1.17.4 ausführen (nicht verpflichtend). Zur Verwendung einer älteren Version erstellen Sie einen neuen Datenbank-Cluster in Aurora MySQL 1.14.4, 1.15.1, 1.16 oder 1.17.3. Sie können dazu die AWS CLI oder die Amazon RDS-API verwenden und die Engine-Version angeben.

Mit Aurora MySQL-Version 1.17.4 verwenden wir ein Cluster-Patching-Modell, bei dem alle Knoten in einem Aurora-DB-Cluster gleichzeitig gepatcht werden.

**Note**

Diese Version ist derzeit in den Regionen AWS GovCloud (US-West) [us-gov-west-1] und China (Peking) [cn-north-1] nicht verfügbar. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

- Replikationsverbesserungen:
  - Weniger Datenverkehr im Netzwerk, weil Binärprotokoll-Datensätze nicht an die Cluster-Replikate übertragen werden. Diese Verbesserung ist standardmäßig aktiviert.
  - Weniger Datenverkehr im Netzwerk, weil die Replikationsnachrichten komprimiert werden. Diese Verbesserung ist standardmäßig für die Instance-Klassen 8xlarge und 16xlarge aktiviert. Diese großen Instances sind für Datenverkehr für Schreiboperationen in großem Umfang geeignet, bei denen aufgrund der Replikationsmeldungen in nennenswertem Umfang zusätzlicher Datenverkehr im Netzwerk auftritt.
  - Korrekturen am Abfrage-Cache für Replikate
- Es wurde ein Problem behoben, bei dem `ORDER BY LOWER(col_name)` bei Verwendung der Kollation `utf8_bin` eine falsche Reihenfolge produzierte.
- Ein Problem wurde behoben, bei dem DDL-Anweisungen (insbesondere `TRUNCATE TABLE`) Probleme mit Aurora-Replicas verursachten, einschließlich Instabilität und fehlender Tabellen.
- Es wurde ein Problem behoben, bei dem Sockets in einem halb geöffneten Zustand verblieben, wenn Speicherknoten neu gestartet wurden.
- Die folgenden neuen DB-Cluster-Parameter sind verfügbar:
  - `aurora_enable_zdr` – Erlauben Sie, dass Verbindungen, die auf einem Aurora-Replica geöffnet wurden, beim Replica-Neustart aktiv bleiben.
  - `aurora_enable_replica_log_compression` – Aktivieren Sie die Kompression von Replikationsnutzlasten, um die Nutzung der Netzwerkbandbreite zwischen Master und Aurora-Replicas zu verbessern.

- `aurora_enable_repl_bin_log_filtering` – Aktivieren Sie die Filterung von Replikationsdatensätzen, die von Aurora-Replicas auf dem Master nicht verwendet werden können.

## Aurora-MySQL-Datenbank-Engine-Updates 05.06.2018 (Version 1.17.3) (veraltet)

Version: 1.17.3

Aurora MySQL 1.17.3 ist allgemein verfügbar. Alle neuen mit MySQL 5.6 kompatiblen Aurora MySQL-Datenbank-Cluster, einschließlich der aus Snapshots wiederhergestellten, werden in Aurora MySQL 1.17.3 erstellt. Für bestehende Datenbank-Cluster können Sie ein Upgrade auf Aurora MySQL 1.17.3 ausführen (nicht verpflichtend). Sie können neue DB-Cluster in Aurora MySQL 1.14.4, Aurora MySQL 1.15.1 oder Aurora MySQL 1.16 erstellen. Sie können dazu die AWS CLI oder die Amazon RDS-API verwenden und die Engine-Version angeben.

Mit Aurora MySQL-Version 1.17.3 verwenden wir ein Cluster-Patching-Modell, bei dem alle Knoten in einem Aurora-DB-Cluster gleichzeitig gepatcht werden.

### Note

Diese Version ist derzeit in den Regionen AWS GovCloud (US-West) [us-gov-west-1] und China (Peking) [cn-north-1] nicht verfügbar. Sobald die Version verfügbar ist, wird dies separat bekanntgegeben.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

- Es wurde ein Problem behoben, bei dem eine Aurora-Replica neu gestartet werden konnte, wenn optimistische Cursor-Wiederherstellungen beim Lesen von Datensätzen verwendet wurden.
- Ein Problem wurde behoben, bei dem ein Aurora-Writer beim Versuch neu gestartet wurde, eine MySQL-Sitzung bei aktiviertem Performance-Schema zu beenden (kill „`<session id>`“).

- Es wurde ein Problem behoben, bei dem ein Aurora Writer neu gestartet wurde, wenn ein Schwellenwert für die Garbage Collection berechnet wurde.
- Ein Problem wurde behoben, bei dem ein Aurora-Writer gelegentlich neu gestartet werden konnte, wenn er den Aurora-Replica-Fortschritt in der Protokollanwendung verfolgt hat.
- Es wurde ein Problem mit dem Abfrage-Cache behoben, das zu veralteten Lesezugriffen führen konnte, wenn das Auto-Commit ausgeschaltet ist.

## Aurora-MySQL-Datenbank-Engine-Updates 27.04.2018 (Version 1.17.2) (veraltet)

Version: 1.17.2

Aurora MySQL 1.17.2 ist allgemein verfügbar. Alle neuen mit MySQL 5.6 kompatiblen Aurora MySQL-Datenbank-Cluster, einschließlich der aus Snapshots wiederhergestellten, werden in Aurora MySQL 1.17.2 erstellt. Für bestehende Datenbank-Cluster können Sie ein Upgrade auf Aurora MySQL 1.17.2 ausführen (nicht verpflichtend). Sie können neue DB-Cluster in Aurora MySQL 1.14.4, Aurora MySQL 1.15.1 oder Aurora MySQL 1.16 erstellen. Sie können dazu die AWS CLI oder die Amazon RDS-API verwenden und die Engine-Version angeben.

Mit Aurora MySQL-Version 1.17.2 verwenden wir ein Cluster-Patching-Modell, bei dem alle Knoten in einem Aurora-DB-Cluster gleichzeitig gepatcht werden.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

### Verbesserungen

- Ein Problem wurde behoben, das während bestimmter DDL-Partitionsoperationen Neustarts verursachte.
- Es wurde ein Problem behoben, das dazu führte, dass die Unterstützung für den Aufruf von AWS Lambda Funktionen über native Aurora MySQL-Funktionen deaktiviert wurde.
- Ein Problem mit der Cache-Invalidierung wurde behoben, das zu Neustarts bei Aurora Replicas führte.
- Ein Problem im Sperren-Manager wurde behoben, das zu Neustarts führte.

# Aurora-MySQL-Datenbank-Engine-Updates 23.03.2018 (Version 1.17.1) (veraltet)

Version: 1.17.1

Aurora MySQL 1.17.1 ist allgemein verfügbar. Alle neuen Datenbank-Cluster, einschließlich der aus Snapshots wiederhergestellten, werden in Aurora MySQL v1.17.1 erstellt. Für bestehende Datenbank-Cluster können Sie ein Upgrade auf Aurora MySQL 1.17.1 ausführen (nicht verpflichtend). Sie können neue DB-Cluster in Aurora MySQL 1.15.1, Aurora MySQL 1.16 oder Aurora MySQL 1.17 erstellen. Sie können dazu die AWS CLI oder die Amazon RDS-API verwenden und die Engine-Version angeben.

Mit Aurora MySQL-Version 1.17.1 verwenden wir ein Cluster-Patching-Modell, bei dem alle Knoten in einem Aurora-DB-Cluster gleichzeitig gepatcht werden. In dieser Veröffentlichung werden einige bekannte Engine-Probleme sowie Regressionen behoben.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Note

Es gibt ein Problem mit der aktuellen Version der Aurora MySQL-Engine. Nach einem Upgrade auf Version 1.17.1 wird die Engine-Version fälschlicherweise als angezeig 1.17. Wenn Sie ein Upgrade auf Version 1.17.1 durchführen, können Sie in der Spalte Maintenance (Wartung) des DB-Clusters in der AWS Management Console überprüfen, ob das Upgrade erfolgreich ausgeführt wurde. Wenn hier none angezeigt wird, wurde ein Engine-Upgrade auf Version 1.17.1 ausgeführt.

## Verbesserungen

- Behebung eines Fehler bei der Wiederherstellung von Binärprotokollen, der im Falle von größeren Binärprotokoll-Indexdateien zu längeren Wiederherstellungszeiten führte (dies kann geschehen, wenn die Binärprotokolle sehr häufig rotieren).
- Behebung eines Fehlers im Abfrageoptimierer, der einen ineffizienten Abfrageplan für partitionierte Tabellen generierte.

- Behebung eines Fehlers im Abfrageoptimierer, bei dem eine Bereichsabfrage zu einem Neustart der Datenbank-Engine führte.

## Aurora-MySQL-Datenbank-Engine-Updates 13.03.2018 (Version 1.17) (veraltet)

Version: 1.17

Aurora MySQL 1.17 ist allgemein verfügbar. Aurora MySQL 1.x-Versionen sind nur mit MySQL 5.6 kompatibel, nicht mit MySQL 5.7. Alle neuen und mit Version 5.6 kompatiblen Datenbank-Cluster, einschließlich der aus Snapshots wiederhergestellten, werden in Aurora v1.1.17 erstellt. Für bestehende Datenbank-Cluster können Sie ein Upgrade auf Aurora 1.17 ausführen (nicht verpflichtend). Sie können neue DB-Cluster in Aurora 1.14.1, Aurora 1.15.1 oder Aurora 1.16 erstellen. Sie können dazu die AWS CLI oder die Amazon RDS-API verwenden und die Engine-Version angeben.

Mit Aurora-Version 1.17 verwenden wir ein Cluster-Patching-Modell, bei dem alle Knoten in einem Aurora-DB-Cluster gleichzeitig gepatcht werden. Wir unterstützen das Zero-Downtime-Patching, das auf Best-Effort-Basis funktioniert, um Client-Verbindungen vor dem Patch-Vorgang zu bewahren. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung.

### Zero-Downtime-Patching (Patchen ohne Ausfallzeiten)

Beim Feature des Patchens ohne Ausfallzeiten (ZDP – Zero-Downtime Patching) wird versucht, Client-Verbindungen auf Best-Effort-Basis vor dem Patch-Vorgang zu bewahren. Weitere Informationen finden Sie unter [Verwendung des Zero-Downtime-Patchings](#) im Amazon-Aurora-Benutzerhandbuch.

### Neue Features

- Aurora MySQL unterstützt nun die Lock-Komprimierung, mit der die Speichernutzung vom Lock Manager optimiert wird. Ab Version 1.17 können Sie diese Funktion auch ohne Aktivierung des Labor-Modus nutzen.



## Verbesserungen

- Ein Fehler wurde behoben, der hauptsächlich bei Instances mit weniger Cores auftrat und bei dem ein einzelner Core auch bei ungenutzter Datenbank eine CPU-Auslastung von 100 % aufweisen konnte.
- Verbesserte Leistung beim Abruf der Binärprotokolle von Aurora-Clustern.
- Behebung eines Fehlers, bei dem Aurora-Replicas während des Versuchs, Tabellenstatistiken in den persistenten Speicher zu schreiben, abstürzten.
- Behebung eines Fehlers, bei dem der Abfrage-Cache auf Aurora-Replicas nicht ordnungsgemäß funktionierte.
- Behebung einer race-Bedingung im Lock-Manager, die zu einem Neustart der Engine führte.
- Behebung eines Fehlers, bei dem Locks, die für schreibgeschützte Transaktionen mit automatischem Commit gelten, zu einem Neustart der Engine führten.
- Behebung eines Fehlers, bei dem einige Abfragen nicht in die Audit-Protokolle geschrieben wurden.
- Behebung eines Fehlers bei der Wiederherstellung bestimmter Partitionswartungsvorgänge bei einem Failover.

## Integration von MySQL-Fehlerbehebungen

- LAST\_INSERT\_ID wird bei Nutzung von Replikationsfiltern fehlerhaft repliziert (Fehler 69861).
- Abfrage gibt abhängig von der INDEX\_MERGE-Einstellung unterschiedliche Ergebnisse zurück (Fehler 16862316).
- Abfrageausführung führt gespeicherte Routine erneut aus, ineffizienter Abfrageplan (Fehler 16346367).
- INNODB-FTS: Assert in FTS\_CACHE\_APPEND\_DELETED\_DOC\_IDS (Fehler 18079671).
- Assert RBT\_EMPTY(INDEX\_CACHE->WORDS) in ALTER TABLE CHANGE COLUMN (Fehler 17536995).
- INNODB-Volltextsuche findet keine Datensätze, wenn Savepoints involviert sind (Fehler 70333, Fehler 17458835)

# Aurora-MySQL-Datenbank-Engine-Updates 11.12.2017 (Version 1.16) (veraltet)

Version: 1.16

Aurora MySQL 1.16 ist allgemein verfügbar. Alle neuen Datenbank-Cluster, einschließlich der aus Snapshots wiederhergestellten, werden in Aurora 1.16 erstellt. Für bestehende Datenbank-Cluster können Sie ein Upgrade auf Aurora 1.16 ausführen (nicht verpflichtend). Sie können neue DB-Cluster in Aurora 1.14.1 oder Aurora 1.15.1 erstellen. Sie können dazu die AWS CLI oder die Amazon RDS-API verwenden und die Engine-Version angeben.

Mit Aurora-Version 1.16 verwenden wir ein Cluster-Patching-Modell, bei dem alle Knoten in einem Aurora-DB-Cluster gleichzeitig gepatcht werden. Wir aktivieren das Zero-Downtime-Patching, das auf Best-Effort-Basis funktioniert, um Client-Verbindungen vor dem Patch-Vorgang zu bewahren. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung.

## Zero-Downtime-Patching (Patchen ohne Ausfallzeiten)

Beim Feature des Patchens ohne Ausfallzeiten (ZDP – Zero-Downtime Patching) wird versucht, Client-Verbindungen auf Best-Effort-Basis vor dem Patch-Vorgang zu bewahren. Weitere Informationen finden Sie unter [Verwendung des Zero-Downtime-Patchings](#) im Amazon-Aurora-Benutzerhandbuch.

## Neue Features

- Aurora MySQL unterstützt jetzt synchrone AWS Lambda Aufrufe über die native Funktion. `lambda_sync()` Außerdem ist die native Funktion `lambda_async()` vorhanden, die als Alternative zu der bestehenden gespeicherten Prozedur für asynchrone Lambda-Aufrufe verwendet werden kann. Weitere Informationen finden Sie unter Aufrufen einer [Lambda-Funktion aus einem Amazon-Aurora-MySQL-DB-Cluster](#) im Amazon-Aurora-Benutzerhandbuch.
- Aurora MySQL unterstützt jetzt Hash-Joins, um Equijoin-Abfragen zu beschleunigen. Der kostenbasierte Optimierer von Aurora kann automatisch entscheiden, wann Hash-Joins verwendet werden sollen. Sie können die Verwendung in einem Abfrageplan auch erzwingen. Weitere

Informationen finden Sie unter [Optimieren großer Aurora-Join-Abfragen mit Hash-Joins](#) im Amazon-Aurora-Benutzerhandbuch.

- Aurora MySQL unterstützt jetzt Scan-Batching, um die speicherinternen, scan-orientierten Abfragen erheblich zu beschleunigen. Die Funktion steigert die Leistung von Tabellenvollscans, Indexvollscans und Indexbereichsscans durch Stapelverarbeitung.

## Verbesserungen

- Ein Problem behoben, bei dem die Read Replicas abgestürzt sind, wenn Abfragen für Tabellen ausgeführt wurden, die gerade auf den Master abgelegt wurden.
- Ein Problem behoben, bei dem der Neustart des Schreibers auf einem Datenbank-Cluster mit einer sehr großen Anzahl an FULLTEXT-Indizes zu einer längeren Wiederherstellungsdauer führt, als erwartet.
- Ein Problem behoben, bei dem das Leeren von Binärprotokollen LOST\_EVENTS-Vorfälle in Binärprotokollereignissen verursacht.
- Stabilitätsprobleme mit dem Scheduler behoben, die aufgetreten sind, wenn das Leistungsschema aktiviert war.
- Ein Problem behoben, bei dem eine Unterabfrage, die temporäre Tabellen verwendet, partielle Ergebnisse zurückgeben konnte.

## Integration von MySQL-Fehlerbehebungen

None

## Aurora-MySQL-Datenbank-Engine-Updates 20.11.2017 (Version 1.15.1) (veraltet)

Version: 1.15.1

Aurora MySQL 1.15.1 ist allgemein verfügbar. Alle neuen Datenbank-Cluster, einschließlich der aus Snapshots wiederhergestellten, werden in Aurora 1.15.1 erstellt. Für bestehende DB-Cluster können Sie ein Upgrade auf Aurora 1.15.1 ausführen (nicht verpflichtend). Sie können neue DB-Cluster in Aurora 1.14.1 erstellen. Sie können dazu die AWS CLI oder die Amazon RDS-API verwenden und die Engine-Version angeben.

Mit Aurora-Version 1.15.1 verwenden wir ein Cluster-Patching-Modell, bei dem alle Knoten in einem Aurora-DB-Cluster gleichzeitig gepatcht werden. Wir aktivieren das Zero-Downtime-Patching, das auf Best-Effort-Basis funktioniert, um Client-Verbindungen vor dem Patch-Vorgang zu bewahren. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Zero-Downtime-Patching (Patchen ohne Ausfallzeiten)

Beim Feature des Patchens ohne Ausfallzeiten (ZDP – Zero-Downtime Patching) wird versucht, Client-Verbindungen auf Best-Effort-Basis vor dem Patch-Vorgang zu bewahren. Weitere Informationen finden Sie unter [Verwendung des Zero-Downtime-Patchings](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

- Es wurde ein Problem im adaptiven Segment-Selektor für eine Leseanforderung behoben, das dazu führen konnte, dass das gleiche Segment zweimal gewählt wurde, was unter bestimmten Bedingungen zu einem Anstieg der Leselatenz führte.
- Ein Problem, das auf eine Optimierung in Aurora MySQL für den Thread-Scheduler zurückzuführen ist, wurde behoben. Dieses Problem manifestiert sich in unechten Fehlern beim Schreiben in das langsame Protokoll, während die zugehörigen Abfragen selbst gut funktionieren.
- Es wurde ein Problem mit der Stabilität von Read Replicas bei großen (> 5 TB) Volumes behoben.
- Es wurde ein Problem behoben, bei dem die Anzahl der Worker-Threads kontinuierlich zunahm, da die Anzahl der Verbindungen nicht korrekt war.
- Es wurde ein Problem mit Tabellensperren behoben, das zu langen Semaphore-Wartezeiten während der Einfüge-Workloads führte.
- Die folgenden, in Aurora MySQL 1.15 enthaltenen MySQL-Bugfixes wurden zurückgesetzt:
  - MySQL-Instance verhindert Ausführung des SYNC-Index (Fehler 73816)
  - Assert RBT\_EMPTY(INDEX\_CACHE->WORDS) in ALTER TABLE CHANGE COLUMN (Fehler 17536995)
  - InnoDB-Volltextsuche findet keine Datensätze, wenn Savepoints involviert sind (Fehler 70333).

## Integration von MySQL-Fehlerbehebungen

None

### Aurora-MySQL-Datenbank-Engine-Updates 24.10.2017 (Version 1.15) (veraltet)

Version: 1.15

Aurora MySQL 1.15 ist allgemein verfügbar. Alle neuen Datenbank-Cluster, einschließlich der aus Snapshots wiederhergestellten, werden in Aurora 1.15 erstellt. Für bestehende DB-Cluster können Sie ein Upgrade auf Aurora 1.15 ausführen (nicht verpflichtend). Sie können neue DB-Cluster in Aurora 1.14.1 erstellen. Sie können dazu die AWS CLI oder die Amazon RDS-API verwenden und die Engine-Version angeben.

Mit Aurora-Version 1.15 verwenden wir ein Cluster-Patching-Modell, bei dem alle Knoten in einem Aurora-DB-Cluster gleichzeitig gepatcht werden. Da Updates den Neustart einer Datenbank erfordern, kommt es zu einem 20- bis 30-minütigen Nutzungsausfall, nach dem Sie wieder mit der Verwendung Ihres DB-Clusters oder -Cluster fortfahren können. Wenn Ihre DB-Cluster derzeit Aurora 1.14 oder Aurora 1.14.1 ausführen, lässt die Zero-Downtime-Patch-Funktion (ZDP) von Aurora MySQL möglicherweise zu, dass Client-Verbindungen mit Ihrer primären Aurora MySQL-Instance während des Upgrades aufrechterhalten bleiben. Dies ist abhängig von Ihrem Workload.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

### Zero-Downtime-Patching (Patchen ohne Ausfallzeiten)

Beim Feature des Patchens ohne Ausfallzeiten (ZDP – Zero-Downtime Patching) wird versucht, Client-Verbindungen auf Best-Effort-Basis vor dem Patch-Vorgang zu bewahren. Weitere Informationen finden Sie unter [Verwendung des Zero-Downtime-Patchings](#) im Amazon-Aurora-Benutzerhandbuch.

### Neue Features

- **Asynchrones Schlüssel-Prefetching** – Asynchrones Schlüssel-Prefetching (Asynchronous Key Prefetch, AKP) ist eine Funktion, die die Performance von nicht gecachten Indexverbindungen verbessern soll, indem Schlüssel bereits aus dem Speicher abgerufen werden, bevor sie

benötigt werden. Der primäre Anwendungsfall für AKP ist eine Indexverbindung zwischen einer kleinen äußeren und einer großen inneren Tabelle, wobei der Index in der größeren Tabelle höchst selektiv ist. Wenn die Multi-Range Read (MRR)-Schnittstelle aktiviert ist, wird AKP zudem für eine Suche vom sekundären zum primären Index genutzt. Kleinere Instances mit Speichergrößenbeschränkungen können AKP in einigen Fällen möglicherweise nutzen, sofern die richtige Schlüsselkardinalität vorliegt. Weitere Informationen finden Sie unter [Optimieren von indizierten Aurora-Join-Abfragen mit asynchronem Schlüssel-Prefetch](#) im Amazon-Aurora-Benutzerhandbuch.

- Schnelle DDL – wir haben die Funktion, die mit [Aurora 1.13](#) veröffentlicht wurde, auf Vorgänge erweitert, die Standardwerte umfassen. Mit dieser Erweiterung kann schnelle DDL für Vorgänge verwendet werden, die eine löschbare Spalte mit oder ohne Standardwerte am Ende einer Tabelle hinzufügen. Die Funktion bleibt dem Aurora-Labor-Modus vorbehalten. Weitere Informationen finden Sie unter [Ändern von Tabellen in Amazon Aurora mithilfe von schneller DDL](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

- Berechnungsfehler während der Optimierung von WITHIN/CONTAINS-Spatial-Abfragen, die zuvor ein leeres Ergebnis lieferten, behoben.
- SHOW VARIABLE-Befehl korrigiert, sodass er den aktualisierten Wert für den Parameter `innodb_buffer_pool_size` zeigt, wenn dieser in der Parametergruppe geändert wird.
- Verbesserte Stabilität der primären Instance bei Masseneinfüfungsvorgängen in eine Tabelle, die mit schneller DDL geändert wurde, wenn die adaptive Hash-Indizierung deaktiviert ist und der einzufügende Datensatz der erste Datensatz einer Seite ist.
- Verbesserte Stabilität von Aurora, wenn der Benutzer versucht, den DB-Cluster-Parameterwert `server_audit_events` auf **default** festzulegen.
- Ein Problem wurde behoben, bei dem eine Änderung am Datenbank-Zeichensatz für die Anweisung ALTER TABLE, die auf der primären Aurora-Instance ausgeführt wurde, erst auf die Aurora-Replicas repliziert wurde, wenn diese neu gestartet wurden.
- Verbesserte Stabilität durch Beheben einer Race Condition auf der primären Instance, die es zuvor ermöglichte, ein Aurora-Replica zu registrieren, auch wenn das Volume der primären Instance geschlossen war.
- Verbesserte Performance der primären Instance während der Indexerstellung in einer großen Tabelle durch Ändern des Sperrprotokolls, sodass gleichzeitige Data Manipulation (DML)-Anweisungen während der Indexerstellung aktiviert sind.

- InnoDB-Metadaten-Inkonsistenz während der Abfrage ALTER TABLE RENAME behoben, wodurch die Stabilität verbessert wurde. Beispiel: Wenn Spalten der Tabelle t1 (c1, c2) innerhalb derselben ALTER-Anweisung zyklisch in t1 (c2, c3) umbenannt werden.
- Verbesserte Stabilität von Aurora-Replicas in Szenarien, in denen ein Aurora-Replica keinen aktiven Workload hat und die primäre Instance nicht reagiert.
- Verbesserte Verfügbarkeit von Aurora-Replicas in Szenarien, in denen ein Aurora-Replica eine Tabelle gesperrt hat und den Replikations-Thread daran hindert, von der primären Instance empfangene DDL-Änderungen anzuwenden.
- Verbesserte Stabilität der primären Instance, wenn ein Fremdschlüssel und eine Spalte aus zwei separaten Sitzungen gleichzeitig zu einer Tabelle hinzugefügt werden und schnelle DDL aktiviert wurde.
- Verbesserte Stabilität des Bereinigungs-Threads in der primären Instance während umfangreicher Schreib-Workloads durch Blockieren der Kürzung von rückgängig-Datensätzen, bis diese bereinigt wurden.
- Verbesserte Stabilität durch Korrektur der Sperrfreigabereihenfolge während des Commit-Prozesses von Transaktionen, in denen Tabellen entfernt werden.
- Fehler bei Aurora-Replicas behoben, bei dem die DB-Instance den Startvorgang nicht abschließen konnte und meldete, dass Port 3306 bereits in Verwendung sei.
- Race Condition behoben, bei der eine SELECT-Abfrage in bestimmten information\_schema-Tabellen (innodb\_trx, innodb\_lock, innodb\_lock\_waits) die Cluster-Instabilität erhöhte.

## Integration von MySQL-Fehlerbehebungen

- CREATE USER akzeptiert Plug-in- und Passwort-Hash, ignoriert jedoch Passwort-Hash (Fehler 78033).
- Die Partitionierungs-Engine fügt Felder zum Lese-Bitsatz hinzu, damit Einträge über einen partitionierten Index sortiert ausgegeben werden können. Dies führt dazu, dass der Verbindungspuffer versucht, nicht benötigte Felder zu lesen. Behoben, indem nicht alle Partitionierungsfelder zu read\_set hinzugefügt werden, sondern stattdessen nur basierend auf den bereits festgelegten Präfix-Feldern in read\_set sortiert wird. DEBUG\_ASSERT hinzugefügt, damit bei der Ausführung von key\_cmp zumindest das erste Feld gelesen werden muss (Fehler 16367691).
- MySQL-Instance verhindert Ausführung des SYNC-Index (Fehler 73816)
- Assert RBT\_EMPTY(INDEX\_CACHE->WORDS) in ALTER TABLE CHANGE COLUMN (Fehler 17536995)

- InnoDB-Volltextsuche findet keine Datensätze, wenn Savepoints involviert sind (Fehler 70333).

## Aurora-MySQL-Datenbank-Engine-Updates 13.03.2018 (Version 1.14.4) (veraltet)

Version: 1.14.4

Aurora MySQL 1.14.4 ist allgemein verfügbar. Sie können neue DB-Cluster in Aurora 1.14.4 erstellen, indem Sie die AWS CLI oder die Amazon RDS-API verwenden und die Engine-Version angeben. Für bestehende 1.14.x-DB-Cluster können Sie ein Upgrade auf Aurora 1.14.4 ausführen (dies ist aber nicht verpflichtend).

Mit Aurora-Version 1.14.4 verwenden wir ein Cluster-Patching-Modell, bei dem alle Knoten in einem Aurora-DB-Cluster gleichzeitig gepatcht werden. Wir unterstützen das Zero-Downtime-Patching, das auf Best-Effort-Basis funktioniert, um Client-Verbindungen vor dem Patch-Vorgang zu bewahren. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

### Zero-Downtime-Patching (Patchen ohne Ausfallzeiten)

Beim Feature des Patchens ohne Ausfallzeiten (ZDP – Zero-Downtime Patching) wird versucht, Client-Verbindungen auf Best-Effort-Basis vor dem Patch-Vorgang zu bewahren. Weitere Informationen finden Sie unter [Verwendung des Zero-Downtime-Patchings](#) im Amazon-Aurora-Benutzerhandbuch.

### Neue Features

- Aurora MySQL unterstützt nun die DB-Instance-Klassen db.r4.

### Verbesserungen

- Behebung eines Fehlers, bei dem LOST\_EVENTS beim Schreiben von umfangreichen Binärprotokollereignissen generiert wurden.



## Integration von MySQL-Fehlerbehebungen

- Ignorierbare Ereignisse funktionieren nicht und werden nicht getestet (Fehler 74683).
- FEHLER DES NEUEN->ALTEN ASSERTS 'GTID\_MODE > 0' (Fehler 20436436).

## Aurora-MySQL-Datenbank-Engine-Updates 22.09.2017 (Version 1.14.1) (veraltet)

Version: 1.14.1

Aurora MySQL 1.14.1 ist allgemein verfügbar. Alle neuen Datenbank-Cluster, einschließlich der aus Snapshots wiederhergestellten, werden in Aurora MySQL 1.14.1 erstellt. Aurora MySQL 1.14.1 ist zudem ein obligatorisches Upgrade für bestehende Aurora MySQL-DB-Cluster. Weitere Informationen finden Sie unter [Ankündigung: Verlängerung des verbindlichen Upgrade-Zeitplans für Amazon Aurora](#) auf der Website der AWS Entwicklerforen.

Mit Aurora MySQL-Version 1.14.1 verwenden wir ein Cluster-Patching-Modell, bei dem alle Knoten in einem Aurora MySQL-DB-Cluster gleichzeitig gepatcht werden. Da Updates den Neustart einer Datenbank erfordern, kommt es zu einem 20- bis 30-minütigen Nutzungsausfall, nach dem Sie wieder mit der Verwendung Ihres DB-Clusters oder -Cluster fortfahren können. Wenn Ihre DB-Cluster derzeit Version 1.13 oder höher ausführen, lässt die Zero-Downtime-Patch-Funktion (ZDP) von Aurora MySQL möglicherweise zu, dass Client-Verbindungen mit Ihrer primären Aurora MySQL-Instance während des Upgrades aufrechterhalten bleiben. Dies ist abhängig von Ihrem Workload.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung.

## Verbesserungen

- Race Conditions im Zusammenhang mit Einfüge- und Bereinigungsvorgängen behoben, um die Stabilität der Funktion für schnelle DDL zu verbessern, die dem Aurora MySQL-Labor-Modus vorbehalten bleibt.

# Aurora-MySQL-Datenbank-Engine-Updates 07.08.2017 (Version 1.14) (veraltet)

Version: 1.14

Aurora MySQL 1.14 ist allgemein verfügbar. Alle neuen Datenbank-Cluster, einschließlich der aus Snapshots wiederhergestellten, werden in Aurora MySQL 1.14 erstellt. Aurora MySQL 1.14 ist zudem ein obligatorisches Upgrade für bestehende Aurora MySQL-DB-Cluster. Wir werden eine separate Ankündigung dazu versenden, zu welchem Zeitpunkt frühere Versionen von Aurora MySQL eingestellt werden.

Mit Aurora MySQL-Version 1.14 verwenden wir ein Cluster-Patching-Modell, bei dem alle Knoten in einem Aurora-DB-Cluster gleichzeitig gepatcht werden. Da Updates den Neustart einer Datenbank erfordern, kommt es zu einem 20- bis 30-minütigen Nutzungsausfall, nach dem Sie wieder mit der Verwendung Ihres DB-Clusters oder -Cluster fortfahren können. Wenn Ihre DB-Cluster derzeit Version 1.13 ausführen, lässt die Zero-Downtime-Patch-Funktion (ZDP) von Aurora möglicherweise zu, dass Client-Verbindungen mit Ihrer primären Aurora-Instance während des Upgrades aufrechterhalten bleiben. Dies ist abhängig von Ihrem Workload.

Wenn Sie Fragen oder Bedenken haben, steht Ihnen der AWS Support in den Community-Foren und über den [AWS Support](#) zur Verfügung.

## Zero-Downtime-Patching (Patchen ohne Ausfallzeiten)

Beim Feature des Patchens ohne Ausfallzeiten (ZDP – Zero-Downtime Patching) wird versucht, Client-Verbindungen auf Best-Effort-Basis vor dem Patch-Vorgang zu bewahren. Weitere Informationen finden Sie unter [Verwendung des Zero-Downtime-Patchings](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

- Falscher Fehler „Datensatz nicht gefunden“ behoben, wenn ein Datensatz im sekundären Index, jedoch nicht im primären Index gefunden wird.
- Stabilitätsproblem behoben, das aufgrund einer zu hoch eingestellten Abwehrrzusicherung (hinzugefügt in 1.12) auftreten kann, wenn ein einzelner Schreibvorgang mehr als 32 Seiten umfasst. Dies kann beispielsweise bei großen BLOB-Werten auftreten.
- Stabilitätsproblem aufgrund von Inkonsistenzen zwischen dem Tabellenraum-Cache und dem Wörterbuch-Cache behoben.

- Problem behoben, bei dem ein Aurora-Replica nicht mehr reagiert, nachdem die maximale Anzahl an Verbindungsversuchen mit der primären Instance überschritten wurde. Ein Aurora-Replica startet nun neu, wenn die Inaktivität länger ist als der Heartbeat-Zeitraum, der für die Zustandsprüfung von der primären Instance verwendet wird.
- Livelock-Problem behoben, das bei sehr hohen Gleichzeitigkeitsstufen auftreten kann, wenn eine Verbindung versucht, eine exklusive Metadaten-Sperre (Meta Data Lock, MDL) zu erzielen, während ein Befehl ausgegeben wird, beispielsweise `ALTER TABLE`.
- Stabilitätsproblem in einer Aurora-Read Replica bei vorliegendem logischen/parallelen Read-Ahead behoben.
- `LOAD FROM S3` in zweierlei Hinsicht verbessert:
  1. Besserer Umgang mit Amazon S3-Timeout-Fehlern durch Verwendung des SDK-Wiederholversuchs zusätzlich zum vorhandenen Wiederholversuch.
  2. Performance-Optimierung beim Laden sehr großer Dateien oder einer großen Dateianzahl durch Cachen und Wiederverwenden des Client-Status.
- Folgende Stabilitätsprobleme mit schneller DDL für `ALTER TABLE`-Vorgänge behoben:
  1. Wenn die Anweisung `ALTER TABLE` mehrere `ADD COLUMN`-Befehle umfasst und die Spaltennamen nicht in aufsteigender Reihenfolge sortiert sind.
  2. Wenn der Name der zu aktualisierenden Spalte und der entsprechende Name, abgerufen aus der internen Systemtabelle, sich durch ein null-terminiertes Zeichen (`/0`) unterscheiden.
  3. Bei bestimmten B-Baum-Split-Operationen.
  4. Wenn die Tabelle über einen Primärschlüssel mit variabler Länge verfügt.
- Stabilitätsproblem mit Aurora-Replicas behoben, bei dem es zu lange dauert, das Index-Cache für die Volltextsuche (Full Text Search, FTS) mit dem der primären Instance konsistent zu machen. Dies kann auftreten, wenn ein großer Teil der neu erstellten FTS-Indexeinträge in der primären Instance noch nicht auf das Laufwerk übertragen wurde.
- Stabilitätsproblem behoben, das während der Indexerstellung auftreten kann.
- Neue Infrastruktur, die den Speicherverbrauch pro Verbindung und die zugehörige Telemetrie aufzeichnet, die zur Entwicklung der OOM-Vermeidung (Out-Of-Memory) verwendet wird.
- Ein Problem wurde behoben, bei dem `ANALYZE TABLE` fälschlicherweise für Aurora-Replicas zugelassen wurde. Dies wurde nun blockiert.
- Stabilitätsproblem behoben, das durch ein seltenes Deadlock-Problem in Folge einer Race Condition zwischen dem logischen Read-Ahead und der Bereinigung verursacht wurde.

## Integration von MySQL-Fehlerbehebungen

- Eine Volltextsuche in Verbindung mit abgeleiteten Tabellen (Unterabfragen in der FROM-Klausel) führte zu einem Austritt des Servers. Wenn nun eine Volltextsuche von einer abgeleiteten Tabelle abhängt, gibt der Server eine Fehlermeldung aus, dass bei einer materialisierten Tabelle keine Volltextsuche durchgeführt werden kann. (Fehler #68751, Fehler #16539903)

## Aurora-MySQL-Datenbank-Engine-Updates 15.05.2017 (Version 1.13) (veraltet)

Version: 1.13

### Note

Wir haben die neue Funktion `SELECT INTO OUTFILE S3` in Aurora MySQL 1.13 nach der Erstversion aktiviert und die Versionshinweise entsprechend aktualisiert.

Aurora MySQL 1.13 ist allgemein verfügbar. Alle neuen Datenbank-Cluster, einschließlich der aus Snapshots wiederhergestellten, werden in Aurora MySQL v1.1.13 erstellt. Für bestehende Datenbank-Cluster können Sie ein Upgrade auf Aurora MySQL 1.13 ausführen (nicht verpflichtend). Mit Aurora-Version 1.13 verwenden wir ein Cluster-Patching-Modell, bei dem alle Knoten in einem Aurora-DB-Cluster gleichzeitig gepatcht werden. Wir aktivieren das Zero-Downtime-Patching, das auf Best-Effort-Basis funktioniert, um Client-Verbindungen vor dem Patch-Vorgang zu bewahren. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Zero-Downtime-Patching (Patchen ohne Ausfallzeiten)

Beim Feature des Patchens ohne Ausfallzeiten (ZDP – Zero-Downtime Patching) wird versucht, Client-Verbindungen auf Best-Effort-Basis vor dem Patch-Vorgang zu bewahren. Weitere Informationen finden Sie unter [Verwendung des Zero-Downtime-Patchings](#) im Amazon-Aurora-Benutzerhandbuch.

## Neue Funktionen:

- **SELECT INTO OUTFILE S3** – Aurora MySQL ermöglicht Ihnen ab sofort das Hochladen von Abfrageergebnissen in eine oder mehrere Dateien in einem Amazon S3-Bucket. Weitere Informationen finden Sie unter [Speichern von Daten aus einem Amazon-Aurora-MySQL-DB-Cluster in Textdateien in einem Amazon-S3-Bucket](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen:

- Implementierte Kürzung von Protokolldateien im CSV-Format beim Engine-Startup, um lange Wiederherstellungszeiten zu vermeiden. Die Tabellen `general_log_backup`, `general_log`, `slow_log_backup` und `slow_log` werden nun nach dem Neustart einer Datenbank verworfen.
- Behebung eines Fehlers, der zum Fehlschlagen der Migration einer Datenbank mit dem Namen **test** führte.
- Verbesserte Stabilität des Garbage Collectors im Lock Manager durch Verwendung der korrekten Sperrsegmente.
- Verbesserte Stabilität des Lock Managers durch Entfernen von ungültigen Aussagen im Deadlock-Detection-Algorithmus.
- Reaktivierte asynchrone Replikation und behobener Fehler, der dazu führte, dass ein falscher Wert für die Replica-Verzögerung für Workloads ohne Ladevorgänge oder schreibgeschützte Workloads ausgegeben wurde. Die Verbesserungen der Replikationspipeline, die mit Version 1.10 eingeführt wurden. Diese Verbesserungen wurden eingeführt, damit Protokoll-Stream-Updates im Buffer-Cache eines Aurora-Replicas angewendet werden konnten. Dies hilft, die Leseleistung und die Stabilität von Aurora-Replicas zu verbessern.
- Behebung eines Fehlers, der dazu geführt hat, dass durch die Einstellung "autocommit=OFF" geplante Ereignisse blockiert und lange Transaktionen offen gehalten wurden, bis der Server neu gestartet wurde.
- Behebung eines Fehlers, der dazu geführt hat, dass allgemeine, Audit- und Slow-Query-Protokolle keine Abfragen protokollieren konnten, die durch einen asynchronen Commit gehandhabt wurden.
- Verbesserung der Leistung der LRA-Funktion (logisches Read-Ahead) auf das bis zu 2,5-Fache. Dies wurde erzielt, indem Vorabrufen erlaubt wurde, über zwischenliegenden Seiten in einem B-Baum weiterhin ausgeführt zu werden.
- Hinzugefügte Parametervalidierung für Audit-Variablen, um unnötigen Raum zu kürzen.

- Regression behoben, eingeführt mit Aurora MySQL-Version 1.11, bei der Abfragen unter Verwendung der SQL\_CALC\_FOUND\_ROWS-Funktion und Aufrufen der FOUND-ROWS()-Funktion falsche Ergebnisse zurückgeben könnten.
- Behebung eines Stabilitätsproblems, wenn die Sperrliste mit Metadaten falsch zusammengestellt wurde.
- Verbesserte Stabilität, wenn sql\_mode auf PAD\_CHAR\_TO\_FULL\_LENGTH festgelegt ist und der Befehl SHOW FUNCTION STATUS WHERE Db='string' ausgeführt wird.
- Behebung eines seltenen Falls, wenn Instances nach einem Upgrade der Aurora-Version nicht mehr erscheinen würden, weil eine falsche Überprüfung auf die Konsistenz des Volumes durchgeführt wurde.
- Ein Leistungsproblem wurde behoben, eingeführt in Aurora MySQL-Version 1.12, bei dem die Leistung eines Aurora-Writers reduziert war, wenn Benutzer über eine große Anzahl an Tabellen verfügten.
- Verbesserte Stabilität, wenn der Aurora-Schreiber als Binärprotokoll-Worker konfiguriert ist und die Anzahl der Verbindungsversuche sich 16.000 nähert.
- Ein seltener Fehler wurde behoben, bei dem ein Aurora-Replica neu starten konnte, wenn eine Verbindung beim Warten auf die Metadaten-Sperre geblockt wurde, wenn DDL auf dem Aurora-Master ausgeführt wurde.

## Integration von MySQL-Fehlerbehebungen

- Mit einer leeren InnoDB-Tabelle ist es nicht möglich, den "auto\_increment"-Wert mithilfe des ALTER TABLE-Statements zu verkleinern, auch wenn die Tabelle leer ist. (Fehler #69882)
- MATCH() ... AGAINST-Abfragen, die lange Zeichenfolgen als ein Argument für AGAINST() verwenden, könnten zu einem Fehler führen, wenn sie in einer InnoDB-Tabelle mit einem Volltext-Suchindex ausgeführt werden. (Fehler #17640261)
- Handhabung von SQL\_CALC\_FOUND\_ROWS in Kombination mit ORDER BY und LIMIT könnte zu falschen Ergebnissen für FOUND\_ROWS() führen. (Fehler #68458, Fehler # 16383173)
- ALTER TABLE erlaubt keine Änderung der Löscharkeit einer Spalte, wenn ein auswärtiger Schlüssel vorhanden ist. (Fehler #77591)

# Aurora-MySQL-Datenbank-Engine-Updates 05.04.2017 (Version 1.12) (veraltet)

Version: 1.12

Aurora MySQL 1.12 ist aktuell die bevorzugte Version für das Erstellen neuer DB-Cluster, einschließlich Wiederherstellungen aus Snapshots.

Dies ist kein verpflichtendes Upgrade für bestehende Cluster. Sie werden die Möglichkeit erhalten, bestehende Cluster auf Version 1.12 upzugraden, nachdem wir den umfassenden Patch auf 1.11 fertiggestellt haben (siehe Aurora 1.11-[Versionshinweise](#) und entsprechende [Ankündigungen im Forum](#)). Mit Aurora-Version 1.12 verwenden wir ein Cluster-Patching-Modell, bei dem alle Knoten in einem Aurora-DB-Cluster gleichzeitig gepatcht werden. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

## Neue Features

- Schnelle DDL – Mit Aurora MySQL können Sie nun die Operation ALTER TABLE tbl\_name ADD COLUMN col\_name column\_definition nahezu in Echtzeit ausführen. Die Operation wird abgeschlossen, ohne dass ein Kopieren der Tabelle erforderlich wäre und ohne eine materielle Auswirkung auf andere DML-Statements zu haben. Da kein temporärer Speicher für eine Tabellenkopie benötigt wird, haben DDL-Statements praktische Vorteile, sogar für große Tabellen auf kleinen Instance-Typen. Fast DDL wird zurzeit nur für das Hinzufügen löschbarer Spalten ohne Standardwert am Ende einer Tabelle unterstützt. Diese Funktion ist aktuell im Aurora-Labor-Modus verfügbar. Weitere Informationen finden Sie unter [Ändern von Tabellen in Amazon Aurora mithilfe von schneller DDL](#) im Amazon-Aurora-Benutzerhandbuch.
- Anzeigen des Volume-Status – Wir haben den neuen Überwachungsbefehl SHOW VOLUME STATUS hinzugefügt, um die Anzahl der Knoten und Festplatten im Volume anzeigen zu lassen. Weitere Informationen finden Sie unter [Anzeigen des Volume-Status für einen Aurora-MySQL-DB-Cluster](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

- Eingeführte Änderungen für das Sperren der Komprimierung, um den zugeteilten Arbeitsspeicher pro Sperrobjekt weiter zu reduzieren. Diese Verbesserung ist im Labor-Modus verfügbar.
- Behebung eines Fehlers, der dazu geführt hat, dass in der `trx_active_transactions`-Metrik eine schnelle Wertabnahme stattfindet, auch wenn sich die Datenbank im Leerlauf befindet.

- Behebung einer ungültigen Fehlermeldung im Zusammenhang mit der Syntax der Fehlersimulationsabfrage, wenn Fehler in Festplatten und Knoten simuliert werden.
- Behebung mehrerer Fehler im Zusammenhang mit Race Conditions und Dead Latches im Lock Manager.
- Behebung eines Fehlers, der dazu geführt hat, dass ein Buffer-Overflow im Abfragen-Optimierer stattfindet.
- Behebung eines Stabilitätsproblems in Aurora Read Replicas, wenn die zugrunde liegenden Speicherknoten wenig verfügbaren Arbeitsspeicher aufweisen.
- Behebung eines Fehlers, der dazu geführt hat, dass inaktive Verbindungen über die Parametereinstellung von `wait_timeout` hinausgingen.
- Behebung eines Fehlers, der dazu geführt hat, dass `query_cache_size` einen unerwarteten Wert zurückgibt, nachdem die Instance neu gestartet wurde.
- Behebung eines Leistungsproblems, das auftritt, wenn Schreibvorgänge nicht an den Speicher übergeben werden und daher das Netzwerk zu häufig durch einen diagnostischen Thread überprüft wird.

## Integration von MySQL-Fehlerbehebungen

- Erneutes Laden einer Tabelle, die während eines Leerlaufs exmittiert wurde, führte zum Zurücksetzen des `AUTO_INCREMENT`-Werts. (Fehler #21454472, Fehler #77743)
- Ein Index-Datensatz wurde beim Rollback aufgrund von Unstimmigkeiten in der `purge_node_t`-Struktur nicht gefunden. Die Unstimmigkeiten führten zu Warn- und Fehlermeldungen wie „Fehler im sec index entry-Update“, „kann einen Datensatz nicht bereinigen“ und „versuchte, einen nicht zur Löschung markierten sec-Indexeintrag zu bereinigen“. (Fehler #19138298, Fehler #70214, Fehler #21126772, Fehler #21065746)
- Falsche Berechnung der Stack-Größe für `qsort`-Operation führt zu Stack-Overflow. (Fehler #73979)
- Datensatz nach Rollback in keinem Index gefunden. (Fehler #70214, Fehler #72419)
- `ALTER TABLE` fügt Spalte `TIMESTAMP` beim Update hinzu, `CURRENT_TIMESTAMP` fügt ZERO-Daten ein (Fehler #17392)



# Aurora-MySQL-Datenbank-Engine-Updates 23.02.2017 (Version 1.11) (veraltet)

Version: 1.11

Wir werden alle Aurora MySQL-DB-Cluster kurz nach der Veröffentlichung mit der neusten Version patchen. DB-Cluster werden mithilfe der Legacy-Prozedur mit einer Ausfallszeit von 5 bis 30 Sekunden gepatcht.

Das Patchen findet während des System-Wartungszeitraums statt, den Sie für jede Ihrer Datenbank-Instances festgelegt haben. Sie können dieses Fenster mithilfe der ansehen oder ändern AWS Management Console. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

Alternativ können Sie den Patch sofort installieren, AWS Management Console indem Sie einen DB-Cluster auswählen, Cluster-Aktionen und dann Jetzt aktualisieren auswählen.

Mit Aurora MySQL-Version 1.11 verwenden wir ein Cluster-Patching-Modell, bei dem alle Knoten in einem Aurora-DB-Cluster gleichzeitig gepatcht werden.

## Neue Funktionen

- MANIFEST-Option für LOAD DATA FROM S3 – LOAD DATA FROM S3 wurde mit Version 1.8 veröffentlicht. Die Optionen für diesen Befehl wurden erweitert und Sie können nun eine Dateiliste festlegen, die mithilfe einer Manifestdatei in ein Aurora-DB-Cluster aus Amazon S3 geladen werden kann. Dies macht es einfacher, Daten aus bestimmten Dateien in einen der mehrere Speicherorte zu laden, in Gegensatz zum Laden von Daten aus einer einzelnen Datei mithilfe der FILE-Option oder zum Laden von Daten aus mehreren Dateien mithilfe der PREFIX-Option, die sich im selben Ort befinden und dasselbe Präfix haben. Das Format der Manifestdatei ist dasselbe wie das von Amazon Redshift verwendete. Weitere Informationen über die Verwendung von LOAD DATA FROM S3 mit der MANIFEST-Option finden Sie unter [Verwenden eines Manifests zum Angeben von zu ladenden Datendateien](#) im Amazon-Aurora-Benutzerhandbuch.
- Räumliche Indizierung standardmäßig aktiviert – Diese Funktion wurde im Labor-Modus in Version 1.10 veröffentlicht und ist jetzt standardmäßig aktiviert. Räumliche Indizierung verbessert die Abfrageleistung in großen Datensätzen für Abfragen, die räumliche Daten verwenden. Weitere Informationen zur Verwendung der räumlichen Indizierung finden Sie unter [Amazon Aurora MySQL und räumliche Daten](#) im Amazon-Aurora-Benutzerhandbuch.

- Änderung der Zeitplanung für erweitertes Auditing – Diese Funktion wurde mit Version 1.10.1 veröffentlicht, um einen Hochleistungsbetrieb für die Auditierung Ihrer Datenbankaktivitäten zu ermöglichen. In dieser Veröffentlichung wurde die Präzision von Protokoll-Zeitstempeln von einer Sekunde in eine Mikrosekunde geändert. Die präziseren Zeitstempel ermöglichen Ihnen bessere Informationen über den Zeitpunkt eines Audit-Ereignisses. Weitere Informationen zu Prüfungen finden Sie unter [Verwenden von Advanced Auditing mit einem Amazon-Aurora-MySQL-DB-Cluster](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

- Der Parameter `thread_handling` wurde geändert, um zu verhindern, dass Sie eine andere Einstellung als **multiple-connections-per-thread** verwenden, die das einzige unterstützte Modell im Aurora-Thread-Pool ist.
- Behebung eines Fehlers, der auftrat, wenn Sie entweder den Parameter `buffer_pool_size` oder den Parameter `query_cache_size` höher als den gesamten Arbeitsspeicher des DB-Clusters festlegten. In diesem Fall legt Aurora den geänderten Parameter auf den Standardwert fest, damit das DB-Cluster hochfahren kann und nicht abstürzt.
- Behebung eines Fehlers im Abfrage-Cache, der dazu geführt hat, dass die Transaktion veraltete Ergebnisse von Lesevorgängen erhält, wenn die Tabelle in einer anderen Transaktion ungültig ist.
- Behebung eines Fehlers, der dazu geführt hat, dass Binärprotokolldateien kurzzeitig verspätet anstatt gleich entfernt werden.
- Behebung eines Fehlers, bei dem eine Datenbank, die mit dem Namen **tmp** erstellt wird, wie eine Systemdatenbank behandelt wird, die im flüchtigen Speicher gespeichert und nicht im verteilten Speicher von Aurora behalten wird.
- Änderung des Verhaltens von SHOW TABLES, um bestimmte systeminterne Tabellen auszuschließen. Diese Änderung hilft, ein unnötiges Failover zu vermeiden, das durch eine Sperrung aller in SHOW TABLES gelisteten Dateien durch mysqldump verursacht wurde, wodurch wiederum Schreibvorgänge in der systeminternen Tabelle verhindert werden und es zum Failover kommt.
- Behebung eines Fehlers, bei dem ein Aurora Replica falsch neu startet, wenn eine temporäre Tabelle aus einer Abfrage erstellt wird, die eine Funktion aufruft, deren Argument eine Spalte in einer InnoDB-Tabelle ist.
- Ein Fehler im Zusammenhang mit einem Sperrkonflikt von Metadaten in einem Aurora-Replica-Knoten wurde behoben, der dazu führt, dass das Aurora-Replica hinter das primäre DB-Cluster zurückfällt und evtl. neu gestartet wird.

- Behebung eines Dead Latch in der Replikations-Pipeline in den Reader-Knoten, der dazu geführt hat, dass ein Aurora Replica in Rückstand gerät und evtl. neu gestartet wird.
- Behebung eines Fehlers, bei dem ein Aurora Replica mit verschlüsselten Volumes, die größer als 1 Terabyte (TB) sind, zu sehr verzögert wird.
- Verbesserung der Dead Latch-Erkennung in einem Aurora Replica, indem eine verbesserte Methode zum Lesen der Systemsperrzeit angewandt wird.
- Behebung eines Fehlers, bei dem ein Aurora Replica zweimal anstatt einmal neu gestartet wird, worauf eine Abmeldung durch den Schreiber erfolgt.
- Behebung eines Fehlers bei der Slow-Query-Leistung in Aurora Replicas, der auftritt, wenn kurzzeitige Statistiken zu Unstimmigkeiten in den Statistiken in uneindeutigen Indexspalten führen.
- Ein Fehler wurde behoben, bei dem ein Aurora Replica ausfallen kann, falls in dem Aurora-Replica ein DDL-Statement repliziert wird, wenn das Aurora-Replica gleichzeitig eine zugehörige Abfrage verarbeitet.
- Änderung der Verbesserungen in der Replikations-Pipeline, die mit Version 1.10 eingeführt wurden, von standardmäßig aktiviert auf standardmäßig deaktiviert. Diese Verbesserungen wurden eingeführt, um Protokoll-Stream-Updates im Buffer-Cache eines Aurora-Replicas anzuwenden. Obwohl diese Funktion hilft, die Leseleistung und Stabilität von Aurora-Replicas zu verbessern, erhöht sie bei einigen Workloads die Replica-Verzögerung.
- Behebung eines Fehlers, bei dem das gleichzeitige Auftreten eines fortlaufenden DDL-Statements und eines ausstehenden parallelen Read-Ahead in derselben Tabelle einen Assertionfehler während der Übergabephase der DDL-Transaktion verursacht.
- Weiterentwicklung des allgemeinen Protokolls und des Slow-Query-Protokolls, um den Neustart eines DB-Clusters zu "überleben".
- Es wurde ein out-of-memory Problem bei bestimmten Abfragen mit langer Laufzeit behoben, indem der Speicherverbrauch im ACL-Modul reduziert wurde.
- Behebung eines Neustart-Problems, das auftritt, wenn eine Tabelle über nicht räumliche Indizes verfügt, räumliche Aussagen in der Abfrage vorliegen, der Planer beschließt, einen nicht räumlichen Index zu wählen, und fälschlicherweise die räumliche Bedingung an den Index weitergibt.
- Behebung eines Problems, bei dem das DB-Cluster neu startet, wenn ein Lösch-, Update- oder Bereinigungsvorgang in sehr großen raumbezogenen Objekten, die extern gespeichert sind (z. B. LOBs), stattgefunden hat.
- Es wurde ein Problem behoben, bei dem die Fehlersimulation mit ALTER SYSTEM SIMULATE ... FOR INTERVAL nicht richtig funktionierte.

- Behebung eines Stabilitätsproblems, das von einer ungültigen Assertion in einer falschen Invariante im Lock Manager ausgelöst wird.
- Deaktivierung der folgenden zwei Verbesserungen in der InnoDB-Volltextsuche, die mit Version 1.10 eingeführt wurden, da diese Stabilitätsprobleme für einige anspruchsvolle Workloads verursachen:
  - Update der Cache-Only-After-A-Read-Anfrage an ein Aurora Replica, damit die Schnelligkeit der Replikation für den Cache des Volltextsuchindex verbessert wird.
  - Abladen des Cache-Sync-Vorgangs in einem separaten Thread, sobald die Größe des Cache 10 % der Gesamtgröße überschreitet, um zu vermeiden, dass MySQL-Abfragen zu lange während des FTS-Cache-Sync-Cache-to-Disk-Vorgangs hinausgezögert werden. (Fehler #22516559, #73816).

## Integration von MySQL-Fehlerbehebungen

- Das gleichzeitige Ausführen von ALTER-Tabelle und Fremdschlüssel-DROP mit einer anderen DROP-Operation führt dazu, dass die Tabelle verschwindet. (Fehler #16095573)
- Einige INFORMATION SCHEMA-Abfragen die ORDER BY verwendet haben, haben nicht wie zuvor Filesort-Optimierung verwendet. (Fehler #16423536)
- FOUND\_ROWS () gibt die falsche Zählung von Zeilen in einer Tabelle zurück. (Fehler #68458)
- Der Server fällt aus, anstatt einen Fehler zurückzugeben, wenn zu viele temp-Tabellen geöffnet sind. (Fehler #18948649)

## Aurora-MySQL-Datenbank-Engine-Updates 12.01.2017 (Version 1.10.1) (veraltet)

Version: 1.10.1

Aurora MySQL 1.10.1 ist eine Opt-In-Version und wird nicht für das Patchen Ihrer Datenbank-Instances verwendet. Sie ist für das Erstellen von neuen Aurora-Instances und für das Upgraden von bestehenden Instances verfügbar. Sie können den Patch anwenden, indem Sie in der [Amazon RDS-Konsole](#) ein Cluster auswählen, Cluster Actions (Cluster-Aktionen) und anschließend Upgrade Now (Jetzt aktualisieren) auswählen. Das Patchen erfordert einen Neustart der Datenbank mit einer Ausfallzeit von normalerweise 5 bis 30 Sekunden, nach der Sie mit der Verwendung Ihrer DB-Cluster fortfahren können. Dieser Patch verwendet ein Cluster-Patching-Modell, bei dem alle Knoten in einem Aurora-Cluster zur selben Zeit gepatcht werden.

## Neue Funktionen

- **Erweitertes Auditing** – Aurora MySQL bietet eine Hochleistungsfunktion für erweitertes Auditing, die Sie für das Kontrollieren von Datenbankaktivitäten verwenden können. Weitere Informationen zur Aktivierung und Nutzung von Advanced Auditing finden Sie unter [Verwenden von Advanced Auditing mit einem Amazon-Aurora-MySQL-DB-Cluster](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

- Behebung eines Fehlers mit räumlicher Indizierung beim Erstellen einer Spalte und Hinzufügen eines Index im selben Statement.
- Behebung eines Fehlers, bei dem räumliche Statistiken während eines Neustarts eines DB-Clusters nicht beständig sind.

## Aurora-MySQL-Datenbank-Engine-Updates 14.12.2016 (Version 1.10) (veraltet)

Version: 1.10

## Neue Funktionen

- **Zero-Downtime-Patch** – Mit dieser Funktion kann eine DB-Instance ohne Auftreten einer Ausfallzeit gepatcht werden. Datenbank-Upgrades werden nämlich durchgeführt, ohne dass Client-Anwendungen getrennt oder die Datenbank neu gestartet wird. Diese Methode erhöht die Verfügbarkeit Ihres Aurora-DB-Clusters während des Wartungszeitraums. Beachten Sie, dass temporäre Daten, wie die im Leistungsschema, während eines Upgrade-Vorgangs zurückgesetzt werden. Diese Funktion gilt für von einem Service zugestellte Patches während des Wartungszeitraums sowie für Patches, die durch einen Benutzer ausgeführt wurden.

Wenn ein Patch gestartet wird, stellt der Service sicher, dass es keine offenen Sperrungen, Transaktionen oder temporären Tabellen gibt, und wartet auf einen passenden Zeitraum, in dem die Datenbank gepatcht und neu gestartet werden kann. Anwendungssitzungen werden bewahrt, obwohl der Durchsatz reduziert wird, während der Patch installiert wird (für ca. 5 Sekunden). Wenn kein passender Zeitraum gefunden werden kann, werden Patches auf das Standardverhalten zurückgesetzt.

Zero-Downtime-Patching läuft auf Best-Effort-Basis mit einigen Einschränkungen ab, wie im Folgenden beschrieben:

- Diese Funktion ist zurzeit für das Patchen eines Single-Knoten-DB-Clusters oder von Schreiber-Instances in Multi-Knoten-DB-Clustern anwendbar.
- SSL-Verbindungen werden in Zusammenhang mit dieser Funktion nicht unterstützt. Wenn aktive SSL-Verbindungen bestehen, wird Amazon Aurora MySQL keinen Zero-Downtime-Patch ausführen und stattdessen in regelmäßigen Abständen nachprüfen, ob die SSL-Verbindungen beendet wurden. Wenn dies der Fall ist, wird das Zero-Downtime-Patching fortgesetzt. Wenn die SSL-Verbindungen nach mehr als einige Sekunden fortbestehen, wird der Standard-Patching-Vorgang mit Ausfallzeit fortgesetzt.
- Diese Funktion ist ab Aurora-Version 1.10 und neuer verfügbar. Im weiteren Verlauf werden wir alle Versionen und Patches nennen, mit denen das Zero-Downtime-Patching nicht verwendet werden kann.
- Diese Funktion ist nicht anwendbar, wenn eine auf Binärprotokollierung basierende Replikation aktiviert ist.
- Räumliche Indizierung – Räumliche Indizierung verbessert die Abfrageleistung in großen Datensätzen für Abfragen, die räumliche Daten verwenden. Weitere Informationen zur Verwendung der räumlichen Indizierung finden Sie unter [Amazon Aurora MySQL und räumliche Daten](#) im Amazon-Aurora-Benutzerhandbuch.

Diese Funktion ist standardmäßig deaktiviert und kann aktiviert werden, indem der Aurora-Labor-Modus aktiviert wird. Weitere Informationen finden Sie unter [Amazon-Aurora-MySQL-Labor-Modus](#) im Amazon-Aurora-Benutzerhandbuch.

- Verbesserungen in der Replikations-Pipeline – Aurora MySQL verwendet jetzt einen verbesserten Mechanismus zur Anwendung von Protokoll-Stream-Updates im Buffer-Cache in einem Aurora Replica. Diese Funktion verbessert die Leseleistung und Stabilität in Aurora Replicas, wenn schreibintensive Workloads in der Haupt-Instance und signifikante leseintensive Workloads im Replica verarbeitet werden. Diese Funktion ist standardmäßig aktiviert.
- Verbesserung des Durchsatzes für Workloads mit gecachten Lesevorgängen – Aurora MySQL verwendet jetzt einen Algorithmus mit riegelloser und simultaner Funktion, um Leseansichten einzubauen, was zu einem besseren Durchsatz von Leseabfragen führt, die von Buffer-Cache bedient werden. Als Ergebnis dieser und anderer Verbesserungen kann Amazon Aurora MySQL einen Durchsatz von bis zu 625.000 Lesevorgängen pro Sekunde erreichen, verglichen mit 164 000 Lesevorgängen pro Sekunde bei MySQL 5.7 für einen SysBench Select-Only-Workload.

- Verbesserung des Durchsatzes für Workloads mit Hot Row Contention – Aurora MySQL verwendet einen neuen Algorithmus mit der Funktion für Veröffentlichungssperrung, der die Leistung verbessert, vor allem wenn Hot Page Contention vorhanden ist (Transaktionen konkurrieren nämlich um die Zeilen auf derselben Seite). Bei Tests mit dem TPC-C-Benchmark kann dies im Vergleich zu MySQL 5.7 einen um bis zu 16-fach besseren Durchsatz in Transaktionen pro Minute bedeuten. Diese Funktion ist standardmäßig deaktiviert und kann aktiviert werden, indem der Aurora-Labor-Modus aktiviert wird. Weitere Informationen finden Sie unter [Amazon-Aurora-MySQL-Labor-Modus](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

- Die Schnelligkeit der Replikation des Volltextsuchindex wurde durch ein Update der Cache-Only-After-A-Read-Anfrage in einem Aurora Replica verbessert. Diese Methode vermeidet jegliche Lesevorgänge der Festplatte durch den Replikations-Thread.
- Behebung eines Problems, bei dem die Außerkraftsetzung des Wörterbuch-Caches in einem Aurora Replica für Tabellen, die ein Sonderzeichen im Datenbank- oder im Tabellennamen enthalten, nicht funktioniert.
- Behebung eines STUCK I/O-Fehlers während des Datenmigrationsvorgangs für verteilte Speicherknoten, wenn Speicher-Wärmemanagement aktiviert ist.
- Behebung eines Fehlers im Lock Manager, bei dem die Überprüfung einer Assertion für den Thread für das Abwarten einer Transaktionssperre fehlschlägt, wenn eine Rollback- oder Commit-Transaktion vorbereitet wird.
- Behebung eines Fehlers beim Öffnen einer beschädigten Tabelle durch das richtige Aktualisieren der Referenzzählung in den Wörterbuchtabelleneinträgen.
- Behebung eines Fehlers, durch den der minimale Lesezeitpunkt des DB-Clusters von langsamen Aurora Replicas beibehalten werden kann.
- Behebung eines potenziellen Speicherlecks im Abfrage-Cache.
- Behebung eines Fehlers, bei dem ein Aurora Replica eine Zeilenebenen-Sperre für eine Tabelle verhängt, wenn eine Abfrage in einem IF-Statement einer gespeicherten Prozedur verwendet wird.

## Integration von MySQL-Fehlerbehebungen

- UNION von abgeleiteten Tabellen gibt die falschen Ergebnisse mit "1=0/false"-Klauseln zurück. (Fehler #69471)

- Serverausfälle in ITEM\_FUNC\_GROUP\_CONCAT::- Vermeidet ein Hinauszögern von MySQL-Abfragen für zu lange andauernde FTS-Cache-Sync-To-Disk-Vorgänge durch das Abladen des Cache-Sync-Vorgangs in einem separaten Thread, sobald die Größe des Cache 10 % der Gesamtgröße überschreitet. (Fehler #22516559, #73816)

## Aurora-MySQL-Datenbank-Engine-Updates 10.11.2016 (Versionen 1.9.0, 1.9.1) (veraltet)

Version: 1.9.0, 1.9.1

### Neue Funktionen

- Verbesserter Index-Build – Die Implementierung für das Aufbauen von sekundären Indizes funktioniert jetzt durch das Aufbauen des Index in einem Bottom-up-Ansatz (von unten nach oben), was eine unnötige Spaltung von Seiten verhindert. Dies kann den erforderlichen Zeitaufwand für das Erstellen eines Index oder Wiederaufbauen einer Tabelle um bis zu 75 % reduzieren (basierend auf der db.r3.8xlarge-DB-Instance-Klasse). Diese Funktion befand sich in Aurora MySQL-Version 1.7 im Labor-Modus und ist jetzt standardmäßig aktiviert in Aurora-Version 1.9 und höher. Weitere Informationen finden Sie unter [Amazon-Aurora-MySQL-Labor-Modus](#) im Amazon-Aurora-Benutzerhandbuch.
- Sperren-Kompression (Labor-Modus) – Diese Anwendung reduziert signifikant die Menge an verbrauchtem Arbeitsspeicher durch Lock Manager um bis zu 66 %. Der Sperrmanager kann mehr Zeilensperren erwerben, ohne dass eine Ausnahme auftritt. out-of-memory Diese Funktion ist standardmäßig deaktiviert und kann aktiviert werden, indem der Aurora-Labor-Modus aktiviert wird. Weitere Informationen finden Sie unter [Amazon-Aurora-MySQL-Labor-Modus](#) im Amazon-Aurora-Benutzerhandbuch.
- Performance-Schema – Aurora MySQL bietet jetzt Unterstützung für Performance-Schema mit minimaler Auswirkung auf die Performance. In unseren Tests konnte die Aktivierung des Leistungsschemas die MySQL-Leistung um bis zu 60% beeinträchtigen. SysBench

SysBench Tests eines Aurora-DB-Clusters ergaben eine viermal geringere Auswirkung auf die Leistung als bei MySQL. Das Ausführen der db.r3.8xlarge-DB-Instance-Klasse ergab 100.000 SQL-Schreibvorgänge/Sek. und sogar über 550.000 SQL-Schreibvorgänge/Sek., wenn das Leistungsschema aktiviert war.



- Verbesserung von Hot Row Contention – Diese Funktion reduziert die CPU-Verwendung und erhöht den Durchsatz, wenn eine große Anzahl an Verbindungen auf eine kleine Anzahl an Hot Rows zugreift. Diese Funktion behebt auch `error 188`, wenn Hot Row Contention besteht.
- Verbesserte out-of-memory Handhabung — Wenn unwichtige, sperrende SQL-Anweisungen ausgeführt werden und der reservierte Speicherpool verletzt wird, erzwingt Aurora ein Rollback dieser SQL-Anweisungen. Diese Funktion gibt Speicherplatz frei und verhindert Engine-Abstürze aufgrund von Ausnahmen. `out-of-memory`
- Intelligenter Leseselektor — Diese Implementierung verbessert die Leselatenz, indem für jeden Lesevorgang das optimale Speichersegment aus verschiedenen Segmenten ausgewählt wird, was zu einem verbesserten Lesedurchsatz führt. SysBench Tests haben eine Leistungssteigerung von bis zu 27% bei Schreib-Workloads ergeben.

## Verbesserungen

- Behebung eines Fehlers, bei dem in einem Aurora Replica eine Lesesperre während des Hochfahrens der Engine auftritt.
- Behebung eines potenziellen Absturzes eines Aurora Replicas, wenn der Adressenverweis für die Leseansicht im Bereinigungssystem NULL ist.

## Aurora-MySQL-Datenbank-Engine-Updates 26.10.2016 (Version 1.8.1) (veraltet)

Version: 1.8.1

## Verbesserungen

- Behebung eines Fehlers, bei dem Bulk-Inserts, die Auslöser verwenden, die AWS Lambda - Prozeduren aufrufen, fehlschlagen.
- Behebung eines Problems, bei dem die Migration von Katalogen fehlschlägt, wenn Autocommit global deaktiviert ist.
- Es wurde ein Verbindungsfehler zu Aurora bei der Verwendung von SSL behoben und die Diffie-Hellman-Gruppe zur Abwehr von Angriffen verbessert. LogJam

## Integration von MySQL-Fehlerbehebungen

- OpenSSL hat aufgrund des Problems die Diffie-Hellman-Schlüssellängen-Parameter geändert. LogJam (Fehler #18367167)

## Aurora-MySQL-Datenbank-Engine-Updates 18.10.2016 (Version 1.8) (veraltet)

Version: 1.8

### Neue Features

- AWS Lambda Integration — Sie können jetzt mithilfe des Verfahrens asynchron eine AWS Lambda Funktion aus einem Aurora-DB-Cluster aufrufen. `mysql.lambda_async` Weitere Informationen finden Sie unter Aufrufen einer [Lambda-Funktion aus einem Amazon-Aurora-MySQL-DB-Cluster](#) im Amazon-Aurora-Benutzerhandbuch.
- Laden von Daten aus Amazon S3 – Sie können jetzt mithilfe der Befehle `LOAD DATA FROM S3` oder `LOAD XML FROM S3` Text- oder XML-Dateien aus einem Amazon S3-Bucket in Ihr Aurora-DB-Cluster laden. Weitere Informationen finden Sie unter [Laden von Daten in einen Amazon-Aurora-MySQL-DB-Cluster aus Textdateien in einem Amazon-S3-Bucket](#) im Amazon-Aurora-Benutzerhandbuch.
- Katalog Migration – Aurora behält jetzt Metadaten von Katalogen im Cluster-Volume bei, um Unterstützung für Versioning zu bieten. Dies ermöglicht eine nahtlose Migration von Katalogen über verschiedene Versionen und Wiederherstellungen hinweg.
- Wartung und Patching auf Cluster-Level – Aurora verwaltet jetzt Wartungsupdates für ein ganzes DB-Cluster. Weitere Informationen finden Sie unter [Verwalten eines Amazon-Aurora-DB-Clusters](#) im Amazon-Aurora-Benutzerhandbuch.

### Verbesserungen

- Behebung eines Problems, bei dem ein Aurora-Cluster abstürzt, wenn einer Inflight-DDL-Tabelle eine Metadaten Sperre nicht gewährt wird.
- Erlaubte Aurora Replicas, Nicht-InnoDB-Tabellen zu ändern, um die Rotation von langsamen und allgemeinen Protokoll-CSV-Dateien mit zu erleichtern `log_output=TABLE`.

- Behebung einer Verzögerung, wenn ein Update von Statistiken aus der primären Instance in ein Aurora Replica durchgeführt wurde. Ohne diese Behebung können die Statistiken des Aurora Replicas die Synchronisation mit den Statistiken in der primären Instance verlieren, was zu einem unterschiedlichen (und möglicherweise leistungsschwachen) Abfrageplan in einem Aurora Replica führt.
- Behebung einer Race-Bedingung, die sicherstellt, dass ein Aurora Replica keine Sperren annimmt.
- Behebung eines seltenen Szenarios, bei dem ein Aurora Replica, das an der primären Instance registriert oder abgemeldet wird, fehlschlagen könnte.
- Behebung einer Race-Bedingung, die zu einem Deadlock in `db.r3.large`-Instances führen könnte, wenn ein Volume geöffnet oder geschlossen wird.
- Es wurde ein out-of-memory Problem behoben, das aufgrund einer Kombination aus einer großen Schreiblast und Ausfällen im Aurora Distributed Storage-Dienst auftreten kann.
- Behebung eines Fehlers mit hohem CPU-Verbrauch aufgrund der Zirkulation des Bereinigungs-Threads bei einer lang laufenden Transaktion.
- Behebung eines Problems, das beim Ausführen von Informationsschema-Abfragen für den Erhalt von Informationen über Sperren unter hoher Workload auftritt.
- Behebung eines Problems mit einem Diagnostik-Vorgang, der in seltenen Fällen dazu führen kann, dass Aurora-Schreibvorgänge auf Speicherknoten verzögert werden und ein Neustart/Failover durchgeführt wird.
- Behebung einer Bedingung, unter der eine erfolgreich erstellte Tabelle während einer Ausfall-Wiederherstellung gelöscht werden konnte, wenn der Ausfall stattfand, während eine `CREATE TABLE [if not exists]`-Anweisung verarbeitet wurde.
- Behebung eines Falls, bei dem die Protokoll-Rotationsprozedur unterbrochen ist, wenn ein allgemeines Protokoll oder ein Slow-Protokoll mithilfe von Katalog-Migration nicht auf der Festplatte gespeichert wurde.
- Behebung eines Ausfalls, der stattfindet, wenn ein Benutzer eine temporäre Tabelle innerhalb einer benutzerdefinierten Funktion erstellt und anschließend die benutzerdefinierte Funktion in einer von der Abfrage ausgewählten Liste verwendet.
- Behebung eines Ausfalls, der auftritt, wenn GTID-Ereignisse wiedergegeben werden. GTID wird von Aurora MySQL nicht unterstützt.

## Integration von MySQL-Fehlerbehebungen:

- Wenn alle Indexe in einer Spalte mit mehreren Indexen verworfen wurden, konnte InnoDB eine DROP INDEX-Operation nicht blockieren, wenn eine Fremdschlüsselbeschränkung einen Index erfordert. (Fehler #16896810)
- Lösung des Ausfalls beim Hinzufügen von Fremdschlüsselbeschränkung. (Fehler #16413976)
- Behebung eines Ausfalls, wenn gleichzeitig ein Cursor in einer gespeicherten Prozedur abgerufen und eine Analyse oder Bereinigung der Tabelle durchgeführt wird. (Fehler #18158639)
- Behebung eines Fehlers bei der automatischen Zählung, wenn ein Benutzer eine Tabelle verändert, um den AUTO\_INCREMENT-Wert in weniger als den maximalen Spaltenwert für automatische Zählung zu ändern. (Fehler #16310273)

## Aurora-MySQL-Datenbank-Engine-Updates 20.09.2016 (Version 1.7.1) (veraltet)

Version: 1.7.1

### Verbesserungen

- Behebt einen Fehler, bei dem ein Aurora Replica abstürzt, wenn der Cache der InnoDB-Volltextsuche voll ist.
- Behebung eines Problems, bei dem die Datenbank-Engine abstürzt, wenn ein Worker-Thread im Thread-Pool auf sich selbst wartet.
- Behebung eines Problems, bei dem ein Aurora Replica abstürzt, wenn eine Metadaten-Sperre in einer Tabelle einen Deadlock verursacht.
- Behebung eines Problems, bei dem die Datenbank-Engine aufgrund einer Race-Bedingung zwischen zwei Worker-Threads im Thread-Pool abstürzt.
- Behebung eines Problems, bei dem ein unnötiges Failover unter hoher Workload auftritt, wenn der Überwachungsagent keinen Fortschritt der Schreib-Operationen im Untersystem des verteilten Speichers feststellen kann.

# Aurora-MySQL-Datenbank-Engine-Updates 30.08.2016 (Version 1.7.0) (veraltet)

Version: 1.7.0

## Neue Funktionen

- NUMA-fähiger Scheduler – Der Task-Scheduler für die Aurora MySQL-Engine ist jetzt NUMA-fähig (Non-Uniform Memory Access). Dies minimiert CPU-Socket-übergreifende Konflikte, was einen verbesserten Leistungsdurchsatz für die `db.r3.8xlarge`-DB-Instance-Klasse bedeutet.
- Paralleles Read-Ahead operiert asynchron im Hintergrund – Paralleles Read-Ahead wurde überarbeitet, um die Leistung zu verbessern, indem ein dedizierter Thread verwendet wird, um die Thread-Konkurrenz zu reduzieren.
- Verbesserter Index-Build (Labor-Modus) – Die Implementierung für das Aufbauen von sekundären Indizes wird nun von unten nach oben ausgeführt, was unnötige Aufteilungen von Seiten verhindert. Dies kann den erforderlichen Zeitaufwand für das Erstellen eines Index oder das Wiederaufbauen einer Tabelle reduzieren. Diese Funktion ist standardmäßig deaktiviert und kann aktiviert werden, indem der Aurora-Labor-Modus aktiviert wird. Weitere Informationen finden Sie unter [Amazon-Aurora-MySQL-Labor-Modus](#) im Amazon-Aurora-Benutzerhandbuch.

## Verbesserungen

- Behebung eines Fehlers, bei dem das Herstellen einer Verbindung sehr lange dauerte, wenn es einen Anstieg an Verbindungsanfragen für eine Instance gab.
- Behebung eines Problems, bei dem ein Absturz auftrat, wenn ALTER TABLE in einer partitionierten Tabelle ausgeführt wurde, in der InnoDB nicht verwendet wurde.
- Behebung eines Fehlers, bei dem von hohen Workloads mit Schreibvorgängen ein Failover ausgelöst werden konnte.
- Behebung einer fehlerhaften Assertion, die einen Ausfall ausgelöst hat, wenn RENAME TABLE auf einer partitionierten Tabelle ausgeführt wurde.
- Verbesserung der Stabilität beim Rollback-Vorgang in einer Transaktion während hoher Workload mit Insert-Vorgängen.
- Behebung eines Problems, bei der Volltextsuchindexe in einem Aurora Replica nicht umsetzbar waren.

## Integration von MySQL-Fehlerbehebungen

- Verbesserte Skalierbarkeit durch Partitionierung der LOCK\_grant-Sperre. (Port WL #8355)
- Das Öffnen des Cursors mit SELECT in einer gespeicherten Prozedur verursacht segfault. (Port-Fehler#16499751)
- MySQL gibt das falsche Ergebnis mit einer ungewöhnlichen Nutzung aus. (Fehler #11751794)
- Ausfall in GET\_SEL\_ARG\_FOR\_KEYPART – verursacht durch Patch für Fehler #11751794. (Fehler #16208709)
- Falsche Ergebnisse für eine einfache Abfrage mit GROUP\_BY. (Fehler #17909656)
- Zusätzliche Zeilen für semijoin-Abfragen mit Bereichseigenschaften. (Fehler #16221623)
- Hinzufügen einer ORDER BY-Klausel gefolgt von einer IN-Unterabfrage könnte dazu führen, dass Duplikatzeilen zurückgegeben werden. (Fehler #16308085)
- Ausfall mit der Erklärung, dass eine Abfrage mit ungenauem Scan für GROUP BY, MyISAM vorliegt. (Fehler #16222245)
- Ungenauer Index-Scan mit Ganzzahl-Prädikat in Anführungszeichen gibt zufällige Daten zurück. (Fehler #16394084)
- Wenn der Optimierer einen ungenauen Index-Scan verwendet hat, konnte der Server während des Versuchs, eine temporäre Tabelle zu erstellen, beendet werden. (Fehler #16436567)
- COUNT(DISTINCT) sollte keine NULL-Werte zählen. Diese wurden jedoch gezählt, als der Optimierer ungenauen Index-Scan verwendet hatte. (Fehler #17222452)
- Wenn eine Abfrage sowohl MIN()/MAX() als auch aggregate\_function(DISTINCT) (z. B. SUM(DISTINCT)) aufwies und mithilfe von ungenauem Index-Scan ausgeführt wurde, wurden die Ergebniswerte von MIN()/MAX() falsch festgelegt. (Fehler #17217128)

## Aurora-MySQL-Datenbank-Engine-Updates 01.06.2016 (Version 1.6.5) (veraltet)

Version: 1.6.5

### Neue Funktionen

- Effizientes Speichern von Binärprotokollen – Effizientes Speichern von Binärprotokollen ist jetzt standardmäßig für alle Aurora MySQL-DB-Cluster aktiviert und nicht konfigurierbar. Effizienter Speicher von Binärprotokollen wurde mit dem Update von April 2016 eingeführt. Weitere

Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 06.04.2016 \(Version 1.6\) \(veraltet\)](#).

## Verbesserungen

- Verbesserte Stabilität für Aurora Replicas, wenn eine primäre Instance einem hohen Workload unterliegt.
- Verbesserte Stabilität für Aurora Replicas beim Ausführen von Abfragen in partitionierten Tabellen und Tabellen mit Sonderzeichen im Tabellennamen.
- Behebung von Verbindungsproblemen, wenn sichere Verbindungen genutzt werden.

## Integration von MySQL-Fehlerbehebungen

- SLAVE CAN'T CONTINUE REPLICATION AFTER MASTER'S CRASH RECOVERY (Port Bug #17632285)

## Aurora-MySQL-Datenbank-Engine-Updates 06.04.2016 (Version 1.6) (veraltet)

Version: 1.6

Dieses Update enthält die folgenden Verbesserungen:

### Neue Funktionen

- Paralleles Read-Ahead – Paralleles Read-Ahead ist jetzt standardmäßig für alle Aurora MySQL-DB-Cluster aktiviert und nicht konfigurierbar. Paralleles Read-Ahead wurde mit dem Update von Dezember 2015 eingeführt. Weitere Informationen finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 03.12.2015 \(Version 1.4\) \(veraltet\)](#).

Zusätzlich zur standardmäßigen Aktivierung von parallelem Read-Ahead sind in dieser Version die folgenden Verbesserungen für paralleles Read-Ahead enthalten:

- Verbesserte Logik, sodass paralleles Read-Ahead weniger aggressiv ist. Dies ist vorteilhaft, wenn Ihr DB-Cluster vielen parallelen Workloads unterliegt.
- Verbesserte Stabilität in kleineren Tabellen.

- Effizienter Speicher von Binärprotokollen (Labor-Modus) – MySQL-Binärprotokolldateien werden jetzt noch effizienter in Aurora MySQL gespeichert. Die neue Speicherimplementierung ermöglicht eine viel schnellere Löschung von Binärprotokolldateien und verbessert die System-Performance für eine Instance in einem Aurora MySQL-DB-Cluster, das ein Binärprotokoll-Replikationsmaster ist.

Legen Sie den Parameter `aurora_lab_mode` in der Parametergruppe für Ihre primäre Instance oder Aurora Replica auf 1 fest, um effizienten Speicher von Binärprotokollen zu aktivieren. Der Parameter `aurora_lab_mode` ist ein Parameter der Instance-Ebene, der sich standardmäßig in der Parametergruppe `default.aurora5.6` befindet. Informationen zum Ändern einer DB-Parametergruppe finden Sie unter [Ändern von Parametern in einer DB-Parametergruppe](#) im Amazon-Aurora-Benutzerhandbuch. Weitere Informationen zu Parametergruppen und Aurora MySQL finden Sie unter [Aurora-MySQL-Konfigurationsparameter](#) im Amazon-Aurora-Benutzerhandbuch.

Aktivieren Sie effizienten Speicher von Binärprotokollen nur für Instances in einem Aurora MySQL-DB-Cluster, die MySQL-Binärprotokoll-Replikationsmaster-Instances sind.

- `AURORA_VERSION`-Systemvariable – Sie können jetzt die Aurora-Version Ihres Aurora MySQL-DB-Clusters abrufen, indem Sie die Systemvariable `AURORA_VERSION` abfragen.

Verwenden Sie die folgenden Abfragen, um die Aurora-Version zu erhalten:

```
select AURORA_VERSION();
select @@aurora_version;
show variables like '%version';
```

Sie können die Aurora-Version auch sehen, AWS Management Console wenn Sie einen DB-Cluster ändern, oder indem Sie den [describe-db-engine-versions](#) AWS CLI Befehl oder die [EngineVersionsDescribeDB-API-Operation](#) aufrufen.

- Metrik der Arbeitsspeicherverwendung von Lock Manager – Informationen über den Arbeitsspeicherverbrauch von Lock Manager sind jetzt als Metrik verfügbar.

Verwenden Sie die folgenden Abfragen, um die Metrik der Arbeitsspeicherverwendung von Lock Manager zu erhalten:

```
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' );
```

## Verbesserungen

- Verbesserte Stabilität während der Wiederherstellung von Binärprotokollen und XA-Transaktionen.
- Behebung eines Arbeitsspeicherproblems, das durch eine hohe Anzahl an Verbindungen ausgelöst wurde.
- Verbesserte Genauigkeit der folgenden Metriken: Read Throughput, Read IOPS, Read Latency, Write Throughput, Write IOPS, Write Latency und Disk Queue Depth.
- Behebung eines Stabilitätsproblems, das den Start großer Instances nach einem Ausfall verlangsamt.
- Verbesserte Nebenläufigkeit im Datenwörterbuch in Hinsicht auf Synchronisationsmechanismen und Cache-Bereinigung.
- Stabilitäts- und Leistungsverbesserungen für Aurora Replicas:
  - Behebung eines Stabilitätsproblems für Aurora Replicas während hoher oder Burst-Workloads mit Schreibvorgängen für die primäre Instance.
  - Nachgebesserte Replica-Verzögerung für db.r3.4xlarge- und db.r3.8xlarge-Instances.
  - Verbesserte Leistung durch Reduzierung von Konflikten zwischen der Anwendung von Protokolldatensätzen und nebenläufigen Lesevorgängen in einem Aurora Replica.
  - Behebung eines Fehlers beim Aktualisieren von Statistiken in Aurora Replicas für neu erstellte oder aktualisierte Statistiken.
  - Verbesserte Stabilität für Aurora Replicas, wenn zu viele Transaktionen für die primäre Instance und nebenläufige Lesevorgänge in den Aurora Replicas für dieselben Daten ausgeführt werden.
  - Verbesserte Stabilität für Aurora Replicas beim Ausführen der Statements UPDATE und DELETE mit JOIN-Statements.
  - Verbesserte Stabilität für Aurora Replicas beim Ausführen von INSERT ... SELECT-Statements.

## Integration von MySQL-Fehlerbehebungen

- BACKPORT-Fehler 18694052 FIX FOR ASSERTION '!M\_ORDERED\_REC\_BUFFER' FAILED TO 5.6 (Port-Fehler 18305270)

- SEGV IN MEMCPY(), HA\_PARTITION::POSITION (Port-Fehler # 18383840)
- FALSCHER ERGEBNISSE BEI PARTITIONIERUNG, INDEX\_MERGE UND KEIN PK (Port-Fehler # 18167648)
- TABELLEN FÜR EXPORT BEREINIGEN: ASSERTION IN HA\_PARTITION::EXTRA (Port-Fehler # 16943907)
- SERVER-AUSFALL IN VIRTUELLEM HA\_ROWS-HANDLER::MULTI\_RANGE\_READ\_INFO\_CONST (Port-Fehler # 16164031)
- BEREICHSOPTIMIERER STÜRZT IN SEL\_ARG::RB\_INSERT() AB (Port-Fehler # 16241773)

## Aurora-MySQL-Datenbank-Engine-Updates 11.01.2016 (Version 1.5) (veraltet)

Version: 1.5

Dieses Update enthält die folgenden Verbesserungen:

### Verbesserungen

- Behebung einer 10-sekündigen Pause von Schreiboperationen für Instances im Leerlauf während Bereitstellungen von Aurora-Speicher.
- Logisches Read-Ahead funktioniert jetzt, wenn `innodb_file_per_table` auf No festgelegt ist. Weitere Informationen zu logischem Read-Ahead finden Sie unter [Aurora-MySQL-Datenbank-Engine-Updates 03.12.2015 \(Version 1.4\) \(veraltet\)](#).
- Behebung von Problemen mit Aurora Replicas, die sich erneut mit der primären Instance verbinden. Diese Verbesserung behebt auch ein Problem bei der Angabe eines großen Wertes für den Parameter `quantity` beim Testen eines Aurora Replicas auf Fehler mithilfe von Fehlersimulationsabfragen. Weitere Informationen finden Sie unter [Testen eines Aurora-Replikat-Ausfalls](#) im Amazon-Aurora-Benutzerhandbuch.
- Verbesserte Überwachung von in Rückstand geratenen oder neu startenden Aurora Replicas.
- Behebung eines Problems, das dazu führte, dass ein Aurora Replica, verzögert, abgemeldet und anschließend neu gestartet wurde.
- Behebung eines Problems, wenn der Befehl `show innodb status` während eines Deadlocks ausgeführt wird.
- Behebung eines Problems mit Failover für größere Instances während eines hohen Durchsatzes an Schreibvorgängen.

## Integration von MySQL-Fehlerbehebungen

- Behebung einer unvollständigen Lösung in der MySQL-Volltextsuche, die sich auf Tabellen auswirkt, bei denen der Datenbankname mit einer Ziffer beginnt. (Port Fehler #17607956)

## Aurora-MySQL-Datenbank-Engine-Updates 03.12.2015 (Version 1.4) (veraltet)

Version: 1.4

Dieses Update enthält die folgenden Verbesserungen:

### Neue Funktionen

- Fast Insert – beschleunigte parallele Inserts (Einsätze) geordnet nach primärem Schlüssel. Weitere Informationen finden Sie unter [Amazon-Aurora-MySQL-Leistungsverbesserungen](#) im Amazon-Aurora-Benutzerhandbuch.
- Performance von Lesevorgängen für große Datensätze – Aurora MySQL erkennt eine I/O-lastige Workload automatisch und startet mehr Threads, um die Performance für das DB-Cluster zu erhöhen. Der Aurora-Scheduler sieht in der I/O-Aktivität nach und führt eine dynamische Anpassung der optimalen Anzahl der Threads im System durch, mit einer schnellen Anpassung zwischen I/O-lastigen und CPU-lastigen Workloads mit geringem Overhead.
- Paralleles Read-Ahead – verbessert die Leistung von B-Baum-Scans, die zu groß für den verfügbaren Arbeitsspeicher Ihrer primären Instance oder Ihres Aurora Replicas sind (einschließlich Bereichsabfragen). Paralleles Read-Ahead erkennt automatisch Muster von Seitenlesevorgängen und lädt Seiten vor ihrem eigentlichen Gebrauch in den Buffer-Cache. Paralleles Read-Ahead funktioniert mit mehreren Tabellen zur selben Zeit innerhalb derselben Transaktion.

### Verbesserungen:

- Behebung von kurzzeitigen Verfügbarkeitsproblemen mit Aurora-Datenbanken während Bereitstellungen von Aurora-Speicher.
- Korrektes Erzwingen des `max_connection`-Limits.
- Verbesserung der Binärprotokollbereinigung, wenn Aurora der Binärprotokollmaster ist und die Datenbank nach einer hohen Workload neu startet.

- Behebung von Verwaltungsproblemen mit dem Arbeitsspeicher im Tabellen-Cache.
- Hinzugefügte Unterstützung für große Seiten im Shared-Memory-Buffer-Cache für schnellere Wiederherstellung.
- Behebung eines Problems, bei dem der lokale Speicher eines Threads nicht initialisiert wird.
- 16.000 Verbindungen sind standardmäßig erlaubt.
- Dynamischer Thread-Pool für I/O-lastige Workloads.
- Behebung eines Problems mit gründlichem Annullieren von Ansichten, die UNION im Abfrage-Cache einbeziehen.
- Behebung eines Stabilitätsproblems mit dem Thread für Wörterbuchstatistiken.
- Behebung eines Lecks im Arbeitsspeicher im Untersystem des Wörterbuchs, das im Zusammenhang mit der Cache-Bereinigung besteht.
- Behebung des Problems mit hoher Latenz von Schreibvorgängen in Aurora Replicas, wenn nur eine sehr geringe Workload mit Schreibvorgängen im Master vorhanden ist.
- Stabilitätsprobleme in Aurora Replicas beim Ausführen von Operationen in DDL-partitionierten Tabellen, wie ALTER TABLE ... wurden behoben. REORGANIZE PARTITION im Master.
- Behebung von Stabilitätsproblemen in Aurora Replicas während des Wachstums von Volumes.
- Behebung eines Problems für Scans in nicht gruppierten Indexen in Aurora Replicas.
- Behobenes Stabilitätsproblem, das dazu führt, dass Aurora Replicas verzögert, evtl. abgemeldet und neu gestartet werden.

## Integration von MySQL-Fehlerbehebungen

- SEGV in FTSPARSE(). (Fehler #16446108)
- InnoDB-Datenwörterbuch wird während der Umbenennung der Spalte nicht aktualisiert. (Fehler #19465984)
- FTS-Ausfall nach der Umbenennung einer Tabelle für eine andere Datenbank. (Fehler #16834860)
- Fehlgeschlagene Vorbereitung eines Auslösers für trunkierte Tabellen verursacht Fehler 1054. (Fehler #18596756)
- Änderungen der Metadaten könnten zu Problemen bei der Ausführung von Auslösern führen. (Fehler #18684393)
- Materialisierung wird nicht für das long-Feld UTF8 VARCHAR ausgewählt. (Fehler #17566396)
- Schlechter Ausführungsplan, wenn ORDER BY mit Limit X. (Fehler #16697792)

- Backport-Fehler #11765744 IN 5.1, 5.5 UND 5.6. (Fehler #17083851)
- Mutex-Problem in SQL/SQL\_SHOW.CC führt zu SIG6. Quelle vermutlich FILL\_VARIABLES. (Fehler #20788853)
- Backport-Fehler #18008907 in Versionen 5.5+. (Fehler #18903155)
- Adaptionfix für einen Stack-Overflow-Fehler in MySQL 5.7. (Fehler #19678930)

## Aurora-MySQL-Datenbank-Engine-Updates 16.10.2015 (Versionen 1.2, 1.3) (veraltet)

Versionen: 1.2, 1.3

Dieses Update enthält die folgenden Verbesserungen:

### Behobene Probleme

- Das out-of-memory Problem im neuen Lock-Manager mit lang andauernden Transaktionen wurde behoben
- Gelöste Sicherheitsschwachstelle beim Replizieren mit Nicht-RDS for MySQL-Datenbanken
- Aktualisiert, um sicherzustellen, dass Quorum-Schreibvorgänge korrekte Wiederholungsversuche bei Speicherfehlern ausführen
- Aktualisiert, um Read Replica-Verzögerung präziser zu melden
- Verbesserte Leistung durch Reduzierung von Konflikten, wenn viele nebenläufige Transaktionen versuchen, dieselbe Zeile zu ändern
- Behobene Annullierung des Abfrage-Cache für Ansichten, die durch die Verknüpfung zweier Tabellen erstellt werden
- Deaktivierter Abfrage-Cache für Transaktionen mit UNCOMMITTED\_READ-Isolation

### Verbesserungen

- Bessere Leistung für Slow-Catalog-Abfragen in "warmen" Caches
- Verbesserte Nebenläufigkeit in Wörterbuchstatistiken
- Bessere Stabilität für den neuen Abfrage-Cache-Ressourcenmanager, erweiterte Verwaltung, Dateien gespeichert im Amazon Aurora Smart-Speicher und Batch-Schreibvorgänge von Protokolldatensätzen

## Integration von MySQL-Fehlerbehebungen

- Beenden einer Abfrage innerhalb innodb verursacht einen Ausfall mit einer Assertion. (Fehler #1608883)
- Bei Fehlschlägen beim Erstellen eines neuen Threads für den Ereignis-Scheduler, die Ereignis-Ausführung oder neue Verbindung, wurde keine Mitteilung in das Fehlerprotokoll geschrieben. (Fehler #16865959)
- Wenn eine Verbindung ihre standardmäßige Datenbank ändert und eine andere Verbindung simultan SHOW PROCESSLIST ausführt, könnte die zweite Verbindung auf ungültigen Arbeitsspeicher zugreifen, wenn sie versucht, den standardmäßigen Datenbankarbeitsspeicher der ersten Verbindung anzuzeigen. (Fehler #11765252)
- PURGE BINARY LOGS entfernt standardmäßig keine Binärprotokolldateien, die verwendet werden oder aktiv sind, gab aber keinen Hinweis, wenn dies stattfand. (Fehler #13727933)
- Bei einigen Statements könnte es zu Lecks im Arbeitsspeicher kommen, wenn der Optimierer unbenötigte Unterabfrage-Klassen entfernt hat. (Fehler #15875919)
- Während des Herunterfahrens könnte der Server versuchen, einen nicht initialisierten Mutex zu sperren. (Fehler #16016493)
- Ein vorbereitetes Statement, das GROUP CONCAT() und eine ORDER BY-Klausel verwendet hat, das mehrere Spalten benannt hat, könnte dazu führen, dass der Server beendet wird. (Fehler #16075310)
- Leistungsschema-Instrumentierung fehlte bei Replikat-Worker-Threads. (Fehler #16083949)
- STOP SLAVE kann einen Deadlock verursachen, wenn sie gleichzeitig mit einer Anweisung wie SHOW STATUS ausgegeben wird, die die Werte für eine oder mehrere der Statusvariablen Slave\_retried\_transactions, Slave\_heartbeat\_period, Slave\_received\_heartbeats, Slave\_last\_heartbeat oder Slave\_running abgerufen hat. (Fehler #16088188)
- Eine vollständige Abfrage mithilfe des booleschen Modus konnte in einigen Fällen keine Ergebnisse liefern, wenn der Suchbegriff in Anführungszeichen stand. (Fehler #16206253)
- Der Versuch des Optimierers, redundante Unterabfrage-Klauseln zu entfernen, erhöhte eine Assertion, wenn ein vorbereitetes Statement mit einer Unterabfrage in der On-Klausel einer Verbindung in einer Unterabfrage ausgeführt wurde. (Fehler #16318585)
- GROUP\_CONCAT instabil, Ausfall in ITEM\_SUM::CLEAN\_UP\_AFTER\_REMOVAL. (Fehler #16347450)

- Der Versuch, die standardmäßige InnoDB-Volltextsuchen-Stoppwortliste (FTS) zu ersetzen, indem eine InnoDB-Tabelle mit derselben Struktur wie INFORMATION\_SCHEMA.INNODB\_FT\_DEFAULT\_STOPWORD erstellt wird, endete in einem Fehler. (Fehler #16373868)
- Nachdem der Client-Thread auf einem Worker ein FLUSH TABLES WITH READ LOCK ausgeführt hat und darauf einige Updates auf dem Master folgten, stürzte der Worker ab, wenn ausgeführt wurde SHOW SLAVE STATUS. (Fehler #16387720)
- Beim Parsen einer beschränkten Suchzeichenfolge wie "abc-def" in einer Volltextsuche verwendet InnoDB jetzt dieselben Trennzeichen wie MyISAM. (Fehler #16419661)
- Ausfall in FTS\_AST\_TERM\_SET\_WILDCARD. (Fehler #16429306)
- SEGFAULT in FTS\_AST\_VISIT() for FTS RQG Test. (Fehler #16435855)
- Wenn bei Debug-Builds der Optimierer ein Item\_ref entfernte, das auf eine Unterabfrage verwies, führte das zum Beenden des Servers. (Fehler #16509874)
- Volltextsuche in InnoDB-Tabellen schlug fehl bei der Suche nach wörtlichen Sätzen in Kombination mit den Operatoren + oder -. (Fehler #16516193)
- START SLAVE schlug fehl, als der Server mit den Optionen -- master-info-repository =TABLE relay-log-info-repository =TABLE gestartet wurde und Autocommit auf 0 gesetzt war, zusammen mit. -- skip-slave-start (Fehler #16533802)
- Sehr große InnoDB-Volltextsuchergebnisse (FTS) konnten in einigen Fällen eine übermäßige Menge an Arbeitsspeicher verbrauchen. (Fehler #16625973)
- Bei Debug-Builds konnte in OPT\_CHECK\_ORDER\_BY eine Assertion auftreten, wenn ein Binärprogramm direkt in der Suchzeichenfolge verwendet wurde, da Binärprogramme NULL-Bytes und andere bedeutungslose Zeichen enthalten können. (Fehler #16766016)
- Bei einigen Statements könnte es zu Lecks im Arbeitsspeicher kommen, wenn der Optimierer unbenötigte Unterabfrage-Klassen entfernt hat. (Fehler #16807641)
- Es war möglich, einen Deadlock zu verursachen, nachdem FLUSH TABLES WITH READ LOCK durch Verwendung von STOP SLAVE in einer neuen Verbindung an den Worker erteilt wurde und anschließend SHOW SLAVE STATUS über die ursprüngliche Verbindung erteilt wurde. (Fehler #16856735)
- GROUP\_CONCAT() mit einem ungültigen Trennzeichen konnte zum Beenden des Servers führen. (Fehler #16870783)
- Der Server führte übermäßiges Sperren für die Mutexe LOCK\_active\_mi und active\_mi->rli->data\_lock jeder SHOW STATUS LIKE 'pattern'-Anweisung aus, selbst wenn das Muster nicht mit den Statusvariablen übereinstimmte, die diese Mutexe verwenden (Slave\_heartbeat\_period,

- Slave\_last\_heartbeat, Slave\_received\_heartbeats, Slave\_retried\_transactions, Slave\_running). (Fehler #16904035)
- Eine Volltextsuche, die den IN BOOLEAN MODE-Modifikator verwendet, konnte in einem Assertionfehler enden. (Fehler #16927092)
  - Volltextsuche in InnoDB-Tabellen schlug fehl bei Suchen, die den booleschen Operator + verwendet haben. (Fehler #17280122)
  - 4-Wege-Deadlock: Zombies, Binärprotokolle bereinigen, Prozesslisten anzeigen, Binärprotokolle anzeigen. (Fehler #17283409)
  - Wenn ein auf ein Commit-Lock wartender SQL-Thread beendet und neu gestartet wurde, führte das zum Überspringen einer Transaktion auf dem Worker. (Fehler #17450876)
  - Ein InnoDB-Volltextsuchen-Fehler konnte aufgrund eines "unbeendeten" Tokens auftreten. Die Zeichenfolge und die Zeichenfolgenlänge sollten für einen Zeichenfolgenvergleich weitergeleitet werden. (Fehler #17659310)
  - Eine große Anzahl an partitionierten InnoDB-Tabellen konnte, wenn diese in MySQL 5.6 oder 5.7 verwendet wurden, viel mehr Arbeitsspeicher verbrauchen als dieselben Tabellen in vorherigen Versionen von MySQL Server. (Fehler #17780517)
  - Bei Volltextabfragen konnte ein Fehler beim Überprüfen, ob num\_token kleiner ist als max\_proximity\_item, in einer Assertion enden. (Fehler #18233051)
  - Bestimmte Abfragen für die INFORMATION SCHEMA TABLES- und COLUMNS-Tabellen konnten zu übermäßigem Arbeitsspeicherverbrauch führen, wenn es eine große Anzahl an leeren InnoDB-Tabellen gab. (Fehler #18592390)
  - Beim Übergeben einer Transaktion wird jetzt eine Kennzeichnung verwendet, um zu überprüfen, ob ein Thread erstellt wurde, anstatt den Thread selbst zu überprüfen, was mehr Ressourcen benötigte, besonders wenn der Server mit master\_info\_repository=TABLE ausgeführt wird. (Fehler #18684222)
  - Wenn ein Client-Thread in einem Worker FLUSH TABLES WITH READ LOCK ausgeführt hat, während der Master eine DML ausführte, wurde das Ausführen von SHOW SLAVE STATUS auf demselben Client blockiert, was zu einem Deadlock führte. (Fehler #19843808)
  - Das Ordnen nach einem GROUP\_CONCAT() -Ergebnis konnte zum Beenden des Servers führen. (Fehler #19880368)



# Aurora-MySQL-Datenbank-Engine-Updates 24.08.2015 (Version 1.1) (veraltet)

Version: 1.1

Dieses Update enthält die folgenden Verbesserungen:

- Verbesserungen der Replikationsstabilität beim Replizieren mit einer MySQL-Datenbank (Binärprotokoll-Replikation). Weitere Informationen über die Aurora-MySQL-Replikation mit MySQL finden Sie unter [Replikation mit Amazon Aurora](#) im Amazon-Aurora-Benutzerhandbuch.
- Eine 1-Gigabyte (GB)-Beschränkung für die Größe von Relay-Protokollen, die für ein Aurora MySQL-DB-Cluster akkumuliert werden, bei dem es sich um einen Replikationsworker handelt. Dies verbessert die Dateiverwaltung für die Aurora-DB-Cluster.
- Stabilitätsverbesserungen in den Bereichen Read-Ahead, Beziehung von rekursiven Fremdschlüsseln und Aurora-Replikation.
- Integration von MySQL-Fehlerbehebungen.
  - InnoDB-Datenbanken mit Namen, die mit einer Zahl beginnen, verursachen einen Fehler im Volltextsuchparser (FTS). (Fehler #17607956)
  - InnoDB-Volltextsuchen schlagen in Datenbanken fehl, deren Namen mit einer Zahl beginnen. (Fehler #17161372)
  - Für InnoDB-Datenbanken in Windows, ist die Objekt-ID für Volltextsuchen (FTS) nicht im erwarteten Hexadezimalformat. (Fehler #16559254)
  - Eine in MySQL 5.6 eingeführte Code-Regression hatte negative Auswirkungen auf die Leistung von DROP TABLE und ALTER TABLE. Dies führt möglicherweise zu einer Leistungsminderung zwischen MySQL Server 5.5.x und 5.6.x. (Fehler #16864741)
- Vereinfachte Protokollierung, um die Größe der Protokolldateien und die Menge des dafür erforderlichen Speicherplatzes zu reduzieren.

# MySQL-Fehlerbehebungen durch Aurora MySQL-Datenbank-Engine-Updates

Die folgenden Abschnitte geben MySQL-Fehler an, die durch Aurora MySQL-Datenbank-Engine-Updates behoben wurden.

## Themen

- [MySQL-Fehlerbehebungen durch Aurora-MySQL-3.x-Datenbank-Engine-Updates](#)
- [MySQL-Fehlerbehebungen durch Aurora MySQL 2.x-Datenbank-Engine-Updates](#)
- [MySQL-Fehlerbehebungen durch Aurora MySQL 1.x-Datenbank-Engine-Updates](#)

## MySQL-Fehlerbehebungen durch Aurora-MySQL-3.x-Datenbank-Engine-Updates

Die mit MySQL 8.0 kompatible Aurora-Version enthält alle MySQL-Fehlerbehebungen bis zu ihrer entsprechenden MySQL-Kompatibilitätsversion. In der folgenden Tabelle werden zusätzliche MySQL-Fehler bezeichnet, die durch Aurora MySQL-Datenbank-Engine-Updates behoben wurden, und es wird angegeben, mit welchem Update sie behoben wurden.

Datenbank-Engine-Update	MySQL-kompatible Version	Version	Behobene MySQL-Fehler
<a href="#">Updates der Aurora MySQL-Datenbank-Engine 2024-06-04 (Version 3.07.0,</a>	<a href="#">8,0,36</a>	3.07.0	<ul style="list-style-type: none"> <li>• Es wurde ein Problem behoben, bei dem der Cache-Zeilwert falsch berechnet werden konnte, was zu einem Fehler beim Datenbankneustart auf Graviton-basierten Instances führte. (Community-Bugfix #35479763)</li> <li>• Es wurde ein Problem behoben, bei dem einige Instanzen von Unterabfragen in gespeicherten Routinen nicht immer korrekt</li> </ul>

Datenbank-Engine-Update	MySQL-kompatible Version	Version	Behobene MySQL-Fehler
<a href="#">kompatibel mit MySQL 8.0.36)</a>			<p>behandelt wurden. (Community-Bugfix #35377192)</p> <ul style="list-style-type: none"> <li>• Es wurde ein Problem behoben, das aufgrund der Rotation von TLS-Zertifikaten im Hintergrund zu einer höheren CPU-Auslastung führen kann (Community Bug Fix #34284186).</li> <li>• Es wurde ein Problem behoben, bei dem InnoDB das Hinzufügen von INSTANT Spalten zu Tabellen im MySQL-Schemaschema in Aurora MySQL-Versionen vor 3.05 erlaubte, was dazu führen konnte, dass der Server nach dem Upgrade auf Aurora MySQL Version 3.05.0 unerwartet geschlossen wurde (Datenbankinstanz neu gestartet). (Community-Bugfix #35625510).</li> </ul>

Datenbank-Engine-Update	MySQL-kompatible Version	Version	Behobene MySQL-Fehler
<a href="#">Updates der Aurora MySQL-Datenbank-Engine 2024-03-07 (Version 3.06.0, kompatibel mit MySQL 8.0.34)</a>	<a href="#">8.0.34</a>	3.06.0	<ul style="list-style-type: none"> <li>• Es wurde ein Problem behoben, bei dem der Cache-Zeilenwert falsch berechnet werden konnte, was zu einem Fehler beim Neustart der Datenbank auf einer Graviton-Instance führte. (Community-Bugfix #35479763)</li> <li>• Es wurde ein Problem behoben, bei dem einige Instanzen von Unterabfragen in gespeicherten Routinen nicht immer korrekt behandelt wurden. (Community-Bugfix #35377192)</li> <li>• Es wurde ein Problem behoben, das aufgrund der Rotation von TLS-Zertifikaten im Hintergrund zu einer höheren CPU-Auslastung führen kann. (Community-Fehlerbehebung #34284186)</li> <li>• Es wurde ein Problem behoben, bei dem InnoDB das Hinzufügen von INSTANT Spalten zu Tabellen im MySQL-Systemschemata in Aurora MySQL-Versionen vor 3.05 erlaubte, was dazu führen konnte, dass der Server nach dem Upgrade auf Aurora MySQL Version 3.05.0 unerwartet geschlossen wurde (Datenbankinstanz neu</li> </ul>

Datenbank-Engine-Update	MySQL-kompatible Version	Version	Behobene MySQL-Fehler
			gestartet). (Community-Fehlerbehebung #35625510)
<a href="#">Aurora MySQL-Datenbank-Engine-Updates 2024-01-31 (Version 3.05.2, kompatibel mit MySQL 8.0.32) Standard</a>	<a href="#">8.0.32</a>	3.05.2	<ul style="list-style-type: none"> <li>Wiederholtes Ausführen einer gespeicherten Routine, bei der als Unterabfrage eine SELECT-Anweisung mit mehreren AND-, oder XOR Bedingungen verwendet wurde, führte zu übermäßigem Verbrauch und möglicherweise schließlich zur Erschöpfung des virtuellen Speichers. (Community-Bugfix #33852530)</li> </ul>

Datenbank-Engine-Update	MySQL-kompatible Version	Version	Behobene MySQL-Fehler
<a href="#">Updates der Aurora MySQL-Datenbank-Engine</a> <a href="#">2023-11-21</a> (Version <a href="#">3.05.1</a> , kompatibel mit <a href="#">MySQL 8.0.32</a> )	<a href="#">8.0.32</a>	3.05.1	<ul style="list-style-type: none"> <li>• Es wurde ein Problem in InnoDB behoben, das zwischen den Aurora MySQL-Versionen 3.01 bis Aurora MySQL-Versionen 3.04 und nach dem Upgrade von Aurora MySQL auf Version 3.05.0 bei DMLs für MySQL-Tabellen, denen in einem Systemschema eine INSTANT_ADD-Spalte hinzugefügt wird, zu einem unerwarteten Schließen des Servers führen kann. (Community-Fehlerbehebung #35625510)</li> </ul>
<a href="#">Aurora MySQL-Datenbank-Engine-Updates</a> <a href="#">25.10.2023</a> (Version <a href="#">3.05.0</a> , kompatibel mit <a href="#">MySQL 8.0.32</a> )	<a href="#">8,0,32</a>	3.05.0	<ul style="list-style-type: none"> <li>• Es wurde ein Problem behoben, das aufgrund der Rotation von TLS-Zertifikaten im Hintergrund zu einer höheren CPU-Auslastung führen kann (Community-Fehlerbehebung #34284186)</li> </ul>

Datenbank-Engine-Update	MySQL-kompatible Version	Version	Behobene MySQL-Fehler
<a href="#">Updates der Aurora MySQL-Datenbank-Engine 2024-03-15 (Version 3.04.2, kompatibel mit MySQL 8.0.28)</a>	<a href="#">8,0,28</a>	3.04,2	<ul style="list-style-type: none"> <li>• Es wurde ein Problem behoben, bei dem der Cache-Zeilenwert falsch berechnet werden konnte, was zu einem Fehler beim Datenbankneustart auf Graviton-basierten Instances führte. (Community-Bugfix #35479763)</li> <li>• Wiederholtes Ausführen einer gespeicherten Routine, bei der als Unterabfrage eine SELECT-Anweisung verwendet wurde, die mehrere AND, OR oder XOR-Bedingungen enthielt, führte zu übermäßigem Verbrauch und möglicherweise schließlich zur Erschöpfung des virtuellen Speichers. (Community-Bugfix #33852530)</li> </ul>

Datenbank-Engine-Update	MySQL-kompatible Version	Version	Behobene MySQL-Fehler
<a href="#">Updates der Aurora MySQL-Datenbank-Engine 2023-11-13 (Version 3.04.1, kompatibel mit MySQL 8.0.28)</a>	<a href="#">8.0.28</a>	3,04,1	<ul style="list-style-type: none"><li>• Es wurde ein Problem behoben, das aufgrund der Rotation von TLS-Zertifikaten im Hintergrund zu einer höheren CPU-Auslastung führen kann (Community-Fehlerbehebung #34284186)</li></ul>



Datenbank-Engine-Update	MySQL-kompatible Version	Version	Behobene MySQL-Fehler
<a href="#">Updates der Aurora MySQL-Datenbank-Engine 2023-07-31 (Version 3.04.0, kompatibel mit MySQL 8.0.28)</a>	<a href="#">8.0,28</a>	3,04,0	<ul style="list-style-type: none"> <li>• Es wurde ein Problem behoben, bei dem ein Pufferblock mit einer systemeigenen temporären Tabellenseite während des Seitendurchlaufs verschoben wurde, was zu einem Assert-Fehler führte (Fehler #33715694)</li> <li>• InnoDB: Verhindern Sie, dass Online-DDL-Operationen auf den out-of-bounds Speicher zugreifen (Bug# 34750489, Bug# 108925)</li> <li>• Es wurde ein Problem behoben, das bei der Verarbeitung komplexer SQL-Anweisungen, die aus mehreren verschachtelten Common Table Expressions (CTEs) bestehen, manchmal zu falschen Abfrageergebnissen führen kann (Fehler #34572040, Fehler #34634469, Fehler #33856374)</li> </ul>

Datenbank-Engine-Update	MySQL-kompatible Version	Version	Behobene MySQL-Fehler
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 08.12.2023 (Version 3.03.3, kompatibel mit MySQL 8.0.26)</a>	<a href="#">8.0.26</a>	3.03,3	<ul style="list-style-type: none"><li>• Es wurde ein Problem behoben, das aufgrund der Rotation von TLS-Zertifikaten im Hintergrund zu einer höheren CPU-Auslastung führen kann (Community-Fehlerbehebung #34284186)</li></ul>

Datenbank-Engine-Update	MySQL-kompatible Version	Version	Behobene MySQL-Fehler
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 29.08.2023 (Version 3.03.2, kompatibel mit MySQL 8.0.26)</a>	<a href="#">8,0,26</a>	3.03,2	<ul style="list-style-type: none"> <li>• Es wurde ein Problem behoben, das bei der Verarbeitung komplexer SQL-Anweisungen, die aus mehreren verschachtelten Common Table Expressions (CTEs) bestehen, manchmal zu falschen Abfrageergebnissen führen kann (Fehler 34572040, Fehler 34634469, Fehler 33856374)</li> <li>• InnoDB: Eine Race-Bedingung zwischen Threads, die versuchten, Statistiken für dieselbe Tabelle zu deinitialisieren und zu initialisieren, führte zu einem Assert-Fehler (Fehler 33135425)</li> <li>• InnoDB: Verhindert, dass Online-DDL-Operationen auf den out-of-bounds Speicher zugreifen (Bug #34750489, Bug #108925)</li> </ul>

Datenbank-Engine-Update	MySQL-kompatible Version	Version	Behobene MySQL-Fehler
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 11.05.2023 (Version 3.03.1, kompatibel mit MySQL 8.0.26)</a>	<a href="#">8.0.26</a>	3,03,1	<ul style="list-style-type: none"><li>• Es wurde ein Problem behoben, bei dem ein Pufferblock mit einer systemeigenen temporären Tabellenseite während des Seitendurchlaufs verschoben wurde, was zu einem Assert-Fehler führte (Fehler 33715694)</li></ul>

Datenbank-Engine-Update	MySQL-kompatible Version	Version	Behobene MySQL-Fehler
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 01.03.2023 (Version 3.03.0, kompatibel mit MySQL 8.0.26) Upgrades auf diese Version werden nicht unterstützt.</a>	<a href="#">8.0,26</a>	3,03,0	<ul style="list-style-type: none"> <li>• Es wurde ein Problem behoben, bei dem einige Spaltentypen, einschließlich JSON und TEXT, manchmal den Sortierpuffer erschöpften, wenn seine Größe nicht mindestens das 15-fache der Größe der größten Zeile in der Sortierung betrug. Jetzt muss der Sortierpuffer nur noch 15-mal so groß sein wie der größte Sortierschlüssel. (Fehler #103325, Fehler #105532, Fehler #32738705, Fehler #33501541)</li> <li>• Es wurde ein Problem behoben, bei dem InnoDB einige legale Namen für Tabellenpartitionen nicht immer korrekt behandelte. (Fehler #32208630)</li> <li>• Es wurde ein Problem behoben, das unter bestimmten Bedingungen aufgrund einer inkorrekten Berechnung der NULL-Eigenschaft bei der Ausführung einer Abfrage mit einer OR-Bedingung zu falschen Ergebnissen führen kann. (Fehler #34060289)</li> <li>• Es wurde ein Problem behoben, das unter bestimmten Bedingungen zu falschen Ergebnissen führen kann, wenn</li> </ul>

Datenbank-Engine-Update	MySQL-kompatible Version	Version	Behobene MySQL-Fehler
			<p>die folgenden beiden Bedingungen erfüllt sind:</p> <ul style="list-style-type: none"> <li>• eine abgeleitete Tabelle wird mit dem äußeren Abfrageblock zusammengeführt.</li> <li>• die Abfrage umfasst einen Left Join und eine IN-Unterabfrage.</li> </ul> <p>(Fehler #34060289)</p> <ul style="list-style-type: none"> <li>• Es wurde ein Problem behoben, bei dem inkorrekte AUTO_INCREMENT-Werte generiert wurden, wenn der maximale ganzzahlige Spaltenwert überschritten wurde. Der Fehler war darauf zurückzuführen, dass der maximale Spaltenwert nicht berücksichtigt wurde. Es hätte der vorherige gültige AUTO_INCREMENT-Wert zurückgegeben werden müssen, was zu einem Fehler mit doppeltem Schlüssel führte. (Fehler #87926, Fehler #26906787)</li> <li>• Es wurde ein Problem behoben, bei dem es nicht möglich war, die DROP-Berechtigung für das Performance-Schema zu widerrufen. (Fehler #33578113)</li> </ul>

Datenbank-Engine-Update	MySQL-kompatible Version	Version	Behobene MySQL-Fehler
			<ul style="list-style-type: none"><li>• Es wurde ein Problem behoben, bei dem eine gespeicherte Prozedur, die eine IF-Anweisung mit EXISTS, die auf eine oder mehrere Tabellen wirkte, die zwischen den Ausführungen gelöscht und neu erstellt wurden, bei den nachfolgenden Aufrufen nach dem ersten nicht korrekt ausgeführt wurde. (Fehler #32855634).</li><li>• Es wurde ein Problem behoben, bei dem eine Abfrage, die auf eine Ansicht in einer Unterabfrage und einem äußeren Abfrageblock verweist, zu einem unerwarteten Neustart führen kann (Fehler #32324234).</li></ul>

Datenbank-Engine-Update	MySQL-kompatible Version	Version	Behobene MySQL-Fehler
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 18.11.2022 (Version 3.02.2, kompatibel mit MySQL 8.0.23)</a> <a href="#">Das Ende des Standard-Supports ist der 15. Januar 2024.</a>	<a href="#">8,0,23</a>	3,02,2	<ul style="list-style-type: none"> <li>• Es wurde ein Problem behoben, das unter bestimmten Bedingungen aufgrund einer inkorrekten Berechnung der NULL-Eigenschaft bei der Ausführung einer Abfrage mit einer OR-Bedingung zu falschen Ergebnissen führen kann. (Fehler #34060289)</li> <li>• Es wurde ein Problem behoben, das unter bestimmten Bedingungen zu falschen Ergebnissen führen kann, wenn die folgenden beiden Bedingungen erfüllt sind:             <ul style="list-style-type: none"> <li>• Eine abgeleitete Tabelle wird mit dem äußeren Abfrageblock zusammengeführt.</li> <li>• Die Abfrage umfasst einen Left Join und eine IN-Unterabfrage. (Fehler #34060289)</li> </ul> </li> <li>• Es wurde ein Problem behoben, bei dem es nicht möglich war, die DROP-Berechtigung für das Performance-Schema zu widerrufen. (Fehler #33578113)</li> <li>• Es wurde ein Problem behoben, bei dem eine gespeicherte Prozedur, die eine IF-Anweisung mit EXISTS, die auf eine oder mehrere Tabellen wirkte,</li> </ul>



Datenbank -Engine- Update	MySQL-kompatible Version	Version	Behobene MySQL-Fehler
			<p>die zwischen den Ausführungen gelöscht und neu erstellt wurden, bei den nachfolgenden Aufrufen nach dem ersten nicht korrekt ausgeführt wurde. (MySQL-Fehler #32855634).</p> <ul style="list-style-type: none"> <li>• Es wurden inkorrekte AUTO_INCREMENT-Werte generiert, wenn der maximale ganzzahlige Spaltenwert überschritten wurde. Der Fehler war darauf zurückzuführen, dass der maximale Spaltenwert nicht berücksichtigt wurde. Es hätte der vorherige gültige AUTO_INCREMENT-Wert zurückgegeben werden müssen, was zu einem Fehler mit doppeltem Schlüssel führte. (Fehler #87926, Fehler #26906787)</li> <li>• Es wurde ein Problem behoben, das zu einem Fehler beim Upgrade eines Aurora-MySQL-Datenbank-Clusters der Version 1 (kompatibel mit MySQL 5.6) führen kann, der vom Benutzer erstellte Tabellen mit bestimmten Tabellen-IDs enthielt. Die Zuweisung dieser Tabellen-IDs kann beim Upgrade von Aurora MySQL</li> </ul>

Datenbank-Engine-Update	MySQL-kompatible Version	Version	Behobene MySQL-Fehler
			Version 2 (kompatibel mit MySQL 5.7) zu Aurora MySQL Version 3 (kompatibel mit MySQL 8.0) dazu führen, dass Datenwörterbuch-Tabellen-IDs in Konflikt geraten. (Fehler 33919635)

Datenbank-Engine-Update	MySQL-kompatible Version	Version	Behobene MySQL-Fehler
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 20.04.2022</a> <a href="#">2</a> <a href="#">(Version 3.02.0, kompatibel mit MySQL 8.0.23).</a> <a href="#">Ende des Standard-Supports 15. Januar 2024.</a> <a href="#">Upgrades auf diese Version werden nicht unterstützt.</a>	<a href="#">8,0,23</a>	3,02,0	<p>Die unsachgemäße Behandlung von temporären Tabellen, die für Cursor in gespeicherten Prozeduren verwendet werden, was zu unerwartetem Serververhalten führen kann, wurde behoben. (<a href="#">Fehler #32416811</a>)</p>

Datenbank-Engine-Update	MySQL-kompatible Version	Version	Behobene MySQL-Fehler
<p><a href="#">Aurora-MySQL-Datenbank-Engine-Updates 15.04.2022</a>  <a href="#">2 (Version 3.01.1, kompatibel mit MySQL 8.0.23)</a>.            Ende des Standard-Supports 15. Januar 2024.  <a href="#">Upgrades auf diese Version werden nicht unterstützt.</a></p>	<p><a href="#">8,0,23</a></p>	<p>3,01,1</p>	<p>Die unsachgemäße Behandlung von temporären Tabellen, die für Cursor in gespeicherten Prozeduren verwendet werden, was zu unerwartetem Serververhalten führen kann, wurde behoben. (<a href="#">Fehler #32416811</a>)</p>

# MySQL-Fehlerbehebungen durch Aurora MySQL 2.x-Datenbank-Engine-Updates

Die mit MySQL 5.7 kompatible Aurora-Version enthält alle MySQL-Fehlerbehebungen bis MySQL 5.7.40. In der folgenden Tabelle werden zusätzliche MySQL-Fehler bezeichnet, die durch Aurora MySQL-Datenbank-Engine-Updates behoben wurden, und es wird angegeben, mit welchem Update sie behoben wurden.

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 28.12.2023 (Version 2.12.1, kompatibel mit MySQL 5.7.40)</a>	2.12.1	<ul style="list-style-type: none"> <li>• Es wurde ein Problem behoben, das dazu führen kann, dass bestehende und neue Remote-Verbindungen blockiert werden, wenn sie gleichzeitig mit der <code>SHOW PROCESSLIST</code> -Anweisung ausgeführt werden (Community-Fehler 34857411)</li> <li>• Replikation: Einige binäre Protokollereignisse wurden nicht immer korrekt behandelt (Fehler 34617506)</li> <li>• Die Verarbeitung von Einzelzeichen-Token durch ein Parser-Plugin für die Volltextsuche (FTS) wurde behoben (Fehler 35432973)</li> </ul>
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 25.07.2023 (Version 2.12.0, kompatibel mit MySQL 5.7.40)</a>	2.12.0	<ul style="list-style-type: none"> <li>• Es wurde ein Problem behoben, das aufgrund einer TLS-Zertifikatrotation im Hintergrund zu einer höheren CPU-Auslastung führen kann. (Community-Fehlerbehebung #34284186)</li> </ul>
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 17.10.2023 (Version 2.11.4, kompatibel mit MySQL 5.7.12)</a>	2.11.4	<ul style="list-style-type: none"> <li>• Replikation: Einige binäre Protokollereignisse wurden nicht immer korrekt behandelt. (Fehler #34617506)</li> <li>• Es wurde ein Problem behoben, das aufgrund einer TLS-Zertifikatrotation im Hintergrund zu einer höheren CPU-Auslastung führen kann. (Community-Fehlerbehebung #34284186)</li> <li>• In vorbereiteten Anweisungen konnten einige Arten von Unterabfragen einen Serverabbruch verursachen. (Fehler #33100586)</li> </ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 25.10.2022 (Version 2.11.0, kompatibel mit MySQL 5.7.12).</a> <a href="#">Diese Version ist für neue Erstellungen nicht verfügbar</a> :	2.11.0	<ul style="list-style-type: none"> <li>• Es wurde ein Problem behoben, bei dem der Code zum Lesen von Zeichensatzinformationen aus den Ereignistabellen von Performance-Schema-Anweisungen (z. B. <code>events_statements_current</code>) das gleichzeitige Schreiben in diese Zeichensatzinformationen nicht verhinderte. Aufgrund dieses Problems kann der Zeichensatz für den SQL-Abfragetext möglicherweise ungültig sein, was zu einem Serverabbruch führen kann. Mit dieser Korrektur führt ein ungültiger Zeichensatz zu einer Kürzung der <code>SQL_TEXT</code>-Spalte und verhindert Serverabbrüche. (Fehler #23540008)</li> <li>• InnoDB: Backport einer Korrektur für Community-Fehler #25189192, Fehler #84038. Es wurde ein Problem behoben, bei dem InnoDB nach einer <code>RENAME TABLE</code>-Operation, bei der eine Tabelle in ein anderes Schema verschoben wurde, die Datenwörterbuchtafel <code>INNODB_SYS_DATAFILES</code> nicht aktualisieren konnte. Dies führte zu einem Fehler beim Neustart, der darauf hinwies, dass die Tablespace-Datendatei nicht gefunden werden konnte.</li> <li>• InnoDB: Es wurde ein Problem behoben, bei dem der Server beim Hinzufügen eines neuen Indexes einen intern definierten Fremdschlüsselindex löschte und versuchte, einen sekundären Index, der für eine virtuell generierte Spalte definiert ist, als Fremdschlüsselindex zu verwenden, was zu einem Serverabsturz führte. InnoDB erlaubt nun einer Fremdschlüsselleinschränkung, auf einen sekundären Index zu verweisen, der für eine virtuell generierte Spalte definiert ist. (Fehler #23533396)</li> <li>• Es wurde ein Problem behoben, bei dem zwei Sitzungen, die gleichzeitig eine <code>INSERT... ON DUPLICATE KEY UPDATE</code>-Operation ausführten, ein Deadlock generierten. Während eines teilweisen Rollbacks eines Tupels konnte es in einer anderen Sitzung aktualisiert werden. Der Fix für diesen Fehler macht die Korrekturen für Fehler #11758237,</li> </ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
		<p>Fehler #17604730 und Fehler #20040791 rückgängig. (Fehler #25966845)</p> <ul style="list-style-type: none"> <li>• Es wurde ein Problem behoben, bei dem die Rechte EXECUTE und ALTER ROUTINE den Erstellern von Routinen nicht korrekt erteilt wurden, selbst wenn automatic_sp_privileges aktiviert war. (Fehler #27407480)</li> <li>• Backport einer Korrektur für Community-Fehler #24671968: Es wurde ein Problem behoben, bei dem eine Abfrage falsche Ergebnisse liefern konnte, wenn die WHERE-Klausel eine abhängige Unterabfrage enthielt, die Tabelle einen sekundären Index für die Spalten in der Auswahlliste hatte, gefolgt von den Spalten in der Unterabfrage, und GROUP BY oder DISTINCT erlaubte, einen Loose Index Scan zu verwenden.</li> <li>• Es wurde ein Problem behoben, bei dem die Replikation unterbrochen wurde, wenn eine Löschanweisung für mehrere Tabellen mit Fremdschlüsseln ausgegeben wurde. (Fehler #80821)</li> <li>• Es wurde ein Problem behoben, bei dem in Sonderfällen bestimmte Slave-Fehler auch dann nicht ignoriert wurden, wenn <a href="#">slave_skip_errors</a> aktiviert war. In Fällen, in denen das Öffnen und Sperren einer Tabelle fehlschlug oder Feldkonvertierungen auf einem Server fehlschlugen, auf dem die zeilenbasierte Replikation ausgeführt wird, wird der Fehler als kritisch angesehen und der Status von <a href="#">slave_skip_errors</a> wird <a href="#">ignoriert</a>. Die Korrektur stellt sicher, dass bei aktivierter Option <a href="#">slave_skip_errors</a> alle Fehler, die beim Anwenden einer Transaktion gemeldet wurden, korrekt behandelt werden. (Fehler #70640, Fehler #17653275)</li> <li>• Es wurde ein Problem behoben, bei dem eine <a href="#">SET PASSWORD</a>-Anweisung von einem MySQL 5.6-Master auf einen MySQL 5.7-Slave oder von einem MySQL 5.7-Master, bei dem die Systemvariable <a href="#">log_builtin_as_identified_by_password</a> auf ON gesetzt war, auf einen MySQL 5.7-Slave</li> </ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
		<p>repliziert wurde. Der Passwort-Hash selbst wurde ebenfalls gehasht, bevor er auf dem Slave gespeichert wurde. Das Problem ist jetzt behoben und der replizierte Passwort-Hash wird so gespeichert, wie er ursprünglich an den Slave übergeben wurde. (Fehler #24687073)</p> <ul style="list-style-type: none"> <li>• Es wurde ein Problem behoben, bei dem die Serialisierung eines JSON-Werts, der aus einem großen Unterdokument besteht, das in viele Ebenen von JSON-Arrays, Objekten oder beidem eingeschlossen ist, manchmal übermäßig viel Zeit in Anspruch nahm. (Fehler #23031146)</li> <li>• Anweisungen, die nicht analysiert werden können (z. B. aufgrund von Syntaxfehlern), werden nicht mehr in das Protokoll für langsame Abfragen geschrieben. (Fehler #33732907)</li> </ul>
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 01.11.2022 (Version 2.10.3) (veraltet)</a>	2.10.3	<ul style="list-style-type: none"> <li>• Es wurde ein Problem behoben, bei dem der Code zum Lesen von Zeichensatzinformationen aus den Ereignistabellen von Performance-Schema-Anweisungen (z. B. <code>events_statements_current</code>) das gleichzeitige Schreiben in diese Zeichensatzinformationen nicht verhinderte. Aufgrund dieses Problems kann der Zeichensatz für den SQL-Abfragetext möglicherweise ungültig sein, was zu einem Serverabbruch führen kann. Mit dieser Korrektur führt ein ungültiger Zeichensatz zu einer Kürzung der <code>SQL_TEXT</code>-Spalte und verhindert Serverabbrüche. (Fehler #23540008)</li> <li>• Es wurde ein Problem behoben, bei dem für ein UPDATE eine temporäre Tabelle mit einem Primärschlüssel von mehr als 1024 Byte erforderlich war und diese Tabelle mit InnoDB erstellt wurde. Der Server konnte dann beendet werden. (Fehler #25153670)</li> </ul>



Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 26.01.2022 (Version 2.10.2) (veraltet)</a>	2.10.2	<ul style="list-style-type: none"> <li>• Es wurde ein Problem in InnoDB behoben, bei dem ein Codefehler im Zusammenhang mit Tabellenstatistiken eine Assertion in der Quelldatei dict0stats.cc (<a href="http://dict0stats.cc/">http://dict0stats.cc/</a>) auslöste. (Fehler #24585978)</li> <li>• Ein sekundärer Index über eine virtuelle Spalte wurde beschädigt, wenn der Index online erstellt wurde. Bei UPDATE-Anweisungen (<a href="https://dev.mysql.com/doc/refman/5.7/en/update.html">https://dev.mysql.com/doc/refman/5.7/en/update.html</a>) beheben wir dies wie folgt: Wenn der virtuelle Spaltenwert des Indexdatensatzes auf NULL festgelegt ist, generieren wir diesen Wert aus dem Clusterindexdatensatz. (Fehler #30556595)</li> <li>• ASSERTION "IOOTHER_LOCK" IN LOCK_REC_ADD_TO_QUEUE (Fehler #29195848)</li> <li>• HANDLE_FATAL_SIGNAL (SIG=11) IN __STRCHR_SSE2 (Fehler #28653104)</li> <li>• Es wurde ein Problem behoben, durch das eine Abfrageunterbrechung während einer Sperrwartezeit einen Fehler in InnoDB verursachen kann. (Fehler #28068293)</li> <li>• Verschachtelte Transaktionen konnten manchmal den Replikat-Applier blockieren, wenn die Transaktionsisolationstufe auf REPEATABLE READ eingestellt wurde. (Bug #25040331)</li> <li>• Es wurde ein Problem behoben, das dazu führen kann, dass Binlog-Replikate aufgrund eines Sperrwartzeitlimits zum Stillstand kamen. (Fehler #27189701)</li> </ul>
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 21.10.2021 (Version 2.10.1) (veraltet)</a>	2.10.1	CURRENT_TIMESTAMP ERZEUGT NULLEN IM AUSLÖSER. (Fehler #25209512)

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 25.05.2021 (Version 2.10.0) (veraltet)</a>	2.10.0	<ul style="list-style-type: none"> <li>• Verschachtelte Transaktionen konnten manchmal den Replikat-Applier blockieren, wenn die Transaktionsisolationstufe auf <a href="#">REPEATABLE READ</a> eingestellt wurde. (Bug #25040331)</li> <li>• Wenn eine gespeicherte Prozedur eine Anweisung enthielt, die auf eine Ansicht verweist, die wiederum auf eine andere Ansicht verweist, konnte die Prozedur nicht mehr als einmal erfolgreich aufgerufen werden. (Fehler #87858, Fehler #26864199)</li> <li>• Bei Abfragen mit vielen OR Bedingungen ist der Optimierer jetzt speichereffizienter und überschreitet weniger wahrscheinlich die Speichergrenze, die durch die Systemvariable <a href="#">range_optimizer_max_mem_size</a> auferlegt wird. Darüber hinaus wurde der Standardwert für diese Variable von 1.536.000 auf 8.388.608 angehoben. (Fehler #79450, Fehler #22283790)</li> <li>• Replikation: In der <code>next_event()</code>-Funktion, die vom SQL-Thread eines Replikats aufgerufen wird, um das nächste Ereignis aus dem Relay-Protokoll zu lesen, hat der SQL-Thread das erfasste <code>relaylog.log_lock</code> nicht freigegeben, als ein Fehler aufgetreten ist (z. B. aufgrund eines geschlossenen Relay-Protokolls), dazu führen, dass alle anderen Threads, die darauf warten, eine Sperre für das Relay-Protokoll zu erhalten, hängen bleiben. Mit dieser Fehlerbehebung wird die Sperre aufgehoben, bevor der SQL-Thread die Funktion unter der Situation verlässt. (Fehler #21697821)</li> <li>• Beheben einer Speicherbeschädigung für ALTER TABLE mit virtueller Spalte. (Fehler #24961167, Fehler #24960450)</li> <li>• Replikation: Multithreaded-Replikate konnten nicht mit kleinen Warteschlangengrößen unter Verwendung von <a href="#">slave_pending_jobs_size_max</a> konfiguriert werden, wenn sie jemals Transaktionen verarbeiten mussten, die größer als</li> </ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
		<p>diese Größe sind. Jedes Paket, das größer als <a href="#">slave_pending_jobs_size_max</a> ist, wurde mit dem Fehler ER_MTS_EVENT_BIGGER_PENDING_JOBS_SIZE_MAX abgelehnt, auch wenn das Paket kleiner war als das von <a href="#">slave_max_allowed_packet</a> festgelegte Limit. Mit dieser Fehlerbehebung wird <a href="#">slave_pending_jobs_size_max</a> eher zu einem sanfteren Limit als zu einem harten Limit. Wenn die Größe eines Pakets <a href="#">slave_pending_jobs_size_max</a> überschreitet, aber kleiner als <a href="#">slave_max_allowed_packet</a> ist, wird die Transaktion solange gehalten, bis alle Replikat-Worker leere Warteschlangen haben und dann verarbeitet. Alle nachfolgenden Transaktionen werden solange gehalten, bis die große Transaktion abgeschlossen ist. Die Warteschlangengröße für Replikat-Worker kann daher begrenzt sein, während gelegentlich größere Transaktionen zugelassen werden. (Fehler #21280753, Fehler #77406)</p> <ul style="list-style-type: none"> <li>• Replikation: Bei Verwendung eines Multithread-Replikats zeigten Applier-Fehler Worker ID-Daten an, die mit Daten, die in den Replikationstabellen des Leistungsschemas externalisierten Daten nicht übereinstimmten. (Fehler #25231367)</li> <li>• Replikation: Bei einem GTID-basierten Replikations-Replikat, das mit <a href="#">-gtid-mode=ON</a>, <a href="#">-log-bin=OFF</a> und der Verwendung von <a href="#">-slave-skip-errors</a> ausgeführt wird, wurde bei einem Fehler, der ignoriert werden sollte, <code>Exec_Master_Log_Pos</code> nicht korrekt aktualisiert, wodurch <code>Exec_Master_Log_Pos</code> die Synchronie mit <code>Read_master_log_pos</code> verliert. Wenn ein <code>GTID_NEXT</code> nicht angegeben wurde, würde das Replikat seinen GTID-Status niemals aktualisieren, wenn es von einer Transaktion mit einer einzelnen Anweisung zurückgesetzt wird. Der <code>Exec_Master_Log_Pos</code> würde nicht aktualisiert werden, denn obwohl die Transaktion abgeschlossen war, zeigte ihr GTID-Status etwas anderes an. Die Fehlerbehebung hebt die Zurückhaltung der Aktualisi</li> </ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
		<p>erung des GTID-Status auf, wenn eine Transaktion nur zurückgesetzt wird, wenn in einer Transaktion GTID_NEXT angegeben wird. (Fehler #22268777)</p> <ul style="list-style-type: none"> <li>• Replikation: Eine teilweise fehlgeschlagene Anweisung verbrauchte eine automatisch generierte oder angegebene GTID nicht korrekt, wenn die binäre Protokollierung deaktiviert wurde. Die Fehlerbehebung stellt sicher, dass eine teilweise fehlgeschlagene <a href="#">DROP TABLE</a>, ein teilweise fehlgeschlagener <a href="#">DROP USER</a> oder ein teilweise fehlgeschlagener <a href="#">DROP VIEW</a> die relevante GTID verbrauchen und sie in der @@GLOBAL.GTID_EXECUTED - und mysql.gtid_executed -Tabelle speichern, wenn die binäre Protokollierung deaktiviert wird. (Fehler #21686749)</li> <li>• Replikation: Replikate, auf denen MySQL 5.7 ausgeführt wird, konnten aufgrund eines Fehlers beim Abrufen von <a href="#">server_uuid</a>, die nicht Teil von MySQL 5.5 ist, keine Verbindung zu einer MySQL 5.5-Quelle herstellen. Dies wurde durch Änderungen in der Methode zum Abrufen von verursacht server_uuid . (Fehler #22748612)</li> <li>• Binlog-Replikation: Der Mechanismus zum Überspringen von GTID-Transaktionen funktionierte vor dieser Fehlerbehebung nicht ordnungsgemäß für die XA-Transaktion. Der Server verfügt über einen Mechanismus zum (stillen) Überspringen einer GTID-Transaktion, wenn diese bestimmte Transaktion bereits in der Vergangenheit ausgeführt wurde. (BUG#25041920)</li> <li>• <a href="#">XA ROLLBACK</a>-Anweisungen, die fehlgeschlagen sind, weil eine falsche Transaktions-ID angegeben wurde, konnten im Binärprotokoll mit der richtigen Transaktions-ID aufgezeichnet und daher durch Replikationsreplikate verwendet werden. Vor der binären Protokollierung wird nun die Fehlersituation geprüft und fehlgeschlagene XA ROLLBACK-Anweisungen nicht protokolliert. (Fehler #26618925)</li> </ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
		<ul style="list-style-type: none"> <li>• Replikation: Wenn ein Replikat mit einer <a href="#">CHANGE MASTER TO</a>-Anweisung ohne Angabe des Quellprotokolldateinamens und der Quellprotokollposition eingerichtet wurde, dann vor der Ausgabe von <a href="#">START SLAVE</a> heruntergefahren und dann mit der Option <a href="#">-relay-log-recovery</a> neu gestartet wurde, wurde die Replikation nicht gestartet. Dies geschah, weil der Empfänger-Thread vor dem Versuch der Wiederherstellung des Relay-Protokolls nicht gestartet wurde, sodass im Relay-Protokoll kein Protokollrotationsereignis verfügbar war, um den Namen der Quellprotokolldatei und die Quellprotokollposition bereitzustellen. In diesem Fall überspringt das Replikat nun die Wiederherstellung des Relay-Protokolls und protokolliert eine Warnung und beginnt dann mit der Replikation. (Fehler #28996606, Fehler #93397)</li> <li>• Replikation: Bei der zeilenbasierten Replikation wurde eine Meldung zurückgegeben, die Feldlängen falsch anzeigte, wenn von einer Tabelle mit einer utf8mb3-Spalte in eine Tabelle derselben Definition repliziert wurde, in der die Spalte mit einem utf8mb4-Zeichensatz definiert wurde. (Fehler #25135304, Fehler #83918)</li> <li>• Replikation: Als eine <a href="#">RESET SLAVE-Anweisung</a> für ein Replikations-Replikat mit verwendeten GTIDs ausgegeben wurde, wurden die vorhandenen Relay-Protokolldateien bereinigt, aber die neue Relay-Protokolldatei für den Kanal wurde generiert, bevor der Satz der empfangenen GTIDs für den Kanal gelöscht wurde. Der frühere GTID-Satz wurde daher als PREVIOUS_GTIDS Ereignis in die neue Relay-Protokolldatei geschrieben, was zu einem schwerwiegenden Fehler bei der Replikation führte, der besagt, dass das Replikat mehr GTIDs als die Quelle hatte, obwohl der für beide Server festgelegte Wert „gtid_executed“ leer war. Wenn nun RESET SLAVE ausgegeben wird, wird der Satz der empfangenen GTIDs gelöscht, bevor die neue Relay-Pro</li> </ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
		<p>tokolldatei generiert wird, damit diese Situation nicht auftritt. (Fehler #27411175)</p> <ul style="list-style-type: none"> <li>• Replikation: Bei Verwendung von GTIDs, die für die Replikation verwendet wurden, konnten Transaktionen einschließlich Anweisungen, die einen Analysefehler verursachten (<a href="#">ER_PARSE_ERROR</a>), nicht manuell durch die empfohlene Methode zum Einspritzen einer leeren oder Ersatztransaktion mit derselben GTID übersprungen werden. Diese Aktion sollte dazu führen, dass das Replikat die GTID als bereits verwendet identifiziert und daher die unerwünschte Transaktion überspringt, die seine GTID geteilt hat. Im Falle eines Parsing-Fehlers wurde der Replikationsanwendungs-Thread jedoch aufgrund des Parsing-Fehlers gestoppt, da die Anweisung geparkt wurde, bevor die GTID überprüft wurde, um zu sehen, ob sie übersprungen werden musste, obwohl die Transaktion trotzdem übersprungen werden sollte. Mit dieser Fehlerbehebung ignoriert der Replikations-Applier-Thread jetzt Parsingfehler, wenn die betreffende Transaktion übersprungen werden muss, da die GTID bereits verwendet wurde. Beachten Sie, dass diese Verhaltensänderung nicht bei Workloads gilt, die aus einer von erzeugten Binärprotokollausgabe bestehe <code>mysqlbinlog</code> . In dieser Situation bestünde die Gefahr, dass eine Transaktion mit einem Parsing-Fehler, der unmittelbar auf eine übersprungene Transaktion folgt, auch stillschweigend übersprungen wird, wenn sie einen Fehler auslösen sollte. (Fehler #27638268)</li> <li>• Replikation: Aktivieren Sie den SQL-Thread, um eine Teiltransaktion mit GTID zu überspringen. (Fehler #25800025)</li> <li>• Replikation: Wenn ein negativer oder fraktionierter Zeitüberschreitungs-Parameter an <code>WAIT_UNTIL_SQL_THREAD_AFTER_GTIDS()</code> angegeben wurde, verhielt sich der Server auf unerwartete Weise. Mit dieser Fehlerbehebung.</li> </ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
		<ul style="list-style-type: none"> <li>• Wird ein fraktionierter Zeitüberschreitungs-Wert unverändert gelesen, ohne Abrunden.</li> <li>• Wird ein negativer Zeitüberschreitungs-Wert mit einem Fehler zurückgewiesen, wenn sich der Server in einem strikten SQL-Modus befindet. Wenn sich der Server nicht in einem strikten SQL-Modus befindet, bewirkt der Wert, dass die Funktion NULL sofort ohne Wartezeit zurückgibt und dann eine Warnung ausgibt. (Fehler #24976304, Fehler #83537)</li> <li>• Replikation: Wenn die <code>WAIT_FOR_EXECUTED_GTID_SET()</code>-Funktion mit einem Zeitüberschreitungs-Wert verwendet wurde, der einen Bruchteil enthält (z. B. 1,5), führte ein Fehler in der Umwandlungslogik dazu, dass die Zeitüberschreitung auf die nächste ganze Sekunde abgerundet wurde und bei Werten unter 1 Sekunde auf null (zum Beispiel 0,1). Die Casting-Logik wurde nun korrigiert, sodass der Zeitüberschreitungs-Wert wie ursprünglich angegeben ohne Rundung angewendet wird. Danke an Dirkjan Bussink für den Beitrag. (Fehler #29324564, Fehler #94247)</li> <li>• Wenn GTIDs aktiviert sind, löste <a href="#">XA COMMIT</a> bei einer nicht verbundenen XA-Transaktion innerhalb einer Transaktion mit mehreren Anweisungen eine Zusicherung aus. (Fehler #22173903)</li> <li>• Replikation: In Debug-Builds wurde eine Zusicherung ausgelöst, wenn eine <a href="#">XA ROLLBACK-Anweisung</a> für eine unbekannte Transaktions-ID ausgegeben wurde, als der <a href="#">gtid_next</a>-Wert manuell festgelegt wurde. Der Server versucht jetzt nicht, den GTID-Status zu aktualisieren, wenn eine ROLLBACK XA-Anweisung mit einem Fehler fehlschlägt. (Fehler #27928837, Fehler #90640)</li> <li>• Behebung des Problems mit der falschen Sortierreihenfolge, wenn mehrere CASE-Funktionen in der ORDER BY-Klausel verwendet werden. (Fehler #22810883)</li> </ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
		<ul style="list-style-type: none"> <li>Einige Abfragen, bei denen die Sortierung verwendet wurde, konnten während der Optimierung auf eine nicht initialisierte Spalte zugreifen und verursachten einen Serverabbruch. (Fehler #27389294)</li> </ul>
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 12.11.2021 (Version 2.09.3) (veraltet)</a>	2.09.3	<ul style="list-style-type: none"> <li>ASERTION! M_PREBUILT-&gt;TRX-&gt;CHECK_FOREIGNS. (Fehler #23533396)</li> <li>Replikation: * Ein Sperrproblem in der Funktion WAIT_FOR_EXECUTED_GTID_SET () könnte dazu führen, dass der Server unter bestimmten Umständen hängen bleibt. Das Problem wurde jetzt behoben. (Fehler #29550513)</li> </ul>
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 11.12.2020 (Version 2.09.1) (veraltet)</a>	2.09.1	<ul style="list-style-type: none"> <li>Replikation: Doppelspieltransaktionen konnten manchmal den Slave-Applier blockieren, wenn die Transaktionsisolationsstufe auf <a href="#">REPEATABLE READ</a> festgelegt wurde. (Bug #25040331)</li> <li>Bei einer Tabelle mit einer <a href="#">TIMESTAMP</a>- oder <a href="#">DATETIME</a>-Spalte mit dem Standardwert <a href="#">CURRENT_TIMESTAMP</a> könnte die Spalte auf 0000-00-00 00:00:00 initialisiert werden, wenn die Tabelle einen BEFORE INSERT-Auslöser hat. (Bug #25209512, Bug #84077)</li> <li>Bei einer <a href="#">INSERT</a>-Anweisung, für die die VALUES-Liste mit einer Unterabfrage, die eine Verknüpfung enthält, Werte für die zweite oder spätere Zeile erzeugte, konnte es zu einer Beendigung des Servers kommen, nachdem die erforderlichen Berechtigungen nicht aufgelöst wurden. (Bug #23762382)</li> </ul>
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 12.11.2020 (Version 2.08.3) (veraltet)</a>	2.08.3	<ul style="list-style-type: none"> <li>Bug #23762382 - WERTE EINFÜGEN ABFRAGE MIT JOIN IN EINEM SELECT VERURSACHT FALSCHES VERHALTEN .</li> <li>Bug #25209512 - CURRENT_TIMESTAMP ERZEUGT NULLEN IM TRIGGER.</li> </ul>



Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 02.06.2020 (Version 2.08.0) (veraltet)</a>	2.08.0	<ul style="list-style-type: none"><li>• <a href="#">Fehler #25289359</a>: Eine bei der Datensynchronisierung vorgenommene Volltext-Cache-Sperre wurde nicht freigegeben, wenn die Volltext-Cache-Größe das Volltext-Cache-Größenlimit überschritt.</li><li>• <a href="#">Fehler #29138644</a>: Das manuelle Ändern der Systemzeit bei ausgeführtem MySQL-Server verursachte Verzögerungen in den Page-Cleaner-Threads.</li><li>• <a href="#">Fehler #25222337</a>: Ein Feldname einer virtuellen NULL-Spalte in einem virtuellen Index führte zu einem Beenden des Servers während eines Feldnamensabgleichs, der beim Befüllen von virtuellen Spalten auftritt, die von einer Fremdschlüsselbeschränkung betroffen sind.</li><li>• <a href="#">Fehler #25053286</a>: Das Ausführen einer gespeicherten Prozedur, die eine Abfrage enthält, die auf eine Ansicht zugreift, konnte Speicher zuweisen, der bis zum Ende der Sitzung nicht freigegeben wurde.</li><li>• <a href="#">Fehler #25586773</a>: Das Ausführen einer gespeicherten Prozedur, die eine Anweisung enthält, die aus dem Inhalt bestimmter <a href="#">SELECT</a>-Anweisungen eine Tabelle erzeugt hat, konnte zu einem Speicherleck führen.</li><li>• <a href="#">Fehler #28834208</a>: Während der Protokollanwendung, nach einer <a href="#">OPTIMIZE TABLE</a>-Operation, füllte InnoDB die virtuellen Spalten nicht auf, bevor nach Aktualisierungen des virtuellen Spaltenindexes gesucht wurde.</li><li>• <a href="#">Fehler #26666274</a>: Endlosschleife im Leistungsschema-Puffercontainer aufgrund eines Überlaufs einer 32-Bit-Ganzzahl ohne Vorzeichen.</li></ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 16.06.2022 (Version 2.07.8) (veraltet)</a>	2.07.8	Wenn für ein UPDATE eine temporäre Tabelle mit einem Primärschlüssel von mehr als 1024 Byte erforderlich war und diese Tabelle mit InnoDB erstellt wurde, konnte ein Serverabbruch erfolgen. (Fehler #25153670)
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 02.09.2021 (Version 2.07.6) (veraltet)</a>	2.07.6	<ul style="list-style-type: none"> <li>• DAS EINFÜGEN VON DATENSÄTZEN DER GRÖSSE 64K NIMMT ZU VIEL ZEIT IN ANSPRUCH. (Fehler#23031146)</li> </ul>
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 04.03.2021 (Version 2.07.4) (veraltet)</a>	2.07.4	<ul style="list-style-type: none"> <li>• Es wurde ein Problem im Volltext-Ngram-Parser behoben, das bei Token mit „ „ (Leerzeichen), „%“ oder „,“ bestand. Kunden sollten ihre FTS-Indizes neu erstellen, wenn sie einen Ngram-Parser verwenden. (Bug #25873310)</li> <li>• Ein Problem wurde behoben, das bei der Ausführung von Abfragen mit verschachtelten SQL-Ansichten zu einem Neustart der Engine führen konnte. (Bug #27214153, Bug #26864199)</li> </ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 10.11.2020 (Version 2.07.3) (veraltet)</a>	2.07.3	<ul style="list-style-type: none"> <li>• InnoDB: Gleichzeitige XA-Transaktionen, die erfolgreich zur XA-Prüfungsphase auf dem Master liefen, führten zu einem Konflikt bei der Wiederholung auf dem Slave, was zu einem Timeout für die Sperrwartzeit im Applier-Thread führte. Der Konflikt war auf den GAP-Sperrbereich zurückzuführen, der sich unterschied, als die Transaktionen auf dem Slave seriell wiedergegeben wurden. Um diese Art von Konflikt zu verhindern, werden GAP-Sperren, die von XA-Transaktionen in der <a href="#">READ COMMITTED</a>-Isolationsstufe übernommen werden, jetzt veröffentlicht (und nicht mehr geerbt), wenn XA-Transaktionen die Vorbereitungsphase erreichen. (Bug #27189701, Bug #25866046)</li> <li>• InnoDB: Eine GAP-Sperre wurde während der Überprüfung von Fremdschlüsseln unnötig durchgeführt, während die <a href="#">READ COMMITTED</a>-Isolationsstufe verwendet wurde. (Bug #25082593)</li> <li>• Replikation: Wenn bei Verwendung von XA-Transaktionen ein Timeout für ein Zeitlimit einer Sperrung oder ein Deadlock für den Applier (SQL) -Thread auf einem Replikations-Slave aufgetreten ist, funktionierte die automatische Wiederholung nicht. Die Ursache war, dass der SQL-Thread zwar einen Rollback durchführen würde, die XA-Transaktion jedoch nicht zurücksetzen würde. Dies bedeutete, dass bei der Wiederholung der Transaktion das erste Ereignis XA START war, was ungültig war, da die XA-Transaktion bereits im Gange war, was zu einem XAER_RMFAIL-Fehler führte. (Bug #24764800)</li> <li>• Replikation: Doppelspieltransaktionen konnten manchmal den Slave-Applier blockieren, wenn die Transaktionsisolationsstufe auf <a href="#">REPEATABLE READ</a> festgelegt wurde. (Bug #25040331)</li> <li>• Replikation: Der von einer <a href="#">SHOW SLAVE STATUS</a>-Anweisung zurückgegebene Wert für die gesamte kombinierte Größe aller vorhandenen Relay-Protokolldateien (Relay_Log_Space) könnte viel größer werden als der tatsächlich von den Relay-</li> </ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
		<p>Protokolldateien belegte Speicherplatz. Der I/O-Thread hat die Variable nicht gesperrt, während sie den Wert aktualisiert hat, so dass der SQL-Thread automatisch eine Relay-Protokolldatei löschen und einen reduzierten Wert schreiben konnte, bevor der I/O-Thread mit der Aktualisierung des Werts fertig war. Der I/O-Thread schrieb dann seine ursprüngliche Größenberechnung, ignorierte das Update des SQL-Threads und fügte so den Platz für die gelöschte Datei hinzu. Der Wert <code>Relay_Log_Space</code> ist jetzt während Updates gesperrt, um gleichzeitige Aktualisierungen zu verhindern und eine genaue Berechnung sicherzustellen. (Bug #26997096, Bug #87832)</p> <ul style="list-style-type: none"> <li>• Bei einer <a href="#">INSERT</a>-Anweisung, für die die VALUES-Liste mit einer Unterabfrage, die eine Verknüpfung enthält, Werte für die zweite oder spätere Zeile erzeugte, konnte es zu einer Beendigung des Servers kommen, nachdem die erforderlichen Berechtigungen nicht aufgelöst wurden. (Bug #23762382)</li> <li>• Bei einer Tabelle mit einer <a href="#">TIMESTAMP</a>- oder <a href="#">DATETIME</a>-Spalte mit dem Standardwert <a href="#">CURRENT_TIMESTAMP</a> könnte die Spalte auf <code>0000-00-00 00:00:00</code> initialisiert werden, wenn die Tabelle einen BEFORE INSERT-Auslöser hat. (Bug #25209512, Bug #84077)</li> <li>• Ein Server-Exit könnte aus gleichzeitigen Versuchen mehrerer Threads resultieren, Metadaten Performance Schema-Objekte zu registrieren und abzumelden. (Bug #26502135)</li> <li>• Das Ausführen einer gespeicherten Prozedur, die eine Anweisung enthält, die aus dem Inhalt bestimmter <a href="#">SELECT</a>-Anweisungen eine Tabelle erzeugt hat, konnte zu einem Speicherleck führen. (Bug #25586773)</li> <li>• Das Ausführen einer gespeicherten Prozedur mit einer auf eine Ansicht zugreifenden Abfrage konnte Speicher zuweisen, der bis zum Ende der Sitzung nicht freigegeben wurde. (Bug #25053286)</li> </ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
		<ul style="list-style-type: none"> <li>• Bestimmte Unterabfragen-Materialisierungen konnten einen Server-Exit verursachen. Diese Abfragen erzeugen nun einen Fehler, der darauf hindeutet, dass die Materialisierung deaktiviert ist. (Bug #26402045)</li> <li>• Abfragen mit zahlreichen Links-Joins waren langsam, wenn die Join-Pufferung verwendet wurde (z. B. bei Verwendung mit dem blockverschachtelten Schleifenalgorithmus). (Bug #18898433, Bug #72854)</li> <li>• Der Optimierer übersprang die zweite Spalte in einem zusammengesetzten Index, als er eine Verknüpfung mit einer LIKE Klausel gegen die zweite Spalte ausführte. (Bug #28086754)</li> </ul>
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 17.04.2020 (Version 2.07.2) (veraltet)</a>	2.07.2	<ul style="list-style-type: none"> <li>• Fehler #23104498: Es wurde ein Problem in Performance Schema behoben, bei dem der gesamte verwendete Speicher gemeldet wurde. (<a href="https://github.com/mysql/mysql-server/commit/20b6840df5452f47313c6f9a6ca075bfbc00a96b">https://github.com/mysql/mysql-server/commit/20b6840df5452f47313c6f9a6ca075bfbc00a96b</a>)</li> <li>• Fehler #22551677: Es wurde ein Problem in Performance Schema behoben, das dazu führen konnte, dass die Datenbank-Engine abstürzte, wenn sie offline geschaltet wurde. (<a href="https://github.com/mysql/mysql-server/commit/05e2386eccd32b6b444b900c9f8a87a1d8d531e9">https://github.com/mysql/mysql-server/commit/05e2386eccd32b6b444b900c9f8a87a1d8d531e9</a>)</li> <li>• Fehler #23550835, Fehler #23298025, Fehler #81464: Es wurde ein Problem in Performance Schema behoben, bei dem eine Datenbank-Engine aufgrund der Überschreitung der Kapazität eines internen Puffers zum Absturz gebracht wurde. (<a href="https://github.com/mysql/mysql-server/commit/b4287f93857bf2f99b18fd06f555bbe5b12debfc">https://github.com/mysql/mysql-server/commit/b4287f93857bf2f99b18fd06f555bbe5b12debfc</a>, <a href="https://github.com/mysql/mysql-server/commit/b4287f93857bf2f99b18fd06f555bbe5b12debfc">https://github.com/mysql/mysql-server/commit/b4287f93857bf2f99b18fd06f555bbe5b12debfc</a>)</li> </ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 25.11.2019 (Version 2.07.0) (veraltet)</a>	2.07.0	<ul style="list-style-type: none"> <li>• Bug #26251621: INCORRECT BEHAVIOR WITH TRIGGER AND GCOL</li> <li>• Bug #22574695: ASSERTION `!TABLE    (!TABLE-&gt;READ_SET    BITMAP_IS_SET(TABLE-&gt;READ_SET, FIEL</li> <li>• Bug #25966845: INSERT ON DUPLICATE KEY GENERATE A DEADLOCK</li> <li>• Bug #23070734: CONCURRENT TRUNCATE TABLES CAUSE STALL</li> <li>• Bug #26191879: FOREIGN KEY CASCADES USE EXCESSIVE MEMORY</li> <li>• Bug #20989615: INNODB AUTO_INCREMENT PRODUCES SAME VALUE TWICE</li> </ul>
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 11.11.2019 (Version 2.05.0) (veraltet)</a>	2.05.0	<ul style="list-style-type: none"> <li>• Fehler #23054591: PURGE BINARY LOGS TO liest die gesamte Binlog-Datei und führt zum Stillstand MySql</li> </ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 14.08.2018 (Version 2.04.9) (veraltet)</a>	2.04.9	<ul style="list-style-type: none"> <li>• Fehler #23070734, Bug #80060: Concurrent TRUNCATE TABLEs cause stalls</li> <li>• Fehler #23103937: PS_TRUNCATE_ALL_TABLES() DOES NOT WORK IN SUPER_READ_ONLY MODE</li> <li>• Fehler #22551677: Wenn der Server offline genommen wurde, konnte eine seltene Bedingung im Performance Schema zu einem Server-Exit führen.</li> <li>• Fehler #27082268: Ungültige FTS-Synchronisierung.</li> <li>• FEHLER #12589870: Es wurde ein Fehler behoben, der einen Neustart mit mehrfacher Abfrage auslöste, wenn der Abfrage-Cache aktiviert ist.</li> <li>• Fehler #26402045: Bestimmte Fälle der Materialisierung von Unterabfragen konnten einen Server-Exit verursachen. Diese Abfragen erzeugen nun einen Fehler, der darauf hindeutet, dass die Materialisierung deaktiviert ist.</li> <li>• Fehler #18898433: Abfragen mit vielen Links-Joins waren langsam, wenn die Join-Pufferung verwendet wurde (zum Beispiel mit dem block-verschachtelten Schleifenalgorithmus).</li> <li>• Fehler #25222337: Ein Feldname einer virtuellen NULL-Spalte in einem virtuellen Index führte zu einem Beenden des Servers während eines Feldnamensabgleichs, der beim Befüllen von virtuellen Spalten auftritt, die von einer Fremdschlüsselbeschränkung betroffen sind. (<a href="https://github.com/mysql/mysql-server/commit/273d5c9d7072c63b6c47dbef6963d7dc491d5131">https://github.com/mysql/mysql-server/commit/273d5c9d7072c63b6c47dbef6963d7dc491d5131</a>)</li> <li>• Fehler #25053286: Das Ausführen einer gespeicherten Prozedur mit einer auf eine Ansicht zugreifenden Abfrage konnte Speicher zuweisen, der bis zum Ende der Sitzung nicht freigegeben wurde. (<a href="https://github.com/mysql/mysql-server/commit/d7b37d4d141a95f577916448650c429f0d6e193d">https://github.com/mysql/mysql-server/commit/d7b37d4d141a95f577916448650c429f0d6e193d</a>)</li> <li>• Fehler #25586773: Das Ausführen einer gespeicherten Prozedur mit einer Anweisung, die eine Tabelle aus dem</li> </ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
		<p>Inhalt bestimmter SELECT (<a href="https://dev.mysql.com/doc/refman/5.7/en/select.html">https://dev.mysql.com/doc/refman/5.7/en/select.html</a>) -Anweisungen erstellt, konnte zu einem Speicherverlust führen. (<a href="https://github.com/mysql/mysql-server/commit/88301e5adab65f6750f66af284be410c4369d0c1">https://github.com/mysql/mysql-server/commit/88301e5adab65f6750f66af284be410c4369d0c1</a>)</p> <ul style="list-style-type: none"> <li>• Fehler #26666274: INFINITE LOOP IN PERFORMANCE SCHEMA BUFFER CONTAINER.</li> <li>• Fehler #23550835, Fehler #23298025, Fehler #81464: SELECT Performance Schema-Tabellen, konnten bei vollem internen Puffer einen Server-Exit verursachen.</li> </ul>
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 19.09.2019 (Version 2.04.6) (veraltet)</a>	2.04.6	<ul style="list-style-type: none"> <li>• Fehler #23054591: PURGE BINARY LOGS TO liest die gesamte Binlog-Datei und führt zum Stillstand MySql</li> </ul>
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 02.05.2019 (Version 2.04.2) (veraltet)</a>	2.04.2	<p>Fehler #24829050 – INDEX_MERGE_INTERSECTION-OPTIMIERUNG FÜHRT ZU FALSCHEN ABFRAGEER GEBNISSEN</p>
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 11.10.2018 (Version 2.03) (veraltet)</a>	2.03	<ul style="list-style-type: none"> <li>• REVERSE-SCAN AUF EINER PARTITIONIERTEN TABELLE FÜHRT ICP AUS – ORDER BY DESC (Fehler #24929748).</li> <li>• JSON_OBJECT ERSTELLT UNGÜLTIGEN JSON-CODE (Fehler #26867509).</li> <li>• DAS EINFÜGEN GROßER JSON-DATEN DAUERT ÜBERMÄßIG LANGE. (Fehler #22843444).</li> <li>• DIE PARTITIONIERTEN TABELLEN NUTZEN UNTER VERSION 5.7 MEHR SPEICHER ALS UNTER VERSION 5.6 (Fehler #25080442).</li> </ul>



Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 21.09.2018 (Version 2.02.4) (veraltet)</a>	2.02.4	<ul style="list-style-type: none"> <li>• BUG#13651665 INNODB MAY BE UNABLE TO LOAD TABLE DEFINITION AFTER RENAME</li> <li>• BUG#21371070 INNODB: CANNOT ALLOCATE 0 BYTES.</li> <li>• BUG#21378944 FTS ASSERT ENC.SRC_ILIST_PTR != NULL, FTS_OPTIMIZE_WORD(), OPTIMIZE TABLE</li> <li>• BUG#21508537 ASSERTION FAILURE UT_A(!VIC TIM_TRX-&gt;READ_ONLY)</li> <li>• BUG#21983865 UNEXPECTED DEADLOCK WITH INNODB_AUTOINC_LOCK_MODE=0</li> <li>• BUG#22679185 INVALID INNODB FTS DOC ID DURING INSERT</li> <li>• BUG#22899305 GCOLS: ASSERTION: !(COL-&gt;PR TYPE &amp; 256).</li> <li>• BUG#22956469 MEMORY LEAK INTRODUCED IN 5.7.8 IN MEMORY/INNODB/OS0FILE</li> <li>• BUG#22996488 CRASH IN FTS_SYNC_INDEX WHEN DOING DDL IN A LOOP</li> <li>• BUG#23014521 GCOL:INNODB: ASSERTION: !IS_V</li> <li>• BUG#23021168 REPLICATION STOPS AFTER TRX IS ROLLED BACK ASYNC</li> <li>• BUG#23052231 ASSERTION: ADD_AUTOINC &lt; DICT_TABLE_GET_N_USER_COLS</li> <li>• BUG#23149683 ROTATE INNODB MASTER KEY WITH KEYRING_OKV_CONF_DIR MISSING: SIGSEGV; SIGNAL 11</li> <li>• BUG#23762382 INSERT VALUES QUERY WITH JOIN IN A SELECT CAUSES INCORRECT BEHAVIOR</li> <li>• BUG#25209512 CURRENT_TIMESTAMP PRODUCES ZEROS IN TRIGGER</li> </ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
		<ul style="list-style-type: none"> <li>• BUG#26626277 BUG IN "INSERT... ON DUPLICATE KEY UPDATE" QUERY</li> <li>• BUG#26734162 INCORRECT BEHAVIOR WITH INSERT OF BLOB + ON DUPLICATE KEY UPDATE</li> <li>• BUG#27460607 INCORRECT WHEN INSERT SELECT'S SOURCE TABLE IS EMPTY</li> </ul>
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 03.05.2018 (Version 2.02) (veraltet)</a>	2.02.0	Left Join gibt falsche Ergebnisse auf der äußeren Seite zurück (Fehlernr. 22833364)

## MySQL-Fehlerbehebungen durch Aurora MySQL 1.x-Datenbank-Engine-Updates

MySQL 5.6-kompatible Version Aurora enthält alle MySQL-Fehlerbehebungen durch MySQL 5.6.10. In der folgenden Tabelle werden zusätzliche MySQL-Fehler bezeichnet, die durch Aurora MySQL-Datenbank-Engine-Updates behoben wurden, und es wird angegeben, mit welchem Update sie behoben wurden.

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 18.03.2021 (Version 1.23.2) (veraltet)</a>	1.23.2	<ul style="list-style-type: none"> <li>• Replikation: Während eine SHOW BINLOG EVENTS Anweisung ausgeführt wurde, wurde jede parallele Transaktion blockiert. Der Fix stellt sicher, dass der SHOW BINLOG EVENTS Prozess jetzt nur noch eine Sperre für die Dauer der Berechnung der Endposition der Datei erwirbt, daher werden parallele Transaktionen nicht für lange Zeiträume blockiert. (Fehler #76618, Bug #20928790)</li> </ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 02.09.2020 (Version 1.23.0) (veraltet)</a>	1.23.0	<ul style="list-style-type: none"> <li>• Binärprotokoll-Ereignisse mit ALTER TABLE ADD COLUMN ALGORITHM=QUICK werden zu ALGORITHM=DEFAULT umgeschrieben, um mit der Community-Edition kompatibel zu sein.</li> <li>• Fehler #22350047: WENN DER CLIENT NACH DEM ROLLBACK AUF DEN SAVEPOINT BEENDET WURDE, WURDEN VORHERIGE STMTS ÜBERTRAGEN</li> <li>• Fehler #29915479: DAS AUSFÜHREN VON COM_REGISTER_SLAVE OHNE COM_BINLOG_DUMP KANN ZU EINEM BEENDEN DES SERVERS FÜHREN</li> <li>• Fehler #30441969: FEHLER #29723340: ABSTURZ DES MYSQL-SERVERS NACH SQL-ABFRAGE MIT DATA ?AST</li> <li>• Bug #30628268: ABSTURZ WEGEN FEHLENDEM SPEICHER</li> <li>• Fehler #27081349: UNERWARTETES VERHALTEN BEIM LÖSCHEN MIT RÄUMLICHER FUNKTION</li> <li>• Fehler #27230859: UNERWARTETES VERHALTEN BEIM HANDLING VON UNGÜLTIGEM POLYGON"</li> <li>• Fehler #27081349: UNERWARTETES VERHALTEN BEIM LÖSCHEN MIT SPATIAL"</li> <li>• Fehler #26935001: „ALTER TABLE AUTO_INCREMENT“ VERSUCHT, DEN INDEX AUS VERWORFENEM TABELLENRAUM ZU LESEN</li> <li>• Fehler #29770705: DER SERVER STÜRZTE BEIM AUSFÜHREN EINER AUSWAHL MIT SPEZIFISCHER WHERE-KLAUSEL AB</li> <li>• Fehler #27659490: AUSWAHL MIT DYNAMISCH EN BEREICH UND INDEX-ZUSAMMENFÜHRUNG VERBRAUCHT ZU VIEL SPEICHER (NICHT GENÜGENDE SPEICHER)</li> </ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
		<ul style="list-style-type: none"> <li>• Fehler #24786290: REPLIKATION BRICHT NACH DEM AUFTRETEN DES FEHLERS #74145 IM MASTER AB</li> <li>• Fehler #27703912: ÜBERMÄSSIGE SPEICHERNUTZUNG MIT VIEL VORBEREITUNG</li> <li>• Fehler #20527363: ABSTURZ TEMPORÄRER TRUNCATE-TABELLE: !DICT_TF2_FLAG_IS_SET(TABELLE, DICT_TF2_TEMPORARY)</li> <li>• Fehler #23103937 PS_TRUNCATE_ALL_TABLES() FUNKTIONIERT NICHT IM MODUS SUPER_READ_ONLY</li> <li>• Fehler #25053286: DIE VERWENDUNG DER ANSICHT MIT BEDINGUNG VERURSACHT FEHLERHAFTES VERHALTEN (behoben in 5.6.36)</li> <li>• Fehler #25586773: FEHLERHAFTES VERHALTEN FÜR DAS ERSTELLEN EINER TABELLENAUSWAHL IN EINER SCHLEIFE IN SP (behoben in 5.6.39)</li> <li>• Fehler #27407480: AUTOMATIC_SP_PRIVILEGES ERFORDERT EINFÜGEBERECHTIGUNGEN FÜR MYSQL.USER-TABELLE</li> <li>• Fehler #26997096: Der relay_log_space -Wert wird nicht synchronisiert aktualisiert, sodass sein Wert manchmal viel höher ist als der tatsächlich von den Relay-Protokollen belegte Festplattenspeicher.</li> <li>• Fehler #15831300 SLAVE_TYPE_CONVERSIONS=ALL_NON_LOSSY FUNKTIONIERT NICHT WIE ERWARTET</li> <li>• SSL-Fehler Backport Fehler #17087862, Fehler #20551271</li> <li>• Fehler #16894092: PERFORMANCE REGRESSION IN 5.6.6+ FOR INSERT INTO ... SELECT ... FROM (behoben in 5.6.15).</li> <li>• Portierung einer Fehlerbehebung im Zusammenhang mit SLAVE_TYPE_CONVERSIONS .</li> </ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 09.11.2020 (Version 1.22.3) (veraltet)</a>	1.22.3	<ul style="list-style-type: none"> <li>• Fehler #26654685: Eine beschädigte Index-ID, die während der Überprüfung eines Fremdschlüssels aufgetreten ist, hat eine Behauptung ausgelöst</li> <li>• Fehler #15831300: Beim Hochsetzen von Ganzzahlen von einem kleineren Typ auf dem Master auf einen größeren Typ auf dem Slave (z. B. von einer <a href="#">SMALLINT-Spalte</a> auf dem Master zu einer <a href="#">BIGINT-Spalte</a> auf dem Slave) werden die hochgesetzten Werte standardmäßig so behandelt, als wären sie signiert. In solchen Fällen ist es nun möglich, dieses Verhalten mit einem oder beiden von <code>ALL_SIGNED</code> und <code>ALL_UNSIGNED</code> aus einer Reihe von Werten, die in der für die Systemvariable <a href="#">slave_type_conversions</a> angegebenen werden, zu ändern oder zu überschreiben. Weitere Informationen finden Sie unter <a href="#">Zeilen-basierte Replikation: Attribuförderung und -herabstufung</a> sowie in der Beschreibung der Variable.</li> <li>• Fehler #17449901: Mit <code>foreign_key_checks=0</code> ermöglicht InnoDB das Löschen eines Index, der für eine Fremdschlüsseinschränkung erforderlich ist, wodurch die Tabelle inkonsistent wurde und die beim Laden der Tabelle aufgetretene Fremdschlüsselprüfung fehlschlug. InnoDB verhindert jetzt, dass ein Index gelöscht wird, der für eine Fremdschlüsseinschränkung erforderlich ist, selbst bei <code>foreign_key_checks=0</code>. Die Fremdschlüsseinschränkung muss entfernt werden, bevor der Fremdschlüsselindex gelöscht wird.</li> <li>• FEHLER #20768847: <a href="#">ALTER TABLE ... Eine DROP INDEX-</a>Operation für eine Tabelle mit Fremdschlüsselabhängigkeiten hat eine Behauptung ausgelöst.</li> </ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 25.11.2019 (Version 1.22.0) (veraltet)</a>	1.22.0	<ul style="list-style-type: none"> <li>• Bug#16346241 - SERVER CRASH IN ITEM_PARA M::QUERY_VAL_STR</li> <li>• Bug#17733850 - NAME_CONST() CRASH IN ITEM_NAME_CONST::ITEM_NAME_CONST()</li> <li>• Bug #20989615 - INNODB AUTO_INCREMENT PRODUCES SAME VALUE TWICE</li> <li>• Bug #20181776 - ACCESS CONTROL DOESN'T MATCH MOST SPECIFIC HOST WHEN IT CONTAINS WILDCARD</li> <li>• Bug #27326796 - MYSQL CRASH WITH INNODB ASSERTION FAILURE IN FILE PARS0PARS.CC</li> <li>• Bug #20590013 - IF YOU HAVE A FULLTEXT INDEX AND DROP IT YOU CAN NO LONGER PERFORM ONLINE DDL</li> </ul>
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 25.11.2019 (Version 1.21.0) (veraltet)</a>	1.21.0	<ul style="list-style-type: none"> <li>• Fehler #19929406: HANDLE_FATAL_SIGNAL (SIG=11) IN __MEMMOVE_SSSE3_BACK FROM STRING::COPY</li> <li>• Fehler #17059925: Für <a href="#">UNION</a>-Anweisungen wurde der zeilengeprüfte Wert falsch berechnet. Dies zeigte sich in Gestalt zu großer Werte für die Spalte ROWS_EXAMINED von Leistungsschema-Anweisungstabellen (wie z. B. <a href="#">events_statements_current</a>).</li> <li>• Fehler #11827369: Einige Abfragen mit verschachtelten SELECT . . . FROM DUAL-Unterabfragen erhöhten eine Assertion.</li> <li>• Fehler #16311231: Falsche Ergebnisse wurden zurückgegeben, wenn eine Abfrage eine Unterabfrage in einer IN-Klausel enthielt, die eine <a href="#">XOR</a>-Operation in der WHERE-Klausel enthielt.</li> </ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 11.11.2019 (Version 1.20.0) (veraltet)</a>	1.20.0	<ul style="list-style-type: none"> <li>• Fehler #19929406: HANDLE_FATAL_SIGNAL (SIG=11) IN __MEMMOVE_SSSE3_BACK FROM STRING::COPY</li> <li>• Fehler #17059925: Für <a href="#">UNION</a>-Anweisungen wurde der zeilengeprüfte Wert falsch berechnet. Dies zeigte sich in Gestalt zu großer Werte für die Spalte ROWS_EXAMINED von Leistungsschema-Anweisungstabellen (wie z. B. <a href="#">events_statements_current</a>).</li> <li>• Fehler #11827369: Einige Abfragen mit verschachtelten SELECT ... FROM DUAL-Unterabfragen erhöhten eine Assertion.</li> <li>• Fehler #16311231: Falsche Ergebnisse wurden zurückgegeben, wenn eine Abfrage eine Unterabfrage in einer IN-Klausel enthielt, die eine <a href="#">XOR</a>-Operation in der WHERE-Klausel enthielt.</li> </ul>
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 19.09.2019 (Version 1.19.5) (veraltet)</a>	1.19.5	<ul style="list-style-type: none"> <li>• <a href="#">CVE-2018-2696</a></li> <li>• <a href="#">CVE-2015-4737</a></li> <li>• Fehler #19929406: HANDLE_FATAL_SIGNAL (SIG=11) IN __MEMMOVE_SSSE3_BACK FROM STRING::COPY</li> <li>• Fehler #17059925: Für <a href="#">UNION</a>-Anweisungen wurde der zeilengeprüfte Wert falsch berechnet. Dies manifestierte sich in zu großen Werten für die Spalte ROWS_EXAMINED von Performance-Schema-Anweisungstabellen (z. B. <a href="#">events_statements_current</a>).</li> <li>• Fehler #11827369: Einige Abfragen mit verschachtelten SELECT ... FROM DUAL-Unterabfragen erhöhten eine Assertion.</li> <li>• Fehler #16311231: Falsche Ergebnisse wurden zurückgegeben, wenn eine Abfrage eine Unterabfrage in einer IN-Klausel enthielt, die eine <a href="#">XOR</a>-Operation in der WHERE-Klausel enthielt.</li> </ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 07.02.2019 (Version 1.19.0) (veraltet)</a>	1.19.0	<ul style="list-style-type: none"> <li>• BUG #32917: DETECT ORPHAN TEMP-POOL FILES, AND HANDLE GRACEFULLY</li> <li>• BUG #63144 CREATE TABLE IF NOT EXISTS METADATA LOCK IS TOO RESTRICTIVE</li> </ul>
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 17.01.2019 (Version 1.17.8) (veraltet)</a>	1.17.8	<ul style="list-style-type: none"> <li>• Fehler 13418638: CREATE TABLE IF NOT EXISTS METADATA LOCK IS TOO RESTRICTIVE</li> </ul>
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 08.10.2018 (Version 1.17.7) (veraltet)</a>	1.17.7	<ul style="list-style-type: none"> <li>• Das Verwerfen eines Index auf einer Fremdschlüsselspalte führt zu einer fehlenden Tabelle. (Fehler #16208542)</li> <li>• Speicherleck in add_derived_key(). (Fehler #76349)</li> <li>• Für partitionierte Tabellen liefern Abfragen unterschiedliche Ergebnisse, je nachdem, ob Index Merge verwendet wurde. (Fehler #16862316)</li> <li>• Abfragen mit der index_merge-Optimierung (siehe <a href="#">Index Merge Optimization</a>) können ungültige Ergebnisse liefern, wenn sie an Tabellen ausgeführt werden, die mithilfe von HASH partitioniert wurden. (Fehler #17588348)</li> </ul>
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 06.09.2018 (Version 1.17.6) (veraltet)</a>	1.17.6	<ul style="list-style-type: none"> <li>• Für eine <a href="#">ALTER TABLE</a>-Anweisung, durch die der Standardwert einer <a href="#">BINARY</a>-Spalte umbenannt oder geändert wurde, ist die Änderung mit einer Tabellenkopie und nicht direkt durchgeführt worden. (Fehler #67141, Fehler #14735373, Fehler #69580, Fehler #17024290)</li> <li>• Eine äußere Join-Operation zwischen einer regulären Tabelle und einer abgeleiteten Tabelle, die implizit aus Gruppen besteht, könnte zum Beenden des Servers führen. (Fehler #16177639)</li> </ul>



Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 13.03.2018 (Version 1.17) (veraltet)</a>	1.17.0	<ul style="list-style-type: none"> <li>• LAST_INSERT_ID wird bei Nutzung von Replikationsfiltern fehlerhaft repliziert (Fehler 69861).</li> <li>• Abfrage gibt abhängig von der INDEX_MERGE-Einstellung unterschiedliche Ergebnisse zurück (Fehler 16862316).</li> <li>• Abfrageausführung führt gespeicherte Routine erneut aus, ineffizienter Abfrageplan (Fehler 16346367).</li> <li>• InnoDB-FTS: Assert in FTS_CACHE_APPEND_D ELETED_DOC_IDS (Fehler 18079671).</li> <li>• Assert RBT_EMPTY(INDEX_CACHE-&gt;WORDS) in ALTER TABLE CHANGE COLUMN (Fehler 17536995)</li> <li>• InnoDB-Volltextsuche findet keine Datensätze, wenn Savepoints involviert sind (Fehler #70333, Fehler #17458835).</li> </ul>
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 20.11.2017 (Version 1.15.1) (veraltet)</a>	1.15.1	<ul style="list-style-type: none"> <li>• Zurückgesetzt: — MySQL-Instance verhindert Ausführung des SYNC-Index (Fehler 73816)</li> <li>• Zurückgesetzt: Assert RBT_EMPTY(INDEX_CACHE-&gt;WORDS) in ALTER TABLE CHANGE COLUMN (Fehler 17536995)</li> <li>• Zurückgesetzt: InnoDB-Volltextsuche findet keine Datensätze, wenn Savepoints involviert sind (Fehler 70333)</li> </ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 24.10.2017 (Version 1.15) (veraltet)</a>	1.15.0	<ul style="list-style-type: none"> <li>• CREATE USER akzeptiert Plug-in- und Passwort-Hash, ignoriert jedoch Passwort-Hash (Fehler 78033).</li> <li>• Die Partitionierungs-Engine fügt Felder zum Lese-Bitsatz hinzu, damit Einträge über einen partitionierten Index sortiert ausgegeben werden können. Dies führt dazu, dass der Verbindungs-Puffer versucht, nicht benötigte Felder zu lesen. Behoben, indem nicht alle Partitionierungsfelder zu read_set hinzugefügt werden, sondern stattdessen nur basierend auf den bereits festgelegten Präfix-Feldern in read_set sortiert wird. DEBUG_ASSERT hinzugefügt, damit bei der Ausführung von key_cmp zumindest das erste Feld gelesen werden muss (Fehler 16367691).</li> <li>• MySQL-Instance verhindert Ausführung des SYNC-Index (Fehler 73816)</li> <li>• Assert RBT_EMPTY(INDEX_CACHE-&gt;WORDS) in ALTER TABLE CHANGE COLUMN (Fehler 17536995)</li> <li>• InnoDB-Volltextsuche findet keine Datensätze, wenn Savepoints involviert sind (Fehler 70333).</li> </ul>
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 13.03.2018 (Version 1.14.4) (veraltet)</a>	1.14.4	<ul style="list-style-type: none"> <li>• Ignorierbare Ereignisse funktionieren nicht und werden nicht getestet (Fehler 74683).</li> <li>• FEHLER DES NEUEN-&gt;ALTEN ASSERTS 'GTID_MODE &gt; 0' (Fehler 20436436).</li> </ul>
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 07.08.2017 (Version 1.14) (veraltet)</a>	1.14.0	Eine Volltextsuche in Verbindung mit abgeleiteten Tabellen (Unterabfragen in der FROM-Klausel) führte zu einem Austritt des Servers. Wenn nun eine Volltextsuche von einer abgeleiteten Tabelle abhängt, gibt der Server eine Fehlermeldung aus, dass bei einer materialisierten Tabelle keine Volltextsuche durchgeführt werden kann. (Fehler #68751, Fehler #16539903)

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 15.05.2017 (Version 1.13) (veraltet)</a>	1.13.0	<ul style="list-style-type: none"><li>• Erneutes Laden einer Tabelle, die während eines Leerlaufs exmittiert wurde, führte zum Zurücksetzen des AUTO_INCREMENT-Werts. (Fehler #21454472, Fehler #77743)</li><li>• Ein Index-Datensatz wurde beim Rollback aufgrund von Unstimmigkeiten in der purge_node_t-Struktur nicht gefunden. Die Unstimmigkeiten führten zu Warn- und Fehlermeldungen wie „Fehler im sec index entry-Update“, „kann einen Datensatz nicht bereinigen“ und „versuchte, einen nicht zur Löschung markierten sec-Indexeintrag zu bereinigen“. (Fehler #19138298, Fehler #70214, Fehler #21126772, Fehler #21065746)</li><li>• Falsche Berechnung der Stack-Größe für qsort-Operation führt zu Stack-Overflow. (Fehler #73979)</li><li>• Datensatz nach Rollback in keinem Index gefunden. (Fehler #70214, Fehler #72419)</li><li>• ALTER TABLE fügt Spalte TIMESTAMP beim Update hinzu, CURRENT_TIMESTAMP fügt ZERO-Daten ein (Fehler #17392)</li></ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 05.04.2017 (Version 1.12) (veraltet)</a>	1.12.0	<ul style="list-style-type: none"> <li>• Erneutes Laden einer Tabelle, die während eines Leerlaufs exmittiert wurde, führte zum Zurücksetzen des AUTO_INCREMENT-Werts. (Fehler #21454472, Fehler #77743)</li> <li>• Ein Index-Datensatz wurde beim Rollback aufgrund von Unstimmigkeiten in der purge_node_t-Struktur nicht gefunden. Die Unstimmigkeiten führten zu Warn- und Fehlermeldungen wie „Fehler im sec index entry-Update“, „kann einen Datensatz nicht bereinigen“ und „versuchte, einen nicht zur Löschung markierten sec-Indexeintrag zu bereinigen“. (Fehler #19138298, Fehler #70214, Fehler #21126772, Fehler #21065746)</li> <li>• Falsche Berechnung der Stack-Größe für qsort-Operation führt zu Stack-Overflow. (Fehler #73979)</li> <li>• Datensatz nach Rollback in keinem Index gefunden. (Fehler #70214, Fehler #72419)</li> <li>• ALTER TABLE fügt Spalte TIMESTAMP beim Update hinzu, CURRENT_TIMESTAMP fügt ZERO-Daten ein (Fehler #17392)</li> </ul>
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 23.02.2017 (Version 1.11) (veraltet)</a>	1.11.0	<ul style="list-style-type: none"> <li>• Das gleichzeitige Ausführen von ALTER-Tabelle und Fremdschlüssel-DROP mit einer anderen DROP-Operation führt dazu, dass die Tabelle verschwindet. (Fehler #16095573)</li> <li>• Einige INFORMATION SCHEMA-Abfragen die ORDER BY verwendet haben, haben nicht wie zuvor Filesort-Optimierung verwendet. (Fehler #16423536)</li> <li>• FOUND_ROWS () gibt die falsche Zählung von Zeilen in einer Tabelle zurück. (Fehler #68458)</li> <li>• Der Server fällt aus, anstatt einen Fehler zurückzugeben, wenn zu viele temp-Tabellen geöffnet sind. (Fehler #18948649)</li> </ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 14.12.2016 (Version 1.10) (veraltet)</a>	1.10.0	<ul style="list-style-type: none"> <li>• UNION von abgeleiteten Tabellen gibt die falschen Ergebnisse mit "1=0/false"-Klauseln zurück. (Fehler #69471)</li> <li>• Serverausfälle in ITEM_FUNC_GROUP_CONCAT::FIX_FIELDS bei 2. Ausführung einer gespeicherten Prozedur. (Fehler #20755389)</li> <li>• Vermeidet ein Hinauszögern von MySQL-Abfragen für zu lange andauernde FTS-Cache-Sync-To-Disk-Vorgänge durch das Abladen des Cache-Sync-Vorgangs in einem separaten Thread, sobald die Größe des Cache 10 % der Gesamtgröße überschreitet. (Fehler #22516559, #73816)</li> </ul>
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 26.10.2016 (Version 1.8.1) (veraltet)</a>	1.8.1	<ul style="list-style-type: none"> <li>• OpenSSL hat aufgrund des Problems die Diffie-Hellman-Schlüssellängen-Parameter geändert. LogJam (Fehler #18367167)</li> </ul>
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 18.10.2016 (Version 1.8) (veraltet)</a>	1.8.0	<ul style="list-style-type: none"> <li>• Wenn alle Indexe in einer Spalte mit mehreren Indexen verworfen wurden, konnte InnoDB eine DROP INDEX-Operation nicht blockieren, wenn eine Fremdschlüsselbeschränkung einen Index erfordert. (Fehler #16896810)</li> <li>• Lösung des Ausfalls beim Hinzufügen von Fremdschlüsselbeschränkung. (Fehler #16413976)</li> <li>• Behebung eines Ausfalls, wenn gleichzeitig ein Cursor in einer gespeicherten Prozedur abgerufen und eine Analyse oder Bereinigung der Tabelle durchgeführt wird. (Fehler #18158639)</li> <li>• Behebung eines Fehlers bei der automatischen Zählung, wenn ein Benutzer eine Tabelle verändert, um den AUTO_INCREMENT-Wert in weniger als den maximalen Spaltenwert für automatische Zählung zu ändern. (Fehler #16310273)</li> </ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 30.08.2016 (Version 1.7.0) (veraltet)</a>	1.7.0	<ul style="list-style-type: none"> <li>• Verbesserte Skalierbarkeit durch Partitionierung der LOCK_grant-Sperre. (Port WL #8355)</li> <li>• Das Öffnen des Cursors mit SELECT in einer gespeicherten Prozedur verursacht segfault. (Port-Fehler#16499751)</li> <li>• MySQL gibt das falsche Ergebnis mit einer ungewöhnlichen Nutzung aus. (Fehler #11751794)</li> <li>• Ausfall in GET_SEL_ARG_FOR_KEYPART – verursacht durch Patch für Fehler #11751794. (Fehler #16208709)</li> <li>• Falsche Ergebnisse für eine einfache Abfrage mit GROUP_BY. (Fehler #17909656)</li> <li>• Zusätzliche Zeilen für semijoin-Abfragen mit Bereichseigenschaften. (Fehler #16221623)</li> <li>• Hinzufügen einer ORDER BY-Klausel gefolgt von einer IN-Unterabfrage könnte dazu führen, dass Duplikatzeilen zurückgegeben werden. (Fehler #16308085)</li> <li>• Ausfall mit der Erklärung, dass eine Abfrage mit ungenauem Scan für GROUP BY, MyISAM vorliegt. (Fehler #16222245)</li> <li>• Ungenauer Index-Scan mit Ganzzahl-Prädikat in Anführungszeichen gibt zufällige Daten zurück. (Fehler #16394084)</li> <li>• Wenn der Optimierer einen ungenauen Index-Scan verwendet hat, konnte der Server während des Versuchs, eine temporäre Tabelle zu erstellen, beendet werden. (Fehler #16436567)</li> <li>• COUNT(DISTINCT) sollte keine NULL-Werte zählen. Diese wurden jedoch gezählt, als der Optimierer ungenauen Index-Scan verwendet hatte. (Fehler #17222452)</li> <li>• Wenn eine Abfrage sowohl MIN()/MAX() als auch aggregate_function(DISTINCT) (z. B. SUM(DISTINCT)) aufwies und mithilfe von ungenauem Index-Scan ausgeführt wurde, wurden die Ergebniswerte von MIN()/MAX() falsch festgelegt. (Fehler #17217128)</li> </ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 01.06.2016 (Version 1.6.5) (veraltet)</a>	1.6.5	<ul style="list-style-type: none"> <li>• SLAVE CAN'T CONTINUE REPLICATION AFTER MASTER'S CRASH RECOVERY (Port Bug #17632285)</li> </ul>
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 06.04.2016 (Version 1.6) (veraltet)</a>	1.6.0	<ul style="list-style-type: none"> <li>• BACKPORT-Fehler 18694052 FIX FOR ASSERTION `! M_ORDERED_REC_BUFFER' FAILED TO 5.6 (Port-Fehler 18305270)</li> <li>• SEGV IN MEMCPY(), HA_PARTITION::POSITION (Port-Fehler # 18383840)</li> <li>• FALSCHER ERGEBNISSE BEI PARTITIONIERUNG, IN DEX_MERGE UND KEIN PK (Port-Fehler # 18167648)</li> <li>• TABELLEN FÜR EXPORT BEREINIGEN: ASSERTION IN HA_PARTITION::EXTRA (Port-Fehler # 16943907)</li> <li>• SERVER-AUSFALL IN VIRTUELLEM HA_ROWS_HANDLER::MULTI_RANGE_READ_INFO_CONST (Port-Fehler # 16164031)</li> <li>• BEREICHSOPTIMIERER STÜRZT IN SEL_ARG::RB_INSERT() AB (Port-Fehler # 16241773)</li> </ul>
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 11.01.2016 (Version 1.5) (veraltet)</a>	1.5.0	<ul style="list-style-type: none"> <li>• Behebung einer unvollständigen Lösung in der MySQL-Volltextsuche, die sich auf Tabellen auswirkt, bei denen der Datenbankname mit einer Ziffer beginnt. (Port Fehler #17607956)</li> </ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 03.12.2015 (Version 1.4) (veraltet)</a>	1.4	<ul style="list-style-type: none"><li>• SEGV in FTSPARSE(). (Fehler #16446108)</li><li>• InnoDB-Datenwörterbuch wird während der Umbenennung der Spalte nicht aktualisiert. (Fehler #19465984)</li><li>• FTS-Ausfall nach der Umbenennung einer Tabelle für eine andere Datenbank. (Fehler #16834860)</li><li>• Fehlgeschlagene Vorbereitung eines Auslösers für trunkierte Tabellen verursacht Fehler 1054. (Fehler #18596756)</li><li>• Änderungen der Metadaten könnten zu Problemen bei der Ausführung von Auslösern führen. (Fehler #18684393)</li><li>• Materialisierung wird nicht für das long-Feld UTF8 VARCHAR ausgewählt. (Fehler #17566396)</li><li>• Schlechter Ausführungsplan, wenn ORDER BY mit Limit X. (Fehler #16697792)</li><li>• Backport-Fehler #11765744 IN 5.1, 5.5 UND 5.6. (Fehler #17083851)</li><li>• Mutex-Problem in SQL/SQL_SHOW.CC führt zu SIG6. Quelle vermutlich FILL_VARIABLES. (Fehler #20788853)</li><li>• Backport-Fehler #18008907 in Versionen 5.5+. (Fehler #18903155)</li><li>• Adaptionfix für einen Stack-Overflow-Fehler in MySQL 5.7. (Fehler #19678930)</li></ul>



Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 16.10.2015 (Versionen 1.2, 1.3) (veraltet)</a>	1.2, 1.3	<ul style="list-style-type: none"> <li>• Beenden einer Abfrage innerhalb innodb verursacht einen Ausfall mit einer Assertion. (Fehler #1608883)</li> <li>• Bei Fehlschlägen beim Erstellen eines neuen Threads für den Ereignis-Scheduler, die Ereignis-Ausführung oder neue Verbindung, wurde keine Mitteilung in das Fehlerprotokoll geschrieben. (Fehler #16865959)</li> <li>• Wenn eine Verbindung ihre standardmäßige Datenbank ändert und eine andere Verbindung simultan SHOW PROCESSLIST ausführt, könnte die zweite Verbindung auf ungültigen Arbeitsspeicher zugreifen, wenn sie versucht, den standardmäßigen Datenbankarbeitsspeicher der ersten Verbindung anzuzeigen. (Fehler #11765252)</li> <li>• PURGE BINARY LOGS entfernt standardmäßig keine Binärprotokolldateien, die verwendet werden oder aktiv sind, gab aber keinen Hinweis, wenn dies stattfand. (Fehler #13727933)</li> <li>• Bei einigen Statements könnte es zu Lecks im Arbeitsspeicher kommen, wenn der Optimierer unbenötigte Unterabfrage-Klassen entfernt hat. (Fehler #15875919)</li> <li>• Während des Herunterfahrens könnte der Server versuchen, einen nicht initialisierten Mutex zu sperren. (Fehler #16016493 )</li> <li>• Ein vorbereitetes Statement, das GROUP CONCAT() und eine ORDER BY-Klausel verwendet hat, das mehrere Spalten benannt hat, könnte dazu führen, dass der Server beendet wird. (Fehler #16075310)</li> <li>• Leistungsschema-Instrumentierung fehlte bei Replikat-Worker-Threads. (Fehler #16083949)</li> <li>• STOP SLAVE kann einen Deadlock verursachen, wenn sie gleichzeitig mit einer Anweisung wie SHOW STATUS ausgegeben wird, die die Werte für eine oder mehrere der Statusvariablen <code>Slave_retried_transactions</code> ,</li> </ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
		<p>Slave_heartbeat_period , Slave_received_heartbeats , Slave_last_heartbeat oder Slave_running abgerufen hat. (Fehler #16088188)</p> <ul style="list-style-type: none"> <li>• Eine vollständige Abfrage mithilfe des booleschen Modus konnte in einigen Fällen keine Ergebnisse liefern, wenn der Suchbegriff in Anführungszeichen stand. (Fehler #16206253)</li> <li>• Der Versuch des Optimierers, redundante Unterabfrage-Klauseln zu entfernen, erhöhte eine Assertion, wenn ein vorbereitetes Statement mit einer Unterabfrage in der On-Klausel einer Verbindung in einer Unterabfrage ausgeführt wurde. (Fehler #16318585)</li> <li>• GROUP_CONCAT instabil, Ausfall in ITEM_SUM::CLEAN_UP_AFTER_REMOVAL. (Fehler #16347450)</li> <li>• Der Versuch, die standardmäßige InnoDB-Volltextsuche-Stoppwortliste (FTS) zu ersetzen, indem eine InnoDB-Tabelle mit derselben Struktur wie INFORMATION_SCHEMA.INNODB_FT_DEFAULT_STOPWORD erstellt wird, endete in einem Fehler. (Fehler #16373868)</li> <li>• Nachdem der Client-Thread auf einem Worker ein FLUSH TABLES WITH READ LOCK ausgeführt hat und darauf einige Updates auf dem Master folgten, stürzte der Worker ab, wenn ausgeführt wurde SHOW SLAVE STATUS. (Fehler #16387720)</li> <li>• Beim Parsen einer beschränkten Suchzeichenfolge wie "abc-def" in einer Volltextsuche verwendet InnoDB jetzt dieselben Trennzeichen wie MyISAM. (Fehler #16419661)</li> <li>• Ausfall in FTS_AST_TERM_SET_WILDCARD. (Fehler #16429306)</li> <li>• SEGFAULT in FTS_AST_VISIT() for FTS RQG Test. (Fehler #16435855)</li> <li>• Wenn bei Debug-Builds der Optimierer ein Item_ref entfernte, das auf eine Unterabfrage verwies, führte das zum Beenden des Servers. (Fehler #16509874)</li> </ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
		<ul style="list-style-type: none"> <li>• Volltextsuche in InnoDB-Tabellen schlug fehl bei der Suche nach wörtlichen Sätzen in Kombination mit den Operatoren + oder -. (Fehler #16516193)</li> <li>• <code>START SLAVE</code> schlug fehl, als der Server mit den Optionen <code>--master-info-repository=TABLE relay-log-info-repository=TABLE</code> gestartet wurde und Autocommit auf 0 gesetzt war, zusammen mit <code>--skip-slave-start</code> (Fehler #16533802)</li> <li>• Sehr große InnoDB-Volltextsuchergebnisse (FTS) konnten in einigen Fällen eine übermäßige Menge an Arbeitsspeicher verbrauchen. (Fehler #16625973)</li> <li>• Bei Debug-Builds konnte in <code>OPT_CHECK_ORDER_BY</code> eine Assertion auftreten, wenn ein Binärprogramm direkt in der Suchzeichenfolge verwendet wurde, da Binärprogramme NULL-Bytes und andere bedeutungslose Zeichen enthalten können. (Fehler #16766016)</li> <li>• Bei einigen Statements könnte es zu Lecks im Arbeitsspeicher kommen, wenn der Optimierer unbenötigte Unterabfrage-Klassen entfernt hat. (Fehler #16807641)</li> <li>• Es war möglich, einen Deadlock zu verursachen, nachdem <code>FLUSH TABLES WITH READ LOCK</code> durch Verwendung von <code>STOP SLAVE</code> in einer neuen Verbindung an den Worker erteilt wurde und anschließend <code>SHOW SLAVE STATUS</code> über die ursprüngliche Verbindung erteilt wurde. (Fehler #16856735)</li> <li>• <code>GROUP_CONCAT()</code> mit einem ungültigen Trennzeichen konnte zum Beenden des Servers führen. (Fehler #16870783)</li> <li>• Der Server führte übermäßiges Sperren für die Mutexe <code>LOCK_active_mi</code> und <code>active_mi-&gt;rli-&gt;data_lock</code> jeder <code>SHOW STATUS LIKE 'pattern'</code>-Anweisung aus, selbst wenn das Muster nicht mit den Statusvariablen übereinstimmte, die diese Mutexe verwenden (<code>Slave_heartbeat_period</code>, <code>Slave_last_heartbeat</code>, <code>Slave_received_hea</code></li> </ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
		<p>rtbeats , Slave_retried_transactions , Slave_running ). (Fehler #16904035)</p> <ul style="list-style-type: none"> <li>• Eine Volltextsuche, die den IN BOOLEAN MODE-Modifikator verwendet, konnte in einem Assertionfehler enden. (Fehler #16927092)</li> <li>• Volltextsuche in InnoDB-Tabellen schlug fehl bei Suchen, die den booleschen Operator + verwendet haben. (Fehler #17280122)</li> <li>• 4-Wege-Deadlock: Zombies, Binärprotokolle bereinigen, Prozesslisten anzeigen, Binärprotokolle anzeigen. (Fehler #17283409)</li> <li>• Wenn ein auf ein Commit-Lock wartender SQL-Thread beendet und neu gestartet wurde, führte das zum Überspringen einer Transaktion auf dem Worker. (Fehler #17450876)</li> <li>• Ein InnoDB-Volltextsuchen-Fehler konnte aufgrund eines "unbeendeten" Tokens auftreten. Die Zeichenfolge und die Zeichenfolgenlänge sollten für einen Zeichenfolgenvergleich weitergeleitet werden. (Fehler #17659310)</li> <li>• Eine große Anzahl an partitionierten InnoDB-Tabellen konnte, wenn diese in MySQL 5.6 oder 5.7 verwendet wurden, viel mehr Arbeitsspeicher verbrauchen als dieselben Tabellen in vorherigen Versionen von MySQL Server. (Fehler #17780517)</li> <li>• Bei Volltextabfragen konnte ein Fehler beim Überprüfen, ob num_token kleiner ist als max_proximity_item, in einer Assertion enden. (Fehler #18233051)</li> <li>• Bestimmte Abfragen für die INFORMATION SCHEMA TABLES- und COLUMNS-Tabellen konnten zu übermäßigem Arbeitsspeicherverbrauch führen, wenn es eine große Anzahl an leeren InnoDB-Tabellen gab. (Fehler #18592390)</li> <li>• Beim Übergeben einer Transaktion wird jetzt eine Kennzeichnung verwendet, um zu überprüfen, ob ein Thread erstellt wurde, anstatt den Thread selbst zu überprüfen, was mehr</li> </ul>

Datenbank-Engine-Update	Version	Behobene MySQL-Fehler
		<p>Ressourcen benötigte, besonderen wenn der Server mit <code>master_info_repository=TABLE</code> ausgeführt wird. (Fehler #18684222)</p> <ul style="list-style-type: none"> <li>• Wenn ein Client-Thread in einem Worker <code>FLUSH TABLES WITH READ LOCK</code> ausgeführt hat, während der Master eine DML ausführte, wurde das Ausführen von <code>SHOW SLAVE STATUS</code> auf demselben Client blockiert, was zu einem Deadlock führte. (Fehler #19843808)</li> <li>• Das Ordnen nach einem <code>GROUP_CONCAT()</code> -Ergebnis konnte zum Beenden des Servers führen. (Fehler #19880368)</li> </ul>
<a href="#">Aurora-MySQL-Datenbank-Engine-Updates 24.08.2015 (Version 1.1) (veraltet)</a>	1.1	<ul style="list-style-type: none"> <li>• InnoDB-Datenbanken mit Namen, die mit einer Zahl beginnen, verursachen einen Fehler im Volltextsuchparser (FTS). (Fehler #17607956)</li> <li>• InnoDB-Volltextsuchen schlagen in Datenbanken fehl, deren Namen mit einer Zahl beginnen. (Fehler #17161372)</li> <li>• Für InnoDB-Datenbanken in Windows, ist die Objekt-ID für Volltextsuchen (FTS) nicht im erwarteten Hexadezimalformat. (Fehler #16559254)</li> <li>• Eine in MySQL 5.6 eingeführte Code-Regression hatte negative Auswirkungen auf die Leistung von <code>DROP TABLE</code> und <code>ALTER TABLE</code>. Dies führt möglicherweise zu einer Leistungsminderung zwischen MySQL Server 5.5.x und 5.6.x. (Fehler #16864741)</li> </ul>

# In Aurora MySQL behobene Sicherheitsschwachstellen

Common Vulnerabilities and Exposures (CVE) ist eine Liste von Einträgen für öffentlich bekannte Cybersicherheitsschwachstellen. Jeder Eintrag enthält eine Identifikationsnummer, eine Beschreibung und mindestens einen öffentlichen Verweis.

Auf dieser Seite finden Sie eine Liste der in Aurora MySQL behobenen Sicherheitsschwachstellen. Allgemeine Informationen zur Sicherheit von Aurora finden Sie unter [Sicherheit in Amazon Aurora](#) im Amazon-Aurora-Benutzerhandbuch. Zusätzliche Informationen zur Sicherheit von Aurora MySQL finden Sie unter [Sicherheit in Amazon Aurora MySQL](#) im Amazon-Aurora-Benutzerhandbuch.

Wir empfehlen Ihnen, immer auf die neueste Version von Aurora zu aktualisieren, um gegen bekannte Schwachstellen geschützt zu sein. Auf dieser Seite können Sie überprüfen, ob mit einer bestimmten Version von Aurora MySQL eine bestimmte Sicherheitsschwachstelle behoben wird. Wenn Ihr Cluster nicht über den Sicherheitspatch verfügt, können Sie sehen, auf welche Aurora MySQL-Version Sie für diesen Patch aktualisieren sollten.

CVEs, die in Aurora MySQL Version 1, 2 und 3 behoben wurden, sind auch in den Versionshinweisen für diese Version aufgeführt:

- [Aktualisierungen der Datenbank-Engine für Amazon-Aurora-MySQL-Version 3](#)
- [Aktualisierungen der Datenbank-Engine für Amazon Aurora MySQL Version 2](#)
- [Aktualisierungen der Datenbank-Engine für Amazon Aurora MySQL Version 1 \(veraltet\)](#)

## Note

Die erste Version von Aurora-MySQL-Version 3 enthält alle CVEs, die für die Community-MySQL 8.0.23 festgelegt sind. Für zukünftige CVEs, die repariert sind, suchen Sie nach ihnen, die hier und in den Versionshinweisen der Aurora-MySQL-Version 3 aufgeführt sind.

CVEs und gepatchte Aurora MySQL-Mindestversionen

- [CVE-2024-20963:2.12.2 , 2.11.5](#)
- [CVE-2023-39975 : 3.07.0, 3.06.0, 3.05.2, 3.04.2, 2.12.2, 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](#), [2.11.5](#)
- [CVE-2023-22084](#):[2.11.5](#)
- [CVE-2023-22053](#): [2.12.1](#)
- [CVE-2023-22028](#):[2.12.1](#) , [2.11.5](#)
- [CVE-2023-22026](#): [2.12.1](#), [2.11.5](#)
- [CVE-2023-22015](#):[2.12.1](#) , [2.11.5](#)
- [CVE-2023-21963](#): [3.04.0](#), [3.03.2](#), [2.12.0](#), [2.11.3](#)
- [CVE-2023-21912](#): [3.04.0](#), [3.03.2](#), [2.12.0](#), [2.11.3](#)
- [CVE-2023-21840](#): [2.12.0](#)
- [CVE-2023-0215](#): [3.04.0](#), [3.03.2](#), [2.12.0](#), [2.11.3](#)
- [CVE-2022-43551](#): [3.04.0](#), [3.03.2](#), [2.12.0](#), [2.11.3](#)
- [CVE-2022-37434](#): [3.05.0](#), [3.04.0](#), [3.03.2](#), [2.12.0](#), [2.11.3](#)
- [CVE-2022-32221](#): [3.03.0](#), [2.12.0](#), [2.11.1](#), [2.07.9](#)
- [CVE-2022-24407](#): [2.12.1](#), [2.11.4](#)
- [CVE-2022-21635](#): [3.04.0](#)
- [CVE-2022-21556](#): [3.04.0](#)
- [CVE-2022-21460](#): [2.11.0](#)
- [CVE-2022-21451](#): [3.03.0](#), [3.02.2](#), [2.11.0](#)
- [CVE-2022-21444](#): [3.03.0](#), [3.02.2](#), [2.11.0](#), [2.10.3](#)
- [CVE-2022-21417](#) : [2.11.0](#)
- [CVE-2022-21352](#): [3.04.0](#)
- [CVE-2022-21344](#): [2.10.3](#)
- [CVE-2022-21304](#): [2.11.0](#), [2.10.3](#)
- [CVE-2022-21303](#): [2.11.0](#)
- [CVE-2022-21245](#): [2.11.0](#), [2.10.3](#), [2.07.8](#)
- [CVE-2022-0778](#): [3.02.1](#), [2.11.0](#)
- [CVE-2021-36222](#): [3.02.2](#), [3.01.1](#), [2.12.0](#), [2.11.1](#), [2.11.0](#), [2.10.3](#), [2.10.2](#), [2.07.8](#)
- [CVE-2021-35630](#): [3.04.0](#)
- [CVE-2021-35624](#): [3.04.0](#), [2.10.2](#)
- [CVE-2021-35604](#): [2.10.2](#)

- [CVE-2021-28196](#): [2.11.0](#)
- [CVE-2021-23841](#): [2.11.0](#), [2.10.0](#), [2.09.3](#), [1.23.3](#)
- [CVE-2021-22946](#): [3.02.0](#), [3.01.1](#), [2.12.0](#)
- [CVE-2021-22926](#): [3.02.2](#), [3.01.1](#), [2.11.1](#), [2.11.0](#), [2.10.3](#), [2.10.22.07.8](#)
- [CVE-2021-3712](#): [2.09.3](#)
- [CVE-2021-3449](#): [2.11.0](#), [2.10.0](#), [2.09.3](#), [1.23.3](#)
- [CVE-2021-2390](#): [2.10.2](#)
- [CVE-2021-2389](#): [2.10.2](#)
- [CVE-2021-2385](#): [2.10.2](#)
- [CVE-2021-2356](#): [2.10.2](#)
- [CVE-2021-2307](#): [2.11.0](#), [2.10.1](#), [2.09.3](#), [1.23.4](#)
- [CVE-2021-2226](#): [2.11.0](#), [2.10.1](#), [2.09.3](#), [1.23.4](#)
- [CVE-2021-2202](#): [2.11.0](#)
- [CVE-2021-2194](#): [2.11.0](#), [2.10.1](#)
- [CVE-2021-2179](#): [2.11.0](#)
- [CVE-2021-2178](#): [2.11.0](#)
- [CVE-2021-2174](#): [2.11.0](#), [2.10.1](#), [2.09.3](#)
- [CVE-2021-2171](#): [2.11.0](#), [2.10.1](#), [2.09.3](#)
- [CVE-2021-2169](#): [2.12.0](#), [2.11.1](#), [2.11.0](#), [2.10.1](#), [2.09.3](#)
- [CVE-2021-2166](#): [2.11.0](#), [2.10.1](#), [2.09.3](#)
- [CVE-2021-2160](#): [2.11.0](#), [2.10.1](#), [1.23.4](#)
- [CVE-2021-2154](#): [2.11.0](#), [2.10.1](#), [2.09.3](#), [1.23.4](#)
- [CVE-2021-2144](#): [2.07.3](#)
- [CVE-2021-2060](#): [2.10.1](#), [2.09.3](#), [1.23.4](#)
- [CVE-2021-2032](#): [2.10.1](#), [2.09.3](#), [1.23.4](#)
- [CVE-2021-2001](#): [2.10.1](#), [2.09.3](#), [1.23.4](#)
- [CVE-2020-28196](#): [2.10.0](#), [2.09.3](#), [1.23.3](#)
- [CVE-2020-14867](#): [1.23.2](#), [1.22.4](#)
- [CVE-2020-14812](#): [2.09.2](#), [2.07.4](#), [1.23.2](#), [1.22.4](#)
- [CVE-2020-14793](#): [2.09.2](#), [2.07.4](#), [1.23.2](#), [1.22.4](#)



- [CVE-2020-14790](#): [2.10.0](#), [2.09.2](#), [2.07.4](#)
- [CVE-2020-14776](#): [2.10.0](#)
- [CVE-2020-14775](#): [2.09.2](#), [2.07.4](#)
- [CVE-2020-14769](#): [2.09.3](#), [2.09.2](#), [2.07.4](#), [1.23.2](#), [1.22.4](#)
- [CVE-2020-14765](#): [2.09.2](#), [2.07.4](#), [1.23.2](#), [1.22.4](#)
- [CVE-2020-14760](#): [2.09.2](#), [2.07.4](#)
- [CVE-2020-14672](#): [2.09.2](#), [2.07.4](#), [1.23.2](#), [1.22.4](#)
- [CVE-2020-14567](#): [2.10.0](#), [2.09.1](#), [2.08.3](#), [2.07.3](#)
- [CVE-2020-14559](#): [2.10.0](#), [2.09.1](#), [2.08.3](#), [2.07.3](#), [1.23.1](#), [1.22.3](#)
- [CVE-2020-14553](#): [2.10.0](#), [2.09.1](#), [2.08.3](#), [2.07.3](#)
- [CVE-2020-14547](#): [2.10.0](#), [2.09.1](#), [2.08.3](#), [2.07.3](#)
- [CVE-2020-14540](#): [2.10.0](#), [2.09.1](#), [2.08.3](#), [2.07.3](#)
- [CVE-2020-14539](#): [2.10.0](#), [1.23.1](#), [1.22.3](#)
- [CVE-2020-11105](#) : [3.07.0](#), [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](#)

# Dokumentverlauf der Aurora-MySQL-Versionshinweise

In der folgenden Tabelle werden die Dokumentationsversionen für die Aurora-MySQL-Versionshinweise beschrieben.

Änderung	Beschreibung	Datum
<a href="#">Aurora MySQL Version 3.07.0, kompatibel mit MySQL 8.0.36</a>	Aurora MySQL Version 3.07.0 ist verfügbar. Diese Version ist mit MySQL 8.0.36 kompatibel. Ausführliche Informationen finden Sie unter <a href="#">Aurora-MySQL-Version 3, kompatibel mit MySQL 8.0</a> .	4. Juni 2024
<a href="#">Aurora MySQL Version 2.11.5, kompatibel mit MySQL 5.7.12</a>	Aurora MySQL Version 2.11.5 ist verfügbar. Diese Version ist mit MySQL 5.7.12 kompatibel. Ausführliche Informationen finden Sie unter <a href="#">Aurora MySQL Version 2, kompatibel mit MySQL 5.7</a> .	26. März 2024
<a href="#">Aurora MySQL Version 2.12.2, kompatibel mit MySQL 5.7.44</a>	Aurora MySQL Version 2.12.2 ist verfügbar. Diese Version ist mit MySQL 5.7.44 kompatibel. Ausführliche Informationen finden Sie unter <a href="#">Aurora MySQL Version 2, kompatibel mit MySQL 5.7</a> .	19. März 2024
<a href="#">Aurora MySQL Version 3.04.2, kompatibel mit MySQL 8.0.28</a>	Aurora MySQL Version 3.04.2 ist verfügbar. Diese Version ist mit MySQL 8.0.28 kompatibel. Ausführliche Informationen finden Sie unter <a href="#">Aurora-My</a>	15. März 2024

---

<a href="#">SQL-Version 3, kompatibel mit MySQL 8.0.</a>		
<a href="#">Aurora MySQL Version 3.06.0, kompatibel mit MySQL 8.0.34</a>	Aurora MySQL Version 3.06.0 ist verfügbar. Diese Version ist mit MySQL 8.0.34 kompatibel. Ausführliche Informationen finden Sie unter <a href="#">Aurora-MySQL-Version 3, kompatibel mit MySQL 8.0.</a>	7. März 2024
<a href="#">Aurora MySQL Version 3.05.2, kompatibel mit MySQL 8.0.32</a>	Aurora MySQL Version 3.05.2 ist verfügbar. Diese Version ist mit MySQL 8.0.32 kompatibel. Ausführliche Informationen finden Sie unter <a href="#">Aurora-MySQL-Version 3, kompatibel mit MySQL 8.0.</a>	31. Januar 2024
<a href="#">Aurora MySQL Version 2.12.1, kompatibel mit MySQL 5.7.40</a>	Aurora MySQL Version 2.12.1 ist verfügbar. Diese Version ist mit MySQL 5.7.40 kompatibel. Ausführliche Informationen finden Sie unter <a href="#">Aurora MySQL Version 2, kompatibel mit MySQL 5.7.</a>	28. Dezember 2023
<a href="#">Aurora MySQL Version 3.03.3, kompatibel mit MySQL 8.0.26</a>	Aurora MySQL Version 3.03.3 ist verfügbar. Diese Version ist mit MySQL 8.0.26 kompatibel. Ausführliche Informationen finden Sie unter <a href="#">Aurora-MySQL-Version 3, kompatibel mit MySQL 8.0.</a>	8. Dezember 2023

---

<a href="#">Aurora MySQL Version 3.05.1, kompatibel mit MySQL 8.0.32</a>	Aurora MySQL Version 3.05.1 ist verfügbar. Diese Version ist mit MySQL 8.0.32 kompatibel. Ausführliche Informationen finden Sie unter <a href="#">Aurora-MySQL-Version 3, kompatibel mit MySQL 8.0</a> .	21. November 2023
<a href="#">Aurora MySQL Version 3.04.1, kompatibel mit MySQL 8.0.28</a>	Aurora MySQL Version 3.04.1 ist verfügbar. Diese Version ist mit MySQL 8.0.28 kompatibel. Ausführliche Informationen finden Sie unter <a href="#">Aurora-MySQL-Version 3, kompatibel mit MySQL 8.0</a> .	13. November 2023
<a href="#">Aurora MySQL Version 3.05.0.1, kompatibel mit MySQL 8.0.32, Beta</a>	Aurora MySQL Version 3.05.0.1 ist verfügbar. Diese Version ist mit MySQL 8.0.32 kompatibel. Ausführliche Informationen finden Sie unter <a href="#">Aurora-MySQL-Version 3, kompatibel mit MySQL 8.0</a> .	30. Oktober 2023
<a href="#">Aurora MySQL Version 3.05.0, kompatibel mit MySQL 8.0.32</a>	Aurora MySQL Version 3.05.0 ist verfügbar. Diese Version ist mit MySQL 8.0.32 kompatibel. Ausführliche Informationen finden Sie unter <a href="#">Aurora-MySQL-Version 3, kompatibel mit MySQL 8.0</a> .	25. Oktober 2023

<a href="#">Aurora MySQL Version 2.12.0.1, kompatibel mit MySQL 5.7.40, Beta</a>	Aurora MySQL Version 2.12.0.1 Beta ist verfügbar. Diese Version ist mit MySQL 5.7.40 kompatibel. Ausführliche Informationen finden Sie unter <a href="#">Aurora MySQL Version 2, kompatibel mit MySQL 5.7.</a>	25. Oktober 2023
<a href="#">Aurora MySQL Version 2.11.4, kompatibel mit MySQL 5.7.12</a>	Aurora MySQL Version 2.11.4 ist verfügbar. Diese Version ist mit MySQL 5.7.12 kompatibel. Ausführliche Informationen finden Sie unter <a href="#">Aurora MySQL Version 2, kompatibel mit MySQL 5.7.</a>	17. Oktober 2023
<a href="#">Aurora MySQL Version 3.03.2, kompatibel mit MySQL 8.0.26</a>	Aurora MySQL Version 3.03.2 ist verfügbar. Diese Version ist mit MySQL 8.0.26 kompatibel. Ausführliche Informationen finden Sie unter <a href="#">Aurora-MySQL-Version 3, kompatibel mit MySQL 8.0.</a>	29. August 2023
<a href="#">Aurora MySQL Version 2.07.10, kompatibel mit MySQL 5.7.12</a>	Aurora MySQL Version 2.07.10 ist verfügbar. Diese Version ist mit MySQL 5.7.12 kompatibel. Ausführliche Informationen finden Sie unter <a href="#">Aurora MySQL Version 2, kompatibel mit MySQL 5.7.</a>	15. August 2023



---

<a href="#"><u>Aurora MySQL Version 3.04.0, kompatibel mit MySQL 8.0.28</u></a>	Aurora MySQL Version 3.04.0 ist verfügbar. Diese Version ist mit MySQL 8.0.28 kompatibel. Ausführliche Informationen finden Sie unter <a href="#"><u>Aurora-MySQL-Version 3, kompatibel mit MySQL 8.0.</u></a>	31. Juli 2023
<a href="#"><u>Aurora MySQL Version 2.12.0, kompatibel mit MySQL 5.7.40</u></a>	Aurora MySQL Version 2.12.0 ist verfügbar. Diese Version ist mit MySQL 5.7.40 kompatibel. Ausführliche Informationen finden Sie unter <a href="#"><u>Aurora MySQL Version 2, kompatibel mit MySQL 5.7.</u></a>	25. Juli 2023
<a href="#"><u>Aurora MySQL Version 2.11.3, kompatibel mit MySQL 5.7.12</u></a>	Aurora MySQL Version 2.11.3 ist verfügbar. Diese Version ist mit MySQL 5.7.12 kompatibel. Ausführliche Informationen finden Sie unter <a href="#"><u>Aurora MySQL Version 2, kompatibel mit MySQL 5.7.</u></a>	9. Juni 2023
<a href="#"><u>Aurora MySQL Version 3.03.1, kompatibel mit MySQL 8.0.26</u></a>	Aurora MySQL Version 3.03.1 ist verfügbar. Diese Version ist mit MySQL 8.0.26 kompatibel. Ausführliche Informationen finden Sie unter <a href="#"><u>Aurora-MySQL-Version 3, kompatibel mit MySQL 8.0.</u></a>	11. Mai 2023

---

<a href="#">Aurora MySQL Version 2.07.9, kompatibel mit MySQL 5.7.12.</a>	Aurora MySQL Version 2.07.9 ist verfügbar. Diese Version ist mit MySQL 5.7.12 kompatibel. Ausführliche Informationen finden Sie unter <a href="#">Aurora MySQL Version 2, kompatibel mit MySQL 5.7.</a>	4. Mai 2023
<a href="#">Aurora MySQL Version 3.02.3, kompatibel mit MySQL 8.0.23</a>	Aurora MySQL Version 3.02.3 ist verfügbar. Diese Version ist mit MySQL 8.0.23 kompatibel. Ausführliche Informationen finden Sie unter <a href="#">Aurora-MySQL-Version 3, kompatibel mit MySQL 8.0.</a>	17. April 2023
<a href="#">Aurora MySQL Version 2.11.2, kompatibel mit MySQL 5.7.12</a>	Aurora MySQL Version 2.11.2 ist verfügbar. Diese Version ist mit MySQL 5.7.12 kompatibel. Ausführliche Informationen finden Sie unter <a href="#">Aurora MySQL Version 2, kompatibel mit MySQL 5.7.</a>	24. März 2023
<a href="#">Aurora MySQL Version 3.03.0, kompatibel mit MySQL 8.0.26</a>	Aurora MySQL Version 3.03.0 ist verfügbar. Diese Version ist mit MySQL 8.0.26 kompatibel. Ausführliche Informationen finden Sie unter <a href="#">Aurora-MySQL-Version 3, kompatibel mit MySQL 8.0.</a>	1. März 2023

---

<a href="#">Aurora MySQL Version 2.11.1, kompatibel mit MySQL 5.7.12.</a>	Aurora MySQL Version 2.11.1 ist verfügbar. Diese Version ist mit MySQL 5.7.12 kompatibel. Ausführliche Informationen finden Sie unter <a href="#">Aurora MySQL Version 2, kompatibel mit MySQL 5.7.</a>	14. Februar 2023
<a href="#">Aurora MySQL Version 3.02.2, kompatibel mit MySQL 8.0.23</a>	Aurora MySQL Version 3.02.2 ist verfügbar. Diese Version ist mit MySQL 8.0.23 kompatibel. Ausführliche Informationen finden Sie unter <a href="#">Aurora-MySQL-Version 3, kompatibel mit MySQL 8.0.</a>	18. November 2022
<a href="#">Aurora MySQL Version 2.10.3, kompatibel mit MySQL 5.7</a>	Aurora MySQL Version 2.10.3 ist verfügbar. Diese Version ist mit MySQL 5.7 kompatibel. Ausführliche Informationen finden Sie unter <a href="#">Aurora MySQL Version 2, kompatibel mit MySQL 5.7.</a>	1. November 2022
<a href="#">Aurora MySQL Version 2.11.0, kompatibel mit MySQL 5.7.12</a>	Aurora MySQL Version 2.11.0 ist verfügbar. Diese Version ist mit MySQL 5.7.12 kompatibel. Ausführliche Informationen finden Sie unter <a href="#">Aurora MySQL Version 2, kompatibel mit MySQL 5.7.</a>	25. Oktober 2022

<a href="#">Aurora MySQL Version 3.02.1, kompatibel mit MySQL 8.0.23</a>	Aurora MySQL Version 3.02.1 ist verfügbar. Diese Version ist mit MySQL 8.0.23 kompatibel. Ausführliche Informationen finden Sie unter <a href="#">Aurora-MySQL-Version 3, kompatibel mit MySQL 8.0</a> .	7. September 2022
<a href="#">Direktes Upgrade für MySQL 5.6-kompatibles Aurora Serverless v1</a>	Sie können ein direktes Upgrade für einen mit MySQL 5.6 kompatiblen Aurora Serverless v1-Cluster vornehmen, um einen vorhandenen Cluster in einen mit MySQL 5.7 kompatiblen Aurora Serverless v1-Cluster zu ändern. Weitere Informationen finden Sie unter <a href="#">Aurora-MySQL-Serverless-5.7-Engine-Updates vom 18.06.2020 (Version 2.07.1)</a> .	16. Juni 2022
<a href="#">Direktes Upgrade für MySQL 5.6-kompatibles Aurora Serverless v1</a>	Sie können ein direktes Upgrade für einen mit MySQL 5.6 kompatiblen Aurora Serverless v1-Cluster vornehmen, um einen vorhandenen Cluster in einen mit MySQL 5.7 kompatiblen Aurora Serverless v1-Cluster zu ändern. Weitere Informationen finden Sie unter <a href="#">Aurora-MySQL-Serverless-5.7-Engine-Updates vom 18.06.2020 (Version 2.07.1)</a> .	16. Juni 2022

---

<a href="#">Aurora-MySQL-Datenbank-Engine-Updates vom 16.06.2022 (Version 2.07.8) sind verfügbar.</a>	Aurora MySQL Version 2.07.8 ist verfügbar.	16. Juni 2022
<a href="#">Aurora MySQL Version 3.02.0, kompatibel mit MySQL 8.0.23</a>	Aurora MySQL Version 3.02.0 ist verfügbar. Diese Version ist mit MySQL 8.0.23 kompatibel. Ausführliche Informationen finden Sie unter <a href="#">Aurora-MySQL-Version 3, kompatibel mit MySQL 8.0.</a>	20. April 2022
<a href="#">Aurora MySQL Version 3.01.1, kompatibel mit MySQL 8.0.23</a>	Aurora MySQL Version 3.01.1 ist verfügbar. Diese Version ist mit MySQL 8.0.23 kompatibel. Ausführliche Informationen finden Sie unter <a href="#">Aurora-MySQL-Version 3, kompatibel mit MySQL 8.0.</a>	15. April 2022
<a href="#">Erstversion</a>	Erste Veröffentlichung der Aurora-MySQL-Versionshinweise.	22. März 2022
<a href="#">Aurora-MySQL-Version 2.10.2</a>	Aurora-MySQL-Version 2.10.2 ist verfügbar.	26. Januar 2022
<a href="#">Aurora-MySQL-Version 2.08.4</a>	Aurora-MySQL-Version 2.08.4 ist verfügbar.	6. Januar 2022
<a href="#">Aurora MySQL Version 2.07.7</a>	Aurora MySQL Version 2.07.7 ist verfügbar.	24. November 2021

---

<a href="#">Aurora MySQL Version 3.01.0, kompatibel mit MySQL 8.0.23</a>	Aurora-MySQL-Version 3.01.0 ist verfügbar. Diese Version ist mit MySQL 8.0.23 kompatibel. Ausführliche Details finden Sie unter <a href="#">Aurora MySQL Version 3, kompatibel mit MySQL 8.0.</a>	18. November 2021
<a href="#">Aurora MySQL Version 2.09.3</a>	Aurora MySQL Version 2.09.3 ist verfügbar.	12. November 2021
<a href="#">Aurora MySQL-Version 2.10.1</a>	Aurora MySQL Version 2.10.1 ist verfügbar.	21. Oktober 2021
<a href="#">Aurora MySQL Version 1.23.4</a>	Aurora MySQL Version 1.23.4 ist verfügbar.	30. September 2021
<a href="#">Aurora MySQL Version 2.07.6</a>	Aurora MySQL Version 2.07.6 ist verfügbar.	2. September 2021
<a href="#">Aurora MySQL Version 2.07.5</a>	Aurora MySQL Version 2.07.5 ist verfügbar.	6. Juli 2021
<a href="#">Aurora MySQL Version 1.23.3</a>	Aurora MySQL Version 1.23.3 ist verfügbar.	28. Juni 2021
<a href="#">Aurora MySQL Version 1.22.5</a>	Aurora MySQL Version 1.22.5 ist verfügbar.	3. Juni 2021

---

<a href="#">Aurora MySQL Version 2.10.0</a>	Aurora MySQL Version 2.10.0 ist verfügbar. Zu den Highlights gehören eine <a href="#">höhere Verfügbarkeit von Reader-Instances während Neustarts des Writers</a> , <a href="#">Verbesserungen des Patches ohne Ausfallzeiten (ZDP)</a> , <a href="#">Verbesserungen des Neustarts ohne Ausfallzeiten (ZDR)</a> und der <a href="#">Binlog-I/O-Cache-Optimierung</a> .	25. Mai 2021
<a href="#">Aurora MySQL Version 1.23.2</a>	Aurora MySQL Version 1.23.2 ist verfügbar.	18. März 2021
<a href="#">Aurora MySQL Version 2.07.4</a>	Aurora MySQL Version 2.07.4 ist verfügbar.	4. März 2021
<a href="#">Aurora MySQL Version 1.22.4</a>	Aurora MySQL Version 1.22.4 ist verfügbar.	4. März 2021
<a href="#">Aurora MySQL Version 2.09.2</a>	Aurora MySQL Version 2.09.2 ist verfügbar.	26. Februar 2021
<a href="#">Aurora MySQL Version 2.09.1</a>	Aurora MySQL Version 2.09.1 ist verfügbar.	11. Dezember 2020
<a href="#">Aurora MySQL Version 1.23.1</a>	Aurora MySQL Version 1.23.1 ist verfügbar.	24. November 2020
<a href="#">Aurora MySQL Version 2.08.3</a>	Aurora MySQL Version 2.08.3 ist verfügbar.	12. November 2020
<a href="#">Aurora MySQL Version 2.07.3</a>	Aurora MySQL Version 2.07.3 ist verfügbar.	10. November 2020
<a href="#">Aurora MySQL Version 1.22.3</a>	Aurora MySQL Version 1.22.3 ist verfügbar.	9. November 2020

---

<a href="#">Aurora MySQL Version 2.09.0</a>	Aurora MySQL-Version 2.09.0 ist verfügbar.	17. September 2020
<a href="#">Aurora MySQL Version 1.23.0</a>	Aurora MySQL-Version 1.23.0 ist verfügbar.	2. September 2020
<a href="#">Aurora MySQL Version 2.08.2</a>	Aurora MySQL Version 2.08.2 ist verfügbar.	28. August 2020
<a href="#">Aurora MySQL Version 2.04.9</a>	Aurora MySQL-Version 2.04.9 ist verfügbar.	14. August 2020
<a href="#">Aurora MySQL Version 2.08.1</a>	Aurora MySQL-Version 2.08.1 ist verfügbar.	18. Juni 2020
<a href="#">Aurora MySQL Version 1.22.2 für Parallelabfrage-Cluster</a>	Aurora MySQL-Version 1.22.2 ist verfügbar, wenn Sie einen Parallelabfragecluster erstellen .	18. Juni 2020
<a href="#">Aurora MySQL Version 1.20.1 für Parallelabfrage-Cluster</a>	Aurora MySQL-Version 1.20.1 ist verfügbar, wenn Sie einen Parallelabfragecluster erstellen .	11. Juni 2020
<a href="#">Aurora MySQL Version 2.08.0</a>	Aurora MySQL-Version 2.08.0 ist verfügbar.	2. Juni 2020
<a href="#">Aurora MySQL Version 1.19.6 für Parallelabfrage-Cluster</a>	Aurora MySQL-Version 1.19.6 ist verfügbar, wenn Sie einen Parallelabfragecluster erstellen .	2. Juni 2020
<a href="#">Aurora MySQL Version 2.07.2</a>	Aurora MySQL Version 2.07.2 ist verfügbar.	17. April 2020
<a href="#">Aurora MySQL Version 1.22.2</a>	Aurora MySQL-Version 1.22.2 ist verfügbar.	5. März 2020



---

<a href="#">Aurora MySQL Version 1.20.1</a>	Aurora MySQL-Version 1.20.1 ist verfügbar.	5. März 2020
<a href="#">Aurora MySQL Version 1.19.6</a>	Aurora MySQL-Version 1.19.6 ist verfügbar.	5. März 2020
<a href="#">Aurora MySQL Version 1.17.9</a>	Aurora MySQL-Version 1.17.9 ist verfügbar.	5. März 2020
<a href="#">Aurora MySQL Version 2.07.1</a>	Aurora MySQL-Version 2.07.1 ist verfügbar.	23. Dezember 2019
<a href="#">Aurora MySQL Version 1.22.1</a>	Aurora MySQL-Version 1.22.1 ist verfügbar.	23. Dezember 2019
<a href="#">Aurora MySQL Version 2.07.0</a>	Aurora MySQL Version 2.07.0 ist verfügbar.	25. November 2019
<a href="#">Aurora MySQL Version 1.22.0</a>	Aurora MySQL Version 1.22.0 ist verfügbar.	25. November 2019
<a href="#">Aurora MySQL Version 1.21.0</a>	Aurora MySQL Version 1.21.0 ist verfügbar.	25. November 2019
<a href="#">Aurora MySQL Version 2.06.0</a>	Aurora MySQL Version 2.06.0 ist verfügbar.	22. November 2019
<a href="#">Aurora MySQL Version 2.04.8</a>	Aurora MySQL Version 2.04.8 ist verfügbar.	20. November 2019
<a href="#">Aurora MySQL Version 2.04.7</a>	Aurora MySQL Version 2.04.7 ist verfügbar.	14. November 2019
<a href="#">Aurora MySQL Version 2.05.0</a>	Aurora MySQL Version 2.05.0 ist verfügbar.	11. November 2019
<a href="#">Aurora MySQL Version 1.20.0</a>	Aurora MySQL Version 1.20.0 ist verfügbar.	11. November 2019

---

<a href="#">Aurora MySQL Version 2.04.6</a>	Aurora MySQL Version 2.04.6 ist verfügbar.	19. September 2019
<a href="#">Aurora MySQL Version 1.19.5</a>	Aurora MySQL Version 1.19.5 ist verfügbar.	19. September 2019
<a href="#">Aurora MySQL Version 2.04.5</a>	Aurora MySQL Version 2.04.5 ist verfügbar.	8. Juli 2019
<a href="#">Aurora MySQL Version 1.19.2</a>	Aurora MySQL Version 1.19.2 ist verfügbar.	5. Juni 2019
<a href="#">Aurora MySQL Version 2.04.4</a>	Aurora MySQL Version 2.04.4 ist verfügbar.	29. Mai 2019
<a href="#">Aurora MySQL Version 2.04.3</a>	Aurora MySQL Version 2.04.3 ist verfügbar.	9. Mai 2019
<a href="#">Aurora MySQL Version 1.19.1</a>	Aurora MySQL Version 1.19.1 ist verfügbar.	9. Mai 2019
<a href="#">Aurora MySQL Version 2.04.2</a>	Aurora MySQL Version 2.04.2 ist verfügbar.	2. Mai 2019
<a href="#">Aurora MySQL Version 2.04.1</a>	Aurora MySQL Version 2.04.1 ist verfügbar.	25. März 2019
<a href="#">Aurora MySQL Version 2.04</a>	Aurora MySQL Version 2.04 ist verfügbar.	25. März 2019
<a href="#">Aurora MySQL Version 2.03.4</a>	Aurora MySQL Version 2.03.4 ist verfügbar.	7. Februar 2019
<a href="#">Aurora MySQL Version 1.19.0</a>	Aurora MySQL Version 1.19.0 ist verfügbar.	7. Februar 2019
<a href="#">Aurora MySQL Version 2.03.3</a>	Aurora MySQL Version 2.03.3 ist verfügbar.	18. Januar 2019

---

<a href="#">Aurora MySQL Version 1.17.8</a>	Aurora MySQL Version 1.17.8 ist verfügbar.	17. Januar 2019
<a href="#">Aurora MySQL Version 2.03.2</a>	Aurora MySQL Version 2.03.2 ist verfügbar.	9. Januar 2019
<a href="#">Aurora MySQL Version 2.03.1</a>	Aurora MySQL Version 2.03.1 ist verfügbar.	24. Oktober 2018
<a href="#">Aurora MySQL Version 2.03</a>	Aurora MySQL Version 2.03 ist verfügbar.	11. Oktober 2018
<a href="#">Aurora MySQL Version 2.02.5</a>	Aurora MySQL Version 2.02.5 ist verfügbar.	8. Oktober 2018
<a href="#">Aurora MySQL Version 1.17.7</a>	Aurora MySQL Version 1.17.7 ist verfügbar.	8. Oktober 2018
<a href="#">Aurora MySQL Version 2.02.4</a>	Aurora MySQL Version 2.02.4 ist verfügbar.	21. September 2018
<a href="#">Aurora MySQL Version 1.18.0</a>	Aurora MySQL Version 1.18.0 ist verfügbar.	20. September 2018
<a href="#">Aurora MySQL Version 1.17.6</a>	Aurora MySQL Version 1.17.6 ist verfügbar.	6. September 2018

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.